Pontificia Universidad Javeriana Cali
Facultad de Ingeniería.
Ingeniería de Sistemas y Computación.
Proyecto de Grado.

Apoyo en la compra y venta de acciones de la bolsa de valores estadounidense utilizando técnicas de aprendizaje por refuerzo

Elkin Jadier Narvaez Paz

Director: Dr. Diego Luis Linares Ospina
Directora: Dra. Gloria Inés Álvarez Vargas

05 de Octubre del 2023

Pontificia Universidad JAVERIANA Cali

Facultad de Ingeniería y Ciencias

Acta de Correcciones al Proyecto de Grado
Ingeniería de Sistemas y Computación

Fecha: 15 de noviembre del 2023

Autores: Elkin Jadier Narvaez Paz

Nombre del Proyecto de Grado: Apoyo en la compra y venta de acciones de la bolsa de valores estadounidense utilizando técnicas de aprendizaje por refuerzo

Director: Diego Luis Linares Ospina

Como indica el artículo 2.27 de las Directrices de Trabajo de Grado, he verificado que los estudiantes indicados arriba han implementado todas las correcciones que los Jurados del Proyecto de Grado definieron que se efectuaran, como consta en el Acta de Calificación correspondiente.

Firma de director(a) del Proyecto de Grado

Nota de Aceptación

Aprobado por el Comité de Trabajo de Grado en cumplimiento de los requisitos exigidos por la Pontificia Universidad Javeriana para optar el título de Ingeniero de Sistemas y Computación.

DR. HENÁN CAMILO ROCHA NIÑO
Decano de la Facultad de Ingeniería

DR. GERARDO MAURICIO SARRIA
Director Carrera Ingeniería Sistemas y Computación.

DR. DIEGO LUIS LINARES OSPINA
Director Trabajo

DRA. GLORIA INÉS ÁLVAEZ VARGAS
Codirectora Trabajo

DR. JULIÁN GIL GONZÁLEZ
Jurado 1

DRA. LUISA FERNANDA RINCÓN PÉREZ
Jurado 2

Santiago de Cali, 05 de Octubre del 2023.

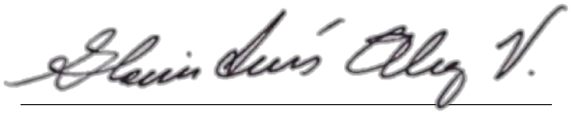
Señores
Pontificia Universidad Javeriana Cali.
Dr. Gerardo Mauricio Sarria
Director Carrera de Ingeniería de Sistemas y Computación.
Cali.

Cordial Saludo.

Por medio de la presente nos permitimos informarle que el estudiante de Ingeniería de Sistemas y Computación Elkin Jadier Narvaez Paz (cod: 8943358) culminó satisfactoriamente bajo nuestra dirección el proyecto de grado titulado "Apoyo en la compra y venta de acciones de la bolsa de valores estadounidense utilizando técnicas de aprendizaje por refuerzo".

Atentamente,

Dr. Diego Luis Linares Ospina



Dra. Gloria Inés Álvarez Vargas

Santiago de Cali, 05 de Octubre del 2023.

Señores
Pontificia Universidad Javeriana Cali.
Dr. Gerardo Mauricio Sarria
Director Carrera de Ingeniería de Sistemas y Computación.
Cali.

Cordial Saludo.

Me permito presentar a su consideración el proyecto de grado titulado "Apoyo en la compra y venta de acciones de la bolsa de valores estadounidense utilizando técnicas de aprendizaje por refuerzo" con el fin de cumplir con los requisitos exigidos por la Universidad para optar al título de Ingeniero de Sistemas y Computación.

Al firmar aquí, doy fe que entiendo y conozco las directrices para la presentación de trabajos de grado de la Facultad de Ingeniería aprobadas el 26 de Noviembre de 2009, donde se establecen los plazos y normas para el desarrollo del anteproyecto y del trabajo de grado.

Atentamente,

Elkin Narvaez

Elkin Jadier Narvaez Paz
Código: 8943358

Abstract

This research explored the development of a reinforcement learning-based trading agent, aiming to evaluate its performance in comparison to the *buy-and-hold* strategy within the dynamic environemnt of financial markets. The core problem at the heart of this investigation was to discern whether an agent trained through reinforcement learning techniques could effectively navigate the complexities of trading, marked by market volatility, evolving trends, and financial uncertainties. Our approach involved a deliberate and iterative progression of experiments, beginning with preliminary phases designed to extract qualitative insights and identify trends. These early experiments were constrained in terms of computational resources and training duration but proved instrumental in steering us toward optimal hyperparameters and training configurations for the last experiment. This last experiment, characterized by an extensive training period, stands as the focal point of our findings, showcasing the culmination of our efforts. Within this extended training, the agent exhibited a remarkable capacity to adapt to dynamic market conditions, resulting in performance that competes favorably with the *buy-and-hold* strategy. These outcomes underscore the agent's adaptability and learning capabilities within the context of trading, revealing its potential for practical application in real financial markets. In conclusion, this research sheds light on the promise of reinforcement learning in trading, emphasizing the importance of a systematic approach to experimentation and paving the way for future refinements to enhance the model's robustness for real-world financial applications.

Keywords: Reinforcement learning, trading agent, buy-and-hold strategy, technical analysis indicators, stock market, financial positions, deep learning.

Resumen

Este proyecto de investigación se adentró en el desarrollo de un agente de trading basado en aprendizaje por refuerzo, con el objetivo de evaluar su desempeño en comparación con la estrategia *buy-and-hold* en el dinámico entorno de los mercados financieros. El problema central de esta investigación era evaluar si un agente entrenado mediante técnicas de aprendizaje por refuerzo podía navegar eficazmente por las complejidades de la compra y venta de activos, marcadas por la volatilidad del mercado, tendencias cambiantes e incertidumbres financieras. Nuestro enfoque implicó una progresión deliberada e iterativa de experimentos, comenzando con fases preliminares diseñadas para extraer información cualitativa e identificar tendencias. Estos primeros experimentos estaban limitados en términos de recursos computacionales y duración del entrenamiento, pero fueron fundamentales para dirigirnos hacia hiperparámetros óptimos y configuraciones de entrenamiento para el último experimento. Este último experimento, caracterizado por un período de entrenamiento extenso, se destaca como el punto central de nuestros hallazgos, exhibiendo la culminación de nuestros esfuerzos. Dentro de este entrenamiento extendido, el agente demostró una notable capacidad para adaptarse a las dinámicas cambiantes del mercado, lo que se tradujo en un desempeño que compite favorablemente con la estrategia *buy-and-hold*. Estos resultados destacan la adaptabilidad y las capacidades de aprendizaje del agente en el contexto de compra y venta de activos, revelando su potencial para su aplicación práctica en los mercados financieros reales. En conclusión, esta investigación arroja luz sobre la promesa del aprendizaje por refuerzo en el trading, enfatizando la importancia de un enfoque sistemático para la experimentación y dejando el camino para futuros refinamientos destinados a mejorar la robustez del modelo para aplicaciones financieras del mundo real.

Palabras Clave: Aprendizaje por refuerzo, agente de trading, estrategia *buy-and-hold*, indicadores técnicos, posiciones financieras, aprendizaje profundo

Índice general

Introducción

La inversión en acciones ha sido reconocida durante mucho tiempo como un poderoso vehículo para la acumulación de riqueza, ofreciendo a los inversionistas una perspectiva de retornos sustanciales a largo plazo. La parte atractiva de las inversiones en el mercado de valores radica en la oportunidad que brinda a las personas para hacer crecer su capital, asegurar su futuro financiero y cumplir sus aspiraciones de inversión. El rendimiento histórico del mercado de valores corrobora su papel como base fundamental de muchas carteras de inversión. Con el potencial de la creación de riqueza y la seguridad financiera, actúa como un motor clave para el bienestar financiero individual.

Sin embargo, detrás de la promesa de ganancias financieras a largo plazo, invertir en acciones puede presentar un gran desafío, especialmente cuando se trata de inversiones a corto plazo. Las acciones son conocidas por su inherente volatilidad, caracterizada por fluctuaciones rápidas y a menudo impredecibles en los precios en periodos cortos. Por ejemplo, considere los abruptos cambios de precios exhibidos por gigantes tecnológicos como Amazon y Tesla, o la inestabilidad experimentada por las compañías farmacéuticas mientras navegan por los resultados de ensayos clínicos cruciales. Estos movimientos abruptos de precios pueden provocar incertidumbre y riesgo significativo en el panorama de inversión, generando preguntas sobre la idoneidad de la participación a corto plazo en el mercado de valores.

Aún así, las implicaciones de la volatilidad del mercado de valores se extienden mucho más allá del ámbito de los inversionistas individuales. Las repercusiones afectan a economías enteras, afectando a millones de personas y sectores más allá del alcance directo de los mercados financieros. El bienestar de los ciudadanos de una nación, la estabilidad de su moneda e incluso el progreso de sus iniciativas de desarrollo están relacionados con el desempeño de sus mercados de valores. En consecuencia, mitigar los desafíos planteados por la volatilidad del mercado de valores no es solo una preocupación para los inversionistas, sino una cuestión de importancia nacional.

En respuesta a estos desafíos, la aparición de agentes de compra y venta de activos impulsados por tecnologías de aprendizaje por refuerzo presenta una oportunidad transformadora. Estos sofisticados agentes, diseñados con capacidades avanzadas de inteligencia artificial, poseen el potencial de descifrar las complejidades de la dinámica del mercado de valores. Pueden ofrecer un apoyo a los inversionistas para tomar decisiones más informadas, estratégicas y basadas en datos. Al aprovechar el análisis colectivo incorporado en datos históricos del mercado y adaptarse dinámicamente a las condiciones del mercado en tiempo real, estos agentes están especialmente preparados para navegar por la inestabilidad a corto plazo de los mercados de valores. Al hacerlo, permiten a los inversionistas aprovechar oportunidades mientras gestionan prudentemente los riesgos asociados con la volatilidad de los cambios abruptos en los precios de los activos. En las siguientes secciones, profundizamos en el mundo de las inversiones en el mercado de valores, examinando los desafíos planteados por la volatilidad a corto plazo y explorando el papel fundamental de los agentes de trading basados en aprendizaje por refuerzo en promover la estabilidad financiera y la prosperidad económica.

Capítulo 1

Análisis

1.1. Planteamiento del Problema

Las acciones son una parte muy importante en cualquier cartera de inversión, la cual a su vez puede estar diversificada en diferentes sectores, tales como el sector financiero, industrial, tecnológico, de la salud, entre otros. Estos sectores pueden categorizarse como defensivos, sensibles o cíclicos, donde cada categoría indica el impacto que podría llegar a causar, ya sea positivo o negativo, si llegaran a ocurrir alteraciones en los ciclos económicos [1].

Adicionalmente a las acciones, una cartera puede estar compuesta por otros tipos de activos, tales como bonos, materias primas, dinero en efectivo, y otros equivalentes al efectivo, incluyendo los fondos cerrados y los fondos cotizados en la bolsa (ETFs) [1]. Sin embargo, incluir una renta variable en una mayor proporción dentro de un portafolio de inversión ha demostrado una mejor rentabilidad al inversor debido a sus grandes retribuciones a largo plazo [2]. Por ejemplo, de acuerdo con el informe de Análisis Cuantitativo del Comportamiento de los Inversores (QAIB) realizado por DALBAR en el 2017 [3], un inversor promedio en fondos de renta variable obtiene una retribución del 20.64 % en un periodo de 12 meses, mientras que un inversor promedio de renta fija obtiene una retribución del 1.52 % durante el mismo periodo.

Aunque una inversión en renta variable presenta ventajas en comparación con la renta fija, también tiene sus limitaciones debido a su alta volatilidad y propensión a altas valoraciones y caídas, lo cual hace que su riesgo de inversión sea elevado [4]. Por esta razón, han surgido diferentes estrategias que ayudan a hacer una mejor distribución de las acciones que son incluidas dentro de un portafolio. La Figura 1.1 muestra un ejemplo de esto, donde podemos observar que el tener una diversificación de acciones entre nacionales y extranjeras dentro de nuestra cartera, produce un menor riesgo de pérdida, ya que los comportamientos negativos de algunos activos son compensados con los comportamientos positivos de otros.

Aunque la renta variable ha demostrado una mejor rentabilidad a largo plazo en comparación con la renta fija, esta ha estado muy por debajo de las expectativas que los inversionistas tienen en periodos prolongados. De acuerdo con el informe QAIB realizado por DALBAR en el 2017 [3], un inversionista promedio de renta variable obtiene una retribución del 4.88 % en un periodo de 10 años, lo cual, si bien representa una ganancia, termina representando un rendimiento muy inferior en comparación con el rendimiento objetivo que se esperaría de sus inversiones. La situación se vuelve un poco más desalentadora para los inversionistas en renta fija, ya que, según este informe,

[1] Un ciclo económico es una serie de fases por las que pasa la economía y que suceden en orden hasta llegar a la fase final en la que el ciclo económico comienza de nuevo.

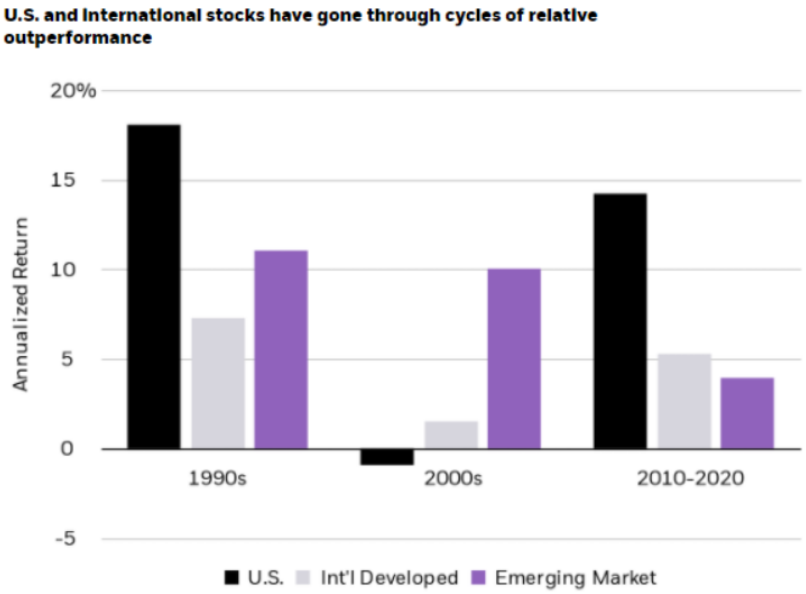


Figura 1.1: Distribución de acciones dentro de los Estados Unidos y extranjeras. *Recuperado de: https://www.lynalden.com/asset-allocation/*

obtienen una retribución del 0.48 % en un periodo de 10 años, lo cual se puede traducir directamente a pérdidas considerando solo el factor de la inflación. De manera similar, las retribuciones al invertir en acciones que se han venido presentando en los últimos 10 años han estado por debajo de índices mundiales reconocidos del mercado de valores, tales como el S&P 500 [2]. En el 2014, la retribución de un inversionista promedio de renta variable tuvo un rendimiento inferior al S&P 500 por un 1.52 % (5.88 % vs 7.40 %) en un periodo de 10 años; y en el 2018, la retribución tuvo un rendimiento inferior al S&P 500 por un 3.62 % (4.88 % vs 8.50 %) durante el mismo periodo [3] [5].

Dejar que las emociones guíen las decisiones a la hora de invertir es una de las razones por la cual los inversionistas ven reflejados rendimientos que están por debajo de sus expectativas [6]. Cuando el precio de una acción sube, el comportamiento natural de un inversionista es comprar lo más rápido posible, ya que es guiado por un sentimiento de euforia que hace que pierda la noción del riesgo. De manera similar, cuando el precio de una acción baja, el comportamiento natural es vender tan pronto como pueda, ya que es gobernado por una cohibición o temor de que posiblemente pueda tener pérdidas. Esto ha creado la necesidad de que existan herramientas automáticas que apoyen en la toma de decisiones al momento de invertir en acciones, dejando a un lado estos puntos débiles que los seres humanos tenemos.

[2] S&P 500 es un índice del mercado de valores que hace seguimiento del desempeño de 500 grandes empresas que cotizan en las bolsas de valores de los Estados Unidos

1.1.1. Formulación

¿Cómo diseñar un modelo de aprendizaje por refuerzo que apoye en la toma de decisiones relacionadas con la inversión en acciones de la bolsa de valores estadounidense?

1.1.2. Sistematización

- ¿Qué información se debe utilizar como fuente para la toma de decisiones al momento de invertir en acciones?
- ¿Cómo hacer el preprocesamiento de los datos?
- ¿Qué técnicas de aprendizaje automático por refuerzo serán utilizadas para construir modelos de toma de decisiones en el la compra y venta de acciones?
- ¿Cómo hacer la estimación de los parámetros e hiperparámetros en las técnicas que lo requieran con el fin de mejorar el desempeño de los modelos?
- ¿Cómo realizar la evaluación de los modelos?

1.2. Objetivos

1.2.1. Objetivo General

Diseñar un modelo de aprendizaje por refuerzo que apoye en la toma de decisiones relacionadas con la inversión en acciones de la bolsa de valores estadounidense

1.2.2. Objetivos Específicos

- Identificar el tipo de información que debe ser recolectada como fuente para la toma de decisiones al momento de invertir en acciones.
- Determinar las tareas de preprocesamiento.
- Determinar las técnicas de aprendizaje automático por refuerzo que serán utilizadas para la construcción de los modelos de toma de decisiones en la compra y venta de acciones.
- Encontrar los parámetros e hiperparámetros en las técnicas que lo requieran de tal forma mejoren el desempeño de los modelos
- Realizar la evaluación de los modelos construidos.

1.3. Justificación

Tener un portafolio de inversiones se ha convertido en una de las formas en la cual millones de personas tratan de obtener ganancias a corto plazo, especialmente cuando se trata de inversiones en renta variable. De acuerdo con Lydia Saad y Jeffrey M. Jones [7], en el 2021 se identificó que el 56 % de personas que residen en los Estados Unidos invierten en acciones, donde se encuentran el 39 % de personas entre 18 y 29 años, el 60 % de personas entre 30 y 49 años, el 62 % de personas entre 50 y 64 años y el 59 % de personas con 50+ años. De esta manera, las malas decisiones que se tomen al momento de invertir podrían llegar a tener un impacto negativo en el estado financiero de una gran proporción de personas, poniendo incluso en riesgo la misma economía de un país.

Este proyecto tiene como propósito apoyar a un inversionista en la gestión de su propio portafolio de acciones, tomando decisiones que traten de disminuir el riesgo y aumentar el rendimiento de sus retribuciones. De esta manera, este trabajo podría beneficiar a muchas personas que posean portafolios de inversiones, ya que ayudará a tomar decisiones que traten de optimizar sus ganancias. Adicionalmente, este proyecto presenta una alta viabilidad ya que no requiere ningún costo al que se deba incurrir para lograr su implementación.

1.4. Delimitaciones y Alcances

Se preveen las siguientes delimitaciones y alcances:

- Todo el desarrollo se llevará a cabo usando el lenguaje de programación Python.
- El proyecto solo abarcará un conjunto de acciones de la bolsa de valores estadounidense pertenecientes al índice S&P 500.
- Los modelos implementados se ejecutarán sobre un ambiente simulado de compra y venta de acciones.

1.4.1. Entregables

Como resultado de este proyecto se entregará lo siguiente:

- Documento de tesis.
- Un script en Python que contendrá el código del preprocesamiento de los datos, el entrenamiento de los modelos y su respectiva evaluación.

1.5. Metodología

1.5.1. Tipo de Estudio

Este proyecto tiene un enfoque de tipo de estudio experimental, el cual se basa en explorar distintas técnicas de aprendizaje automático por refuerzo basadas en datos de análisis bursátil.

Para este estudio exploratorio se hará una evaluación de las distintas métricas de desempeño y error que los modelos presenten, y de igual forma se evaluará el comportamiento que los agentes tengan en un entorno de compra y venta de acciones simulado.

1.5.2. Actividades

1. Obtención de datos
   - Identificar repositorios públicos de datos financieros
   - Obtener datos técnicos
2. Pre procesamiento de los datos
   - Analizar los datos
   - Identificar atributos a considerar
   - Crear indicadores basados en análisis técnico fundamental
   - Limpiar datos
   - Transformar los datos
   - Realizar una separación de los datos
3. Selección de los modelos
   - Realizar una revisión de los modelos más usados para la predicción en el mercado de acciones
   - Realizar pruebas de funcionamiento de los modelos
   - Seleccionar los modelos
4. Desarrollo de los modelos
   - Estudiar librerías que implementan los modelos
   - Hacer una estimación de parámetros e hiper parámetros
   - Implementar un ambiente simulado de compra y venta de acciones
5. Evaluación de desempeño y análisis
   - Calcular métricas de desempeño para los modelos
   - Comparar las métricas obtenidas para los distintos modelos
   - Analizar el comportamiento que los modelos presentan en el ambiente simulado de compra y venta de acciones
   - Documentación y presentación del proyecto

Capítulo 2

Marco de referencia

2.1. Áreas Temáticas

Este proyecto de grado abarca dos áreas temáticas principales, las cuales son **aprendizaje automático** y **análisis bursátil**. Por un lado, el aprendizaje automático es un área que hace parte de la disciplina de ciencias de la computación, y en particular este trabajo se enfoca en la aplicación y uso de técnicas de **aprendizaje por refuerzo**. Por otro lado, el análisis bursátil es una disciplina que, aunque no está completamente relacionada con el área de estudio de este trabajo de grado, esta representa el área de aplicación sobre la cual se van a aplicar las metodologías propias de la disciplina.

2.2. Marco Teórico

2.2.1. Términos y conceptos

Análisis bursátil: Es una disciplina que estudia los movimientos de las cotizaciones bursátiles a través de gráficos, balances, series temporales o series estadísticas [8].

Análisis técnico: Es una disciplina empleada para evaluar inversiones e identificar oportunidades de trading analizando tendencias estadísticas obtenidas por la actividad que se presenta en el mercado, tales como el movimiento del precio o volumen [9].

Aprendizaje por refuerzo: El aprendizaje por refuerzo es un método de entrenamiento de aprendizaje automático que se basa en recompensar los comportamiento deseados y/o castigar los no deseados. De esta manera, un agente de aprendizaje por refuerzo es capaz de percibir e interpretar su ambiente, tomar acciones y aprender a través de prueba y error [10].

Bolsa de valores: El mercado de valores se refiere al conjunto de mercados e intercambios donde se llevan a cabo las actividades regulares de compra, venta y emisión de valores, tales como acciones, bonos, entre otros instrumentos financieros [11].

Posición financiera: Una posición es la cantidad de un valor, activo o moneda que posee una persona o entidad. Hay dos tipos básicos de posición: una larga (manteniendo una cantidad positiva del activo) y una corta (manteniendo una cantidad negativa del activo). [12].

2.2.2. Aprendizaje por refuerzo

De acuerdo con Fischer [13], la mayor parte de la investigación con respecto a la aplicación de aprendizaje automático en la predicción del mercado de acciones se ha dedicado al aprendizaje

supervisado. Entre los métodos más usados para esta tarea se encuentran: máquinas de vectores de soporte [14, 15], algoritmos basados en árboles [16, 17, 18, 19] y clasificación de vecinos más cercanos [20]. Según Fischer [13], la idea general de todos estos trabajos se basa en entrenar un modelo predictivo basado en datos históricos para prever el cambio de precio de una acción usando un conjunto de variables explicativas (características), para luego alimentar un modulo de trading que deriva la acción que se debe realizar, e.g., comprar el activo financiero en caso de que la predicción sobrepase un cierto umbral. Sin embargo, Fischer señala que este enfoque tiene distintas limitaciones que pueden llevar a que se tenga un rendimiento subóptimo [21]. En primer lugar, el objetivo de optimización en el modelo predictivo no está necesariamente alineado con el objetivo final del inversionista. Por ejemplo, si un inversionista quiere maximizar el rendimiento por unidad de riesgo, no hay forma de que el modelo pueda adaptarse a ese requerimiento, ya que su objetivo principal es minimizar el error de la predicción. En segundo lugar, en la mayoría de los casos, solo la predicción es usada como entrada del modulo que realiza las acciones de compra y venta, lo cual causa que se descarte información adicional valiosa que podría obtenerse del espacio de características [21]. En tercer lugar, las restricciones impuestas por factores externos del entorno escasamente se incorporan en la optimización del componente de trading, o en muchos de los casos no se toma en cuenta para nada. Entre estos factores externos podemos encontrar cosas como los costos que tienen las trasacciones o la falta de liquidez[1]. De esta manera, Fischer [13] expone que el aprendizaje por refuerzo hace a un lado estas limitaciones, ya que este se basa en "aprender qué hacer y cómo asignar situaciones a acciones, con el fin de maximizar una recompensa" [23]. Las aplicaciones más recientes de aprendizaje por refuerzo (RL, por sus siglas en inglés) en mercados financieros consideran espacios de acción y estado discretos o continuos, para lo cual se emplean los siguientes enfoques: critic-only, actor-only y actor-critic [24]. El enfoque critic-only se centra en estimar el valor de las acciones, actor-only selecciona acciones directamente basadas en una política, y actor-critic combina ambos enfoques. El enfoque critic-only es la aplicación más frecuente de aprendizaje por refuerzo en los mercados financieros. La Figura 2.1 muestra la frecuencia que cada uno de estos enfoques ha tenido en los últimos 25 años (1993-2018).

La naturaleza interactiva del aprendizaje por refuerzo hace que sea particularmente adecuado para el dominio de compra y venta de activos dentro de un portafolio de inversión. El aprendizaje por refuerzo trata de modelar un aprendizaje dirigido por un agente que interactúa con un ambiente, generalmente estocástico, del que tiene una información incompleta o desconocida. El objetivo es que el agente pueda tomar decisiones que permitan lograr un objetivo a largo plazo al aprender el valor de estados y acciones basadas en una señal de recompensa. Este aprendizaje continuo por medio de recompensas deriva una política que codifica las reglas de comportamiento del agente asignando estados a acciones. Este tipo de aprendizaje difiere del aprendizaje supervisado ya que optimiza el comportamiento de un agente basado en experiencia de intento y error por medio de una señal de recompensa, en lugar de generalizar usando ejemplos representativos correctamente etiquetados de la variable objetivo. De igual manera, el aprendizaje por refuerzo no solo se limita en hacer predicciones, sino que también incluye acciones y sus consecuencias dentro del ambiente.

[1] Situación financiera que se caracteriza por la falta de efectivo o activos que sean fácilmente convertibles en efectivo [22]

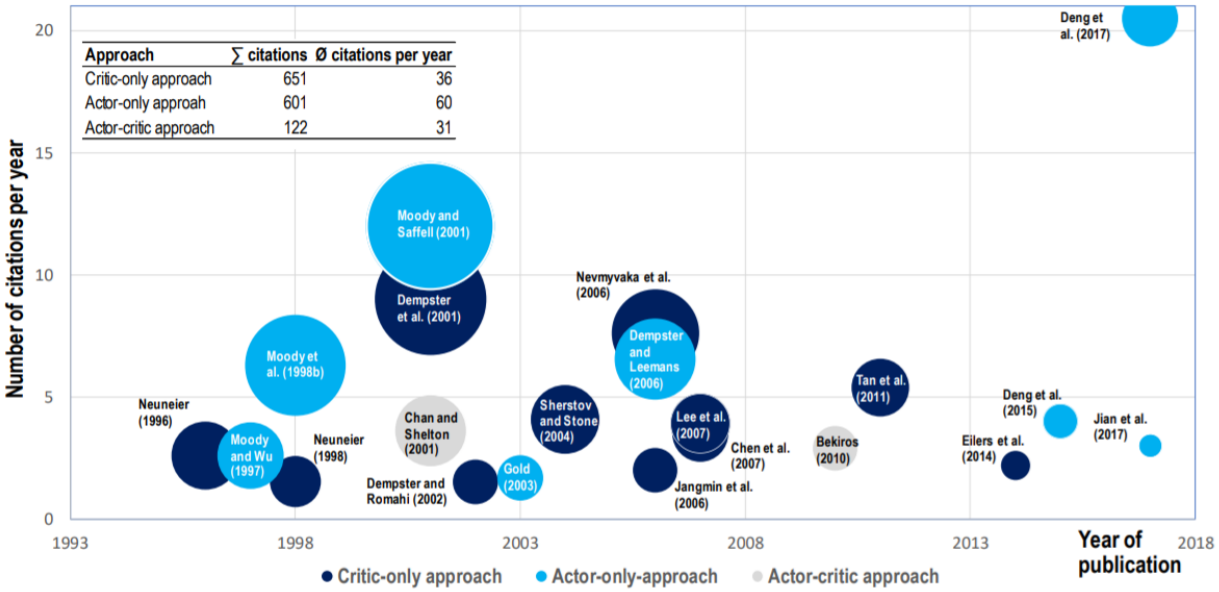


Figura 2.1: Descripción general de los trabajos de investigación que aplican aprendizaje por refuerzo en los mercados financieros [24]

La estructura del ambiente sobre el cual el agente interactúa tiene un impacto muy importante en la habilidad que tiene para aprender una tarea en específico. Es por ello que la construcción de un ambiente adecuado también requiere una tarea de exploración y de diseño con el fin de facilitar el proceso de entrenamiento. De una manera general, el objetivo del ambiente es 1) presentar información acerca de su estado al agente, 2) asignar recompensas a acciones y 3) llevar el agente a nuevos estados, sujeto a distribuciones de probabilidad que pueda llegar a saber o no. La representación del estado y el espacio de acciones del agente dentro de un ambiente puede ser discreto o continuo. Acciones y estados continuos requieren aprendizaje automático para aproximar una relación funcional entre estados, acciones y sus valores. De esta manera, también se requiere un grado de generalización ya que el agente solo va a poder obtener experiencia de un subconjunto potencialmente infinito de estados continuos durante la fase de entrenamiento. La figura 2.2 ilustra la relación que existe entre el agente y el ambiente.

2.2.2.1. Política

La política es aquella que define el comportamiento del agente en cualquier momento del tiempo. Esta crea una relación entre cualquier estado y una o más acciones que eventualmente llevarán al agente a un nuevo estado. Cuando se tiene un ambiente con un número finito de estados y acciones, la política se puede representar con una simple tabla de búsqueda que ira cambiando durante el entrenamiento. Sin embargo, cómo veremos más adelante, aunque el ambiente de compra y venta

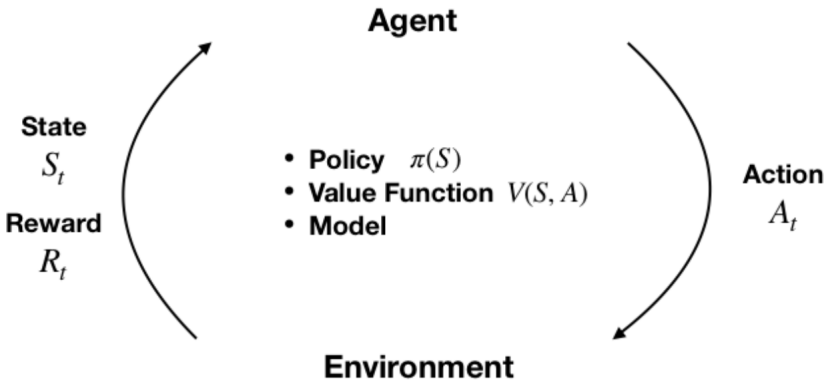


Figura 2.2: Interacción entre agente y ambiente [25]

de acciones de este proyecto tiene un número discreto de acciones (compra, retención y venta), este tiene un número continuo (i.e., infinito) de estados. Por esta razón, la política toma la forma de una función que es aproximada haciendo uso de técnicas de aprendizaje automático.

2.2.2.2. Recompensas

La recompensa es un valor enviado por el ambiente cada vez que el agente cambia de estado. El objetivo del agente es maximizar las recompensas que va obteniendo durante su entrenamiento. Generalmente las recompensas son consideradas con un factor de descuento que depende del momento de tiempo en que son recibidas y que van reflejando un decaimiento en el tiempo (entre más distantes del momento inicial, van a ir decreciendo). Esta recompensa es usada por el agente para aprender el valor de sus decisiones en un estado dado y modificar su política como corresponda. La definición de la recompensa está fuertemente ligada con el diseño del ambiente, lo cual va a ser explicado en un capítulo posterior. Ya que nuestro ambiente es de compra y venta de acciones, se tuvieron que considerar recompensas para las operaciones de **compra**, **retención** y **venta**.

2.2.2.3. Función de valor

Vimos en la sección anterior que la recompensa es una retroalimentación inmediata que el ambiente provee agente. En su lugar, la función de valor permite que el agente pueda tomar decisiones óptimas en el largo plazo. La función de valor trata de resumir la utilidad de estados o acciones en un estado dado en términos de su recompensa futura. La importancia de la función de valor radica en que el agente también tiene que dar cuenta de aquellos casos en los cuales recompensas bajas conducen a mejores resultados futuros (o viceversa). Al tener conocimiento sobre todas las probabilidades de las transiciones y recompensas dentro el ambiente, podemos expresar la función de valor como se muestra en las ecuaciones 2.1 y 2.2. La ecuación 2.1, también conocida como

función de valor estado-acción, estima la recompensa a largo plazo dado un estado y una acción; en su lugar, la función de valor expresada en la ecuación 2.2 estima la recompensa a largo plazo para cada estado. Los valores r_t representan las recompensas obtenidas, mientras que γ representa un valor de descuento temporal aplicado a las recompensas.

$$Q^{\pi}(s,a) = E_{\pi}[r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \cdots | s, a] \tag{2.1}$$

$$v_{\pi}(s) = E_{\pi}[r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \cdots | S_t = s] \tag{2.2}$$

2.2.3. Retos al usar aprendizaje por refuerzo

El aprendizaje por refuerzo generalmente presenta dos problemas principales: la **asignación de crédito** y la **compensación entre exploración y explotación**.

2.2.3.1. Asignación de crédito

El problema de asignación de crédito es el reto de estimar los beneficios y costos de acciones en un estado dado, sin importar si su contribución se ve reflejada en acciones futuras. De esta manera, los algoritmos de aprendizaje por refuerzo necesitan buscar una forma de distribuir el crédito para resultados positivos y negativos entre todas las decisiones que pudieron llegar a contribuir para llegar ahí.

2.2.3.2. Exploración vs explotación

Un agente debe estimar el valor de los estados y las acciones a medida que va experimentando diferentes trayectorias en el ambiente. Estas trayectorias se van definiendo con base a las decisiones que el agente va tomando en cada paso. Sin embargo, estas decisiones están basadas en un aprendizaje incompleto, razón por la cual si el agente solo usa su experiencia pasada para tomarlas, se corre en el riesgo de dejar de explorar nuevas trayectorias que pueden conducir a mejores resultados. De esta manera, dejar que el agente solo explote su experiencia para la toma de decisiones limita la exposición que tiene en el ambiente y le impide aprender una política óptima.

Por esta razón, un algoritmo de aprendizaje por refuerzo necesita balancear su exploración y explotación. En el capítulo siguiente se explica cómo se maneja la exploración y la explotación para este proyecto en particular.

2.2.4. Enfoques al usar aprendizaje por refuerzo

Existen diferentes enfoques para resolver problemas de aprendizaje por refuerzo, sin embargo todos tienen en común la búsqueda un conjunto de reglas para lograr un comportamiento óptimo en el agente. Los enfoques más conocidos son los métodos de **Programación Dinámica (DP)**, de **Monte Carlo (MC)** y el aprendizaje de **Diferencia Temporal (TD)**.

2.2.4.1. Métodos de Programación Dinámica (DP)

Los métodos de programación dinámica asumen un conocimiento completo del ambiente, lo cual en este caso no es muy realista ya que se requeriría conocer todas la probabilidades de las transiciones y recompensas. Sin embargo, este enfoque es la base para la mayor parte de otros enfoques existentes.

2.2.4.2. Métodos de Monte Carlo (MC)

Este tipo de métodos aprenden acerca del ambiente y del costo y beneficio de diferentes decisiones muestreando secuencias completas de estado-acción-recompensa. En el caso de un ambiente de compra y venta de activos, este método tampoco sería útil debido a la naturaleza continua del espacio de estados.

2.2.4.3. Diferencia Temporal (TD)

Este enfoque mejora significativamente la eficiencia de muestreo al aprender por secuencias más cortas. Se basa en **bootstrapping**, lo cual se define como la refinación de sus estimaciones basado en sus propias estimaciones previas. De este tipo de aprendizaje nace un algoritmo llamado **Q-learning**, el cuál será la base para este trabajo. Este algoritmo tiene que someterse a una serie de modificaciones para poder manejar un conjunto de estados continuos, entre las cuales se encuentra la aproximación de la función de valor por medio de aprendizaje supervisado, más precisamente haciendo uso de **deep learning**. Cabe resaltar que este tipo de métodos que abordan espacios de acciones o estados continuos presentan diferentes limitaciones en el contexto de aprendizaje por refuerzo, tales como no reflejar directamente el concepto objetivo (como si lo haría un conjunto de datos de entrenamiento debidamente etiquetado), o hacer que la distribución de las observaciones dependa de las acciones guiadas por la política, la cual en si misma es el sujeto de aprendizaje.

2.2.5. Q-learning

Q-learning fue un algoritmo desarrollado por Chris Watkins en su tesis doctoral [26] (1989). Un proceso de decisión de Markov (MDP, por sus siglas en inglés) es un modelo matemático utilizado en aprendizaje por refuerzo, en el cual un agente toma decisiones secuenciales en un entorno incierto. En el trabajo realizado por Chris Watkins se introduce la programación dinámica incremental para aprender a controlar un MDP sin necesidad de modelar las matrices de transiciones y recompensas, eliminando la limitación que los métodos de programación dinámica tradicionales presentan.

El algoritmo de Q-learning optimiza directamente la función de valor q para aproximar q^*. La convergencia de este algoritmo requiere que todas las parejas estado-acción sean actualizadas durante el proceso de entrenamiento. Para asegurar esto se hace uso de una política ε-greedy.

2.2.5.1. Política ε-greedy

La política ε-greedy asegura que se haga una exploración de nuevas acciones en un estado dado mientras que al mismo tiempo se haga uso de la experiencia de aprendizaje que se ha obtenido hasta el momento. Para hacerlo se selecciona una acción aleatoria con una probabilidad de ε, y la mejor acción de acuerdo a la función de valor en caso contrario. A esta política también se le aplica un factor de descuento en el tiempo, causando que a medida que el agente va dando más pasos tienda a elegir con una mayor probabilidad la mejor acción de acuerdo con la función de valor.

2.2.5.2. Algoritmo Q-learning

El algoritmo de Q-learning mejora progresivamente la función de valor estado-acción después de una inicialización aleatoria por un número dado de episodios. En cada paso que el agente realiza, selecciona una acción basada en la política ε-greedy, y usa una taza de aprendizaje α para actualizar la función de valor de la siguiente manera:

$$Q(S_t, A_t) \longleftarrow Q(S_t, A_t) + \alpha(R_t + \gamma max_a Q(S_{t+1}, a) - Q(S_t, A_t)) \tag{2.3}$$

En la ecuación anterior γ representa el factor de descuento aplicado a la función de valor para el siguiente estado y α es la tasa de aprendizaje. El usar la estimación de la función de valor para mejorar su misma estimación se conoce como *bootstrapping* [27].

2.2.6. Deep RL

La función de valor estado-acción descrita por la ecuación 2.3 nos permite calcular la recompensa a largo plazo en un ambiente con estados y acciones discretas. Sin embargo, aunque las acciones del agente dentro de nuestro ambiente de compra y venta de activos son discretas, el conjunto de estados es continuo, lo cual nos impide hacer uso esta función en su versión original. De este modo, en esta sección se va a explicar como modificar la función de valor teniendo estados continuos y manteniendo las acciones discretas.

El hecho de que los estados sean continuos implica que no podemos simplemente tabular en un arreglo parejas estado-acción con su respectivo valor. En su lugar, necesitamos aproximar q^* usando una red neuronal (NN), generando así una *deep Q-network*.

2.2.6.1. Aproximación de la función de valor usando redes neuronales

La aproximación de la función Q implica el aprendizaje de un mapeo parametrizado continuo de ejemplos de entrenamiento. Particularmente, las redes neuronales han mostrado unos buenos resultados en la aproximación de funciones de valor. Sin embargo, usar técnicas de aprendizaje automático en un contexto de aprendizaje por refuerzo trae consigo un conjunto de retos, especialmente cuando los datos son generados por medio de la interacción entre el modelo y el ambiente usando una política. Algunos de estos retos son:

- Ya que los estados son continuos, el agente no va a poder visitar la mayor cantidad de estados posibles, lo cual crea la necesidad de generalizar.
- En el contexto de aprendizaje por refuerzo solo se obtiene un ejemplo por cada paso que el agente realiza, por lo cual el aprendizaje tiene que ocurrir en linea. En contraste, en el aprendizaje supervisado se generaliza basado en un conjunto de ejemplos independientemente distribuidos que son representativos y están correctamente etiquetados.
- Los ejemplos pueden estar altamente correlacionados cuando los estados secuenciales son similares.

2.2.6.2. Algoritmo Deep Q-Learning

El algoritmo Deep Q-learning es una adaptación del algoritmo Q-learning que nos permite hacer uso de estados continuos en nuestro ambiente. Su objetivo es estimar el valor de las acciones disponibles para un estado dado usando una red neuronal profunda.

El algoritmo Deep Q-learning aproxima la función de valor aprendiendo un conjunto de pesos θ de una red Q profunda (DQN) que relaciona estados con acciones de tal forma que

$$q(s, a, \theta) \approx q^*(s, a)$$

El algoritmo aplica gradiente descendente estocástico basado en una función de pérdida que calcula la diferencia al cuadrado entre el objetivo estimado por la DQN y la predicción del valor Q de la pareja estado-acción actual. La ecuación 2.5 expresa esta función de pérdida.

$$y_i = \mathbb{E}[r + \gamma max_{a'} Q(s', a'; \theta_{i-1}|s, a)] \tag{2.4}$$

$$L_i(\theta_i) = (y_i - Q(s, a; \theta))^2 \tag{2.5}$$

Podemos notar que en la función de pérdida tanto el objetivo como la estimación actual dependen de los pesos de la DQN. Esto es precisamente lo que lo diferencia del aprendizaje supervisado, donde los objetivos son fijos antes del entrenamiento.

Para explorar el espacio de parejas estado-acción, el agente usa una política ε-greedy que selecciona una acción aleatoria con probabilidad ε y sigue su política óptima que selecciona la acción con el valor q más alto en caso contrario. Esto asegura que el agente obtenga una experiencia que no esté ligada solo con su política óptima, sino también con un componente de exploración.

2.2.6.3. Adaptaciones de la arquitectura DQN

La arquitectura DQN ha ido adoptando a una serie de cambios que permiten que el proceso de aprendizaje sea más efectivo. A continuación se explican algunas de estás adaptaciones expuestas por S. Jansen [25].

- **Reproducción de experiencia**: Guarda una historia del estado, acción, recompensa y estado siguiente que el agente va experimentando cada vez que da un paso en el ambiente. Aleatoriamente toma mini-batches de su misma experiencia para actualizar los pesos de la red. Al tener una historia de la experiencia previa del agente, se incrementa la eficiencia de muestreo y se reduce la correlación de los ejemplos que se toman durante el aprendizaje en linea.

- **Red objetivo**: Cómo se mencionó en la sección anterior, la red Q es la encargada de estimar tanto los objetivos y_i (2.4), como también los valores de la pareja estado-acción actual $Q(s, a; \theta)$. En su lugar, lo que busca esta adaptación es decorrelacionar el proceso de aprendizaje separando estás dos estimaciones en redes distintas. Ahora va a existir otra red, llamada **red objetivo**, que tiene la misma arquitectura de la red Q, pero sus pesos θ^- son actualizados periódicamente cada τ pasos copiándolos desde la red Q y manteniéndolos constantes en caso contrario. De esta manera, esta nueva red genera las predicciones de los objetivos, tomando el lugar de la red Q para estimar 2.4.

- **Doble deep Q-learning**

  Q-learning tiende a sobreestimar los valores de las acciones ya que toma la estimaciones máximas de sus valores. Con el fin de desligar la estimación de los valores de las acciones de su misma selección, el algoritmo Double DQN (DDQN) usa dos redes con pesos distintos, una para seleccionar la mejor acción dado el estado siguiente, y otra para dar el valor estimado correspondiente. De esta manera, la ecuación 2.4 se convierte en:

$$y_i = \mathbb{E}[r + \gamma Q(s', argmax_{a'} Q(S_{t+1}, a, \theta_t); \theta'_t)] \tag{2.6}$$

  Para que los pesos de las dos redes difieran, se va a hacer uso de la **red objetivo** para proporcionar θ'_t

2.2.7. Análisis técnico

En el capítulo 3 se da una explicación de los datos recolectados que van a alimentar a los agentes de aprendizaje durante su entrenamiento. Entre las características de este conjunto de datos se incluyen:

- *open*: Primer precio al cual se llevó a cabo una transacción después de que el mercado abriera.
- *close*: Último precio de transacción antes de que el mercado cerrara para un período dado.
- *high*: Valor máximo que alcanzó la acción durante un período de tiempo.
- *low*: Valor más bajo que alcanzó la acción durante un período de tiempo.
- *volume*: Número total de acciones o contratos negociados de una acción en particular durante un período específico.

Sin embargo, estas características no le dan suficiente información a los agentes para que puedan tener una visión más amplia durante cada paso que realizan en el ambiente, teniendo en cuenta que durante cada paso solo tendrán acceso a un registro del conjunto de datos, el cual representará *su estado actual*.

Es por ello que para este el desarrollo de este proyecto se ha decido hacer uso de análisis técnico, específicamente centrándose en **indicadores técnicos**. Tal y como es expuesto por S. Donadio y S. Ghosh [28], los indicadores técnicos se basan en cálculos matemáticos para pronosticar la dirección del mercado financiero. De esta manera, cuándo el agente se encuentre en un punto dado en el ambiente (i.e., su estado actual), va a tener características que lo ayuden a tener un pronóstico más amplio para la toma de decisiones. Existe una larga lista de indicadores técnicos, categorizados en 8 tipos (e.g., volatilidad, volumen, momentum, ...) y 4 clases (e.g., acumulativos, índices, osciladores y de superposición). Sin embargo, en las secciones siguientes solo se explicarán de manera general los indicadores técnicos que se decidió usar en este proyecto, entre los cuales se incluyen indicadores de *momentum*, *volatilidad* y *volumen*.

2.2.7.1. Relative Strength Indicator (RSI)

El indicador *RSI* es un *oscilador de momentum* que mide la velocidad y cambio de los movimientos de precio [29]. Está basado en los cambios de precio a lo largo de periodos fijos para capturar la fuerza/magnitud sus movimientos. Su oscilación varia entre 0 y 100. Se considera que el RSI está sobrecomprado cuando está por encima de 70 y sobrevendido cuando está por debajo de 30. En la ecuación 2.7 se encuentra la formula para calcular este indicador. En esta ecuación se calcula la magnitud del promedio de incrementos (ganancias) de precio a lo largo de un periodo, así como también la magnitud del promedio de decrementos (pérdidas) a lo largo del mismo periodo [28]. Luego se normaliza este valor entre 0 y 100, capturando si hubieron muchas ganancias con respecto a pérdidas, o muchas pérdidas con respecto a ganancias.

$$\text{RSI} = 100 - \left(\frac{100}{1 + \frac{\text{Average of Upward Price Change}}{\text{Average of Downward Price Change}}}\right) \tag{2.7}$$

En la figura 2.3 podemos observar el cálculo de este indicador para los precios de cierre de un activo cualquiera. Aquí se puede notar que en un periodo de tiempo el indicador está por debajo de 30, lo cual indica que ocurrió una sobreventa del activo (i.e., una gran cantidad de personas tomó una posición corta); y en otro periodo de tiempo el indicador está por encima de 70, indicando que ocurrió una sobrecompra (i.e, una gran cantidad de personas tomó una posición larga).

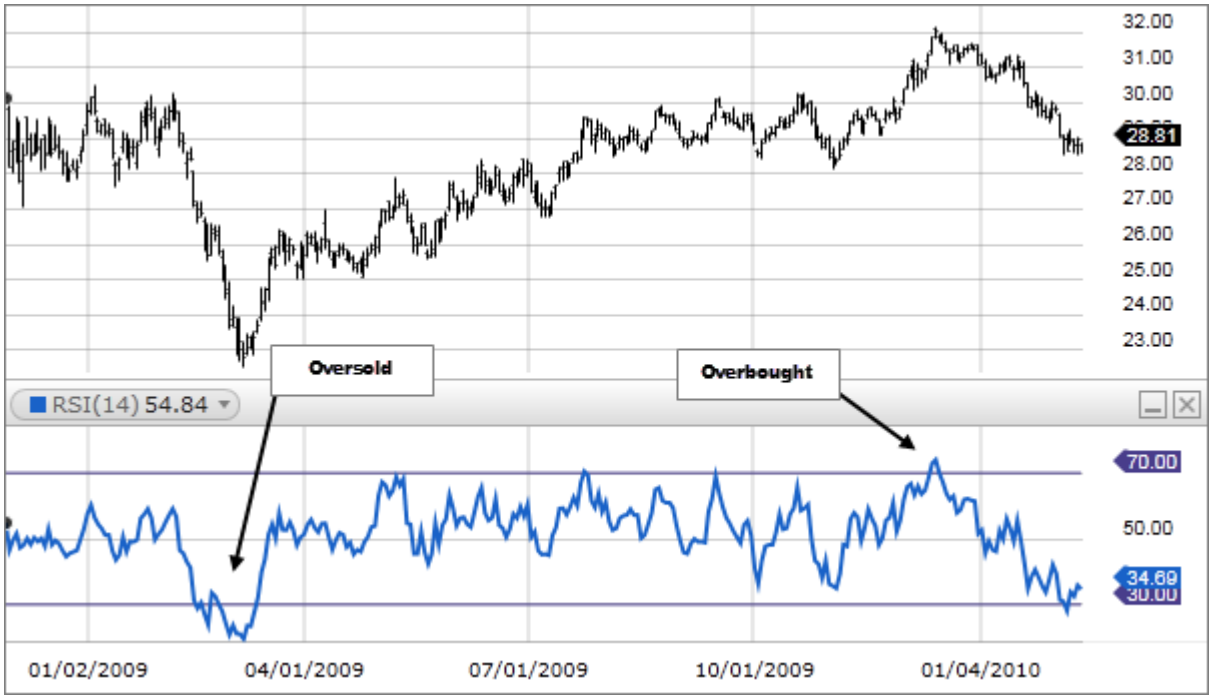


Figura 2.3: Cálculo del RSI [29]

2.2.7.2. Moving Average Convergence/Divergence (MACD)

El indicador *MACD* es un *oscilador de momentum* usado principalmente para realizar operaciones de trading basadas en tendencias [30]. Aunque este indicador también es un oscilador, generalmente no es usado para identificar condiciones de sobreventa o sobrecompra como lo hace el RSI. Este se basa en la clase de indicadores que se construyen usando la media móvil de los precios de cierre de un activo. Más específicamente, el calculo de la señal de salida del indicador *MACD* proviene de aplicar una media móvil exponencial suavizada a la diferencia entre una media móvil exponencial rápida (EMA_{fast}) y una media móvil exponencial lenta (EMA_{slow}). Adicionalmente, si calculamos la diferencia entre el indicador $MACD$ y la media móvil de sus valores ($MACD_{signal}$), podemos visualizarlo como un histograma. Un indicador $MACD$ que se adapta muy bien para capturar la dirección, magnitud y duración de la tendencia del precio de un activo, generalmente se calcula al substraer una media móvil exponencial de 26 periodos (EMA_{slow}) de una media móvil exponencial de 12 periodos (EMA_{fast}). Las ecuaciones 2.8, 2.9 y 2.10 muestran la forma de calcular el indicador $MACD$, su media móvil exponencial, y el histograma de su diferencia, respectivamente.

$$\text{MACD} = APO = \text{EMA}_{fast} - \text{EMA}_{slow} \tag{2.8}$$

$$\text{MACD}_{signal} = \text{EMA}_{MACD} \tag{2.9}$$

$$\text{MACD}_{histogram} = \text{MACD} - \text{MACD}_{signal} \tag{2.10}$$

En la figura 2.4 podemos observar los gráficos que generan estos cálculos para los precios de un activo.

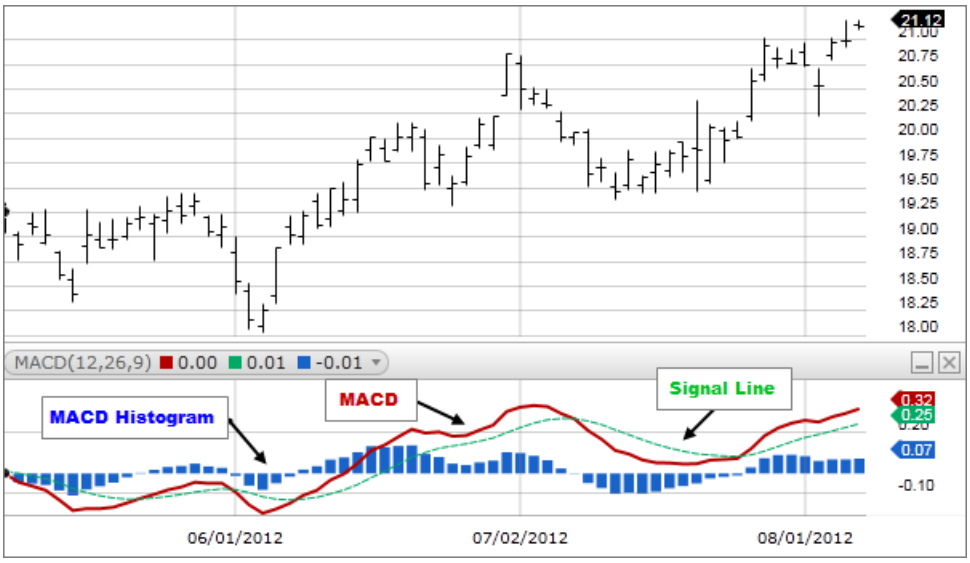


Figura 2.4: Cálculo del indicador MACD [30]

Este indicador nos proporciona información importante acerca de la subida o caída del precio de un activo. A continuación se muestran diferentes formas en que se puede interpretar.

- Si el MACD cruza de negativo a positivo, se considera como una tendencia de subida del precio del activo. En su lugar, si cruza de positivo a negativo, se considera como una tendencia de caída. Cuando el MACD sube mientras se encuentra por debajo de cero, se considera como un tendencia de subida en el precio; y cuando cae mientras se encuentra por encima de cero, se considera como como una tendencia de bajada.

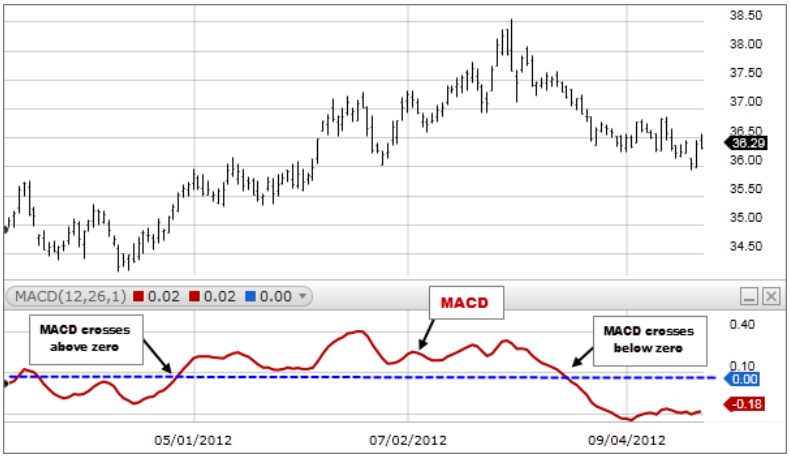


Figura 2.5: MACD [30]

- Cuando la linea del MACD cruza de abajo a arriba la linea de la señal, se considera como una tendencia de subida. Cuanto más por debajo de la línea cero, más fuerte es la señal.
- Cuando la linea del MACD cruza de arriba a abajo la linea de la señal, se considera como una tendencia de bajada. Cuanto más por encima de la línea cero, más fuerte es la señal.

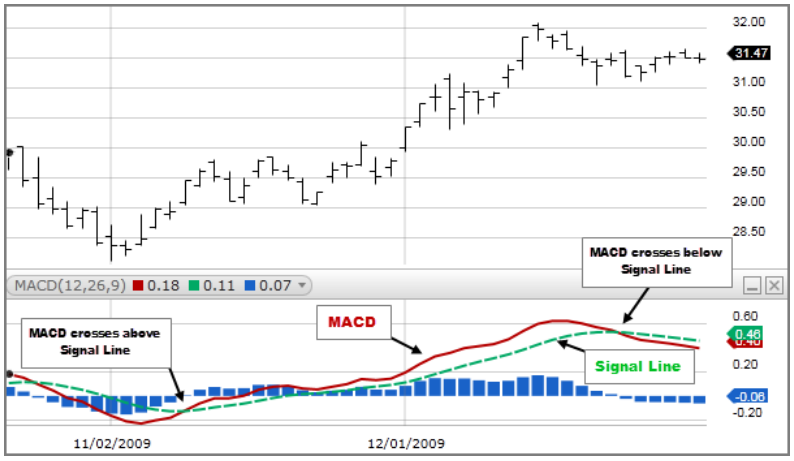


Figura 2.6: MACD [30]

2.2.7.3. Average True Range (ATR)

El *ATR* indica el promedio de rangos verdaderos para un periodo dado. Este es un indicador de *volatilidad*, el cual es medido tomando en cuenta los espacios o diferencias en el movimiento del precio del activo [31]. Para medir una volatilidad a largo plazo, generalmente se usan periodos de 20 a 50 días/semanas/meses (dependiendo de la recurrencia de los precios con los que está siendo calculada). En su lugar, para medir volatilidades a corto plazo, se usan promedios más cortos con periodos de 2 a 10 días/semanas/meses. En la ecuación 2.11 se muestra la formula para calcular este indicador

$$\text{ATR} = \frac{\text{ATR previo} * (n-1) + \text{TR}}{n}, \tag{2.11}$$

donde,

- TR: Rango verdadero
- n: número de periodos
- ATR: Rango verdadero promedio

El rango verdadero (TR) para un día dado se calcula como el valor máximo de:

- *high actual* − *low actual*
- |*high actual - close previo*|
- |*low actual - close previo*|

En la figura 2.7 se puede observar el gráfico generado al calcular este indicador para un activo dado. Un valor de ATR bajo indica una serie de periodos con rangos pequeños (i.e., días tranquilos). Un periodo prolongado de valores ATR bajos puede indicar una zona donde el precio se consolida y la probabilidad de un patrón de continuación. Por otro lado, un ATR en expansión indica un aumento de la volatilidad del activo, con el rango de cada barra haciéndose cada vez más grande. Este indicador no es direccional, por lo cual su aumento puede indicar una presión de compra o una presión de venta. Es poco probable que los valores altos del ATR se mantengan durante periodos prolongados.

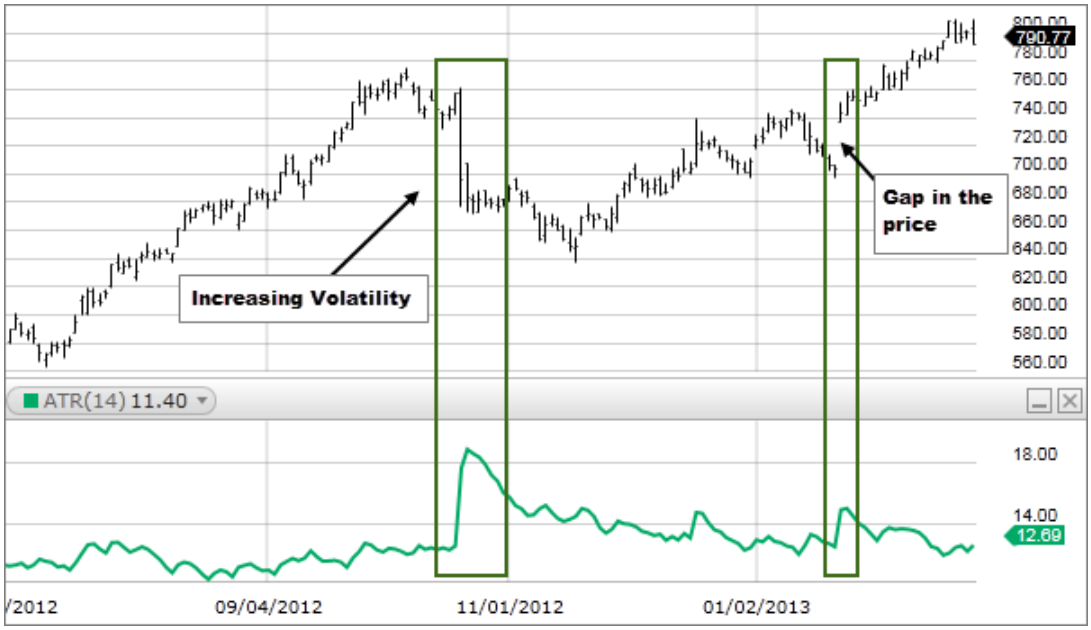


Figura 2.7: Average True Range [31]

2.2.7.4. Slow Stochastic (STOCH)

El indicador *STOCH* es un *indicador de momentum* que muestra la posición del precio de cierre con respecto al rango high-low durante un número determinado de periodos. Este indicador puede oscilar entre 0 y 100 [32]. El precio de cierre tiende a cerrar cerca del máximo en una tendencia de alza y cerca del mínimo en una tendencia de baja. Si el precio de cierre se aleja del mínimo o máximo, significa que su *momentum* está disminuyendo.

En la figura 2.8, el área por encima de 80 indica una región de sobre compra, mientras que el área por debajo de 20 es considerada una región de sobre venta. Cuando este oscilador está por encima de 80 y luego cruza por debajo de este límite, se puede considerar como una señal de venta. Del mismo modo, cuando este oscilador está por debajo de 20 y luego cruza por encima de este límite, se puede considerar como una señal de compra. Los límites de 20 y 80 son los que normalmente se usan, pero pueden ser ajustados de acuerdo a las necesidades.

2.2.7.5. Ultimate Oscillator (ULTOSC)

El indicador *ULTOSC* usa 3 periodos de tiempo diferentes (7, 14 y 28) para representar tendencias del mercado a corto, mediano y largo plazo [33]. Una señal de compra ocurre cuando se produce una divergencia de alza, tal y como se muestra en la figura 2.9. Del mismo modo, una señal de venta ocurre cuando se produce una divergencia de baja, tal y como se muestra en la figura 2.10.

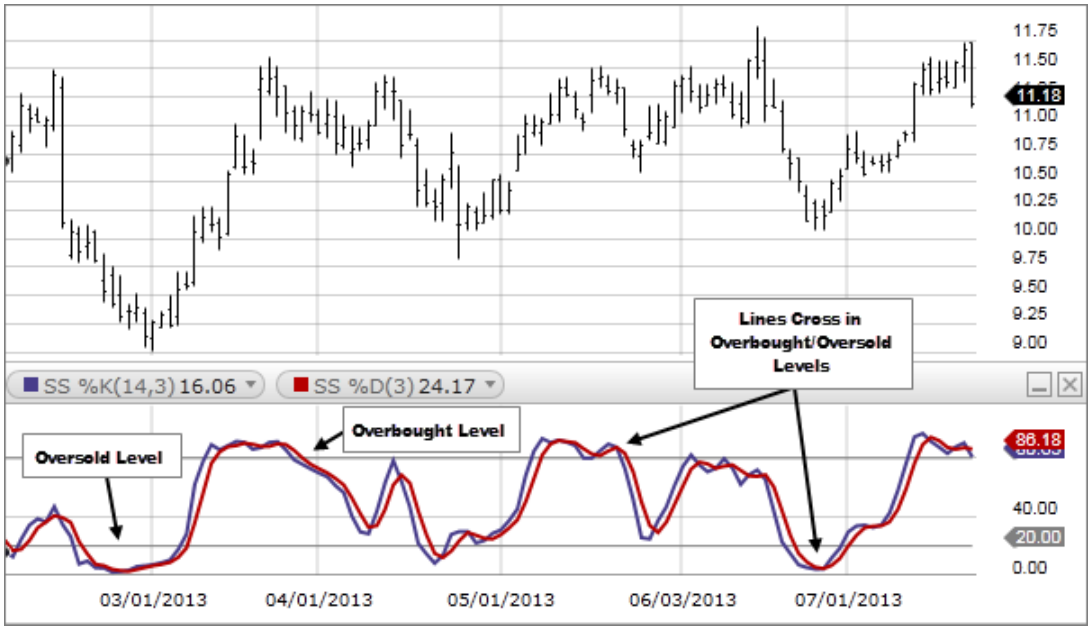


Figura 2.8: Slow Stochastic [32]

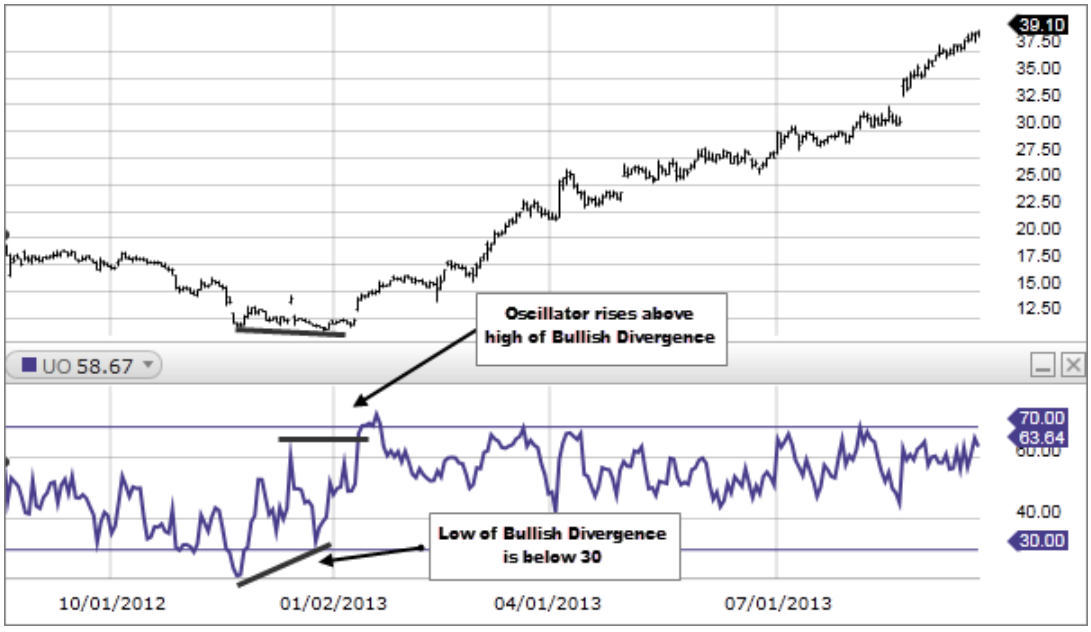


Figura 2.9: Ultimate Oscillator [33]

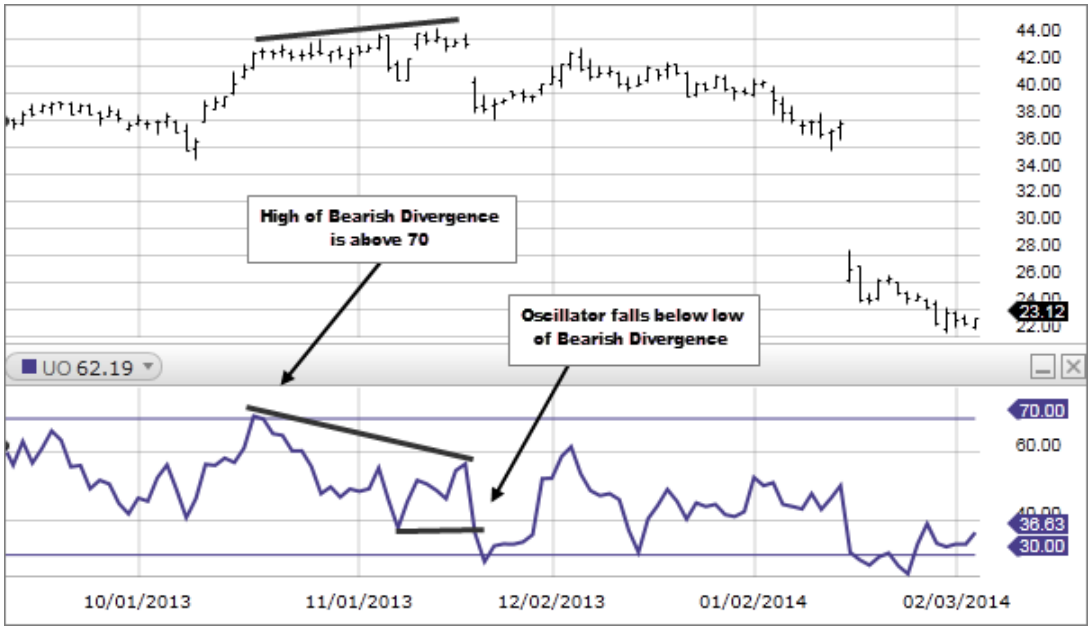


Figura 2.10: Ultimate Oscillator [33]

2.2.7.6. Bollinger Bands (BBANDS)

El indicador *BBANDS* crea una banda superior e inferior a un nivel de desviación estándar por encima y por debajo de una media móvil simple del precio. Esta banda representa la volatilidad esperada de los precios, tratando la media móvil como precio de referencia [28] [34]. En la figura 2.11 se puede observar la creación de las bandas para una media móvil de 20 periodos.

Cuando las bandas se estrechan durante un período de baja volatilidad, aumenta la probabilidad de que se produzca un movimiento brusco del precio en cualquier dirección. Esto puede iniciar un movimiento de tendencia. Por otro lado, cuando las bandas se separan por una cantidad inusualmente grande, la volatilidad aumenta y cualquier tendencia existente puede estar terminando. En la figura 2.12 podemos observar estos comportamientos.

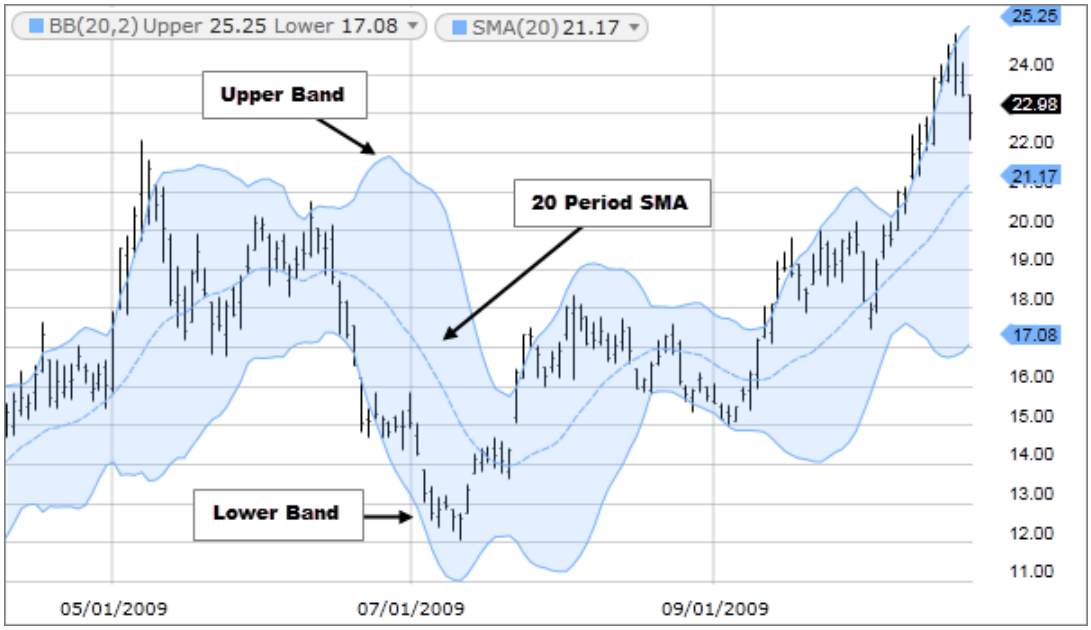


Figura 2.11: Bollinger Bands [34]

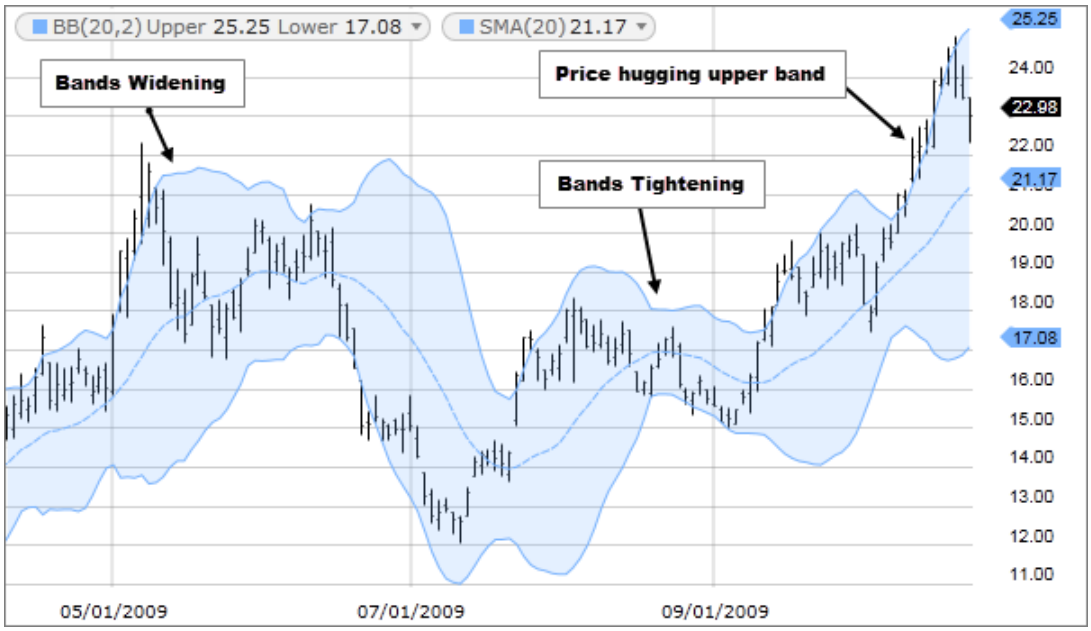


Figura 2.12: Bollinger Bands [34]

2.2.7.7. On Balance Volume (OBV)

El indicador *OBV* mide la presión de compra y venta como un indicador acumulativo que agrega volumen en los días de alza y resta volumen en los días que se presente una baja. Cuando el valor cierra más alto que el cierre anterior, todo el volumen del día se considera volumen al alza. Cuando el valor cierra por debajo del cierre anterior, todo el volumen del día se considera como volumen de bajada [35]

- Cuando tanto el precio como el OBV alcanzan máximos y mínimos más altos, es probable que la tendencia de alza continúe.
- Cuando tanto el precio como el OBV registran máximos y mínimos más bajos, es probable que continúe la tendencia de baja.

En la figura 2.13 se puede observar una tendencia de baja en el precio.

Figura 2.13: On Balance Volume [35]

2.3. Trabajos Relacionados

La relación entre el aprendizaje por refuerzo y el trading ha captado considerable atención en la literatura reciente debido a su potencial para revolucionar los procesos de toma de decisiones financieras. Un aspecto fundamental de esta investigación radica en la aplicación del Doble Aprendizaje Profundo Q (DDQN) para entrenar agentes de trading autónomos. Mnih et al. [36] fueron pioneros en el uso de DDQN, demostrando su eficacia para mejorar la estabilidad y convergencia de

los modelos de aprendizaje profundo por refuerzo. Basándonos en este fundamento, nuestro proyecto aprovecha DDQN para entrenar un agente de trading dentro de un entorno diseñado a medida.

Una gran parte del trabajo realizado en este campo de estudio, incluyendo el de Lillicrap et al. [37] y Haarnoja et al. [38], ha explorado la adaptabilidad de los agentes de aprendizaje por refuerzo a los mercados financieros. Lillicrap et al. [37] introdujeron el concepto de políticas deterministas, mejorando la precisión de la toma de decisiones en entornos volátiles. De manera similar, Haarnoja et al. [38] extendieron estas ideas, enfatizando la importancia de la eficiencia de muestras en el entrenamiento de agentes de aprendizaje por refuerzo para aplicaciones del mundo real. Estas contribuciones destacan la búsqueda continua de algoritmos para aumentar la eficacia de los agentes de trading.

Al personalizar nuestro entorno de trading, nos inspiramos en los trabajos de Jiang et al. [39] y Powell et al. [40], quienes destacaron la importancia de adaptar los entornos a los desafíos específicos de los mercados financieros. Jiang et al. [39] optaron por la inclusión de costos de transacción e impacto en el mercado en el entorno de entrenamiento, reconociendo la naturaleza del trading en configuraciones realistas. Powell et al. [40] ampliaron esto al introducir el concepto de microestructura del mercado, enfatizando el papel de la dinámica del libro de órdenes en la formación del comportamiento del agente. Nuestro entorno diseñado a medida sigue estos principios, capturando las complejidades de escenarios de trading del mundo real.

Además, la comparación del rendimiento de nuestro agente con una estrategia de buy-and-hold se alinea con los hallazgos de Ganti et al. [41] y Kearns et al. [42], quienes evaluaron la efectividad de los agentes de aprendizaje por refuerzo para superar estrategias de referencia. Ganti et al. [41] exploraron la dinámica del trading en diversas condiciones del mercado, enfatizando la importancia de estrategias que se adapten al mercado. Kearns et al. [42] contribuyeron abordando los desafíos de costos de transacción, ofreciendo ideas sobre la evaluación realista de agentes de trading. Nuestro análisis comparativo con una estrategia de buy-and-hold resuena con estos estudios, proporcionando un punto de referencia para evaluar la eficacia de nuestro agente entrenado con DDQN.

En el artículo Deep Reinforcement Learning for Automated Stock Trading: An Ensemble Strategy [24] se plantea una solución que emplea una estrategia de trading de aprendizaje por refuerzo profundo conjunto que incluye tres algoritmos basados en actores críticos: Proximal Policy Optimization (PPO), Advantage Actor Critic (A2C), y Deep Deterministic Policy Gradient (DDPG). El rendimiento del agente es evaluado usando Sharpe ratio, como también comparandolo con el indice Dow Jones Industrial Average y la estrategia de asignación de cartera de min-variance. La Figura 2.14 muestra un esquema general de los agentes y el ambiente, donde los agentes, basados en observaciones del ambiente y una recompensa (ganancia o pérdida), pueden llevar a cabo las acciones de vender, mantener o comprar una determinada acción. Las Figuras 2.15 y 2.16 muestran algunos de los resultados obtenidos en este estudio, tomando el periodo desde el 04/01/2016 hasta el 08/05/2020. Por un lado, en la Figura 2.15 podemos observar una tabla comparativa entre el método conjunto implementado (PPO, A2C y DDPG) y cada método de forma individual. Por otro lado, la Figura 2.16 muestra la retribución acumulada con costo de transacción del método implementado, también comparándolo con los otros métodos.

Por otro lado, en el artículo Global Stock Market Prediction Based on Stock Chart Images Using

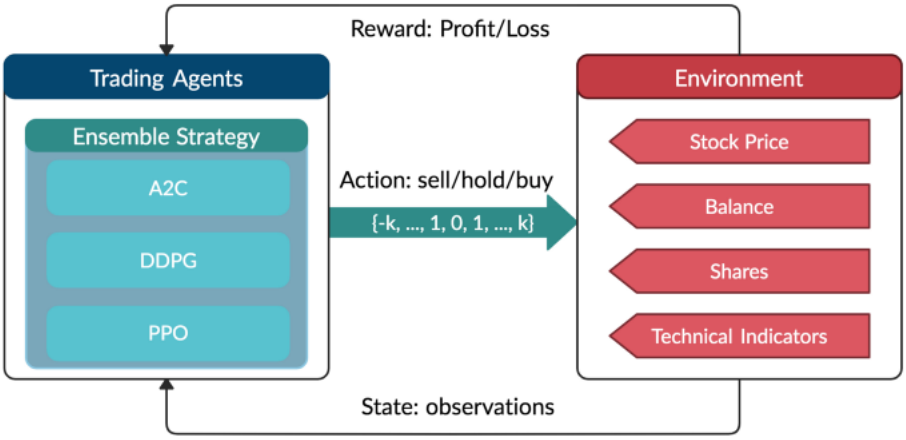


Figura 2.14: Esquema de los agentes de trading y el ambiente [24]

(2016/01/04-2020/05/08)	Ensemble (Ours)	PPO	A2C	DDPG	Min-Variance	DJIA
Cumulative Return	70.4%	83.0%	60.0%	54.8%	31.7%	38.6%
Annual Return	13.0%	15.0%	11.4%	10.5%	6.5%	7.8%
Annual Volatility	9.7%	13.6%	10.4%	12.3%	17.8%	20.1%
Sharpe Ratio	1.30	1.10	1.12	0.87	0.45	0.47
Max Drawdown	-9.7%	-23.7%	-10.2%	-14.8%	-34.3%	-37.1%

Figura 2.15: Metrics comparison [24]

Deep Q-Network [43], se construyó un modelo en el cual se aplica Deep Q-Network, haciendo uso de una red neuronal convolucional como técnica de estimación. Esta red toma como entrada imágenes de gráficos de acciones, con el fin de hacer una predicción global del mercado de acciones.

En resumen, nuestro proyecto se basa en trabajos fundamentales en aprendizaje por refuerzo, incorporando técnicas DDQN para entrenar agentes de trading dentro de un entorno diseñado a la medida. La personalización de este entorno se inspira en ideas de estudios que enfatizan el realismo de simulaciones de mercados financieros. Además, nuestro análisis comparativo se alinea con investigaciones previas que evalúan el rendimiento de agentes de trading frente a estrategias de referencia establecidas. Estas contribuciones posicionan nuestro trabajo en la vanguardia de la aplicación de técnicas avanzadas de aprendizaje por refuerzo para un trading algorítmico más efectivo y adaptativo.

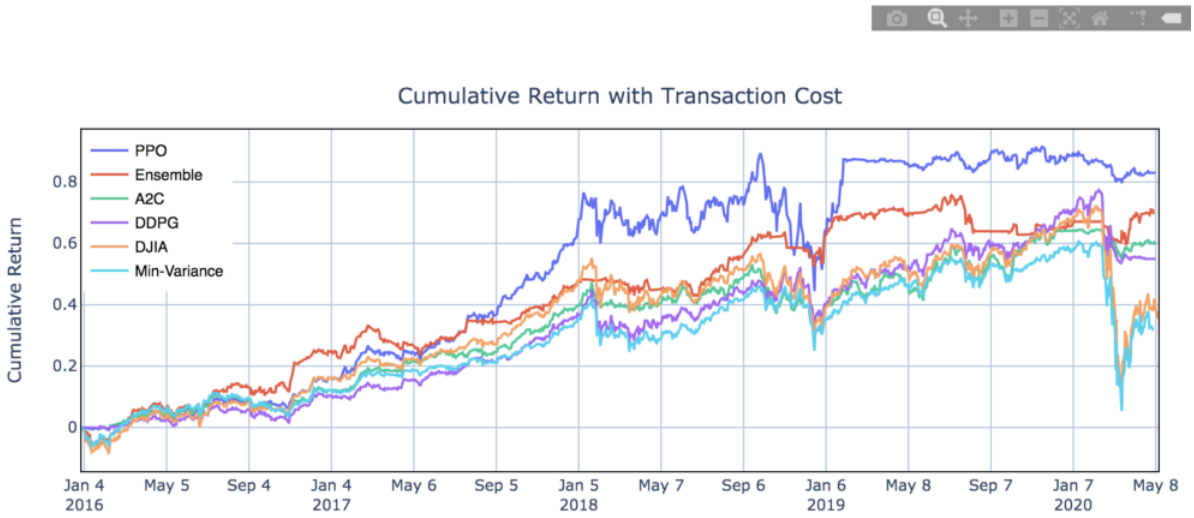


Figura 2.16: Cumulative Return with Trasaction Cost [24]

Capítulo 3

Conjunto de datos

3.1. Recolección de datos

Para este trabajo se hizo uso de un conjunto de datos de acciones estadounidenses que provee NASDAQ. Estos datos pueden ser obtenidos por medio de este enlace (*se requiere primero crear una cuenta para acceder*).

Esta fuente de datos ofrece precios de acciones, dividendos y divisiones para 3000 empresas estadounidenses que cotizan en bolsa de valores. Entre estas empresas se encuentran aquellas pertenecientes al índice S&P 500. Este conjunto de datos contiene datos desde el año 1962 hasta el año 2018 y es actualizado cada día de la semana a las 9:15 PM ET. Cabe precisar que en este conjunto de datos que NASDAQ nos provee, los precios ajustados están originalmente escalados. Esta transformación al conjunto de datos ayuda a que la diferencia entre los precios día con día se vea con mayor claridad.

Con el fin de optimizar la posterior lectura del conjunto de datos, este es almacenado en formato fast HDF como se muestra en el anexo A.1.

3.2. Especificación y entendimiento de datos

En la tabla 3.1 se describen las características del conjunto de datos. Este conjunto contiene registros históricos para más de 3000 activos, pero para propósitos de este proyecto se han seleccionado las acciones de 4 compañías pertenecientes al índice S&P 500, las cuales son Apple Inc (AAPL), Miscrosoft Corporation (MSFT), Amazon Inc (AMZN) y Pepsico Inc (PEP). En la tabla 3.2 se muestra, para cada activo, la capitalización de mercado actual, el año de la oferta pública inicial y el sector e industria a la que pertenece.

Feature	Non-null count	Type
Adj open	15388776 records	Float64
Adj high	15389259 records	Float64
Adj low	15389259 records	Float64
Adj close	15389313 records	Float64
Adj volume	15389314 records	Float64

Total columns: 5
Number of records: 15389314
Memory usage: 1.4+ GB

Cuadro 3.1: Características

Ticker	Name	Market cap	IPO year	Sector	Industry
AAPL	Apple Inc	2.88 trillion	1980	Technology	Computer Manufacturing
MSFT	Microsoft	2.44 trillion	1986	Technology	Computer Software: Prepackaged Software
AMZN	Amazon Inc	1.39 trillion	1997	Consumer Services	Catalog/Specialty Distribution
PEP	Pepsico Inc	249.06 billion	1972	Consumer Non-Durables	Beberages (Production/Distribution)

Cuadro 3.2: Acciones S&P 500

Una gran limitación que las características mostradas en la tabla 3.1 presentan es que están muy correlacionadas entre sí, lo cual hace prácticamente inviable que este conjunto de datos sea usado tal y como está para los entrenamientos. De igual manera, estas características no proporcionan una información suficiente a los agentes para que puedan tener un proceso de entrenamiento óptimo, teniendo en cuenta que cada uno de los registros representa el estado que el agente podrá observar en cada paso que realice dentro del ambiente. Por esta razón, se calculan diferentes indicadores técnicos con el propósito de disminuir la correlación del conjunto de datos, así cómo también proveer a los agentes una visión más amplia de un estado. En la sección 2.2.7 se presentan los indicadores técnicos incluidos en el conjunto de datos.

Las secciones 3.2.1, 3.2.2, 3.2.3 y 3.2.4 presentan de manera más detallada una descripción y entendimiento de los datos para cada uno de los activos.

3.2.1. Apple Inc

El año en que Apple Inc realizó la oferta publica inicial (IPO) de sus acciones fue en 1980. Ya que la frecuencia de recolección de los datos es diaria (cada día de la semana), se tiene un total aproximado de 9400 registros en total, hasta el 27 de marzo del 2018.

En las figuras 3.1 y 3.2 se pueden observar los precios ajustados de apertura, cierre, bajo y alto para este activo a lo largo la historia. Se puede notar que existe una correlación muy grande entre estos precios, lo cual también puede ser observado en la tabla de correlación de la figura 3.3. Por otro lado, en la figura 3.4 se puede observar el volumen histórico para este activo.

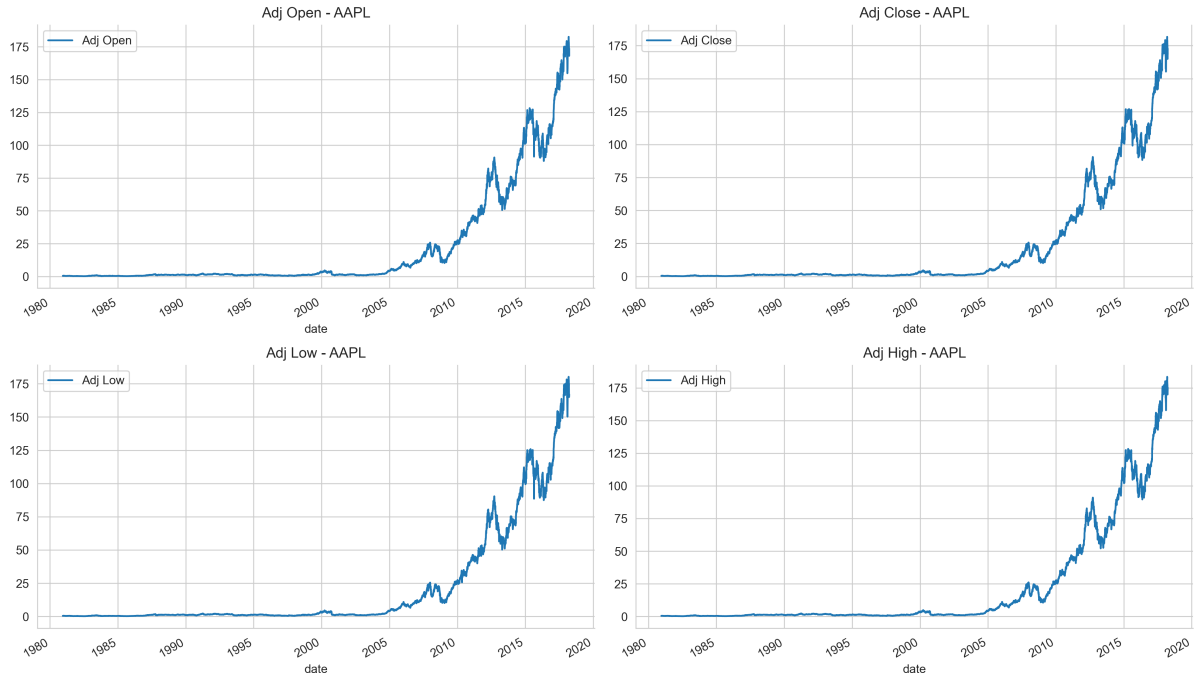


Figura 3.1: Precios históricos AAPL

En la tabla 3.3 se presenta el calculo de algunas medidas estadísticas para las diferentes características de este conjunto de datos.

	Adj open	**Adj close**	**Adj volume**	**Adj low**	**Adj high**
Count	9400	9400	9400	9400	9400
Mean	21,571019	21,567664	$8{,}860156 \times 10^7$	21,351252	21,774929
Standard deviation	39,272529	39,271266	$8{,}704777 \times 10^7$	38,942651	39,584888
Min	0,163495	0,161731	$2{,}503760 \times 10^5$	0,161731	0,163495
25 %	0,923453	0,922730	$3{,}461080 \times 10^7$	0,904096	0,940280
50 %	1,437461	1,437445	$6{,}069700 \times 10^7$	1,410762	1,468272
75 %	20,270182	20,294924	$1{,}109031 \times 10^8$	19,905845	20,565604
Max	182,590000	181,720000	$1{,}855410 \times 10^9$	180,210000	183,50000

Cuadro 3.3: Medidas estadísticas AAPL

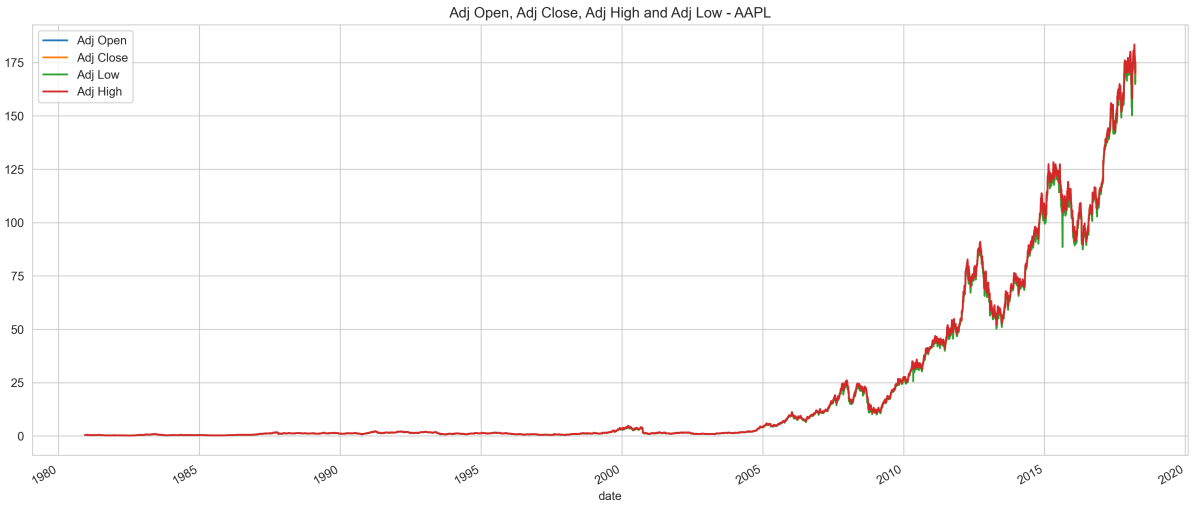


Figura 3.2: Precios históricos AAPL

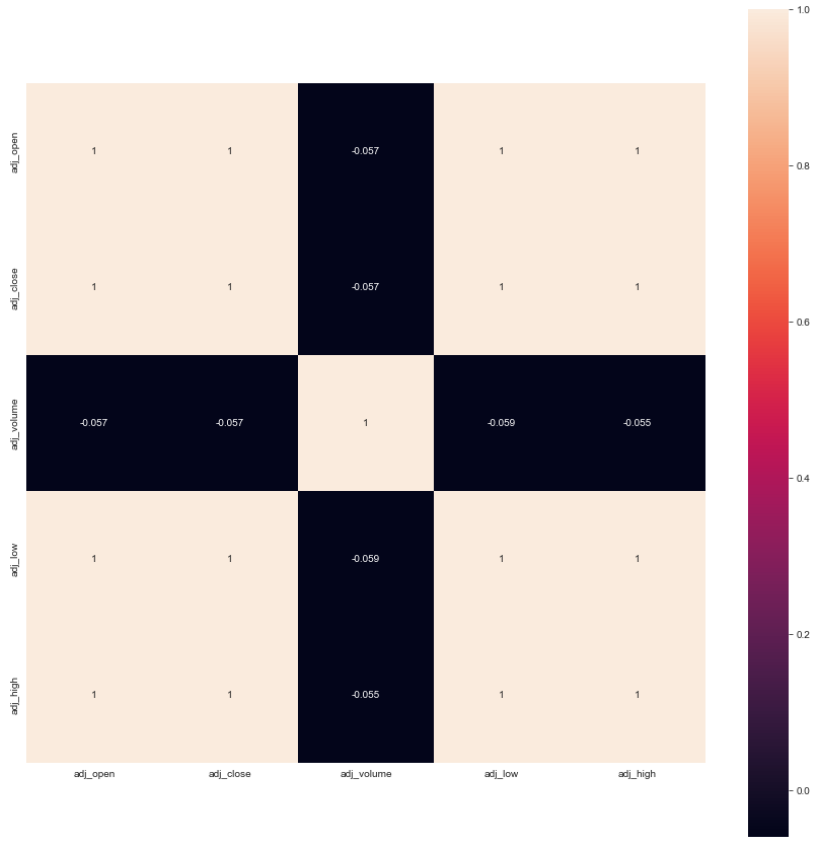


Figura 3.3: Correlación entre características AAPL

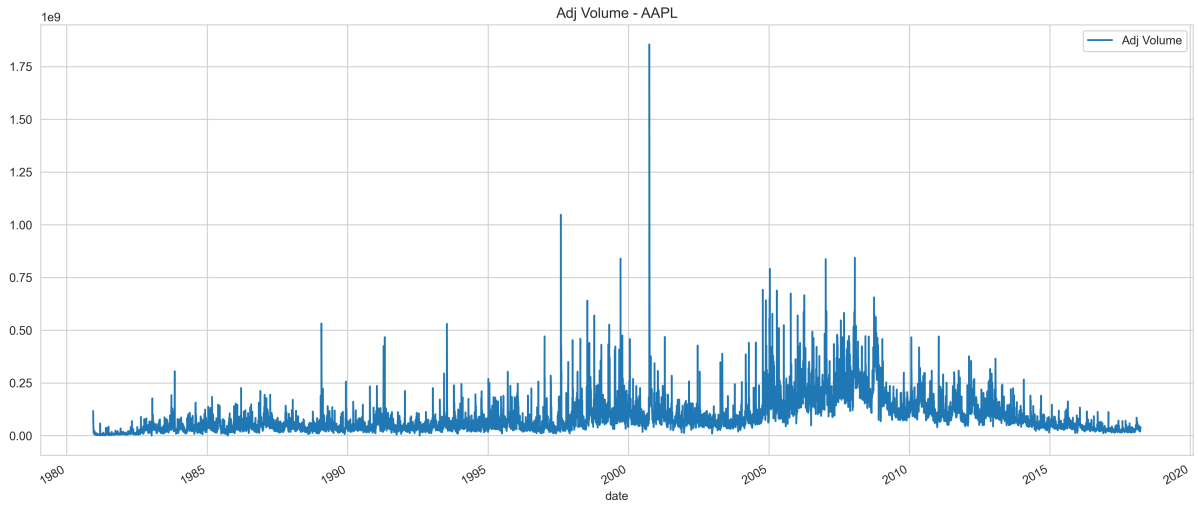


Figura 3.4: Volumen ajustado AAPL

3.2.2. Microsoft

El año en que Microsoft realizó la oferta pública inicial (IPO) de sus acciones fue en 1986. Ya que la frecuencia de recolección de los datos es diaria (cada día de la semana), se tiene un total aproximado de 8076 registros en total, hasta el 27 de marzo del 2018.

En las figuras 3.5 y 3.6 se pueden observar los precios ajustados de apertura, cierre, bajo y alto para este activo a lo largo la historia. Se puede notar que existe una correlación muy grande entre estos precios, lo cual también puede ser observado en la tabla de correlación de la figura 3.7. Por otro lado, en la figura 3.8 se puede observar el volumen histórico para este activo.

La tabla 3.4 presenta el calculo de algunas medidas estadísticas para las diferentes características de este conjunto de datos.

	Adj open	**Adj close**	**Adj volume**	**Adj low**	**Adj high**
Count	8076	8076	8076	8076	8076
Mean	18,763583	18,768430	$6{,}223490 \times 10^7$	18,564394	18,964604
Standard deviation	17,756714	17,760955	$3{,}909512 \times 10^7$	17,607812	17,901221
Min	0,058941	0,060097	$2{,}304000 \times 10^6$	0,058941	0,061253
25 %	1,908662	1,908662	$3{,}907406 \times 10^7$	1,882658	1,934665
50 %	18,581468	18,594782	$5{,}531360 \times 10^7$	18,366597	18,856998
75 %	24,045268	24,060766	$7{,}561875 \times 10^7$	23,802299	24,273922
Max	97,000000	96,770000	$1{,}031789 \times 10^9$	96,040000	97,240000

Cuadro 3.4: Medidas estadísticas MSFT

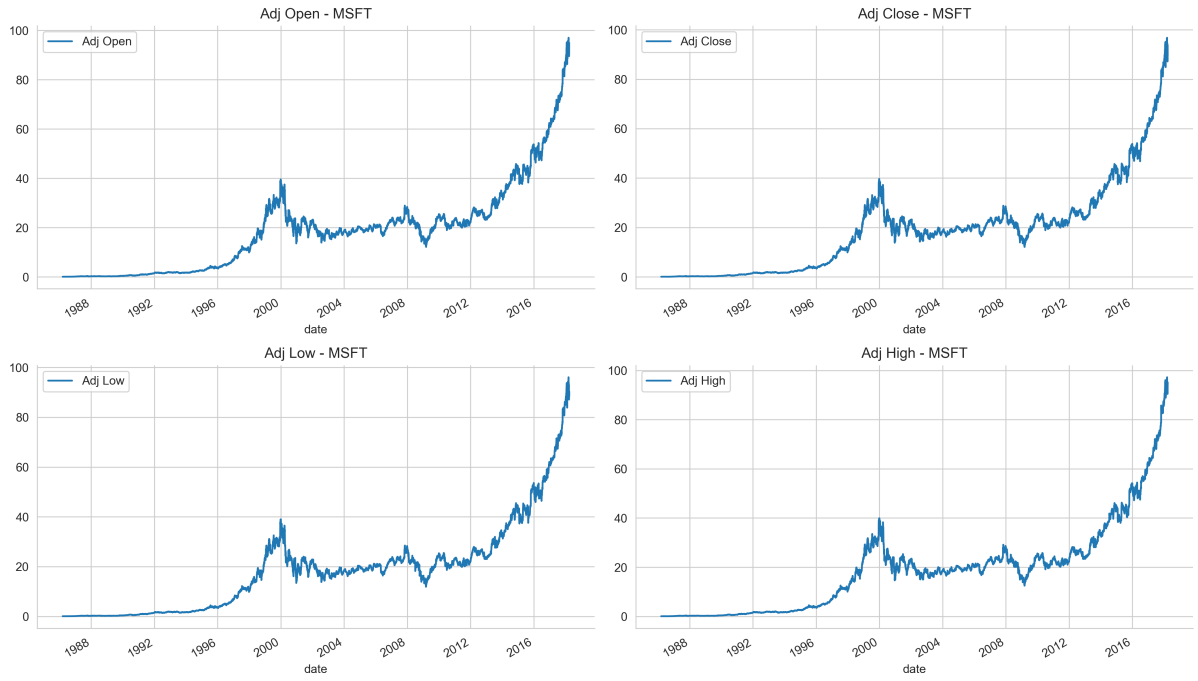


Figura 3.5: Precios históricos MSFT

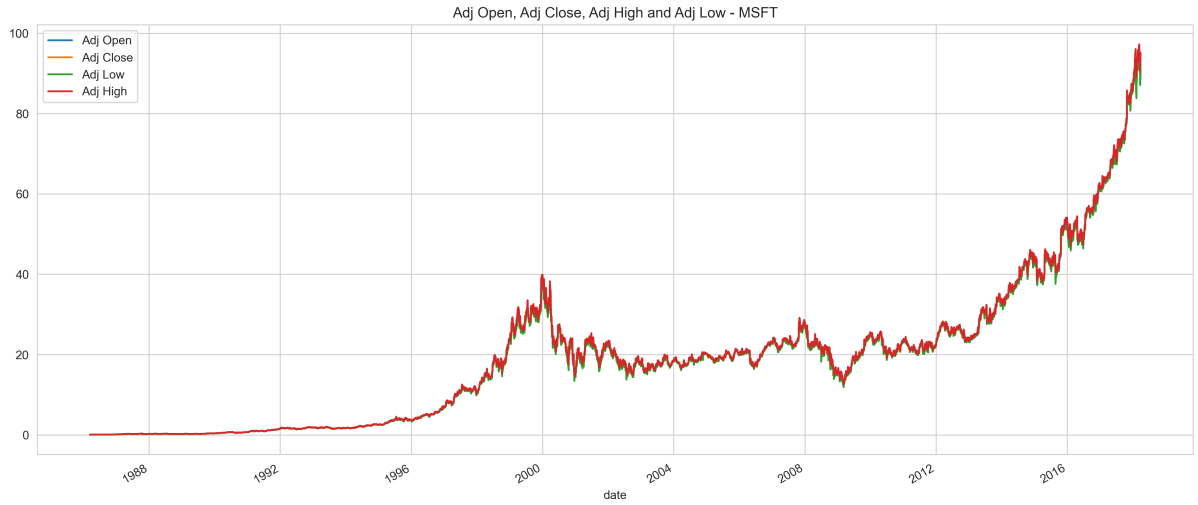


Figura 3.6: Precios históricos MSFT

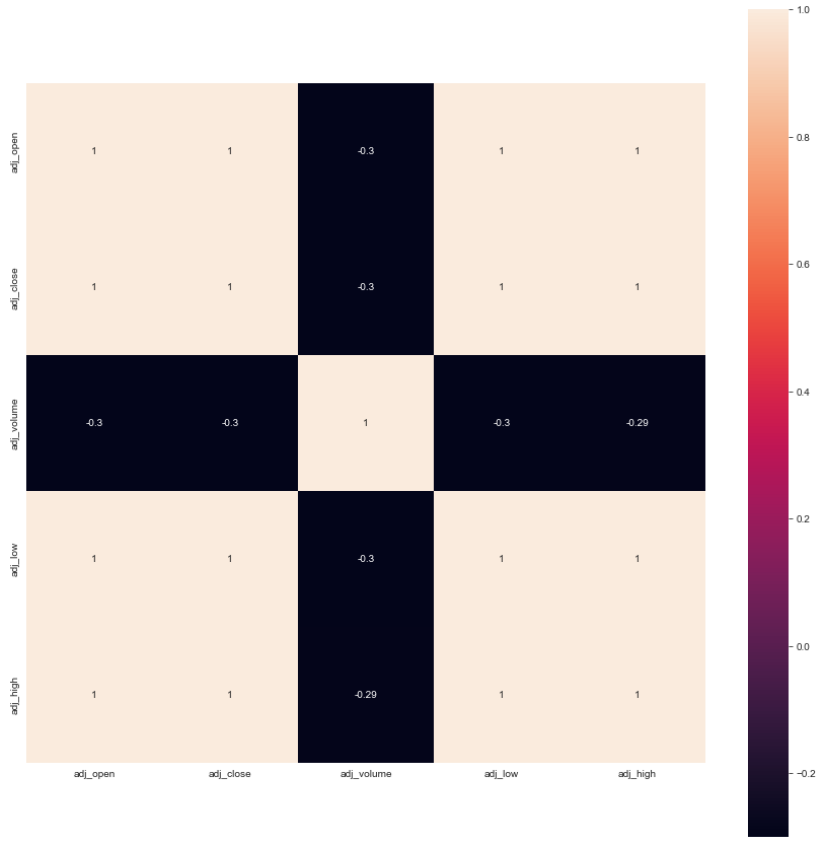


Figura 3.7: Correlación entre características MSFT

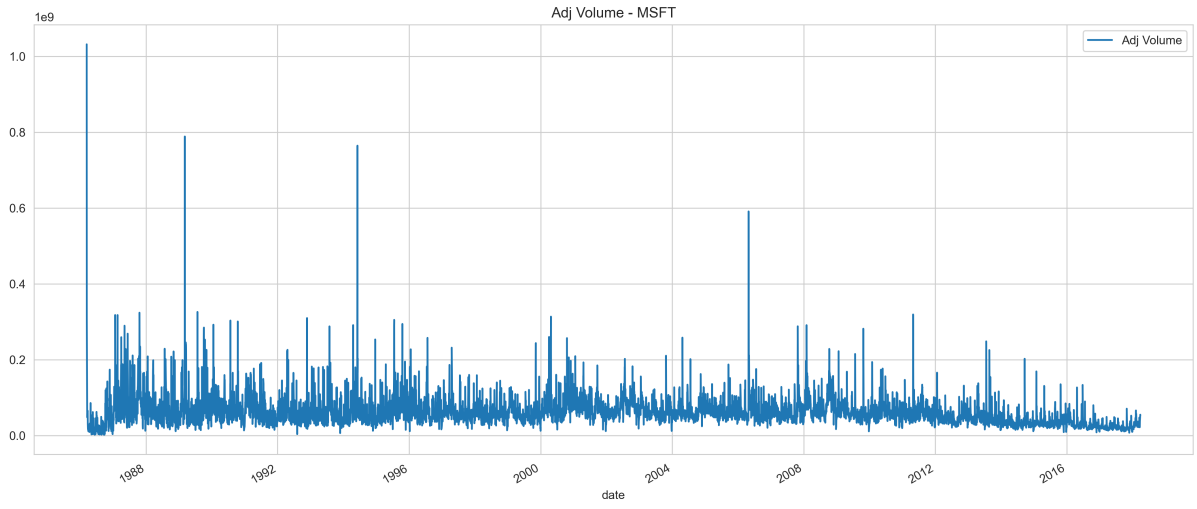


Figura 3.8: Volumen ajustado MSFT

3.2.3. Amazon Inc

El año en que Amazon realizó la oferta pública inicial (IPO) de sus acciones fue en 1997. Ya que la frecuencia de recolección de los datos es diaria (cada día de la semana), se tiene un total aproximado de 5248 registros en total, hasta el 27 de marzo del 2018.

En las figuras 3.9 y 3.10 se pueden observar los precios ajustados de apertura, cierre, bajo y alto para este activo a lo largo la historia. Se puede notar que existe una correlación muy grande entre estos precios, lo cual también puede ser observado en la tabla de correlación de la figura 3.11. Por otro lado, en la figura 3.12 se puede observar el volumen histórico para este activo.

La tabla 3.5 presenta el cálculo de algunas medidas estadísticas para las diferentes características de este conjunto de datos.

	Adj open	**Adj close**	**Adj volume**	**Adj low**	**Adj high**
Count	5248	5248	5248	5248	5248
Mean	201,484144	201,504436	$7{,}803634 \times 10^6$	198,96019	203,817765
Standard deviation	281,862817	281,781054	$7{,}494447 \times 10^6$	279,17354	284,047276
Min	1,406667	1,395833	$4{,}872000 \times 10^5$	1,31250	1,448333
25 %	35,737500	35,750000	$3{,}820240 \times 10^6$	35,16750	36,500000
50 %	72,910000	72,660000	$5{,}882000 \times 10^6$	70,79915	74,510000
75 %	257,877500	258,012500	$8{,}851525 \times 10^6$	255,57750	260,040000
Max	1615,960000	1598,390000	$1{,}043292 \times 10^8$	1590,89000	1617,540000

Cuadro 3.5: Medidas estadísticas AMZN

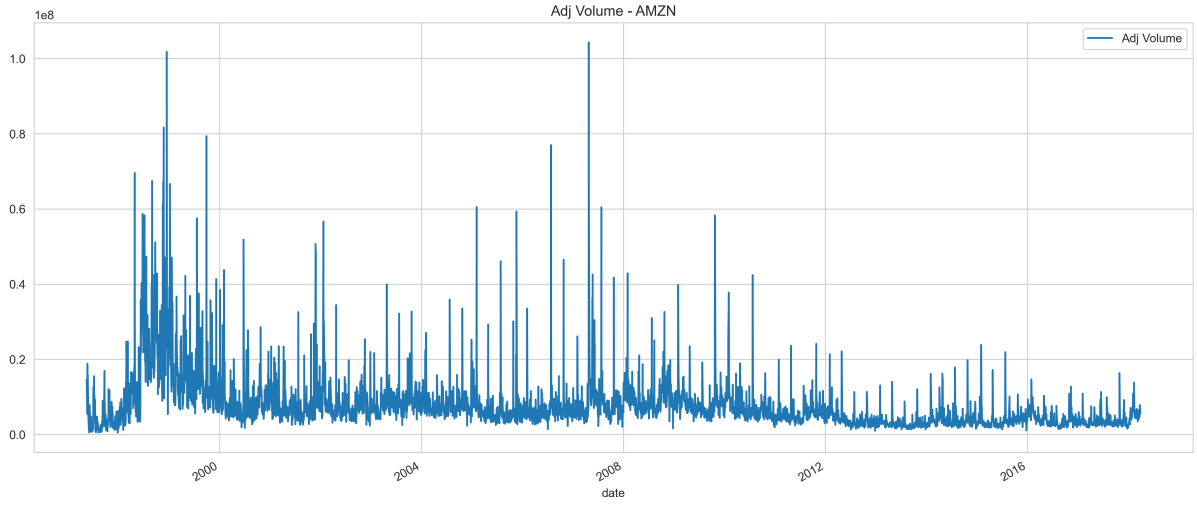


Figura 3.12: Volumen ajustado AMZN

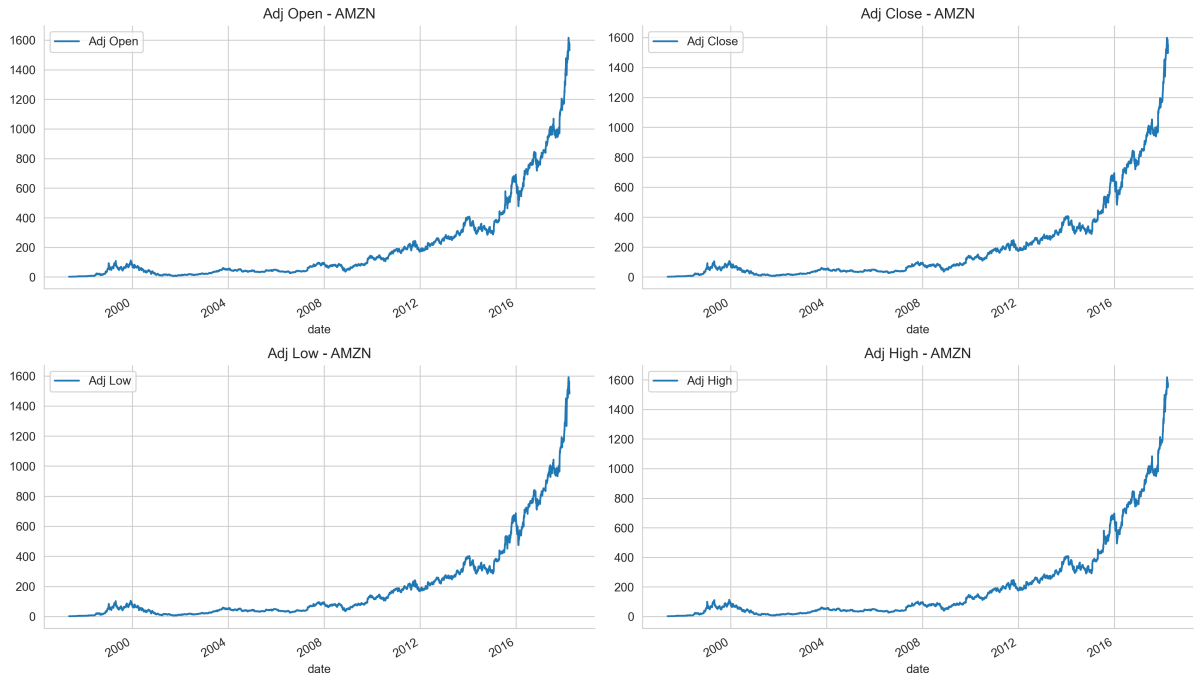


Figura 3.9: Precios históricos AMZN

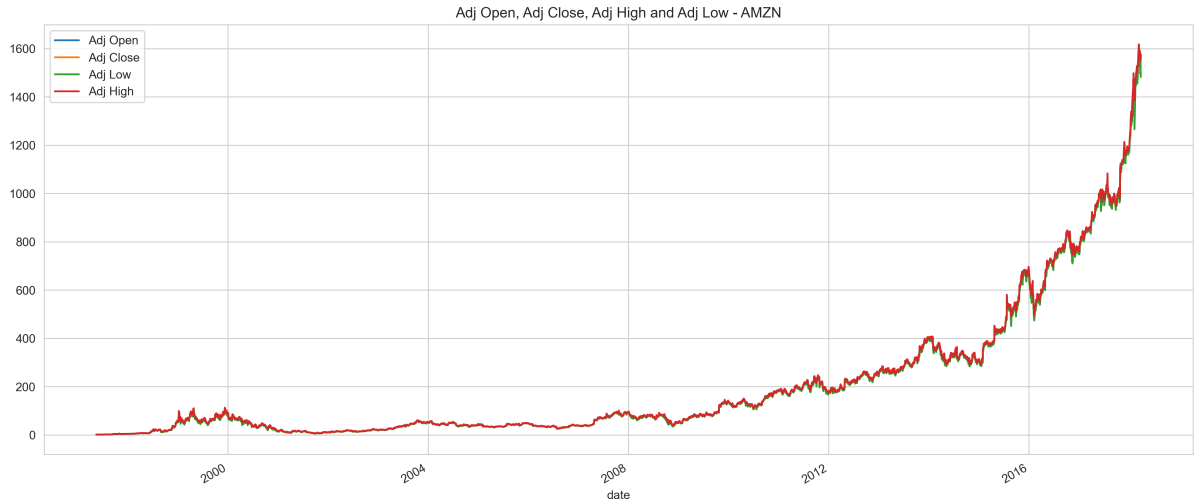


Figura 3.10: Precios históricos AMZN

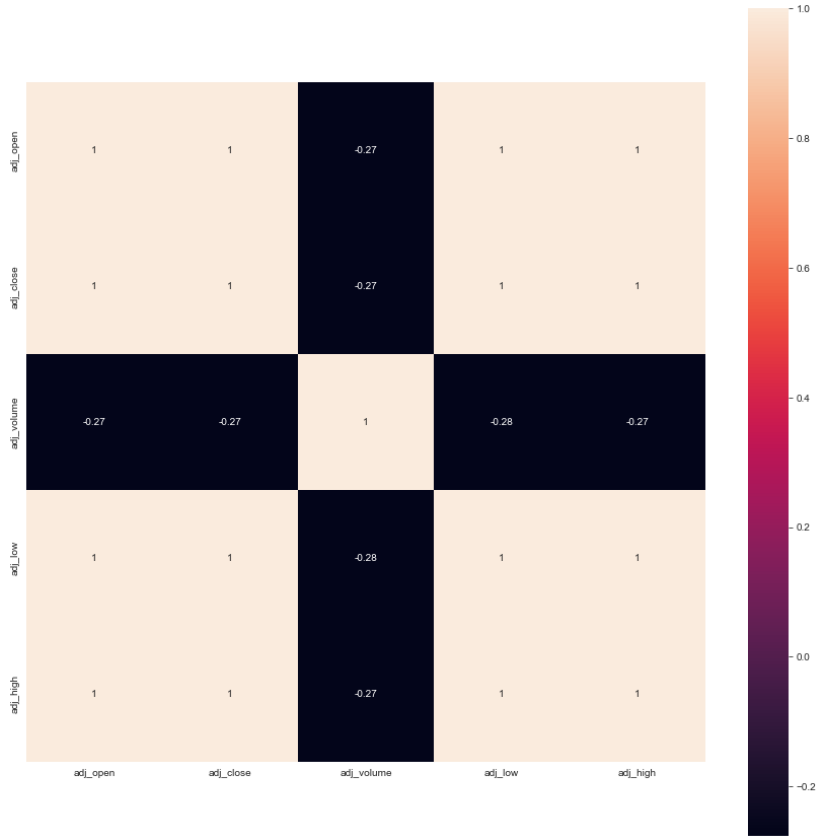


Figura 3.11: Correlación entre características AMZN

3.2.4. Pepsico Inc

El año que Pepsico realizó la oferta pública inicial (IPO) de sus acciones fue en 1972. Ya que la frecuencia de recolección de los datos es diaria (cada día de la semana), se tiene un total aproximado de 11556 registros en total, hasta el 27 de marzo del 2018.

En las figuras 3.13 y 3.14 se pueden observar los precios ajustados de apertura, cierre, bajo y alto para este activo a lo largo de la historia. Se puede notar que existe una correlación muy grande entre estos precios, lo cual también puede ser observado en la tabla de correlación de la figura 3.15. Por otro lado, en la figura 3.16 se puede observar el volumen histórico para este activo.

La tabla 3.6 presenta el cálculo de algunas medidas estadísticas para las diferentes características este conjunto de datos.

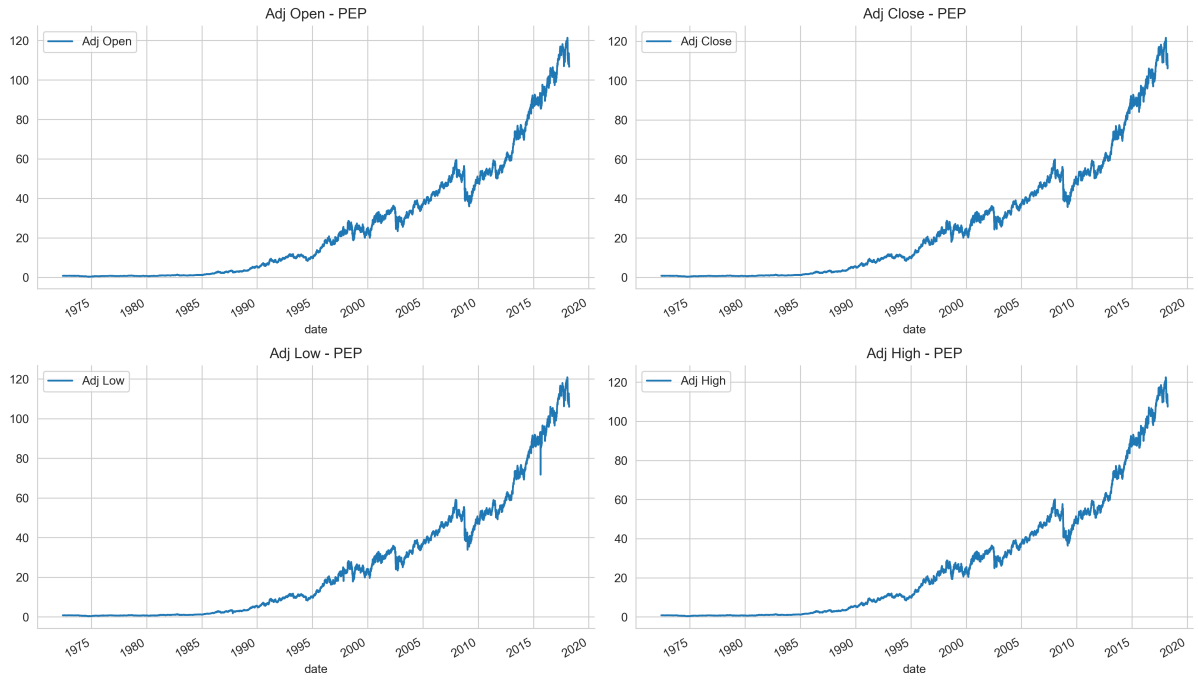


Figura 3.13: Precios históricos PEP

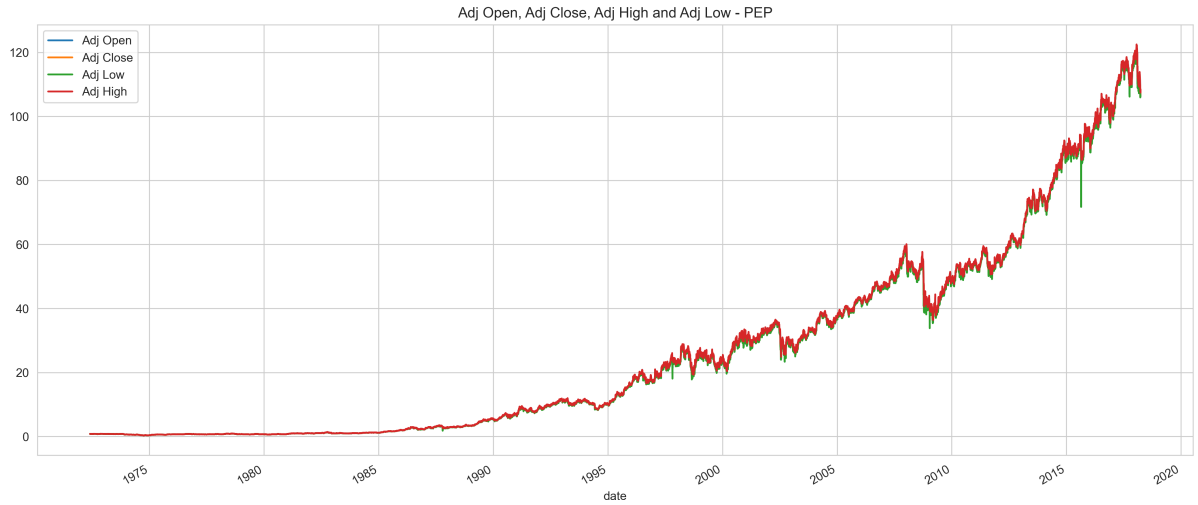


Figura 3.14: Precios históricos PEP

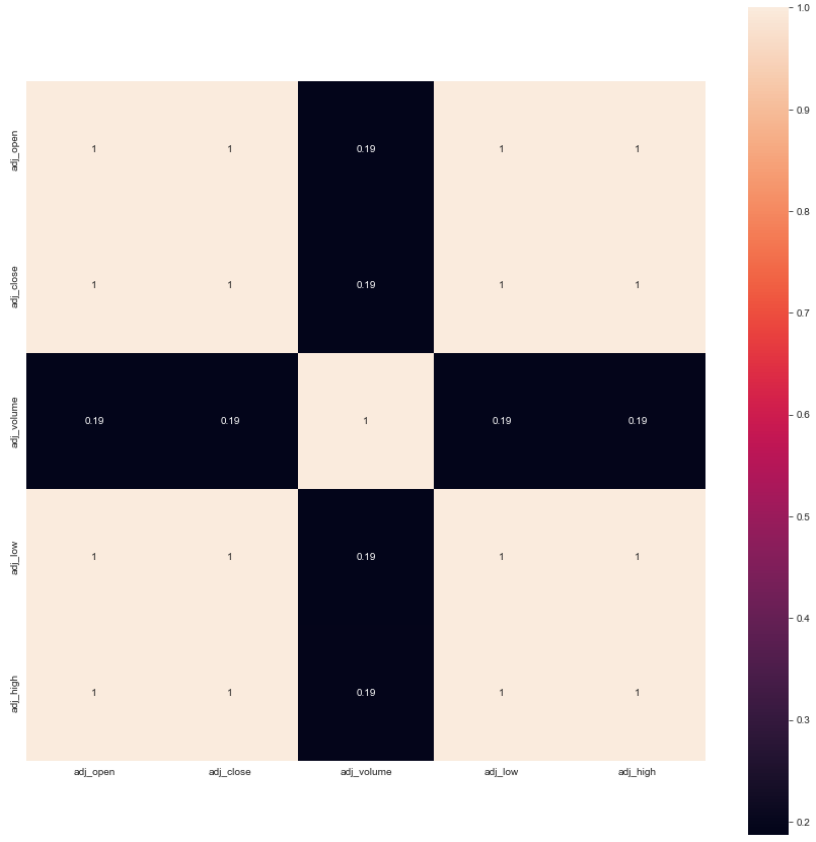


Figura 3.15: Correlación entre características PEP

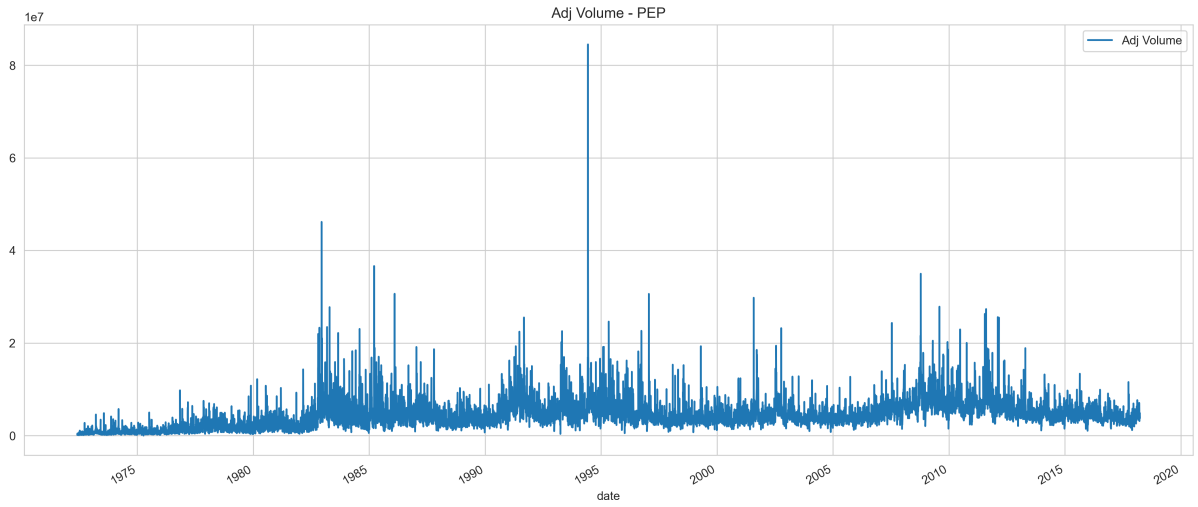


Figura 3.16: Volumen ajustado PEP

	Adj open	Adj close	Adj volume	Adj low	Adj high
Count	11556	11556	11556	11556	11556
Mean	26,171279	26,189184	$4{,}444373 \times 10^6$	25,974429	26,384153
Standard deviation	29,999609	30,010414	$3{,}163817 \times 10^6$	29,830450	30,177871
Min	0,272051	0,272051	0	0,272051	0,282518
25 %	1,086258	1,083757	$2{,}566800 \times 10^6$	1,073198	1,097652
50 %	11,666353	11,699193	$3{,}929285 \times 10^6$	11,572034	11,785645
75 %	42,624870	42,658789	$5{,}642544 \times 10^6$	42,280115	43,037780
Max	121,380000	121,760000	$8{,}450800 \times 10^7$	120,850000	122,510000

Cuadro 3.6: Medidas estadísticas PEP

3.3. Pre-procesamiento

Como se mencionó en la sección anterior, el conjunto de datos debe someterse a un pre procesamiento para su posterior uso ya que sus características están altamente correlacionadas y no presentan una información suficiente a los agentes en un estado dado. En esta sección se van a explicar las trasformaciones que se le aplicaron al conjunto de datos. El pre procesamiento explicado en esta sección aplica para los conjuntos de datos de cada activo ya que se aplican las mismas transformaciones en cada uno de estos.

A continuación se describen las tareas realizadas durante esta fase de pre procesamiento. Algunas de las tareas se van a encontrar más de una vez, pero estas son realizadas en dos instantes de tiempo distintas (i.e., se ejecutan en el orden numérico presentado).

1. Eliminar todos los registros que tengan algún elemento faltante. Esto se hace ya que en un paso posterior se necesita que todos los registros tengan valores numéricos validos.

2. Ordenar registros de acuerdo a su fecha. Esto se hace para facilitar que el ambiente de trading pueda proveer observaciones diarias secuenciales a los agentes durante los entrenamientos.

3. Limitar el conjunto de datos de acuerdo a un rango de fechas establecido. Esto se hace para tener un grado de generalidad más alto con respecto a los datos que son usados en cada experimento.

4. Agregar nuevas características que representen el porcentaje de cambio en el precio de la acción en 1, 2, 5, 10 y 21 periodos/días. El porcentaje de cambio de un solo periodo representa que tanto subió o bajó el precio de un día a otro. Los porcentajes de cambio de más de un periodo se incluyen para que los agentes tengan una visión más amplia del precio de la acción en un estado dado.

5. Agregar nuevas características de los indicadores técnicos descritos en la sección 2.2.7. Esto se hace con el fin de que los agentes tengan más información del conjunto de datos en un estado dado, así como también disminuir la correlación del conjunto de datos.

6. Eliminar características altamente correlacionadas en el conjunto de datos. Las características eliminadas en este paso son: *Adj high*, *Adj low*, *Adj close*, *Adj open* y *Adj volume*.

7. Eliminar registros con algún elemento faltante. Es necesario realizar esta tarea una vez más ya que durante el calculo de las nuevas características se generaron nuevos elementos faltantes.

8. Estandarizar las características del conjunto de datos. Esta tarea se encarga de centrar los datos a la media y escalarlos a una desviación estándar unitaria. La única característica que no se estandariza es aquella que guarda el porcentaje de cambio del precio de la acción de un día a otro. Esta característica necesita mantener su escala ya que es usada en el ambiente de trading como parte de la función de recompensa.

La implementación de esta fase de pre procesamiento se presenta en el anexo A.3.1 (linea 43 a 80). El cálculo de los indicadores técnicos mencionados en la sección 2.2.7 se realizó usando TA-Lib, una librería de análisis técnico para conjuntos de datos financieros de series de tiempo.

En la tabla 3.7 se describen las características del conjunto de datos después de su pre procesamiento.

Feature	Non-null count	Type
returns	26457 records	Float64
ret_2	26457 records	Float64
ret_5	26457 records	Float64
ret_10	26457 records	Float64
ret_21	26457 records	Float64
rsi	26457 records	Float64
macd	26457 records	Float64
atr	26457 records	Float64
stoch	26457 records	Float64
ultosc	26457 records	Float64
bbp	26457 records	Float64
obv	26457 records	Float64
adx	26457 records	Float64

Total columns: 13
Number of records: 26457

Cuadro 3.7: Características

Las secciones 3.3.1, 3.3.2, 3.3.3 y 3.3.4 presentan una descripción y entendimiento más detallado de los datos preprocesados para cada activo.

3.3.1. Apple Inc

En las tablas 3.8, 3.9 y 3.10 se presenta el cálculo de algunas medidas estadísticas para las nuevas características del conjunto de datos después de su pre procesamiento. Por otro lado, en la figura 3.17 se muestra la correlación entre estas características.

	returns	ret_2	ret_5	ret_10	ret_21
Count	7080	7080	7080	7080	7080
Mean	0.001128	0	0	0	0
Standard deviation	0.028265	1	1	1	1
Min	-0.518692	-1.376293e+01	-9.495508	-7.188429	-5.467957
25 %	-0.012772	-5.039776e-01	-5.357532e-01	-5.560235e-01	-5.687230e-01
50 %	0.000061	-2.032818e-02	2.337294e-03	-1.839008e-02	-2.223004e-02
75 %	0.014401	4.811800e-01	5.185993e-01	5.387684e-01	5.771181e-01
Max	0.332152	1.192623e+01	1.060189e+01	9.390914	8.484525

Cuadro 3.8: Medidas estadísticas AAPL (1)

	rsi	macd	atr	stoch	ultosc
Count	7080	7080	7080	7080	7080
Mean	0	0	0	0	0
Standard deviation	1	1	1	1	1
Min	-1.543478	-4.776145	-0.730216	-2.682760	-3.201159
25 %	-8.983911e-01	-2.418962e-01	-0.686738	-6.640832e-01	-7.011542e-01
50 %	6.795900e-03	-1.815889e-01	-0.565944	1.459833e-02	2.651079e-04
75 %	9.291321e-01	6.391707e-02	0.466728	6.609958e-01	7.244232e-01
Max	1.495061	4.935953	4.819790	2.640374	2.891175

Cuadro 3.9: Medidas estadísticas AAPL (2)

	bbp	obv	adx
Count	7080	7080	7080
Mean	0	0	0
Standard deviation	1	1	1
Min	-1.803843	-1.344825	-1.770972
25 %	-8.977810e-01	-1.114218	-7.608256e-01
50 %	8.601923e-02	-2.939808e-01	-1.778192e-01
75 %	9.000861e-01	1.142343	5.645452e-01
Max	1.662346	1.441184	4.023681

Cuadro 3.10: Medidas estadísticas AAPL (3)

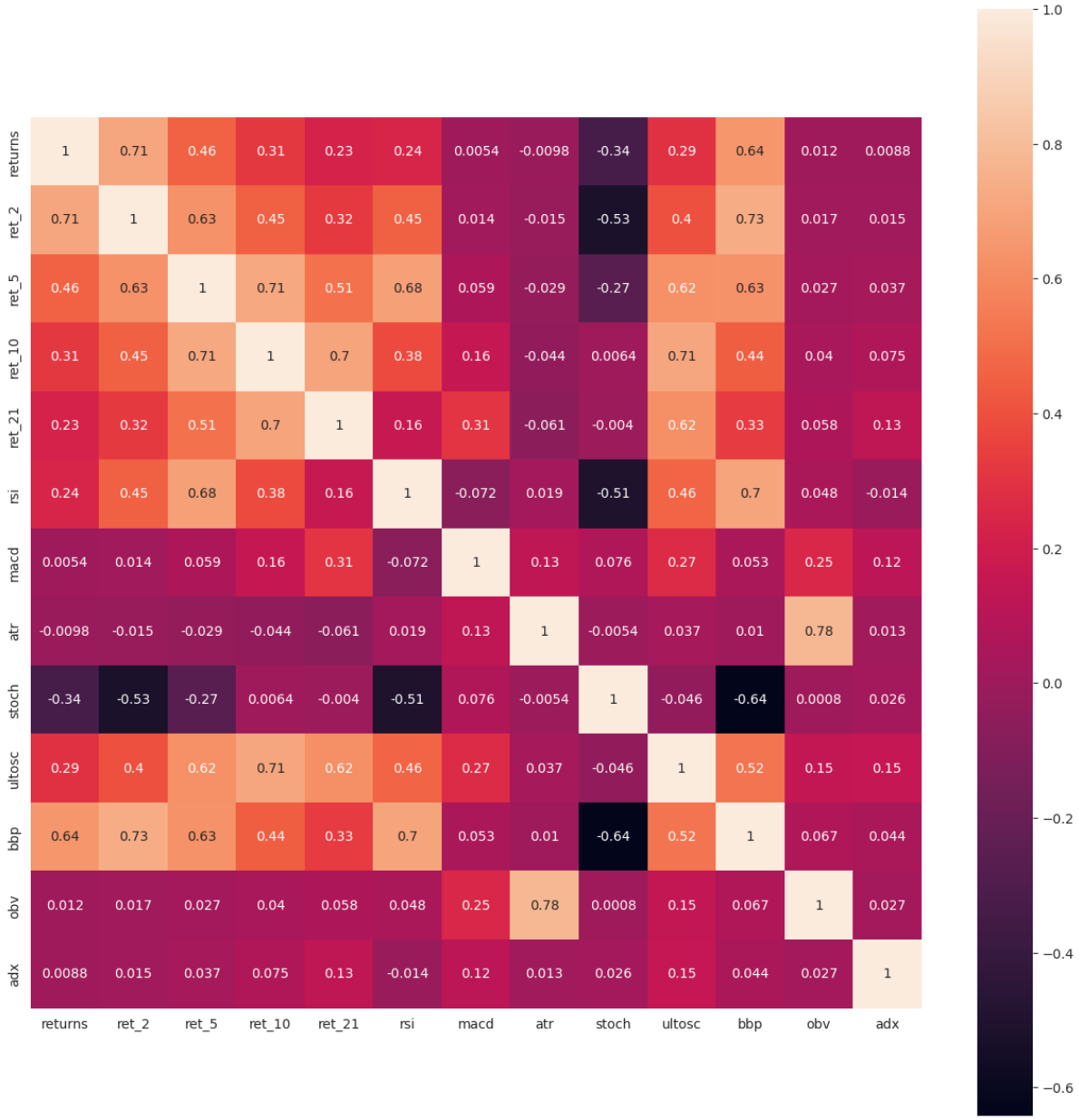


Figura 3.17: Correlación entre características AAPL

3.3.2. Microsoft

En las tablas 3.11, 3.12 y 3.13 se presenta el cálculo de algunas medidas estadísticas para las nuevas características del conjunto de datos después de su pre procesamiento. Por otro lado, en la figura 3.18 se muestra la correlación entre estas características.

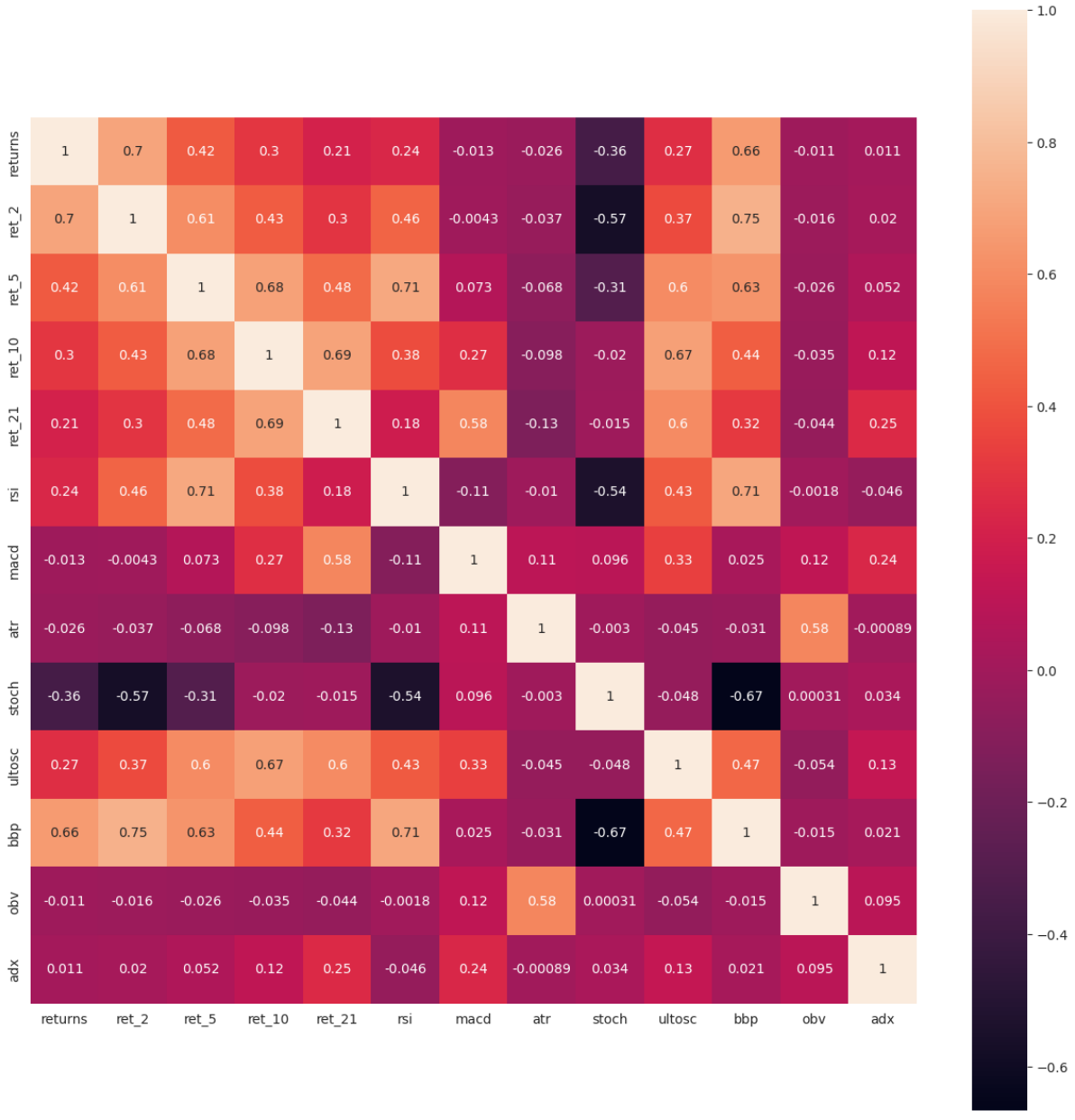


Figura 3.18: Correlación entre características MSFT

	returns	ret_2	ret_5	ret_10	ret_21
Count	7081	7081	7081	7081	7081
Mean	0.000952	0	0	0	0
Standard deviation	0.020202	1	1	1	1
Min	-0.156068	-5.949706	-6.509584	-5.257661	-4.869139
25 %	-0.009149	-5.245287e-01	-5.481315e-01	-5.501072e-01	-5.962262e-01
50 %	0.000172	-2.943024e-02	-3.228792e-02	-2.447977e-02	-2.091058e-02
75 %	0.010644	4.941718e-01	5.267003e-01	5.306335e-01	5.214883e-01
Max	0.195749	9.108488	5.317003	6.096444	5.043782

Cuadro 3.11: Medidas estadísticas MSFT (1)

	rsi	macd	atr	stoch	ultosc
Count	7081	7081	7081	7081	7081
Mean	0	0	0	0	0
Standard deviation	1	1	1	1	1
Min	-1.544166	-5.271256	-1.291806	-2.638003	-3.326276
25 %	-8.853617e-01	-3.737190e-01	-7.193131e-01	-6.712463e-01	-6.656505e-01
50 %	-8.212335e-03	-1.309703e-01	-1.614662e-01	-6.256030e-03	2.411653e-02
75 %	8.885976e-01	4.169755e-01	5.656713e-01	6.765073e-01	6.929088e-01
Max	1.518519	4.743616	6.114323	2.662734	3.319944

Cuadro 3.12: Medidas estadísticas MSFT (2)

	bbp	obv	adx
Count	7081	7081	7081
Mean	0	0	0
Standard deviation	1	1	1
Min	-1.810138	-2.729591	-1.800049
25 %	-9.088611e-01	-3.623333e-01	-7.463557e-01
50 %	7.385662e-02	2.491425e-01	-1.904070e-01
75 %	9.008779e-01	6.941459e-01	5.620910e-01
Max	1.670560	1.990624	4.209490

Cuadro 3.13: Medidas estadísticas MSFT (3)

3.3.3. Amazon Inc

En las tablas 3.14, 3.15 y 3.16 se presenta el cálculo de algunas medidas estadísticas para las nuevas características del conjunto de datos después de su pre procesamiento. Por otro lado, en la figura 3.19 se muestra la correlación entre estas características.

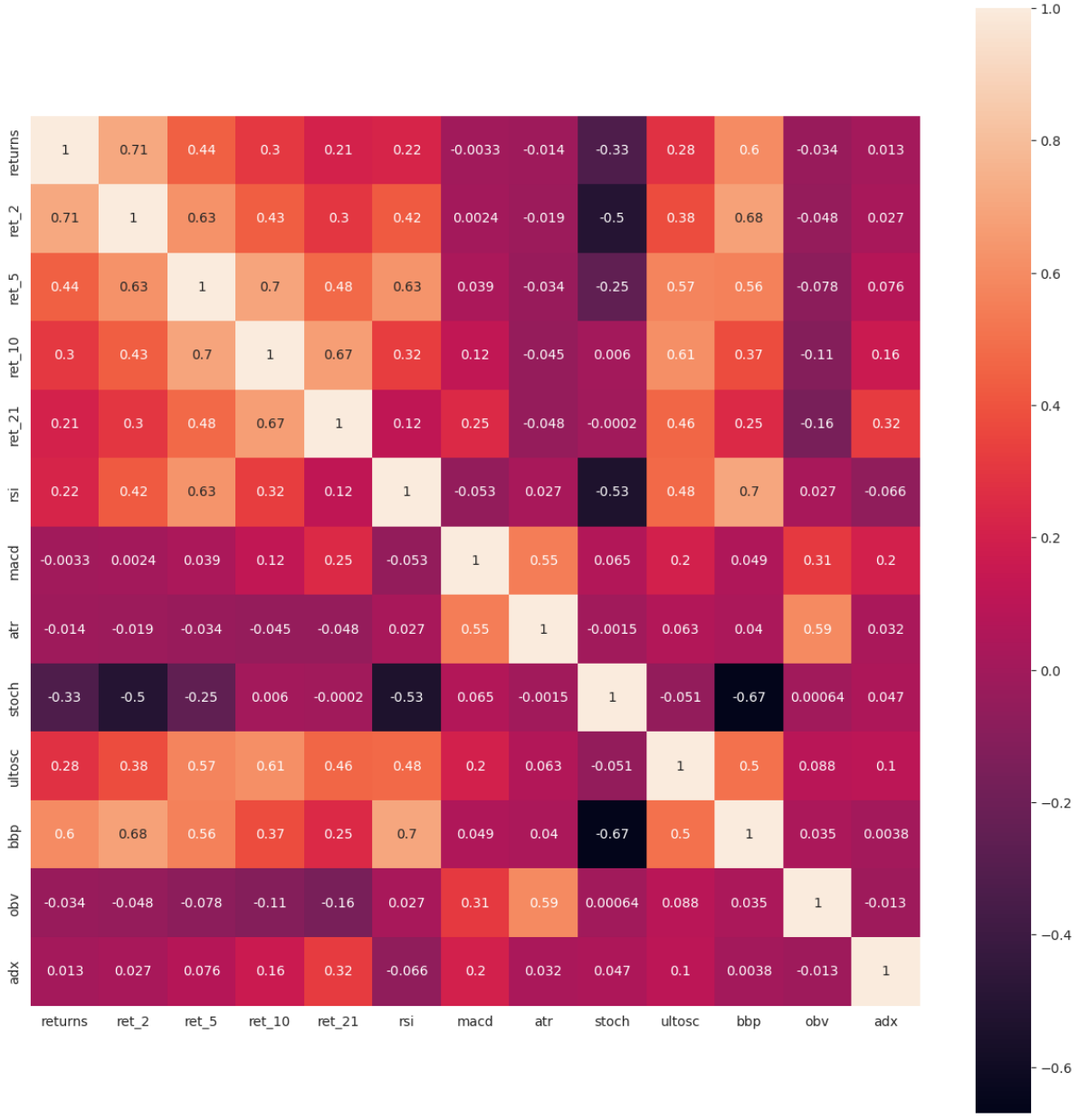


Figura 3.19: Correlación entre características AMZN

	returns	ret_2	ret_5	ret_10	ret_21
Count	5215	5215	5215	5215	5215
Mean	0.002037	0	0	0	0
Standard deviation	0.038436	1	1	1	1
Min	-0.247661	-5.494876	-4.615600	-3.298791	-2.821343
25 %	-0.013847	-4.274698e-01	-4.535417e-01	-4.880894e-01	-5.079941e-01
50 %	0.000305	-4.601486e-02	-5.027278e-02	-7.971849e-02	-7.655540e-02
75 %	0.015851	3.496544e-01	3.650108e-01	3.694796e-01	3.504929e-01
Max	0.344714	7.881814	8.202917	7.290285	1.121715e+01

Cuadro 3.14: Medidas estadísticas AMZN (1)

	rsi	macd	atr	stoch	ultosc
Count	5215	5215	5215	5215	5215
Mean	0	0	0	0	0
Standard deviation	1	1	1	1	1
Min	-1.529606	-4.499429	-9.045331e-01	-2.537204	-3.014219
25 %	-9.217623e-01	-3.800796e-01	-6.948254e-01	-6.932462e-01	-0.716715
50 %	7.685649e-04	-2.097629e-01	-2.842103e-01	9.670799e-03	-0.009672
75 %	9.259535e-01	1.231389e-01	2.771582e-01	6.876114e-01	0.662474
Max	1.494536	7.505979	8.670974	2.731888	3.703857

Cuadro 3.15: Medidas estadísticas AMZN (2)

	bbp	obv	adx
Count	5215	5215	5215
Mean	0	0	0
Standard deviation	1	1	1
Min	-1.807562	-3.114294	-1.912072
25 %	-9.144577e-01	-6.876765e-01	-7.424615e-01
50 %	6.355409e-02	2.384041e-01	-1.871104e-01
75 %	9.046517e-01	7.684198e-01	5.566342e-01
Max	1.650981	1.539117	4.032954

Cuadro 3.16: Medidas estadísticas AMZN (3)

3.3.4. Pepsico Inc

En las tablas 3.17, 3.18 y 3.19 se presenta el cálculo de algunas medidas estadísticas para las nuevas características del conjunto de datos después de su pre procesamiento. Por otro lado, en la figura 3.20 se muestra la correlación entre estas características.

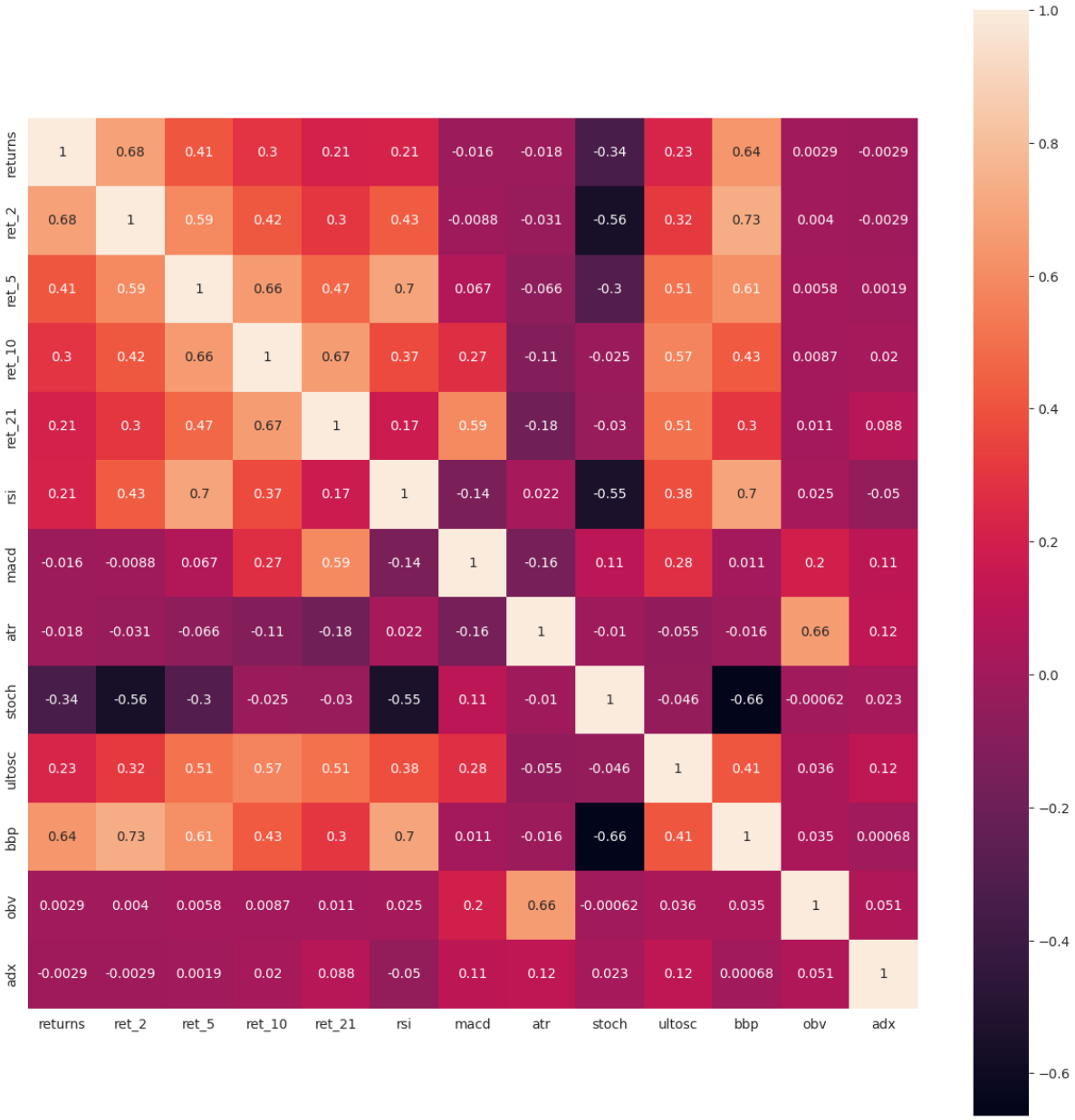


Figura 3.20: Correlación entre características PEP

	returns	ret_2	ret_5	ret_10	ret_21
Count	7081	7081	7081	7081	7081
Mean	0.000537	0	0	0	0
Standard deviation	0.014720	1	1	1	1
Min	-0.119314	-8.504268	-6.624936	-7.385382	-5.735748
25 %	-0.006742	-4.813976e-01	-5.101583e-01	-0.527744	-5.519724e-01
50 %	0	-2.540863e-02	6.812782e-03	0.015727	4.945849e-03
75 %	0.007574	4.635929e-01	5.143288e-01	0.539161	5.371733e-01
Max	0.161406	8.033337	6.674492	6.367952	4.942512

Cuadro 3.17: Medidas estadísticas PEP (1)

	rsi	macd	atr	stoch	ultosc
Count	7081	7081	7081	7081	7081
Mean	0	0	0	0	0
Standard deviation	1	1	1	1	1
Min	-1.585669	-6.965898	-1.579821	-2.706859	-3.165739
25 %	-8.780776e-01	-4.711230e-01	-6.277099e-01	-6.824644e-01	-6.903185e-01
50 %	1.911359e-02	-8.616393e-03	-1.185704e-01	4.360037e-03	1.716376e-02
75 %	9.022857e-01	5.606920e-01	5.916651e-01	6.942559e-01	7.120108e-01
Max	1.545007	3.886454	5.189983	2.643158	2.953605

Cuadro 3.18: Medidas estadísticas PEP (2)

	bbp	obv	adx
Count	7081	7081	7081
Mean	0	0	0
Standard deviation	1	1	1
Min	-1.832564	-2.685346	-1.695270
25 %	-9.286149e-01	-6.263737e-01	-7.709613e-01
50 %	8.644973e-02	-1.652564e-01	-2.155846e-01
75 %	8.845628e-01	5.471554e-01	6.311406e-01
Max	1.665757	2.542102	3.977426

Cuadro 3.19: Medidas estadísticas PEP (3)

Capítulo 4

Desarrollo e implementación

4.1. Agente de red doble de Q-learning profundo

En esta sección se destacan los elementos más importantes de la implementación del agente de red doble de Q-learning profundo (DDQN, por sus siglas en inglés). Una implementación con detalles mucho más extensos se presenta en el anexo A.2. De igual manera, la explicación teórica acerca de las redes neuronales construidas y la forma en que son entrenadas se puede encontrar en la sección 2.2.6.

4.1.1. Creación del agente DDQN

Un agente DDQN se crea o inicializa con la siguiente información:

- **Características del ambiente:** Número de dimensiones de los estados observados por el agente (*state_dim*), número de acciones disponibles (*num_actions*) y el símbolo del activo para el cual va a ser entrenado (*ticker*).
- **Información para la política ε-greedy:** Cómo se mencionó en la sección 2.2.5.1, para garantizar un balance entre exploración y explotación, el algoritmo Q hace uso de una política ε-greedy que selecciona una acción aleatoria con una probabilidad de ε, y sigue su política óptima (que selecciona la acción con el valor q más alto) en caso contrario. Adicionalmente, el valor ε se va a ver afectado por un factor de descuento que es aplicado después de cada episodio. Este descuento va a ser lineal hasta alcanzar un valor dado (*epsilon_end*), el cual será aplicado en una cantidad fija de pasos (*epsilon_decay_steps*). Después de eso, el descuento aplicado va a ser exponencial y está dado por el valor *epsilon_exponential_decay*. El valor inicial de ε está dado por *epsilon_start*. Usando los valores de *epsilon_start*, *epsilon_end* y *epsilon_decay_steps* podemos calcular el descuento lineal que se va a reflejar al final de cada episodio. La ecuación 4.1 muestra este cálculo.

$$epsilon\_decay = \frac{epsilon\_start - epsilon\_end}{epsilon\_decay\_steps} \tag{4.1}$$

  Por otro lado, el constructor también recibe un parámetro que indica si el agente va a ser entrenado o no (*train*). En el caso donde este parámetro sea verdadero, al atributo *epsilon* se le asignará el valor indicado por *epsilon_start*, y va a ser descontado como se explicó anteriormente. En el caso contrario, el valor del atributo *epsilon* va a ser fijado como 0, esto

con el fin de que el agente siempre tome su política óptima. Además, en este caso el descuento no será aplicado ya que este solo tiene sentido cuando las redes actualizan sus pesos durante entrenamiento.

- **Arquitectura de la red:** El constructor recibe la arquitectura de la red neuronal, la cual establece el número de capas que va a tener, como también el número de nodos asignados a cada capa. La dimensión de entrada de la primera capa corresponderá a la dimensión de los estados observados por el agente.
- **Parámetros de entrenamiento:**
  - Replay capacity (*replay_capacity*): Representa la capacidad máxima que va a tener el agente en la acumulación de su propia experiencia durante entrenamiento. Cómo se explicó en la sección 2.2.6.3, está experiencia va a ser usada para crear mini-batches aleatorios que servirán para actualizar los pesos de la red neuronal.
  - Regularización L2 (*l2_reg*): Factor de regularización de los pesos de la red neuronal para evitar overfitting.
  - τ (*tau*): Frecuencia de actualización de la red objetivo (*target_network*).
  - Tamaño de batch (*batch_size*): Tamaño de los mini-batches que serán creados para el entrenamiento de la red neuronal que predice la estimación actual (*online_network*).
  - Tasa de aprendizaje (*learning_rate*): Tasa de aprendizaje que determinará que tanto cambiar el modelo en respuesta al error estimado cada vez que se actualizan los pesos.
  - Factor de descuento (*gamma*): Factor de descuento que será aplicado a las recompensas a largo plazo obtenidas por el agente durante cada paso que realice.

  Usando la arquitectura de la red y los hiper parámetros descritos anteriormente, se construyen los modelos de la dos redes neuronales que van a guiar al agente durante su aprendizaje: *online_network* y *target_network*. En la sección 4.1.2 se describe la construcción de estos modelos.

4.1.2. Construcción del modelo

Este método es el encargado de construir los modelos para las dos redes neuronales que el agente va a usar durante su entrenamiento: *online_network* y *target_network*.

La red en linea se crea de tal forma que pueda ser entrenada (con el parámetro *trainable* como verdadero), mientras que en el caso de la red objetivo se crea como una red no entrenable. Esto se debe a que los pesos de red objetivo solo serán actualizados copiando periódicamente los pesos de la red en linea (cada τ pasos).

Inicialmente se crean capas densamente conectadas basadas en la arquitectura con la cual el agente fue instanciado. Esta arquitectura toma la forma de una lista o tupla de números enteros, donde su longitud indica el número de capas que va a tener la red, y cada valor indica el número de

nodos. A la primera capa se le asigna como dimensionalidad de entrada la dimensión del espacio de estados observados por el agente. Todas las capas especificadas en la arquitectura van a tener una función de activación *relu* y un regularizador de kernel *l2*, para el cual su factor de regularización es especificado como hiper parámetro. Después de haber creado estas capas, se crean otras dos capas adicionales. Primero, se crea una capa de *Dropout*, la cual se va a encargar de ignorar neuronas seleccionadas de forma aleatoria a lo largo del entrenamiento. Esta es una técnica de regularización que previene el sobreajuste de los modelos. Segundo, se agrega otra capa densamente conectada que tendrá como espacio de salida el número de acciones que el agente puede realizar en el ambiente. Esta es la capa de salida que estará compuesta por los valores Q para cada acción en un estado dado. Finalmente, se compila el modelo dejándolo así listo para entrenamiento. La red es entrenada en una función de pérdida que calcula el error cuadrático medio usando el optimizador *Adam*.

La implementación detallada de la construcción del modelo se presenta en el anexo A.2.2.

4.1.3. Manejo de pesos de la red en linea

Con el fin de que el entrenamiento completo de los agentes pueda ser llevado a cabo en una serie de entrenamientos más cortos, se han creado dos métodos auxiliares en el agente que nos permiten guardar y cargar los pesos en cualquier momento previo, durante o después del entrenamiento. Estos métodos son *load_model_weights()* y *save_model_weights()*. Detalles de su implementación pueden ser encontrados en el anexo A.2.6.

4.1.4. Política ε-greedy

Este método es el encargado de seleccionar la acción que el agente va a realizar en un determinado estado. A continuación se presenta su implementación:

```
1 def epsilon_greedy_policy(self, state):
2     self.total_steps += 1
3     if np.random.rand() <= self.epsilon:
4         return np.random.choice(self.num_actions)
5     q = self.online_network.predict(state, verbose=0)
6     return np.argmax(q, axis=1).squeeze()
```

Ya que el algoritmo DDQ-learning hace uso de una política ε-greedy para garantizar un balance entre exploración y explotación, en la línea 3 y 4 se puede ver reflejado el caso en el cual el agente decide tomar una acción aleatoria con una probabilidad de ε, en lugar de seguir su política óptima. Por otro lado, cuando este caso no se da, se procede a obtener los valores q para todas las acciones del estado actual, para luego retornar la acción con el valor q mayor.

4.1.5. Memorizar transición

Este método se encarga de memorizar cada transición de estado que el agente va experimentando en el ambiente en consecuencia de sus acciones. Esto permitirá crear mini-batches aleatorios que luego serán usados para actualizar los pesos de la red neuronal. El método recibe la observación

del estado actual (s), la acción del agente (a), el estado siguiente (s'), la recompensa obtenida (r) y un parámetro que indica si el episodio ha terminado o no (*done*). Si un episodio ha finalizado, este método también es el encargado de aplicar el factor de descuento al valor ε usado en la política ε-greedy. La aplicación de este descuento se explica con mayor detalle en la sección 4.1.1. La implementación detallada de este método se presenta en el anexo A.2.4.

4.1.6. Reproducción de experiencia

Este método se encarga de actualizar los pesos de las dos redes neuronales que se han construido para el agente: *online_network* y *target_network*. Detalles acerca del objetivo y función de cada una de estas redes se presentan en la sección 2.2.6.3. La reproducción de la experiencia comienza tan pronto cuando existe la cantidad suficiente de transiciones para crear un batch completo.

El método predice los valores Q para los siguientes estados usando la red de estimación actual (*online_network*) y selecciona las mejores acciones. Luego, selecciona los valores Q que se predicen para estas acciones de la red objetivo para llegar a los objetivos TD. Finalmente, se entrena la red online usando un unico batch de observaciones del estado actual como entrada, los objetivos de TD como resultado y el error cuadrático medio como función de pérdida.

La red objetivo se actualiza cada τ pasos copiando los pesos de la red en linea. La implementación detallada de este método se presenta en el anexo A.2.5.

4.2. Ambiente de trading

Un ambiente de trading, en el contexto de este proyecto, es un sistema simulado que replica las condiciones y dinámicas del mercado financiero en el que un agente de trading basado en aprendizaje por refuerzo opera y toma decisiones. Este entorno sirve como un campo de prueba virtual donde el agente puede interactuar con datos históricos, realizar acciones como comprar, vender o mantener activos financieros, y recibir retroalimentación en forma de recompensas y resultados de sus acciones.

La importancia de un entorno de trading en el desarrollo de este proyecto es fundamental. En primer lugar, proporciona un ambiente controlado y seguro para entrenar y evaluar el agente antes de desplegarlo en mercados financieros reales. Esto es esencial para mitigar riesgos financieros y desarrollar estrategias efectivas sin exponer recursos reales. Además, permite reproducir condiciones históricas y explorar diversos escenarios para evaluar el desempeño del agente en una variedad de condiciones del mercado.

Desarrollar un ambiente de trading personalizado ofrece muchas ventajas. En primer lugar, permite una adaptación precisa de las características del ambiente a los objetivos específicos del proyecto, como la elección de activos, la inclusión de indicadores técnicos y la modelización de la volatilidad. Además, facilita la incorporación de desafíos realistas que el agente puede encontrar en el mundo real, lo que promueve un aprendizaje más efectivo. Un ambiente personalizado también proporciona un mayor grado de control, lo que resulta esencial para experimentar con escenarios y condiciones que pueden no estar disponibles en entornos comerciales existentes. En resumen, un ambiente de trading personalizado no solo es esencial para el desarrollo y entrenamiento del agente,

sino que también ofrece la flexibilidad necesaria para adaptarse a los objetivos y desafíos específicos del proyecto de investigación, lo que lo convierte en una herramienta invaluable en el camino hacia la comprensión y mejora de las estrategias de trading basadas en aprendizaje por refuerzo.

En la sección anterior se explicó de manera general la implementación de un agente. Para el propósito del desarrollo de este proyecto, se decidió que cada activo tendrá su propio agente DDQN independiente, razón por la cual contaremos con 4 agentes durante el entrenamiento. Sin embargo, todos estos agentes van a interactuar con un ambiente de trading en común. En la imagen 4.1 se muestra la interacción entre los agentes y el ambiente.

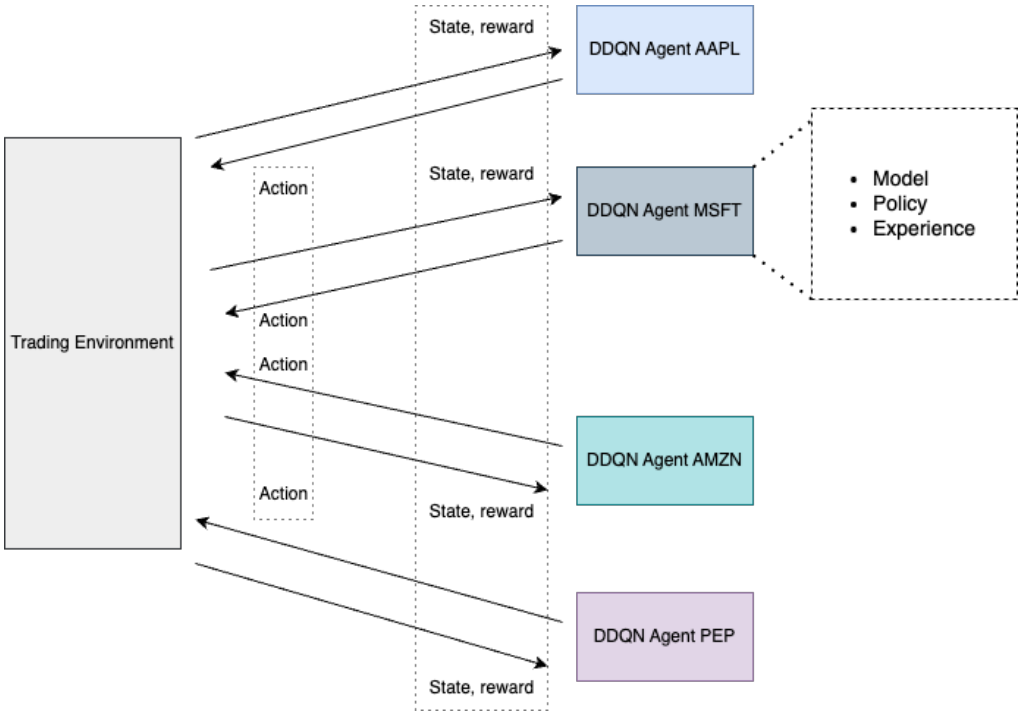


Figura 4.1: Interacción entre los agentes y el ambiente

El ambiente genera una serie de tiempo del precio para varias acciones usando una fecha de inicio aleatoria para simular un periodo de trading que contiene 252 días (1 año). Cada día representa un paso donde el ambiente provee a los agentes con una observación del estado que está compuesta de retornos históricos para periodos de 1, 2, 5, 10 y 21 días, así como también indicadores técnicos que se describen en la sección 2.2.7. Durante cada paso los agentes tiene la posibilidad de realizar una de las siguientes acciones:

- **Comprar**: Invertir todo el capital para una posición larga en el activo.
- **Mantener**: Mantener el capital.
- **Vender en corto**: Tomar una posición corta igual a la cantidad del capital.

En la acción de compra se tiene la expectativa de beneficiarse del incremento del precio del activo. De esta manera, se obtienen ganancias cuando se vende el activo a un precio superior al precio de compra. Por otro lado, en la acción de venta en corto se tiene la expectativa de beneficiarse de una bajada del precio del activo. De esta manera, se obtienen ganancias cuando se recompra el activo a un precio más bajo.

El cambiar de posición implica un costo de trading. Aunque este costo esta parametrizado en el ambiente, para todas las simulaciones realizadas se ha establecido un costo de trading de 10bps[1]. El no cambiar de posición también implica un costo de tiempo, el cual se ha establecido como 1bps para todas las simulaciones.

Este ambiente de trading no lleva registro de una cantidad de capital en particular, sino del valor que representa ese capital en el tiempo. En finanzas, este concepto se conoce como Valor Liquidativo Neto (VL, o NAV por sus siglas en inglés), el cual hace referencia a la representación de valor por unidad de un activo. De esta manera, un episodio comienza con un NAV de 1 unidad de efectivo.

Note que este ambiente tiene dos simplificaciones importantes:

- Las acciones de venta y compra siempre se hacen sobre todo el capital.
- No se lleva un registro de la cantidad específica de capital sino del valor unitario que este representa.

Estas simplificaciones reducen la complejidad del ambiente ya que evitan la necesidad de tener acciones continuas o un mayor numero de acciones discretas y facilitan el registro de la contabilidad. Determinar la cantidad de capital que es asignada a una operación de trading en particular se conoce en finanzas como tamaño de posicion, lo cual es sugerido como trabajo futuro en la sección 6.2.

En las subsecciones siguientes se destacan los elementos más importantes de la implementación del ambiente. Sin embargo, una implementación con detalles mucho más extensos se presenta en el anexo A.3. El ambiente se compone de tres módulos principales:

- **DataSource**: Se encarga de cargar las series de tiempo de cada activo, generar nuevas características y proveer observaciones a los agentes en cada paso.
- **TradingSimulator**: Lleva registro de las posiciones, costos y retribuciones obtenidas por las acciones que los agentes realizan en cada paso. También se implementa y lleva registro de los resultados de la estrategia *buy-and-hold*, la cual es una estrategia de inversión a largo plazo donde un inversionista compra un activo con la intención de mantenerlo por un periodo extendido, por lo general varios meses, años o décadas. Esta estrategia servirá como referencia para comparar los resultados que cada agente obtenga. Como se expone en la sección 1.1, las inversiones en renta variable son muy volátiles en periodos cortos, por lo cual las retribuciones más altas se ven reflejas en inversiones a largo plazo. Por esa razón, la estrategia de *buy-and-hold* nos sirve como un buen marco de referencia para comparar los resultados obtenidos por los agentes.

[1] En finanzas, un punto básico (bp) es una unidad de medida que describe cambios porcentuales muy pequeños. 1bp es igual a una centésima de punto porcentual, o 0.01 %. En forma decimal, es representado como 0.0001.

- **TradingEnviroment**: Se encarga de orquestar todo el proceso usando las dos clases anteriores.

4.2.1. DataSource

En esta clase se cargan y preprocesan los datos históricos de los activos. Detalles acerca del pre procesamiento pueden encontrarse en la sección 3.3.

Por otro lado, la clase también se encarga de mantener registro del progreso de cada episodio, proveer observaciones de los estados a los agentes en cada paso e indicar el final de los episodios. A continuación se presenta la implementación del método *take_ step* para este fin. Detalles de la implementación completa de esta clase se encuentran en el anexo A.3.1.

```
def take_step(self):
    observations = {}
    for ticker in self.tickers:
        observations[ticker] = self.data.loc[(
            slice(None), ticker), :].iloc[self.offset[ticker] + self.step].values
    self.step += 1
    done = self.step > self.trading_days
    return observations, done
```

4.2.2. TradingSimulator

Esta clase se encarga de calcular las recompensas que los agentes obtienen en cada paso. La función de recompensa es una parte fundamental en el diseño de un ambiente de aprendizaje por refuerzo, ya que tiene un impacto directo en el aprendizaje que los agentes experimentan por cada acción que tomen en un paso dado. Para este ambiente, la función de recompensa consiste en los **retornos diarios** menos los **costos generados** para llegar a la posición que generó estos retornos. Los retornos diarios dependen de la posición tomada el día anterior y de los retornos del mercado en el día actual. Las posiciones se abstraen en el ambiente como -1 para representar una posición corta, 0 para una posición neutra y 1 para una posición larga. La ecuación 4.2 muestra el calculo de los retornos diarios.

$$daily\ returns = prev\ position \times actual\ market\ returns \tag{4.2}$$

Por otro lado, los costos en un día dado equivalen a los costos de trading, en caso de haber cambiado de posición con respecto al día anterior, o los costos de tiempo en caso de haber mantenido la misma posición. Como se mencionó en una sección anterior, los cosos de trading equivalen a $10bps$ y los costos de tiempo a $1bps$. Ir de una posición corta a larga, o viceversa, implica dos operaciones, razón por la cual los costos de trading es ese caso serian $2 \times 10bps$. La ecuación 4.3 muestra los cálculos de los costos.

$$costs = number\ of\ trades \times trading\ cost + time\ cost \tag{4.3}$$

De esta manera, con ayuda de las ecuaciones 4.2 y 4.3, la ecuación 4.4 muestra el calculo de la recompensa.

$$reward = daily\ returns - prev\ costs \tag{4.4}$$

Esta clase también se encarga de llevar registro del NAV de los agentes y el mercado, así como también de las posiciones, costos y numero de operaciones realizadas en cada paso. El NAV del mercado representa la ejecución de la estrategia *buy-and-hold*. A continuación se muestra una versión simplificada del método *take_ step*. Detalles de la implementación completa de esta clase se encuentran ene el anexo A.3.2.

```
def take_step(self, actions, market_returns):
    ticker_rewards = {}
    for ticker in self.tickers:
        # get previous step indicators
        prev_step = max(0, self.step - 1)
        prev_nav = self.navs[ticker][prev_step]
        prev_market_nav = self.market_navs[ticker][prev_step]
        prev_position = self.positions[ticker][prev_step]
        prev_costs = self.costs[ticker][prev_step]

        # Track positions, number of trades, market returns, actions and costs
        end_position = actions[ticker] - 1  # short, neutral, long
        n_trades = end_position - prev_position
        trade_costs = abs(n_trades) * self.trading_cost_bps
        time_cost = 0 if n_trades else self.time_cost_bps
        self.positions[ticker][self.step] = end_position
        self.trades[ticker][self.step] = n_trades
        self.market_returns[ticker][self.step] = market_returns[ticker]
        self.actions[ticker][self.step] = actions[ticker]
        self.costs[ticker][self.step] = trade_costs + time_cost

        # Compute reward
        daily_returns = prev_position * market_returns[ticker]
        reward = daily_returns - prev_costs
        self.strategy_returns[ticker][self.step] = reward

        # Compute and track agent and market navs
        if self.step != 0:
            self.navs[ticker][self.step] = prev_nav * (1 + self.strategy_returns[
    ticker][self.step])
            self.market_navs[ticker][self.step] = prev_market_nav * (1 + self.
    market_returns[ticker][self.step])
        ticker_rewards[ticker] = reward
    self.step += 1
    return ticker_rewards
```

4.2.3. TradingEnvironment

Esta clase se encarga de orquestar los procesos descritos en las dos secciones anteriores, determinando así la dinámica del ambiente en cada episodio. Los métodos más importantes de esta clase son *reset* y *take_ step*. Por un lado, el método *reset* se ejecuta cada vez que un episodio ha finalizado, causando que

1. la instancia de la clase *DataSource* proporcione nuevos índices iniciales en la series temporal,
2. la instancia de la clase *TradingSimulator* inicialice todos los registros calculados hasta ese momento y
3. se retorne la primera observación, ejecutando el método *take_ step* de la clase *DataSource*.

Por otro lado, el método *take_ step* proporciona una observación del estado (ejecutando el método *take_ step* de la clase *DataSource*) y recompensa las acciones tomadas por los agentes (ejecutando el método *take_ step* de la clase *TradingSimulator*). A continuación se muestra la implementación de estos dos métodos. Detalles de la implementación completa de esta clase se encuentran en el anexo A.3.3.

```
def reset(self):
    self.data_source.reset()
    self.simulator.reset()
    return self.data_source.take_step()[0]
```

```
def step(self, actions):
    observations, done = self.data_source.take_step()
    rewards self.simulator.take_step(actions=actions,
                                     market_returns={ticker: observations[ticker][0]
    for ticker in self.tickers})
    return observations, rewards, done
```

4.3. Entrenamiento

En este proyecto, el enfoque principal del entrenamiento se centra en el desarrollo y perfeccionamiento de un agente de trading basado en aprendizaje por refuerzo. Este agente será sometido a un extenso entrenamiento dentro de un ambiente simulado, donde aprenderá a tomar decisiones de forma autónoma relacionadas con la compra, venta o retención de activos financieros. El objetivo de entrenamiento del agente es adquirir la capacidad de navegar por las complejidades de los mercados financieros, analizar datos de mercado históricos, identificar patrones y tendencias, y ejecutar acciones de trading que maximicen los retornos. A lo largo del proceso de entrenamiento, el agente se adaptará y optimizará continuamente sus estrategias de toma de decisiones, aprovechando el poder del aprendizaje por refuerzo, y en particular de la técnica DDQ-learning, para mejorar progresivamente su competencia en el desafiante ámbito de la compra y venta de acciones. En última

instancia, se espera que el agente entrenado demuestre la capacidad de tomar decisiones de trading informadas y estratégicas, incluso frente a la volatilidad del mercado y las rápidas fluctuaciones de precios.

Durante el entrenamiento ocurren dos cosas principales:

1. Los agentes interactúan con el ambiente (i.e, los agentes tomando acciones y el ambiente ejecutándolas).

2. Los agentes son entrenados de acuerdo a la experiencia que han obtenido en cada paso.

Como se mostró en figura 4.1, existen agentes independientes para cada activo, los cuales son inicializados con sus respectivos hiper parámetros antes de iniciar el ciclo de entrenamiento.

```
ddqn = {ticker: DDQNAgent(state_dim=state_dim,
                  num_actions=num_actions,
                  learning_rate=learning_rate,
                  gamma=gamma,
                  epsilon_start=epsilon_start,
                  epsilon_end=epsilon_end,
                  epsilon_decay_steps=epsilon_decay_steps,
                  epsilon_exponential_decay=epsilon_exponential_decay,
                  replay_capacity=replay_capacity,
                  architecture=architecture,
                  l2_reg=l2_reg,
                  tau=tau,
                  batch_size=batch_size,
                  train=train,
                  ticker=ticker) for ticker in tickers}
```

El entrenamiento se lleva a cabo por un numero fijo de episodios, donde en cada uno de estos los agentes los agentes toman acciones que forman parte de la ejecución de pasos en el ambiente por un total de 252 días. Mientras un episodio no ha terminado, cada agente toma una acción recomendada por su política, la cual puede ser basada en su política óptima o una acción aleatoria de exploración (ir a sección 4.1.4 para detalles acerca de la política).

```
actions = {}
for ticker in tickers:
    action = int(ddqn[ticker].epsilon_greedy_policy(current_states[ticker].reshape(-1,
     state_dim)))
    actions[ticker] = action
```

Estas acciones generan transiciones de estados en el ambiente, las cuales son memorizadas por los agentes para luego entrenar la red en linea con la experiencia memorizada hasta ese punto.

```
for ticker in tickers:
    ddqn[ticker].memorize_transition(current_states[ticker],
                                actions[ticker],
```

```
                                rewards[ticker],
                                next_states[ticker],
                                done)
    ddqn[ticker].experience_replay()
```

Detalles de la implementación del ciclo de entrenamiento se encuentran en el anexo A.4.

4.3.1. Estrategia de entrenamiento

Una de las limitaciones que se encontraron durante el desarrollo de este proyecto fue la falta de recursos computacionales, ya que los entrenamientos requerían un uso de memoria bastante significativo (aproximadamente 1GB por episodio). De esta manera, ejecutar un experimento de más de 500 episodios requería un uso de memoria de 500GB+, lo cual era una cantidad de recursos que no se tenían. Por esta razón, la ejecución de los experimentos fue llevada a cabo en una serie de entrenamientos con un número menor de episodios. En la figura 4.2 se muestra como se llevó a cabo este proceso.

4.3.2. Exploración de hiper-parámetros

En las secciones 4.1 y 4.2 se describieron una serie de hiper parámetros sobre los cuales se puede hacer una exploración. Estos hiper-parámetros se describen en la tabla 4.1.

Hiper-parámetro	**Descripción**
gamma	Factor de descuento
tau	Frecuencia de actualización de la red objetivo
architecture	Arquitectura del modelo de la red en linea y objetivo
learning_rate	Tasa de aprendizaje
l2_reg	Factor de regularización
replay_capacity	Capacidad máxima de acumulación de experiencia
batch_size	Tamaño de los mini-batches para entrenamiento de la red

Cuadro 4.1: Hiper-parámetros

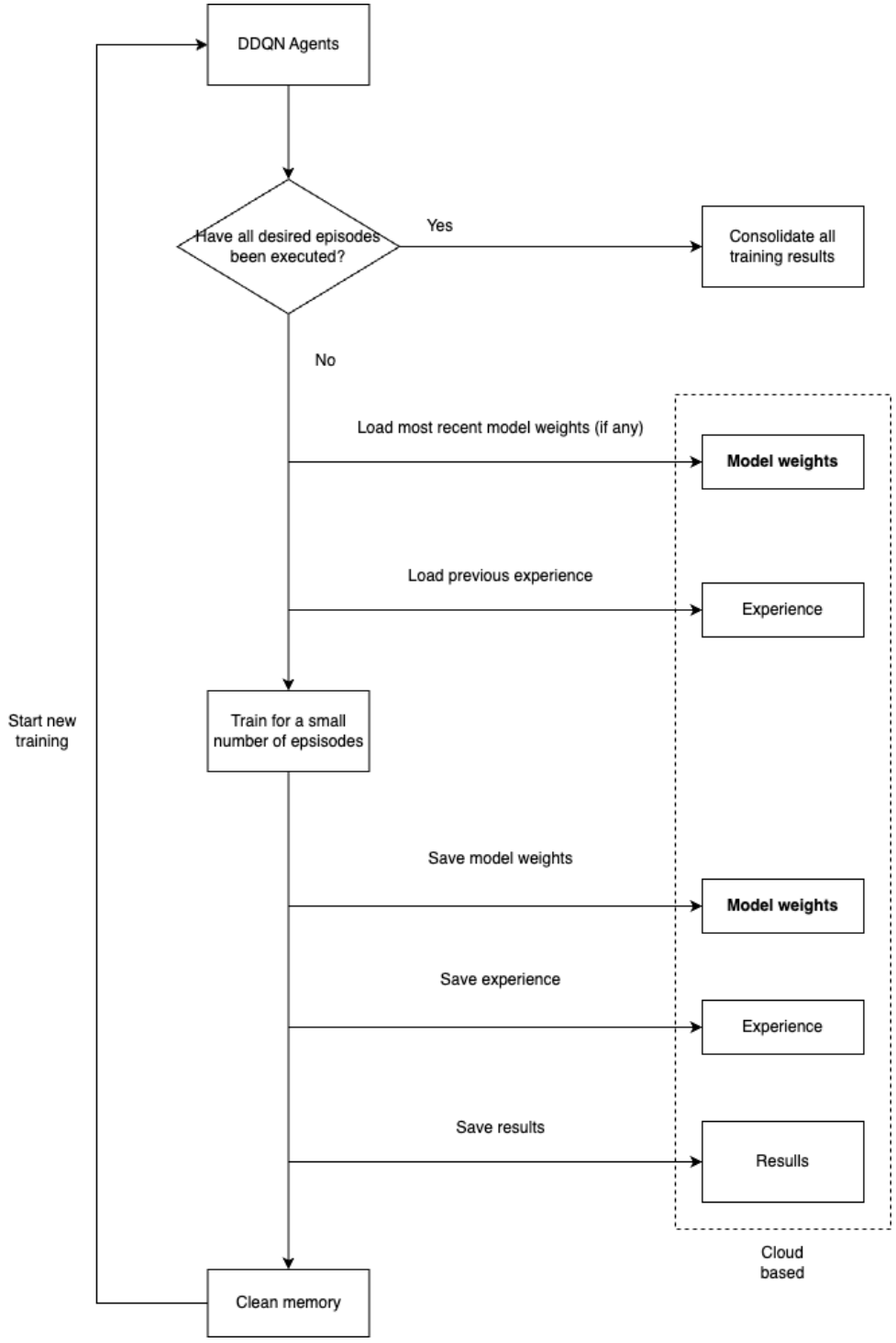


Figura 4.2: Proceso de entrenamiento

Debido a que las ejecuciones de los entrenamientos toman mucho tiempo, se ha decido explorar solamente sobre un subconjunto de estos hiper-parámetros. En particular, se decidió dejar *gamma* y *tau* con valores constantes de 0,99 y 100, respectivamente, para todos los experimentos realizados. Para los hiper-parámetros restantes se ha planteado el siguiente espacio de búsqueda:

Hiper-parámetro	**Espacio de búsqueda**
architecture	(64, 64), (256, 128, 64), (128, 64, 32), (256, 256)
learning_rate	0.01, 0.001
l2_reg	0.001, 0.000001
replay_capacity	1000, 10000, 100000, 1000000
batch_size	128, 256, 1024, 2048, 4096

Ya que este espacio de búsqueda sigue siendo muy grande para explorarlo completamente, se hace una elección de hiper-parámetros aleatoriamente dirigida. De esta manera, se generan los siguientes experimentos:

Hiper-parámetro	**Valor**
architecture	(128, 64, 32)
learning_rate	0.0001
l2_reg	1e-06
replay_capacity	1000000
batch_size	256

Cuadro 4.2: Experimento 1

Hiper-parámetro	**Valor**
architecture	(64, 64)
learning_rate	0.001
l2_reg	1e-06
replay_capacity	10000
batch_size	128

Cuadro 4.3: Experimento 2

Hiper-parámetro	**Valor**
architecture	(256, 128, 64)
learning_rate	0.0001
l2_reg	1e-03
replay_capacity	10000
batch_size	2048

Cuadro 4.4: Experimento 3

Hiper-parámetro	**Valor**
architecture	(256, 256)
learning_rate	0.0001
l2_reg	1e-06
replay_capacity	1000000
batch_size	4096

Cuadro 4.5: Experimento 4

4.3.3. Ejecución de experimentos

En esta sección se van a presentar con mayor detalle los experimentos descritos en la sección anterior. En particular, se van a explicar más detalles acerca de los hiper parámetros de cada experimento, así como también los tiempos de ejecución que cada uno de estos presentaron.

Durante los entrenamientos se aplica un factor de descuento al valor ε, el cual es un descuento lineal de 1,0 a 0,01 durante 250 episodios y un descuento exponencial de 0,99 después de eso. La figura 4.3 muestra el comportamiento del valor ε a lo largo de la ejecución de cada episodio.

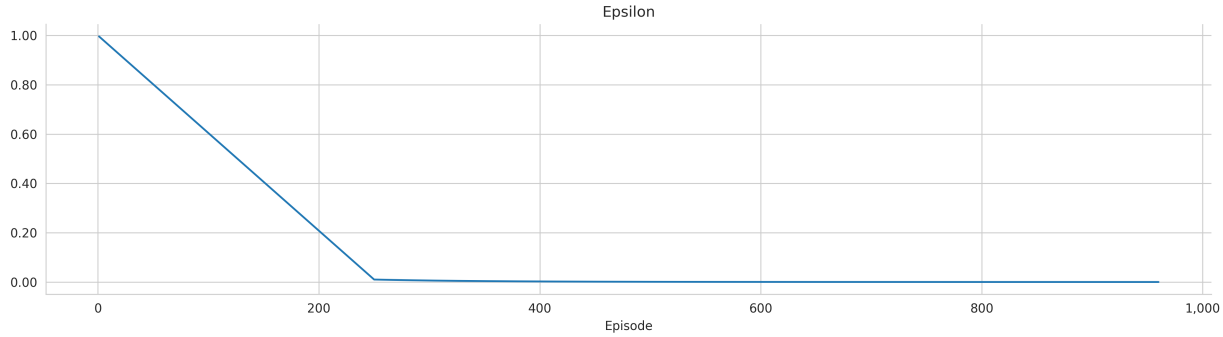


Figura 4.3: Factor de descuento a ε

Los experimentos 1, 2 y 3 son ejecutados sobre un numero bajo de episodios, por lo cual no se esperan buenos resultados en términos de rendimientos acumulados. El propósito de estos 3 primeros experimentos es observar tendencias de aprendizaje que guíen la selección de hiper-parámetros de un cuarto experimento que si sea ejecutado sobre un numero considerable de episodios. En este último experimento si se esperan buenos resultados en términos de rendimientos acumulados, incluso con expectativas de sobrepasar los rendimientos obtenidos por la estrategia *buy-and-hold*.

4.3.3.1. Experimento 1

Los valores de los hiper-parámetros de este experimento se encuentran en la tabla 4.2. En primer lugar, se tiene una arquitectura de 3 capas ocultas, donde la primera tiene 128 nodos, la segunda 64 y la última 32. Adicionalmente, el modelo también está compuesto por una última capa de regularización para reducir el overfitting (ver sección 4.1.2 para más detalles). La figura 4.4 muestra la dimensión de entrada y de salida de cada capa. En segundo lugar, tenemos una tasa de aprendizaje de 0,0001, la cual resultará en actualizaciones pequeñas de los pesos del modelo durante cada iteración de entrenamiento. En tercer lugar, tenemos un factor de regularización $l2$ de 0,000001, el cual resultará en actualizaciones pequeñas de los pesos en el modelo. En cuarto lugar, tenemos una capacidad de reproducción de experiencia de 1000000, indicando que los agentes tendrán acceso a las ultimas 1000000 transiciones para formar los mini-batches aleatorios durante su entrenamiento. Finalmente, tenemos un tamaño de batch de 256, indicando que de la experiencia acumulada hasta un punto dado se van a formar mini-batches de tamaño 256 para la actualización de los pesos de la red en linea.

Para este experimento se permitió que la exploración continuara por un total de 480 episodios de 252 pasos/días, resultando así en una ejecución de alrededor de 121000 pasos. El tiempo de ejecución de este entrenamiento fue de aproximadamente 13 horas.

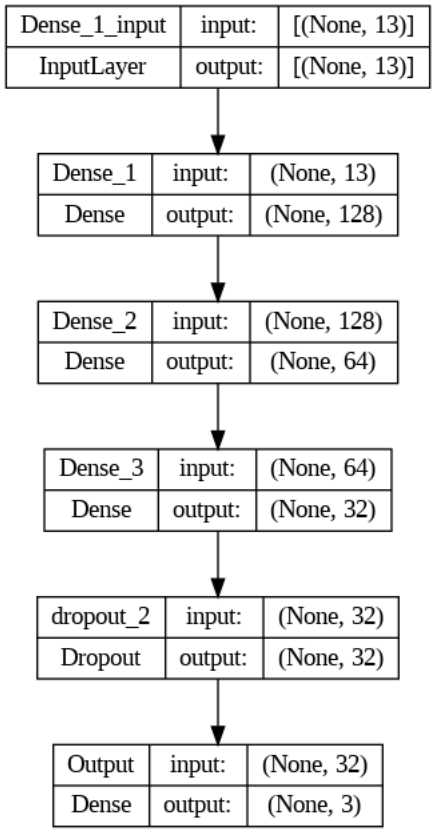


Figura 4.4: Arquitectura red neuronal experimento 1

4.3.3.2. Experimento 2

Los valores de los hiper-parámetros de este experimento se encuentran en la tabla 4.3. En primer lugar, se tiene una arquitectura de 2 capas ocultas de 64 nodos cada una. Adicionalmente, el modelo también está compuesto por una última capa de regularización para reducir el overfitting (ver sección 4.1.2 para más detalles). La figura 4.5 muestra la dimensión de entrada y de salida de cada capa. En segundo lugar, tenemos una tasa de aprendizaje de 0,001, la cual resultará en actualizaciones ligeramente más grandes de los pesos del modelo durante cada iteración de entrenamiento con respecto al experimento anterior. En tercer lugar, tenemos un factor de regularización $l2$ de 0,000001, el cual resultará en actualizaciones pequeñas de los pesos en el modelo. En cuarto lugar, tenemos una capacidad de reproducción de experiencia de 10000, indicando que los agentes tendrán acceso a las ultimas 10000 transiciones para formar los mini-batches aleatorios durante su entrenamiento. Finalmente, tenemos un tamaño de batch de 128, indicando que de la experiencia acumulada hasta un punto dado se van a formar mini-batches de tamaño 128 para la actualización de los pesos de la red en linea.

Para este experimento se permitió que la exploración continuara por un total de 480 episodios

de 252 pasos/días, resultando así en una ejecución de alrededor de 121000 pasos. El tiempo de ejecución de este entrenamiento fue de aproximadamente 13 horas.

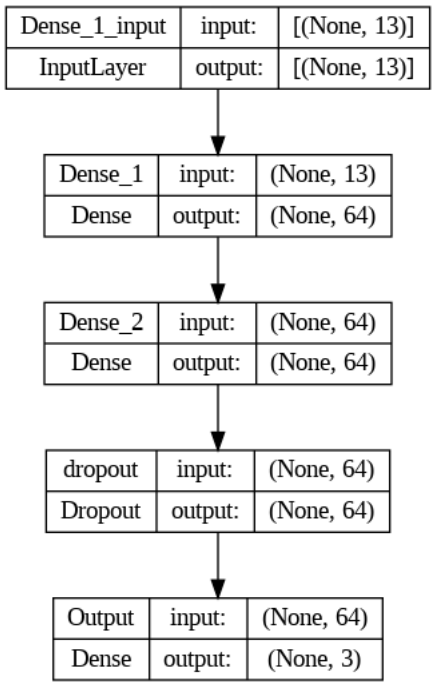


Figura 4.5: Arquitectura red neuronal experimento 2

4.3.3.3. Experimento 3

Los valores de los hiper-parámetros de este experimento se encuentran en la tabla 4.4. En primer lugar, se tiene una arquitectura de 3 capas ocultas de 256, 128 y 64 nodos, respectivamente. Adicionalmente, el modelo también está compuesto por una última capa de regularización para reducir el overfitting (ver sección 4.1.2 para más detalles). La figura 4.6 muestra la dimensión de entrada y de salida de cada capa. En segundo lugar, tenemos una tasa de aprendizaje de 0,0001, la cual, al igual que en el experimento 1, resultará en actualizaciones pequeñas de los pesos del modelo durante cada iteración de entrenamiento. En tercer lugar, tenemos un factor de regularización $l2$ de 0,001, el cual al ser un factor más grande que en los experimentos anteriores, resultará en actualizaciones de pesos ligeramente más grandes en el modelo. En cuarto lugar, tenemos una capacidad de reproducción de experiencia de 10000, indicando que los agentes tendrán acceso a las ultimas 10000 transiciones para formar los mini-batches aleatorios durante su entrenamiento. Finalmente, tenemos un tamaño de batch de 2048, indicando que de la experiencia acumulada hasta un punto dado se van a formar mini-batches de tamaño 2048 para la actualización de los pesos de la red en linea.

Para este experimento se permitió que la exploración continuara por un total de 480 episodios

de 252 pasos/días, resultando así en una ejecución de alrededor de 121000 pasos. El tiempo de ejecución de este entrenamiento fue de aproximadamente 13 horas.

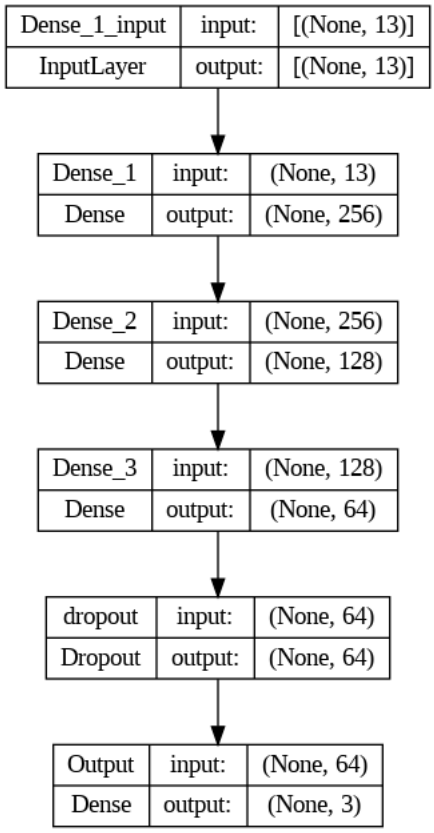


Figura 4.6: Arquitectura red neuronal experimento 3

4.3.3.4. Experimento 4

Los valores de los hiper-parámetros de este experimento se encuentran en la tabla 4.5. En primer lugar, se tiene una arquitectura de 2 capas ocultas de 256, nodos cada una. Adicionalmente, el modelo también está compuesto por una última capa de regularización para reducir el overfitting (ver sección 4.1.2 para más detalles). La figura 4.7 muestra la dimensión de entrada y de salida de cada capa. En segundo lugar, tenemos una tasa de aprendizaje de 0,0001, la cual, al igual que en el experimento 1 y 3, resultará en actualizaciones pequeñas de los pesos del modelo durante cada iteración de entrenamiento. En tercer lugar, tenemos un factor de regularización $l2$ de 0,000001, el cual resultará en actualizaciones pequeñas de los pesos en el modelo. En cuarto lugar, tenemos una capacidad de reproducción de experiencia de 1000000, indicando que los agentes tendrán acceso a las ultimas 1000000 transiciones para formar los mini-batches aleatorios durante su entrenamiento. Finalmente, tenemos un tamaño de batch de 4096, indicando que de la experiencia acumulada hasta un punto dado se van a formar mini-batches de tamaño 4096 para la actualización de los pesos de

la red en linea.

Para este experimento se permitió que la exploración continuara por un total de 960 episodios de 252 pasos/días, resultando así en una ejecución de alrededor de 242000 pasos. El tiempo de ejecución de este entrenamiento fue de aproximadamente 30 horas.

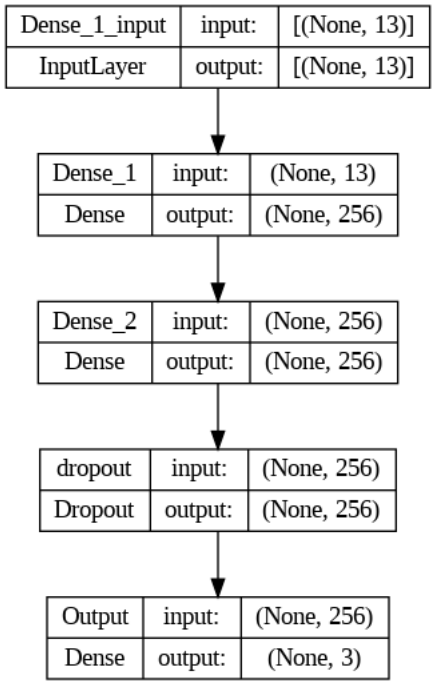


Figura 4.7: Arquitectura red neuronal experimento 4

Capítulo 5

Resultados

En este capitulo se pone un énfasis particular en los resultados de nuestro cuarto experimento, el cual es descrito en la sección 4.3.3.4. Es importante destacar que los tres experimentos iniciales, si bien valiosos por sí mismos, cumplían un propósito distinto en el desarrollo iterativo de nuestro agente de trading. Estos experimentos preliminares, diseñados para explorar y comprender mejor los matices del aprendizaje por refuerzo en el contexto financiero, se llevaron a cabo con un número limitado de episodios de entrenamiento. Esta elección se debió a la necesidad de gestionar eficazmente el tiempo y los recursos, ya que el entrenamiento de agentes de aprendizaje por refuerzo en entornos complejos es una tarea que consume tiempo. Su principal objetivo era descubrir tendencias emergentes y obtener una comprensión cualitativa del comportamiento del agente en diferentes configuraciones. A través de estos experimentos iniciales, pudimos adquirir valiosas intuiciones que, a su vez, informaron nuestras decisiones clave en términos de arquitectura del agente y estrategia de entrenamiento. Resultados para estos experimentos se presentan como referencia para el lector en el anexo B. No obstante, es el cuarto experimento el que asume un papel central en nuestra investigación, ya que en este asignamos recursos sustanciales y extendimos significativamente la duración del entrenamiento, lo que permitió al agente acumular una experiencia profunda, perfeccionar sus estrategias y adaptarse a las dinámicas del mercado. Como resultado, el cuarto experimento representa la culminación de nuestros esfuerzos y brinda una evaluación exhaustiva y sólida de nuestro enfoque de aprendizaje por refuerzo en el contexto del trading. Sus resultados, por lo tanto, proporcionan la base más sólida para evaluar las capacidades del agente y su potencial aplicación práctica en los mercados financieros reales. En particular, en este capitulo se van a presentar resultados de:

1. Rendimientos
2. Recompensas
3. Costos
4. Operaciones de trading
5. Función de pérdida

5.1. Rendimientos

Los rendimientos se refieren a las ganancias (o pérdidas) de capital que se lograron obtener en periodo dado. En nuestro caso, se hace una evaluación sobre los valores de retorno acumulativo

(NAVs) obtenidos por los agentes y por mercado, donde estos últimos representan la ejecución de la estrategia *buy-and-hold*. En particular, se hará un análisis de:

1. Rendimientos anuales logrados al final de cada episodio.

2. Promedio de los rendimientos logrados cada día durante un episodio.

3. Proporción en la que los agentes superaron la estrategia *buy-and-hold*.

5.1.1. Apple Inc

La figura 5.1 muestra la media móvil sobre los últimos 100 episodios de los valores de retorno acumulativo para los 960 periodos de entrenamiento, así como también la media móvil de la proporción de los últimos 100 episodios en los que el agente superó la estrategia *buy-and-hold*. De manera similar, en la figura 5.2 se muestra, bajo esta misma configuración, el promedio de los valores de retorno acumulativo obtenidos cada día durante un episodio. Finalmente, en la figura 5.3 se muestra la media móvil sobre los últimos 50 episodios de la proporción de días en los que el agente superó la estrategia *buy-and-hold* durante un episodio.

Con respecto a la figura 5.1 podemos observar que a partir del episodio 600 nuestro agente siempre termina con un rendimiento acumulado que sobrepasa a la estrategia *buy-and-hold* (lado izquierdo). Esto indica que el agente esta capturando correctamente las tendencias del mercado realizando operaciones de trading diarias, a diferencia de la estrategia *buy-and-hold* cuyo rendimiento depende únicamente de la valorización del activo lograda durante cada episodio. De igual forma, notamos que a partir del episodio 500 nuestro agente tiene una proporción mayor de ganancias con respecto a la estrategia *buy-and-hold* (lado derecho). Esto nos indica que la proporción superior de ganancias de nuestro agente tuvo una influencia en los rendimientos superiores generados durante cada episodio. Aunque estas dos métricas van muy de la mano, una no implica a la otra, por lo cual puede ocurrir un caso en el que se tenga un rendimiento acumulado que sobrepase a la estrategia *buy-and-hold*, pero que la proporción de ganancias no lo haga. Un ejemplo de esto se presenta más adelante, lo cual va a requerir un análisis más granular de los rendimientos obtenidos. La figura 5.2 corrobora las tendencias observadas en la figura 5.1, pero esta vez llevando un registro de los rendimientos anuales promedio logrados durante cada episodio (en lugar de los acumulados). El fin de tener un registro de ambos indicadores es poder analizar posibles resultados incongruentes o inconsistentes, los cuales en este caso no se presentan. Finalmente, la figura 5.3 nos muestra una proporción de ganancias entre episodios (*ganancias en episodio/pasos totales*), la cual presenta una tendencia creciente a medida que el agente avanza en los episodios de entrenamiento.

De manera general se puede decir que nuestro agente logró capturar muy bien las tendencias del mercado para este activo, lo cual se tradujo en una toma de decisiones inteligente que lo llevó a sobrepasar las retribuciones que se lograron con una estrategia *buy-and-hold*.

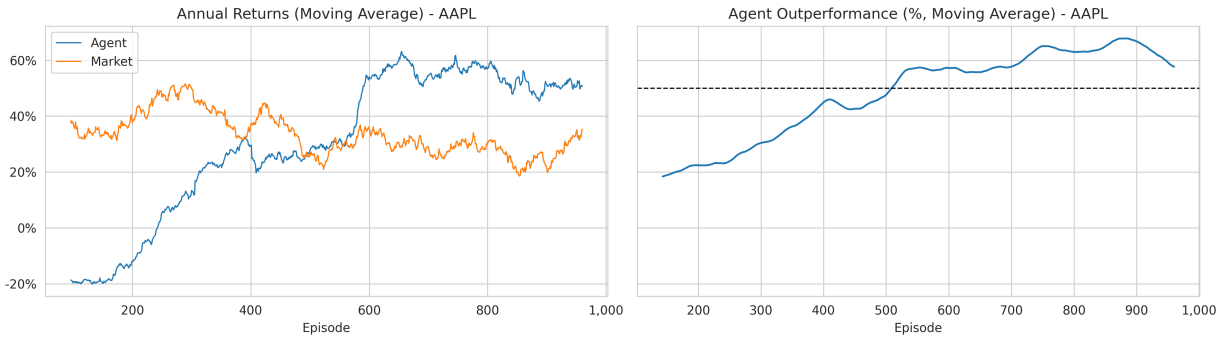


Figura 5.1: Media móvil de rendimientos anuales

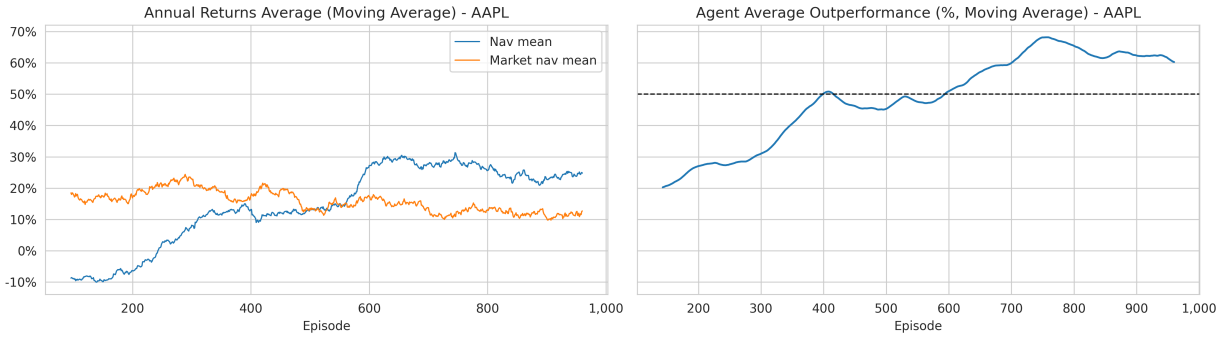


Figura 5.2: Media móvil de rendimientos anuales promedio

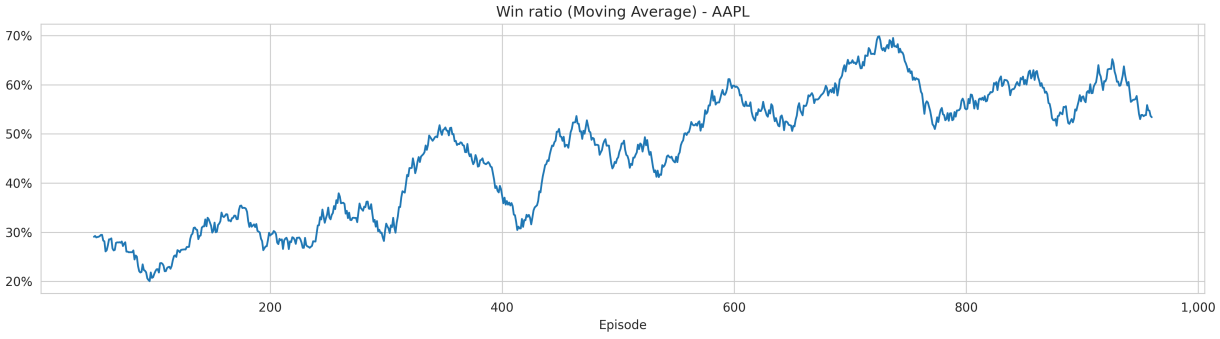


Figura 5.3: Media móvil de proporción de ganancias

5.1.2. Microsoft

La figura 5.4 muestra la media móvil sobre los últimos 100 episodios de los valores de retorno acumulativo para los 960 periodos de entrenamiento, así como también la media móvil de la proporción de los últimos 100 episodios en los que el agente superó la estrategia *buy-and-hold*. De manera similar, en la figura 5.5 se muestra, bajo esta misma configuración, el promedio de los valores de retorno acumulativo obtenidos cada día durante un episodio. Finalmente, en la figura 5.6 se muestra la media móvil sobre los últimos 50 episodios de la proporción de días en los que el agente superó la estrategia *buy-and-hold* durante un episodio.

Con respecto a la figura 5.4 podemos observar que aunque existe una tendencia de subida en el rendimiento acumulado de nuestro agente, este no logra sobrepasar los rendimientos logrados por la estrategia *buy-and-hold* (lado izquierdo). Esto indica que es probable que se necesiten más episodios de entrenamiento que lleven a que se obtengan retribuciones superiores, ya que se puede notar una aproximación muy cercana a la estrategia en los últimos episodios. De igual forma, la proporción de ganancias de nuestro agente tampoco logra superar a la estrategia *buy-and-hold* (lado derecho), lo cual hace sentido con base a los rendimientos inferiores obtenidos por el agente para este activo. La figura 5.5 corrobora las tendencias observadas en la figura 5.4, pero esta vez llevando un registro de los rendimientos anuales promedio logrados durante cada episodio (en lugar de los acumulados). Finalmente, la figura 5.6 nos muestra que la proporción de ganancias en cada episodio (*ganancias en episodio/pasos totales*) tiene una tendencia creciente a medida que el agente avanza en los episodios de entrenamiento, lo cual sustenta más fuertemente la idea de que con más episodios de entrenamiento es posible que el agente obtenga retribuciones superiores a la estrategia *buy-and-hold*.

De manera general se puede decir que aunque nuestro agente no logró superar los rendimientos obtenidos por la estrategia *buy-and-hold* para este activo, este mostró una tendencia de mejora a lo largo de los episodios de entrenamiento, lo cual sugiere que más episodios podrían resultar en retribuciones que sobrepasen a la estrategia.

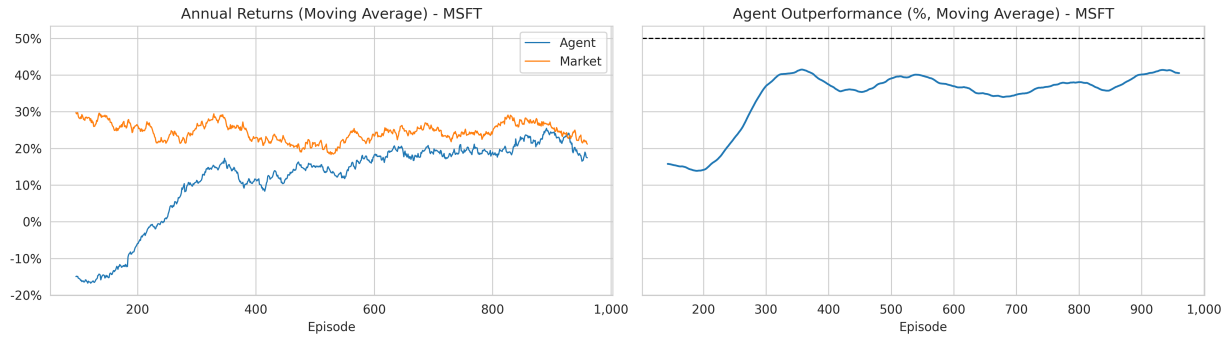


Figura 5.4: Media móvil de rendimientos anuales

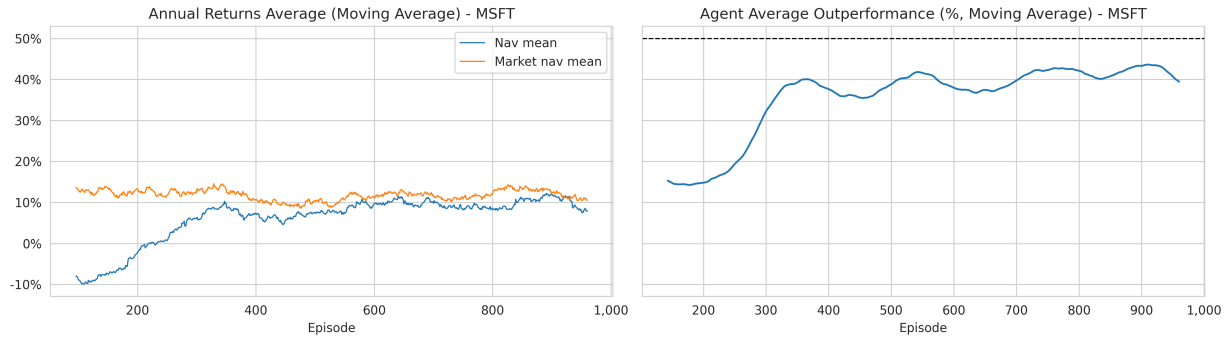


Figura 5.5: Media móvil de rendimientos anuales promedio

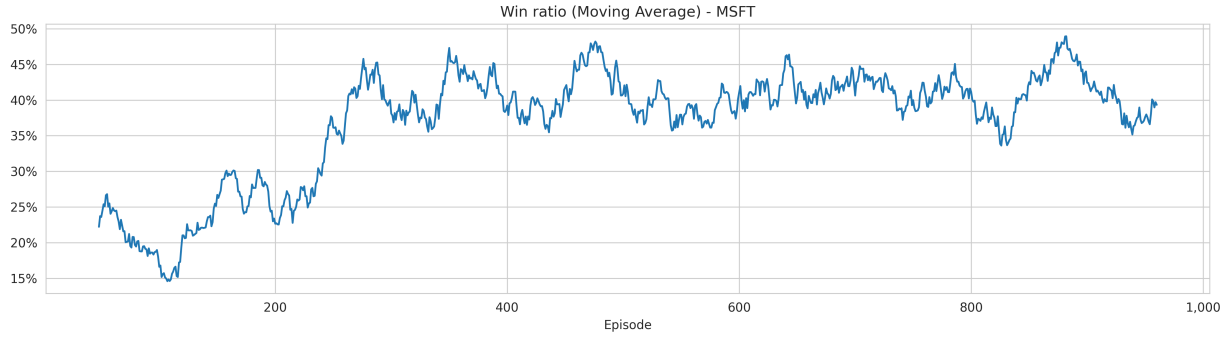


Figura 5.6: Media móvil de proporción de ganancias

5.1.3. Amazon Inc

La figura 5.7 muestra la media móvil sobre los últimos 100 episodios de los valores de retorno acumulativo para los 960 periodos de entrenamiento, así como también la media móvil de la proporción de los últimos 100 episodios en los que el agente superó la estrategia *buy-and-hold*. De manera similar, en la figura 5.9 se muestra, bajo esta misma configuración, el promedio de los valores de retorno acumulativo obtenidos cada día durante un episodio. Finalmente, en la figura 5.10 se muestra la media móvil sobre los últimos 50 episodios de la proporción de días en los que el agente superó la estrategia *buy-and-hold* durante un episodio.

Con respecto a la figura 5.7 podemos observar que a partir del episodio 500 nuestro agente siempre termina con un rendimiento acumulado que sobrepasa a la estrategia *buy-and-hold* (lado izquierdo). Sin embargo, podemos notar que la proporción de ganancias de nuestro agente con respecto a la estrategia no sobrepasa el 50 % en ningún momento (lado derecho), lo cual puede ser explicado con ayuda de la figura 5.8. En esta figura podemos observar que en ciertos episodios existen picos altos de retribuciones acumuladas de nuestro agente (especialmente después del episodio 500), los cuales hacen que la media móvil de retribuciones acumuladas se vea beneficiada sin

que necesariamente la proporción de ganancias entre episodios sea el factor que influya en estos rendimientos superiores generados durante cada episodio. La figura 5.9 corrobora las tendencias observadas en la figura 5.7, pero esta vez llevando un registro de los rendimientos anuales promedio logrados durante cada episodio (en lugar de los acumulados). Finalmente, la figura 5.10 nos muestra que la proporción de ganancias del agente con respecto a la estrategia *buy-and-hold* en cada episodio (*ganancias en episodio/pasos totales*) tiene una tendencia ligeramente creciente a medida que el agente avanza en los episodios de entrenamiento. Sin embargo, podemos observar una caída abrupta de este indicador entre los episodios 400 y 500.

De manera general se puede decir que nuestro agente realizó una toma decisiones que contribuyó positivamente a que los rendimientos acumulados anuales sobrepasaran a la estrategia *buy-and-hold*.

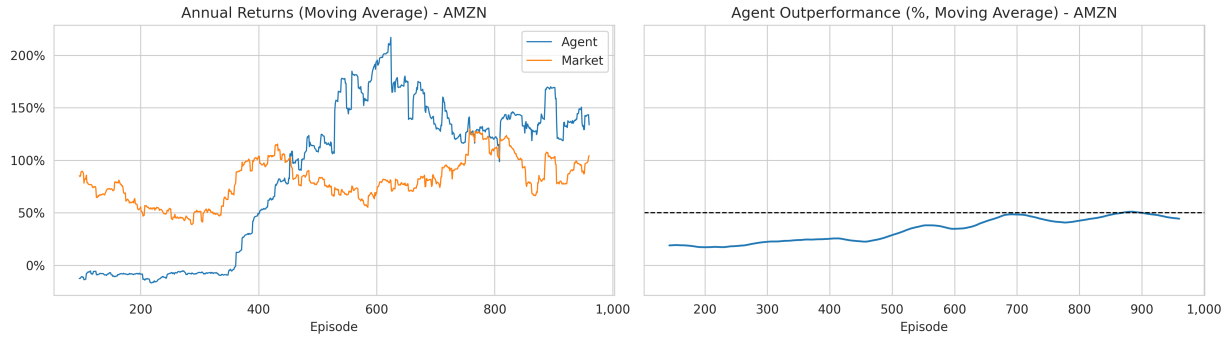


Figura 5.7: Media móvil de rendimientos anuales

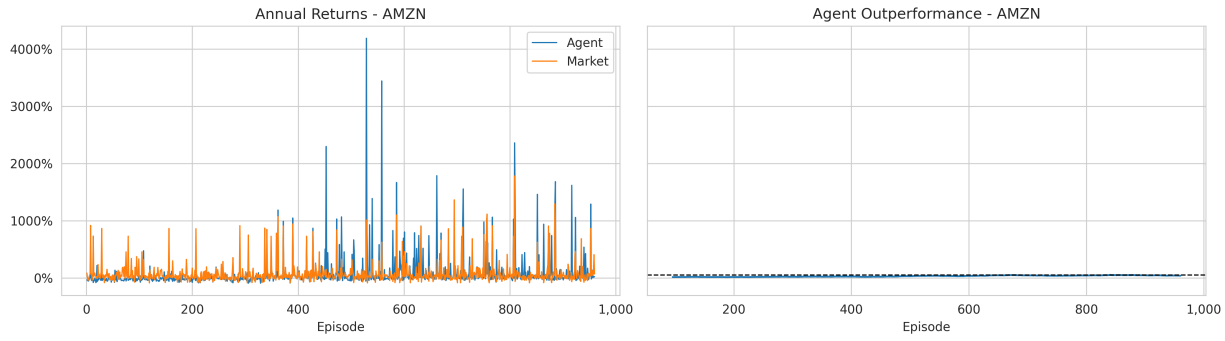


Figura 5.8: Rendimientos anuales

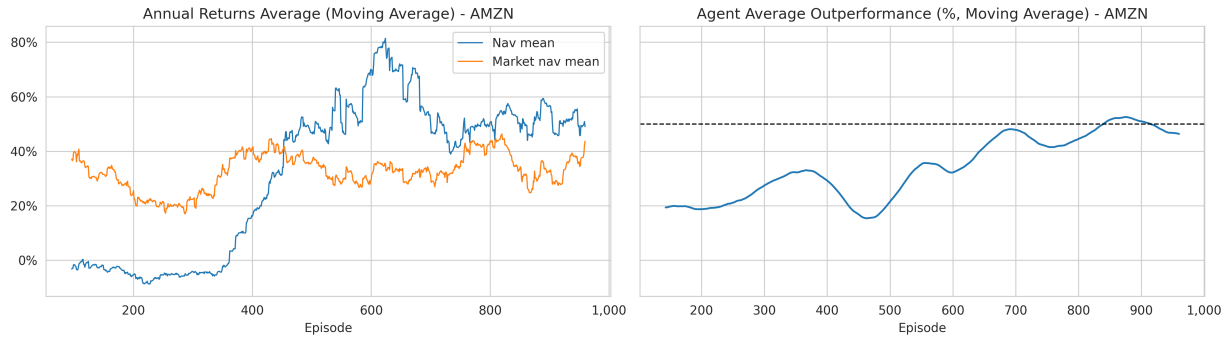


Figura 5.9: Media móvil de rendimientos anuales promedio

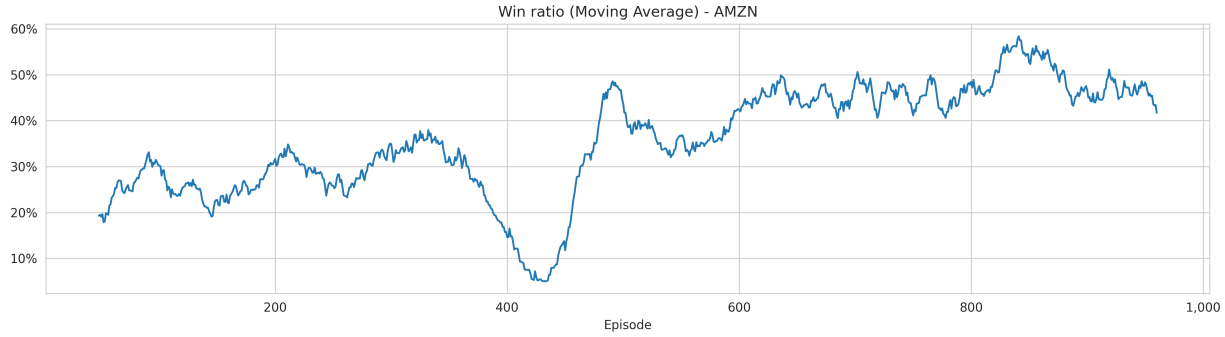


Figura 5.10: Media móvil de proporción de ganancias

5.1.4. Pepsico Inc

La figura 5.11 muestra la media móvil sobre los últimos 100 episodios de los valores de retorno acumulativo para los 960 periodos de entrenamiento, así como también la media móvil de la proporción de los últimos 100 episodios en los que el agente superó la estrategia *buy-and-hold*. De manera similar, en la figura 5.12 se muestra, bajo esta misma configuración, el promedio de los valores de retorno acumulativo obtenidos cada día durante un episodio. Finalmente, en la figura 5.13 se muestra la media móvil sobre los últimos 50 episodios de la proporción de días en los que el agente superó la estrategia *buy-and-hold* durante un episodio.

Con respecto a la figura 5.11 podemos observar que aunque existe una tendencia de subida en el rendimiento acumulado de nuestro agente, este no logra sobrepasar los rendimientos logrados por la estrategia *buy-and-hold* (lado izquierdo). De igual forma, la proporción de ganancias de nuestro agente tampoco logra superar a la estrategia en ningún episodio, pero la tendencia de aumento sigue estando presente. Esto nos puede sugerir que con un entrenamiento de más episodios se logre el objetivo de sobrepasar a la estrategia *buy-and-hold*. La figura 5.12 corrobora las tendencias observadas en la figura 5.11, pero esta vez llevando un registro de los rendimientos anuales promedio logrados

durante cada episodio (en lugar de los acumulados). Finalmente, la figura 5.13 nos muestra que la proporción de ganancias en cada episodio (*ganancias en episodio/pasos totales*) sigue manteniendo una tendencia creciente a medida que el agente avanza en los episodios de entrenamiento, lo cual sustenta más fuertemente la idea de que con más episodios de entrenamiento es posible que el agente obtenga retribuciones superiores a la estrategia *buy-and-hold*.

De manera general se puede decir que aunque nuestro agente no logró superar los rendimientos obtenidos por la estrategia *buy-and-hold*, este mostró una tendencia de mejora a lo largo de los episodios de entrenamiento, lo cual sugiere que más episodios de entrenamiento podrían eventualmente resultar en retribuciones que sobrepasen la estrategia.

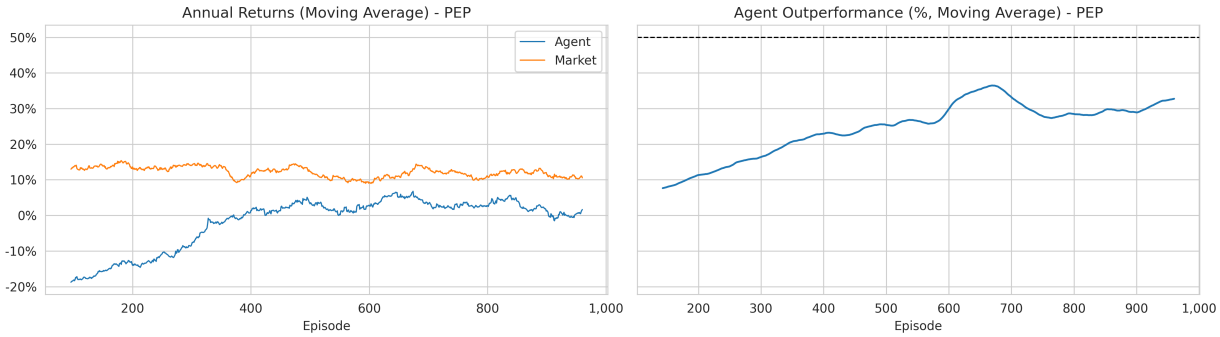


Figura 5.11: Media móvil de rendimientos anuales

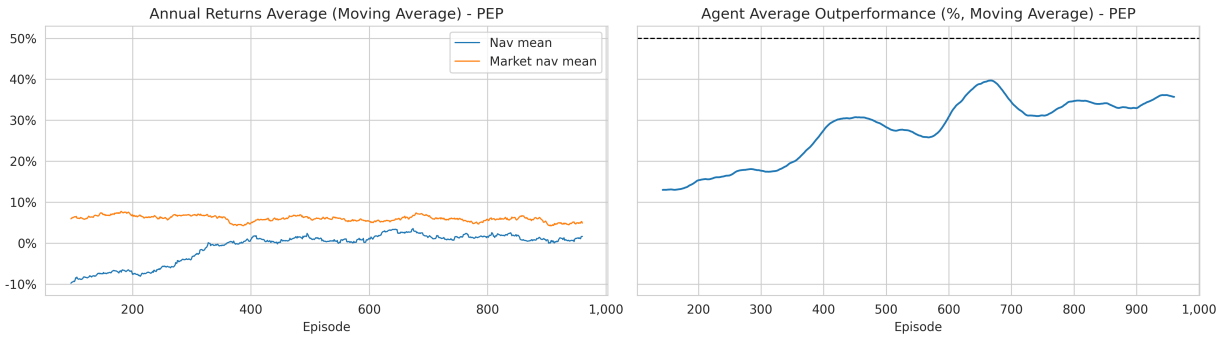


Figura 5.12: Media móvil de rendimientos anuales promedio

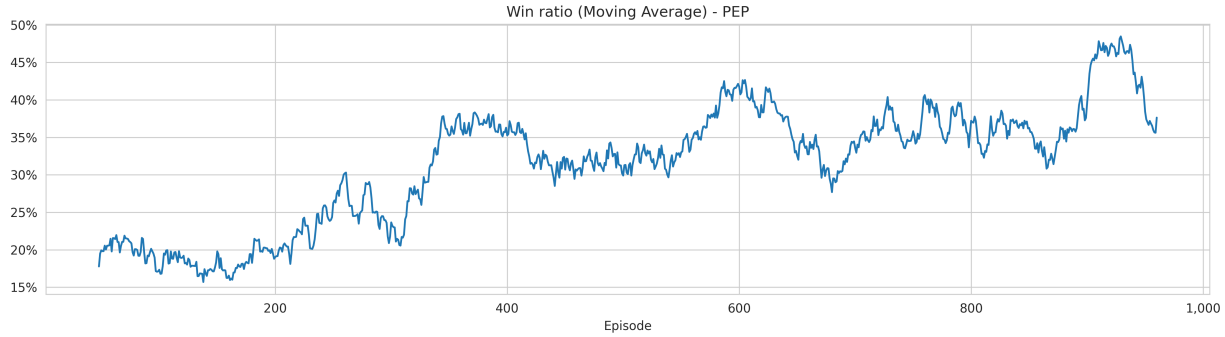


Figura 5.13: Media móvil de proporción de ganancias

5.2. Recompensas

Cada vez que los agentes realizan una acción sobre el ambiente, estos reciben una recompensa que trata de cuantificar que tan buena fue esa acción con base a los rendimientos que se obtuvieron. Un valor mayor en la función de recompensa representa un mejor rendimiento. En la sección 4.2.2 se explica a mayor detalle el diseño de esta función y el rol tan importante que tiene dentro del ambiente. En esta sección se va a realizar una evaluación de las recompensas obtenidas, y en particular se presentarán resultados para:

1. La suma acumulada de recompensas obtenidas durante un episodio.
2. El promedio de recompensas obtenidas durante un episodio.
3. La diferencia entre la mayor y menor recompensa obtenida durante un episodio.

5.2.1. Apple Inc

La figura 5.14 muestra la media móvil sobre los últimos 50 episodios de la suma acumulada de recompensas obtenidas durante un episodio para los 960 periodos de entrenamiento. De manera similar, en la figura 5.15 se muestra, bajo esta misma configuración, el promedio de recompensas obtenidas durante un episodio. Finalmente, en la figura 5.16 se muestra la media móvil sobre los últimos 50 episodios de la diferencia entre la mayor y menor recompensa obtenidas durante un episodio.

Con base a estos resultados y a los rendimientos obtenidos para este activo, podemos concluir que la tendencia de incremento en la suma acumulada y promedio de recompensas indica que el agente está mejorando gradualmente su toma decisiones. Así mismo, el agente está adaptando su estrategia en función de experiencias pasadas y aprendiendo de sus errores, ya que identifica patrones en los datos del mercado y ajusta sus acciones para aprovechar oportunidades rentables mientras evita las no rentables. En este caso en particular, estas recompensas ayudaron al agente a navegar por el ambiente hasta un punto de sobrepasar a la estrategia *buy-and-hold*, lo cual representa un

gran logro en este trabajo. Por otro lado, los resultados de la figura 5.16 nos indican que el agente está asumiendo niveles de riesgo volátiles durante cada episodio, lo cual, aunque en este caso no afecta significativamente en las retribuciones obtenidas, sugiere una comprensión mas detallada de la exposición al riego para lograr una estrategia más estable y robusta.

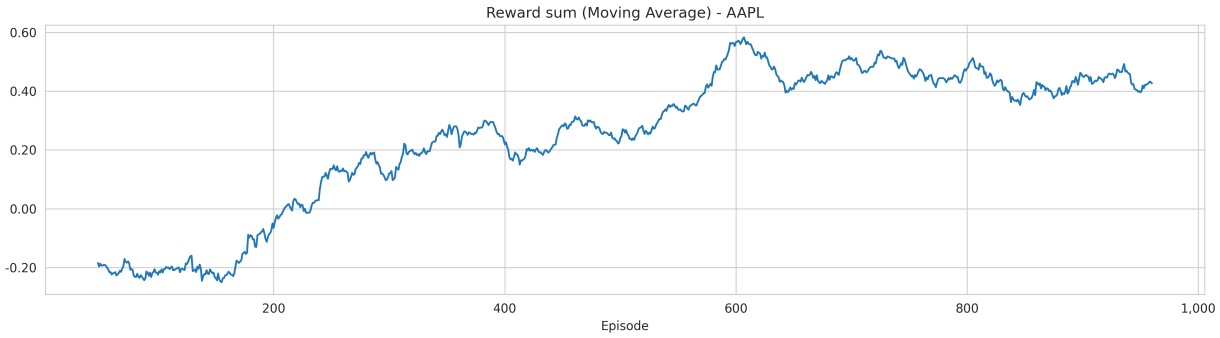


Figura 5.14: Media móvil de suma acumulada de recompensas

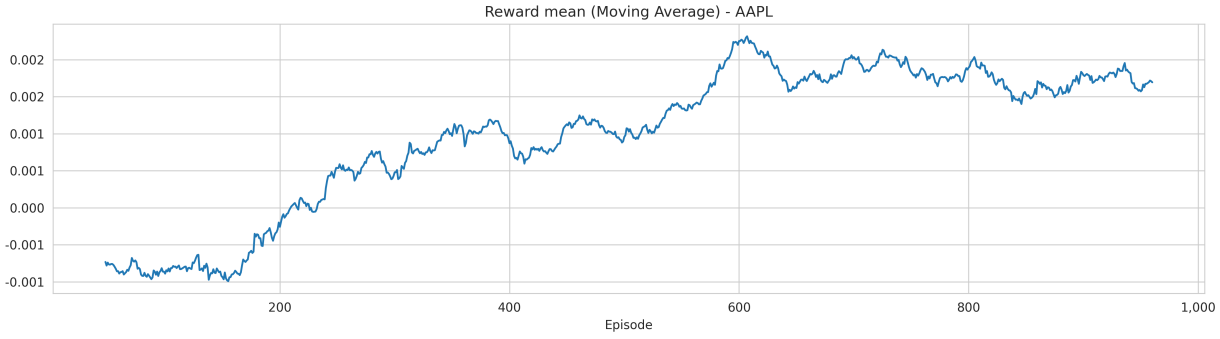


Figura 5.15: Media móvil de promedio de recompensas

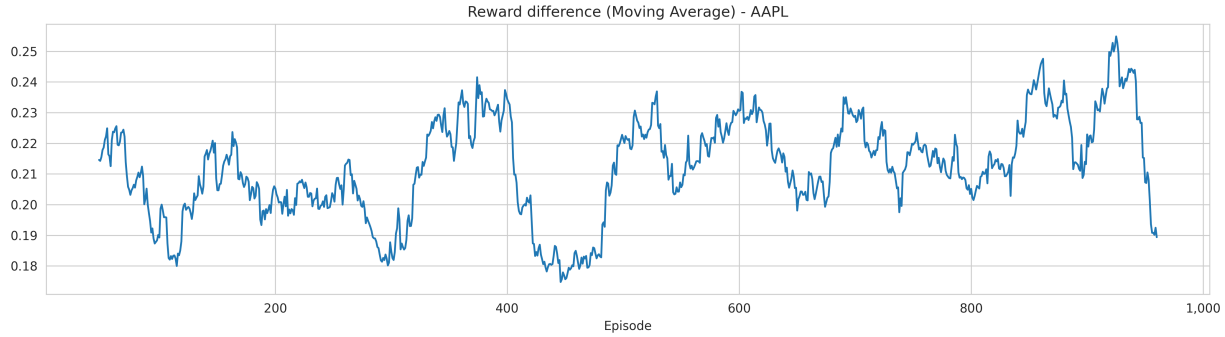


Figura 5.16: Media móvil de diferencia de recompensa

5.2.2. Microsoft

La figura 5.17 muestra la media móvil sobre los últimos 50 episodios de la suma acumulada de recompensas obtenidas durante un episodio para los 960 periodos de entrenamiento. De manera similar, en la figura 5.18 se muestra, bajo esta misma configuración, el promedio de recompensas obtenidas durante un episodio. Finalmente, en la figura 5.19 se muestra la media móvil sobre los últimos 50 episodios de la diferencia entre la mayor y menor recompensa obtenidas durante un episodio.

Con base a estos resultados y a los rendimientos obtenidos para este activo, nos encontramos en una situación similar a la anterior, ya que observamos una tendencia de incremento en la suma acumulada y promedio de recompensas, sugiriendo así un mejoramiento continuo en la toma de decisiones del agente. En este caso en particular, las recompensas obtenidas no lograron que el agente pueda sobrepasar a la estrategia *buy-and-hold*, pero claramente demostraron una gran mejora en su rendimiento a lo largo del tiempo. Por otro lado, los resultados de la figura 5.19 también nos sugieren en este caso una exposición al riesgo inestable por parte del agente. Como se podrá observar más adelante, esto es un factor común para todos los activos explorados en este trabajo. Esta volatilidad en los niveles de riesgo se debe principalmente a que no se integran estrategias de gestión de riesgo en la función de recompensa, lo cual hace que los riegos inherentes asociados con las ventas en corto no se tomen en cuenta. En la sección 6.2.2 se exponen estrategias de manejo de riesgo que ayudarían a disminuir la volatilidad de los episodios.

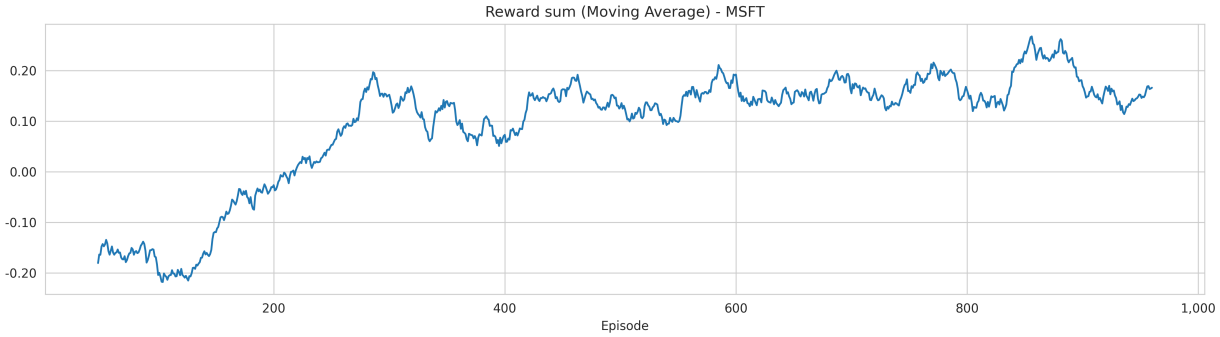


Figura 5.17: Media móvil de suma acumulada de recompensas

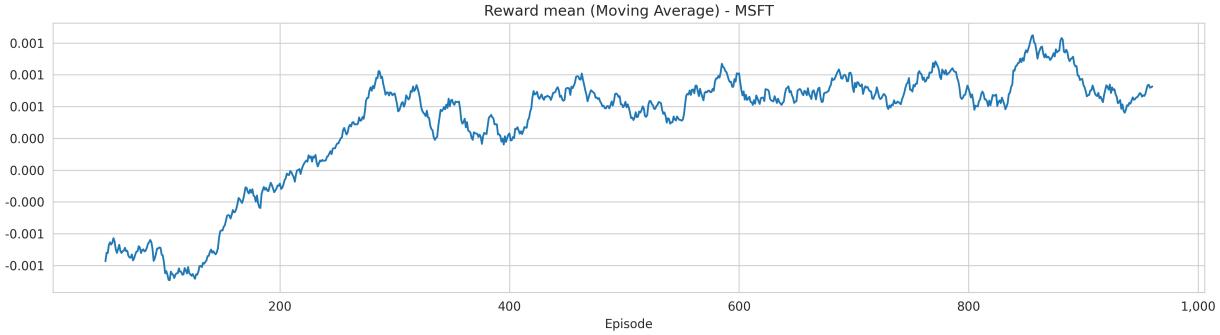


Figura 5.18: Media móvil de promedio de recompensas

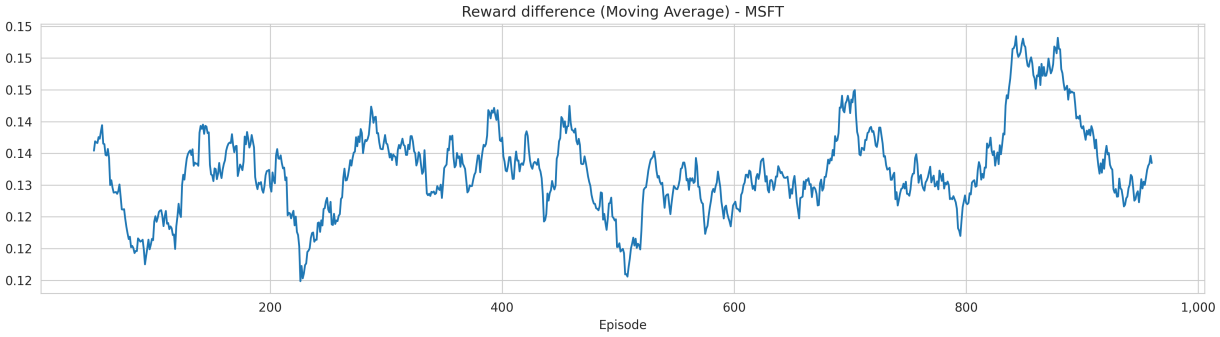


Figura 5.19: Media móvil de diferencia de recompensa

5.2.3. Amazon Inc

La figura 5.20 muestra la media móvil sobre los últimos 50 episodios de la suma acumulada de recompensas obtenidas durante un episodio para los 960 periodos de entrenamiento. De manera similar, en la figura 5.21 se muestra, bajo esta misma configuración, el promedio de recompensas obtenidas durante un episodio. Finalmente, en la figura 5.22 se muestra la media móvil sobre los últimos 50 episodios de la diferencia entre la mayor y menor recompensa obtenidas durante un episodio.

Con base a los resultados y a los rendimientos obtenidos para este activo, se puede observar una tendencia de incremento significativa en la suma acumulada y promedio de recompensas. Esto indica una mejora continua en la toma de decisiones del agente, lo cual se traduce en rendimientos acumulados que sobrepasan a la estrategia *buy-and-hold*. Por otro lado, la figura 5.22 una vez más nos demuestra una exposición al riesgo inestable por parte del agente en la mayoría de episodios, pero que en este caso tiende a estabilizarse ligeramente en los últimos episodios, lo cual no ocurrió con lo otros activos.

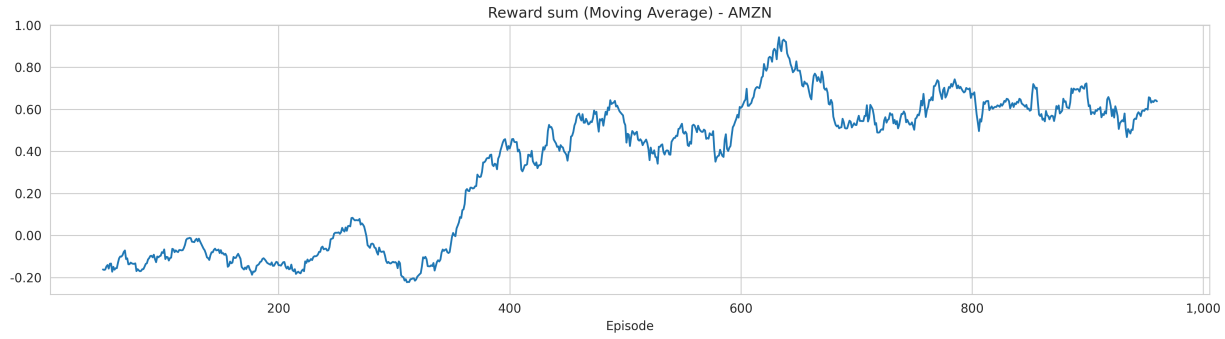


Figura 5.20: Media móvil de suma acumulada de recompensas

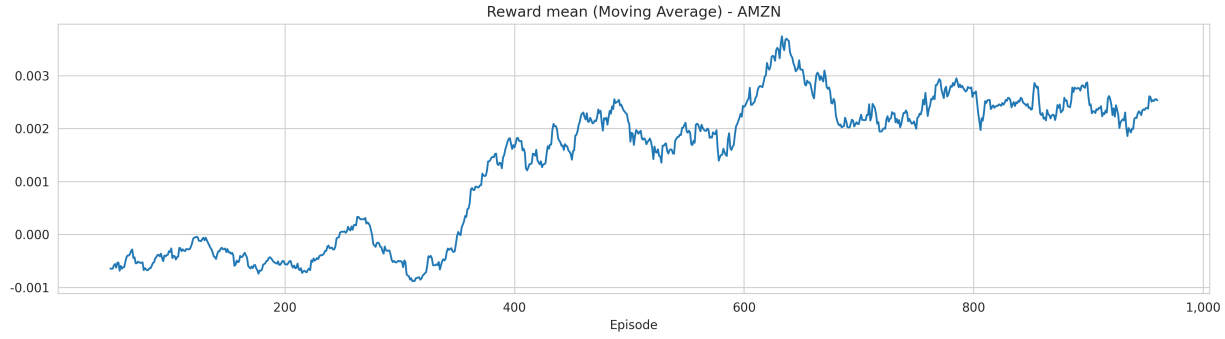


Figura 5.21: Media móvil de promedio de recompensas

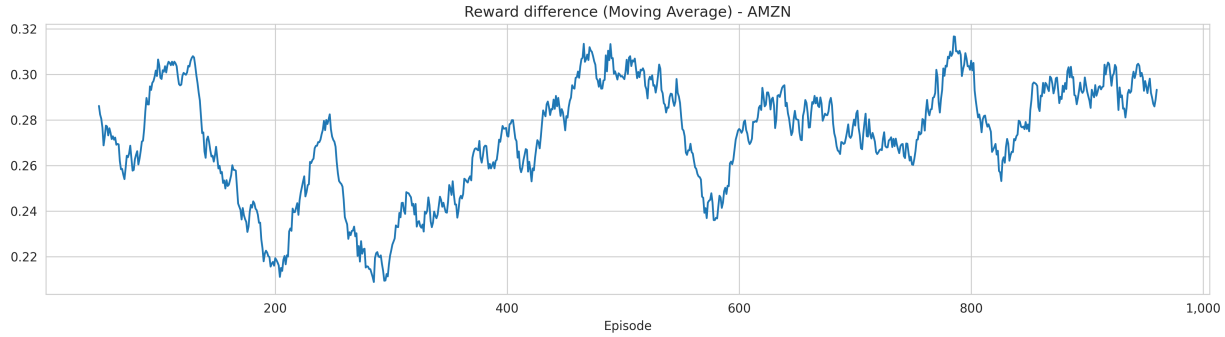


Figura 5.22: Media móvil de diferencia de recompensa

5.2.4. Pepsico Inc

La figura 5.23 muestra la media móvil sobre los últimos 50 episodios de la suma acumulada de recompensas obtenidas durante un episodio para los 960 periodos de entrenamiento. De manera similar, en la figura 5.24 se muestra, bajo esta misma configuración, el promedio de recompensas obtenidas durante un episodio. Finalmente, en la figura 5.25 se muestra la media móvil sobre los últimos 50 episodios de la diferencia entre la mayor y menor recompensa obtenidas durante un episodio.

Con base a los resultados y rendimientos obtenidos para este activo, se puede observar que, aunque existe una tendencia de incremento en la suma acumulada y promedio de recompensas, los valores de recompensa son pequeños en comparación con los otros activos evaluados. Esto sustenta en gran medida la razón por la cual las retribuciones acumuladas para este activo no mejoraron significativamente a lo largo de los episodios. Sin embargo, es importante resaltar la tendencia de mejora que se logró obtener, la cual da luz a mejores retribuciones si los periodos de entrenamiento aumentan.

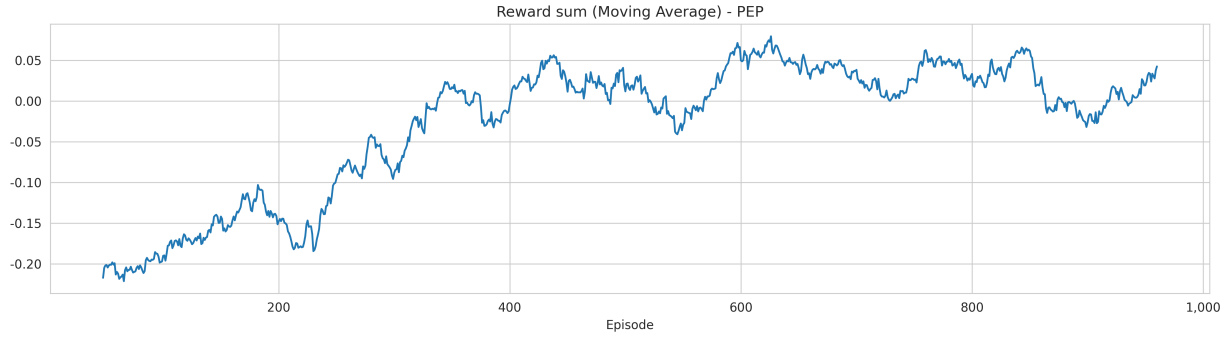


Figura 5.23: Media móvil de suma acumulada de recompensas

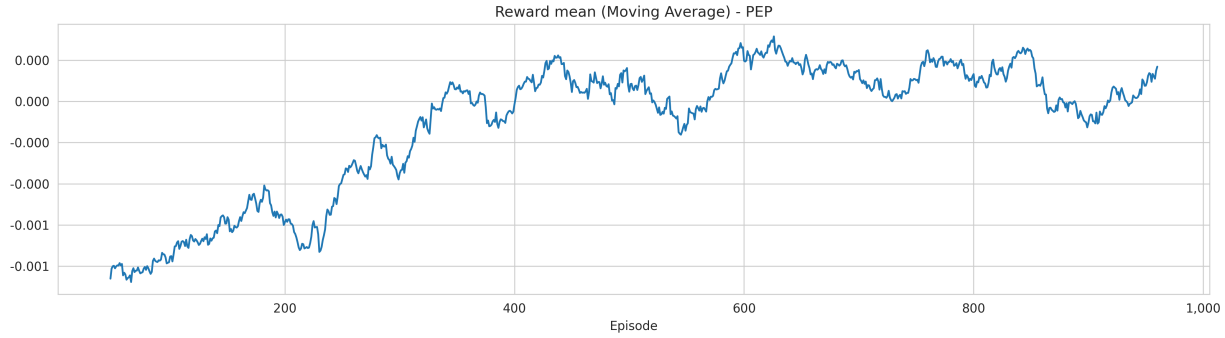


Figura 5.24: Media móvil de promedio de recompensas

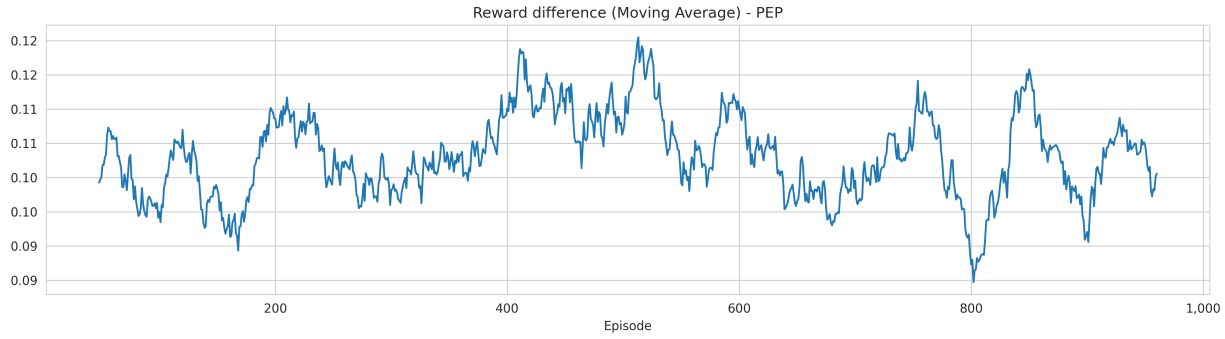


Figura 5.25: Media móvil de diferencia de recompensa

5.3. Costos

Cada acción que los agentes realizan en el ambiente implican un costo de trading (10bps), en caso de cambiar de posición, o un costo de tiempo (1bps) si se mantienen en la misma posición. Estos costos hacen parte de la función de recompensa descrita en la sección 4.2.2, lo cual hace particularmente importante observar el comportamiento que presentan durante el entrenamiento. En esta sección se presentarán resultados para:

1. La suma acumulada de costos obtenidos durante un episodio.

2. El promedio de costos obtenidos durante un episodio.

5.3.1. Apple Inc

La figura 5.26 muestra la media móvil sobre los último 50 episodios de la suma acumulada de costos obtenidos durante un episodio para los 960 periodos de entrenamiento. De manera similar, en

la figura 5.27 se muestra, bajo esta misma configuración, el promedio de costos obtenidos durante un episodio.

Con respecto a estos resultados podemos observar que la suma y promedio acumulado de costos presenta una tendencia decreciente hasta que el agente alcanza el episodio 300 en su entrenamiento. Después de esto, los costos de trading logran estabilizarse. Esto indica que que el agente logró encontrar un punto óptimo en los cambios de posiciones que realiza en cada paso durante un episodio, ya que se minimizan los costos y maximizan las recompensas. Esto se traduce en una tendencia de aumento en los rendimientos obtenidos, como se puede observar en la figura 5.1.

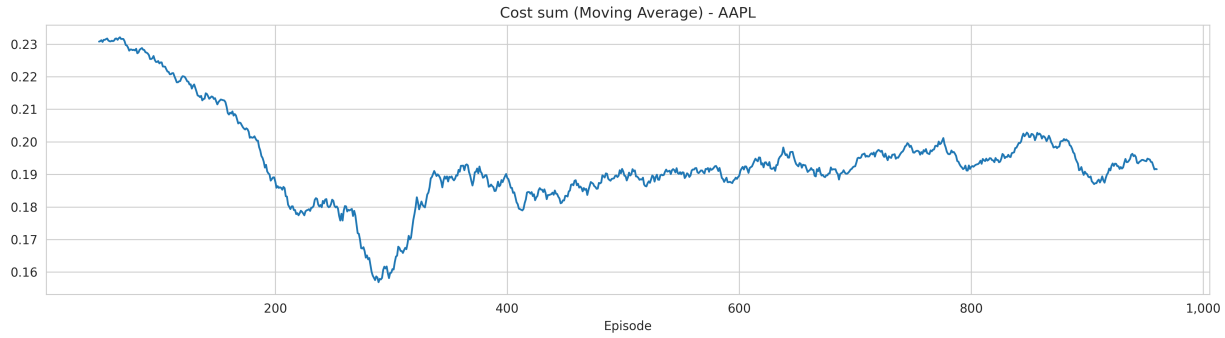


Figura 5.26: Media móvil de suma acumulada de costos

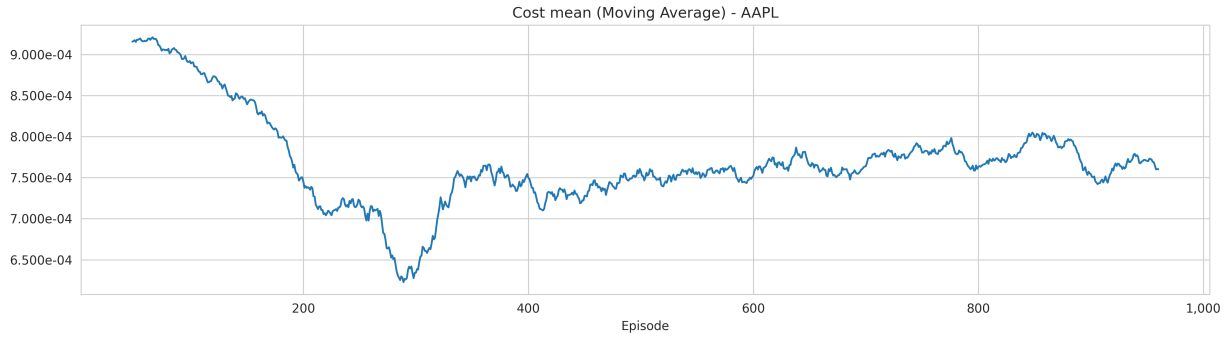


Figura 5.27: Media móvil de promedio de costos

5.3.2. Microsoft

La figura 5.28 muestra la media móvil sobre los último 50 episodios de la suma acumulada de costos obtenidos durante un episodio para los 960 periodos de entrenamiento. De manera similar, en la figura 5.29 se muestra, bajo esta misma configuración, el promedio de costos obtenidos durante un episodio.

Con respecto a estos resultados podemos observar un comportamiento similar al descrito en el

activo anterior, ya que se presenta una tendencia decreciente en la suma y promedio acumulado de costos de operaciones. Estos costos logran estabilizarse después del episodio 400 de entrenamiento del agente, lo cual indica una robustez en la estrategia que minimiza los costos y maximiza las recompensas. Aunque para este activo el agente no logra sobrepasar los rendimientos obtenidos por la estrategia *buy-and-hold*, estos resultados indican un buen camino de aprendizaje ya que el agente demuestra que está capturando correctamente las tendencias del mercado en términos de los cambios de posiciones diarias que realiza a lo largo de los episodios.

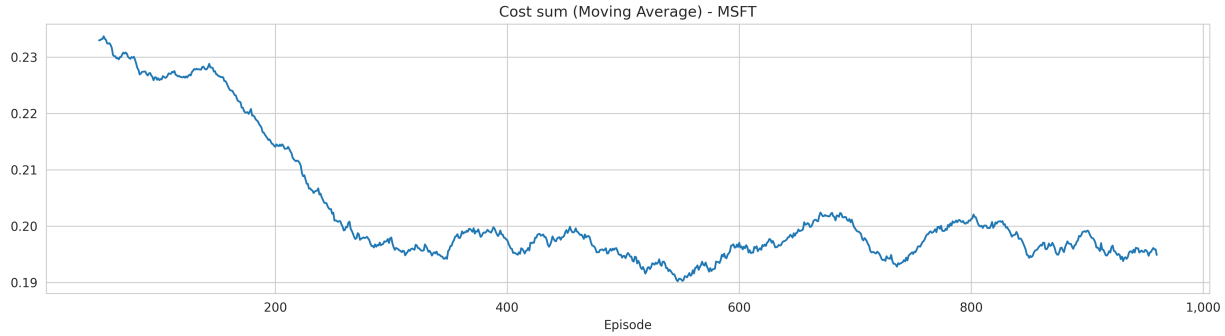


Figura 5.28: Media móvil de suma acumulada de costos

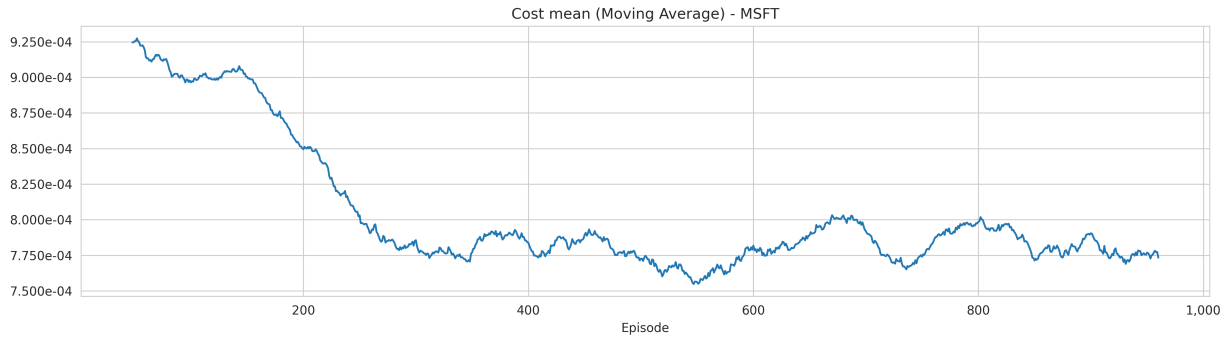


Figura 5.29: Media móvil de promedio de costos

5.3.3. Amazon Inc

La figura 5.30 muestra la media móvil sobre los último 50 episodios de la suma acumulada de costos obtenidos durante un episodio para los 960 periodos de entrenamiento. De manera similar, en la figura 5.31 se muestra, bajo esta misma configuración, el promedio de costos obtenidos durante un episodio.

Con respecto a estos resultados podemos observar que la suma y promedio acumulado de costos presenta una tendencia decreciente hasta que el agente alcanza el episodio 400 en su entrenamiento.

Después de esto, la suma de costos se estabiliza entre el rango de 0.10 y 0.12. Esto indica que el agente logró encontrar una configuración de cambio de posiciones que hace que la estrategia se vuelva estable, y al mismo tiempo haciendo que los costos se minimicen y las recompensas acumuladas aumenten. Finalmente, esto se traduce en una tendencia de aumento en los rendimientos obtenidos, como se puede observar en la figura 5.7.

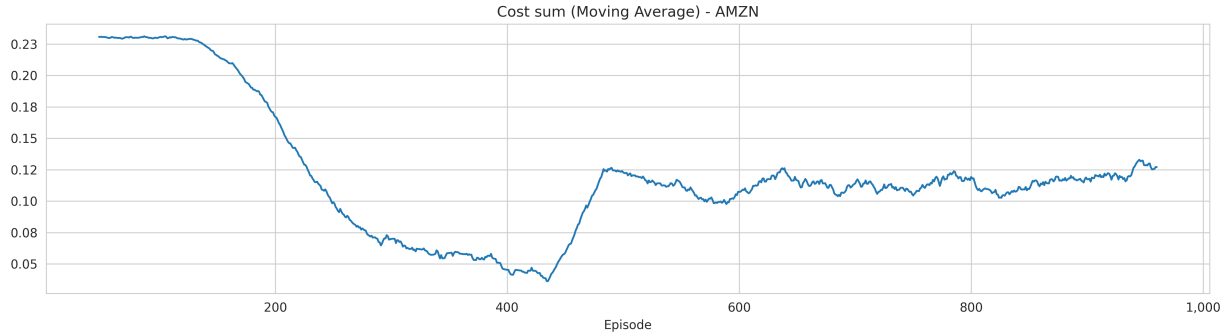


Figura 5.30: Media móvil de suma acumulada de costos

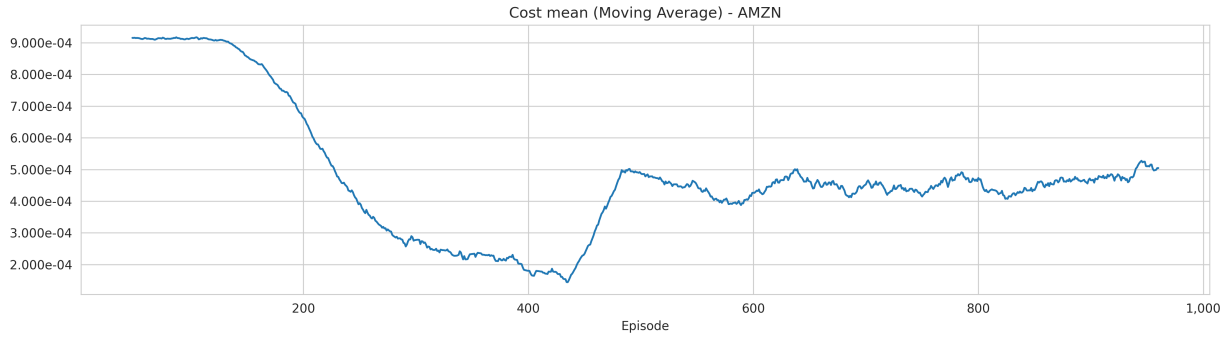


Figura 5.31: Media móvil de promedio de costos

5.3.4. Pepsico Inc

La figura 5.32 muestra la media móvil sobre los último 50 episodios de la suma acumulada de costos obtenidos durante un episodio para los 960 periodos de entrenamiento. De manera similar, en la figura 5.33 se muestra, bajo esta misma configuración, el promedio de costos obtenidos durante un episodio.

Con respecto a estos resultados podemos observar que la suma y promedio acumulado de costos presenta una tendencia decreciente a lo largo de todos los episodios de entrenamiento del agente, logrando estabilizarse ligeramente en los episodios finales. Esto sugiere que solo en los últimos episodios el agente estaba logrando encontrar una configuración de cambio de posiciones que hace que

la estrategia se vuelva estable, por lo cual los rendimientos y recompensas obtenidas no fueron muy buenas a lo largo de los episodios entrenamiento, como se puede observar en las figuras 5.11 y 5.23. Sin embargo, cabe destacar que esta tendencia de decrecimiento en los costos indica un aprendizaje continuo del agente, dando una luz a mejores rendimientos acumulados con un entrenamiento más extenso.

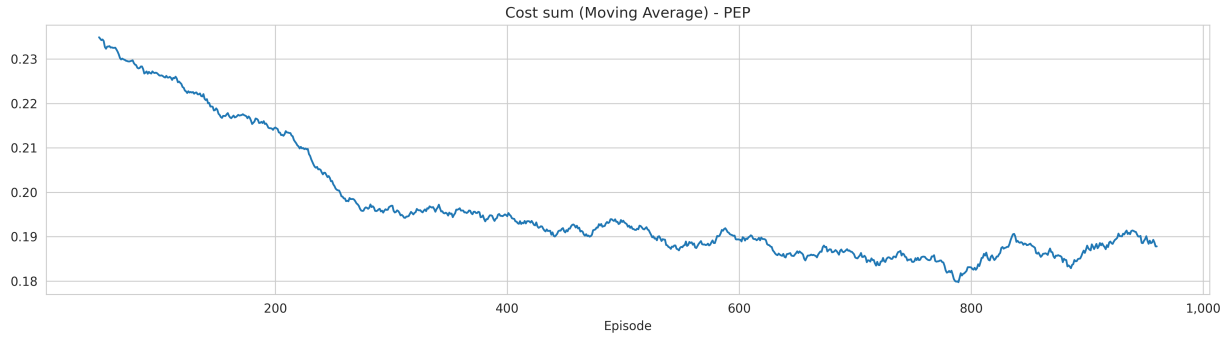


Figura 5.32: Media móvil de suma acumulada de costos

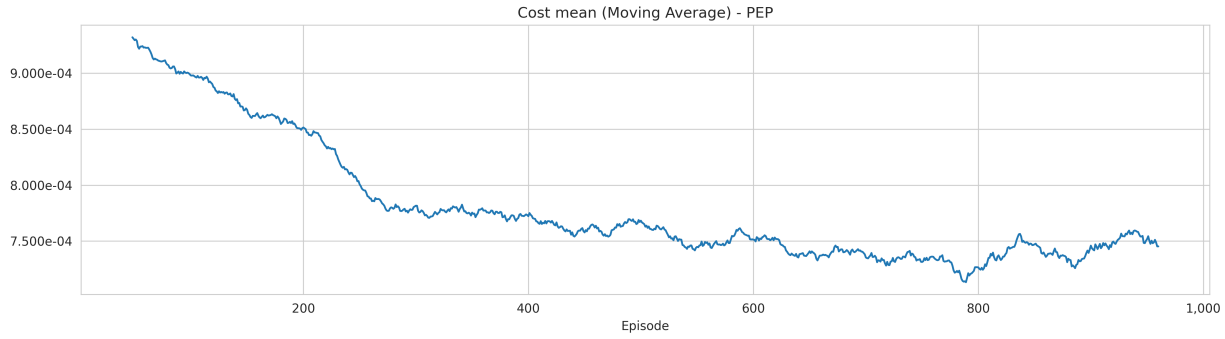


Figura 5.33: Media móvil de promedio de costos

5.4. Operaciones de trading

Las operaciones de trading reflejan los cambios de posición que los agentes realizan en cada paso durante su entrenamiento. El número de operaciones en un día dado puede tomar el valor de 0 si se mantiene la misma posición del día anterior, 1 si se cambia de una posición neutra a corta o larga (o viceversa), y 2 si se cambia de una posición corta a larga (o viceversa). En esta sección se presentarán resultados para:

1. La suma acumulada de todas las operaciones de trading realizadas durante un episodio.
2. El promedio del número de operaciones de trading realizadas cada día durante un episodio.

5.4.1. Apple Inc

La figura 5.34 muestra la media móvil sobre los últimos 50 episodios de la suma acumulada de operaciones de trading durante un episodio para los 960 periodos de entrenamiento. De manera similar, en la figura 5.35 se muestra, bajo esta misma configuración, el promedio de operaciones de trading realizadas cada día durante un episodio.

Con respecto a estos resultados podemos observar una estabilización de la suma acumulada de operaciones de trading a partir del episodio 400. Este resultado esta fuertemente relacionado con la suma acumulada de costos expuesta en la figura 5.26, ya que el número total de operaciones realizadas durante un episodio va a determinar los costos que se generan. Por esta razón se presenta esta similitud tan grande en el comportamiento de estos dos indicadores.

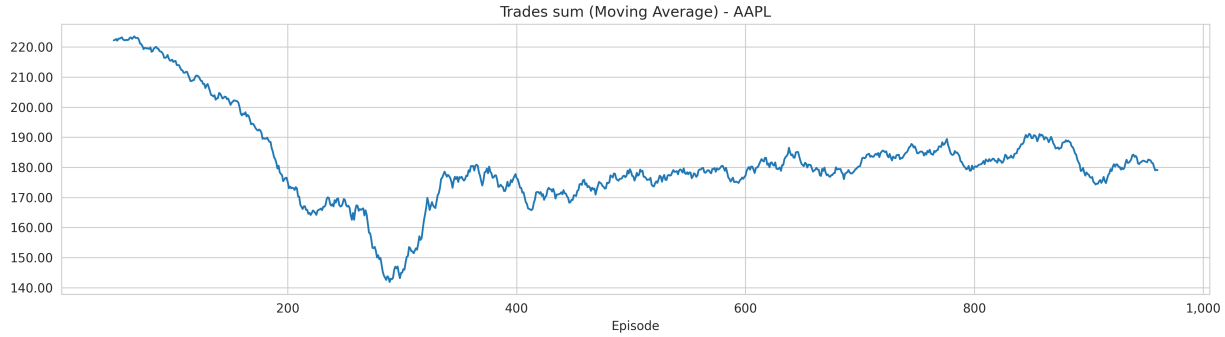


Figura 5.34: Media móvil de suma acumulada de operaciones

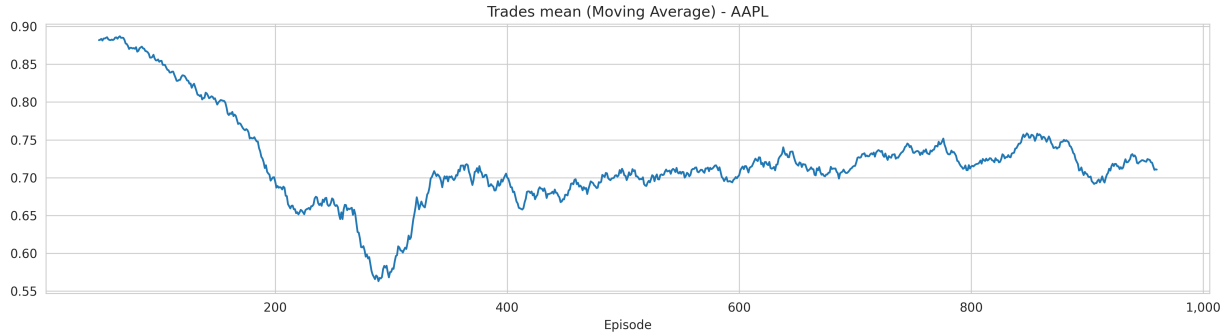


Figura 5.35: Media móvil de promedio de operaciones

5.4.2. Microsoft

La figura 5.36 muestra la media móvil sobre los últimos 50 episodios de la suma acumulada de operaciones de trading durante un episodio para los 960 periodos de entrenamiento. De manera

similar, en la figura 5.37 se muestra, bajo esta misma configuración, el promedio de operaciones de trading realizadas cada día durante un episodio.

Con respecto a estos resultados podemos observar una estabilización de la suma acumulada de operaciones de trading a partir del episodio 400. Este resultado esta fuertemente relacionado con la suma acumulada de costos expuesta en la figura 5.28, ya que el número total de operaciones realizadas durante un episodio va a determinar los costos que se generan. Por esta razón se presenta esta similitud tan grande en el comportamiento de estos dos indicadores.

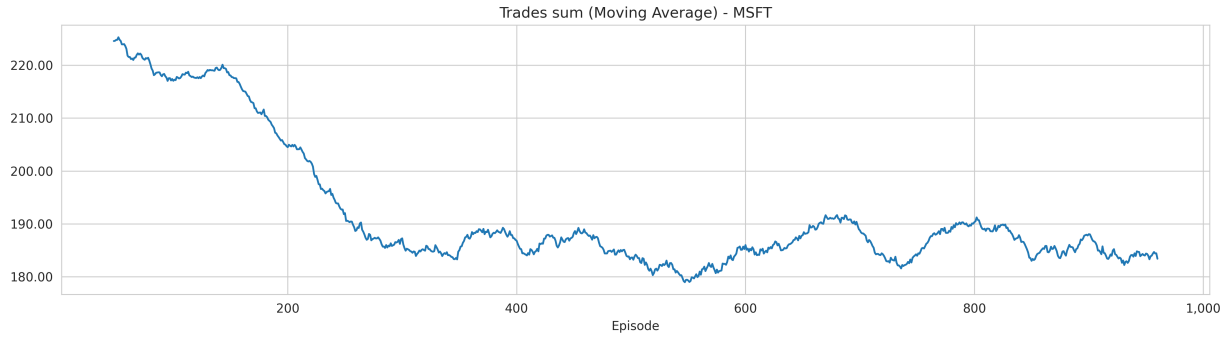


Figura 5.36: Media móvil de suma acumulada de operaciones

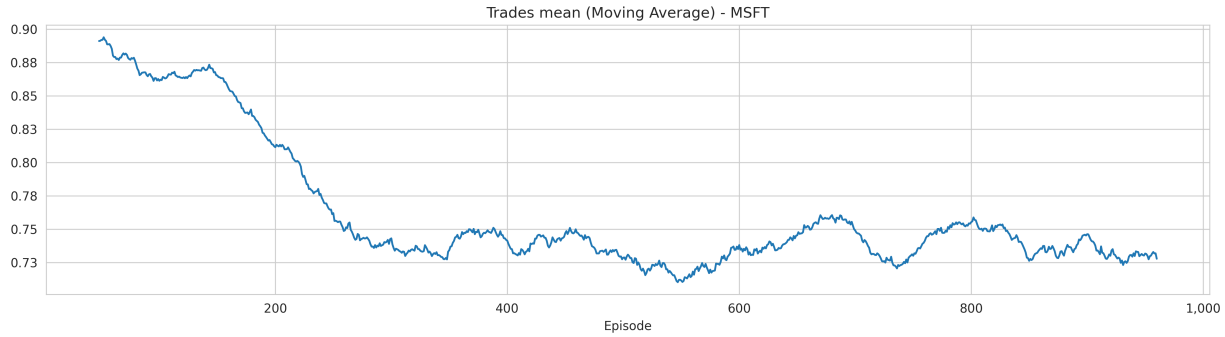


Figura 5.37: Media móvil de promedio de operaciones

5.4.3. Amazon Inc

La figura 5.38 muestra la media móvil sobre los últimos 50 episodios de la suma acumulada de operaciones de trading durante un episodio para los 960 periodos de entrenamiento. De manera similar, en la figura 5.39 se muestra, bajo esta misma configuración, el promedio de operaciones de trading realizadas cada día durante un episodio.

Con respecto a estos resultados podemos observar una estabilización de la suma acumulada de operaciones de trading a partir del episodio 500. Este resultado esta fuertemente relacionado con

la suma acumulada de costos expuesta en la figura 5.30, ya que el número total de operaciones realizadas durante un episodio va a determinar los costos que se generan. Por esta razón se presenta esta similitud tan grande en el comportamiento de estos dos indicadores.

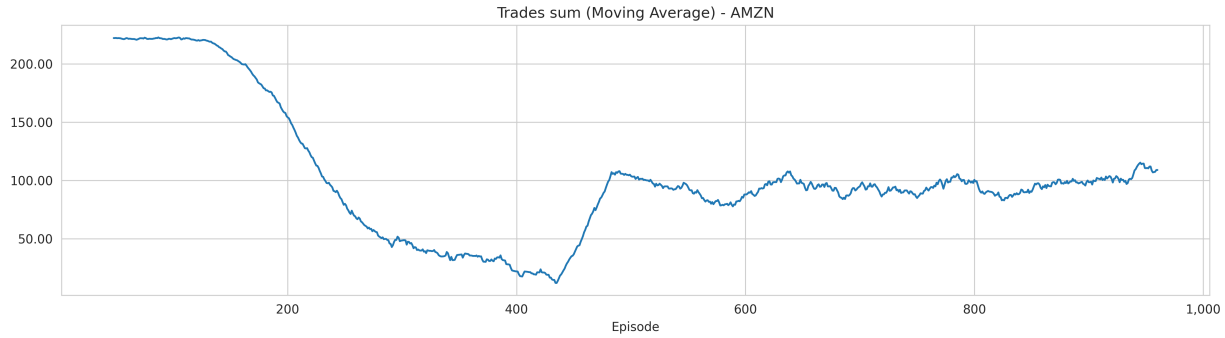


Figura 5.38: Media móvil de suma acumulada de operaciones

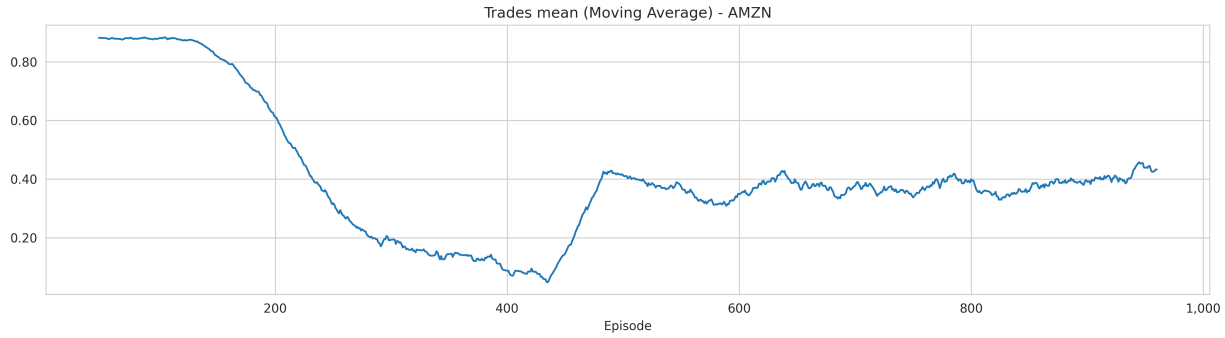


Figura 5.39: Media móvil de promedio de operaciones

5.4.4. Pepsico Inc

La figura 5.40 muestra la media móvil sobre los últimos 50 episodios de la suma acumulada de operaciones de trading durante un episodio para los 960 periodos de entrenamiento. De manera similar, en la figura 5.41 se muestra, bajo esta misma configuración, el promedio de operaciones de trading realizadas cada día durante un episodio.

Con respecto a estos resultados podemos observar una estabilización de la suma acumulada de operaciones de trading en los últimos episodios de entrenamiento. Este resultado esta fuertemente relacionado con la suma acumulada de costos expuesta en la figura 5.32, ya que el número total de operaciones realizadas durante un episodio va a determinar los costos que se generan. Por esta razón se presenta esta similitud tan grande en el comportamiento de estos dos indicadores.

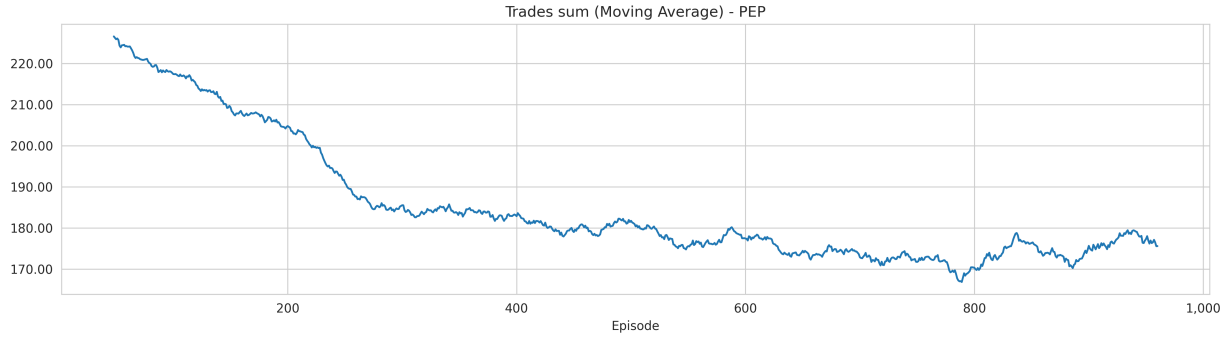


Figura 5.40: Media móvil de suma acumulada de operaciones

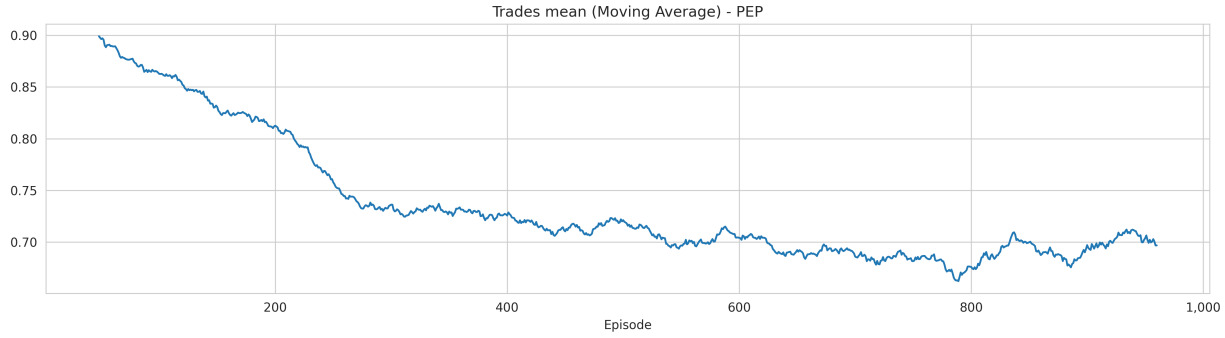


Figura 5.41: Media móvil de promedio de operaciones

5.5. Función de pérdida

En esta sección se presenta el comportamiento de la función de pérdida descrita en la sección 2.2.6.2. En particular, se presentará el promedio de pérdidas acumuladas durante entrenamiento.

La función de pérdida desempeña un papel crucial en el entrenamiento de la red neuronal encargada de estimar los valores Q. Al comienzo del entrenamiento, la función de pérdida suele ser bastante alta porque los parámetros de la red Q se inicializan de manera aleatoria y sus predicciones de valores Q están lejos de ser precisas. Esta alta pérdida refleja la diferencia sustancial entre los valores Q que se predicen y los valores Q óptimos.

A medida que avanza el entrenamiento, el objetivo es minimizar esta función de pérdida. La idea clave para esta minimización es el uso de las dos redes Q: *online_network* (red Q en línea) y *target_network* (red Q objetivo). La red Q objetivo se actualiza con menos frecuencia y proporciona una estimación más estable de los valores Q con el tiempo.

A medida que continúan las iteraciones de entrenamiento, la función de pérdida suele mostrar una tendencia decreciente. Esto significa que la red Q está aprendiendo a aproximar los valores Q con

mayor precisión. La red neuronal ajusta sus parámetros en respuesta a los datos de entrenamiento y la retroalimentación del entorno, lo que le permite hacer mejores predicciones sobre las recompensas esperadas para diferentes pares de estado-acción. En consecuencia, los valores Q que predice la red Q en línea se alinean progresivamente con los valores Q generados por la red Q objetivo, reduciendo el error TD (Diferencia Temporal) entre ellos.

En las figuras 5.42, 5.43, 5.44 y 5.45 se muestra, para cada activo, la media móvil sobre los últimos 50 episodios del promedio de pérdidas acumuladas para los 960 periodos de entrenamiento. Para todos los activos presentados se puede observar una tendencia de disminución de la función de pérdida con el tiempo, lo cual es un indicio prometedor de que los agentes están mejorando su capacidad para evaluar y seleccionar acciones en el ambiente. Esto implica que la red Q está convergiendo hacia una mejor representación de la función Q óptima, lo que finalmente conduce a una toma de decisiones más efectiva y fundamentada.

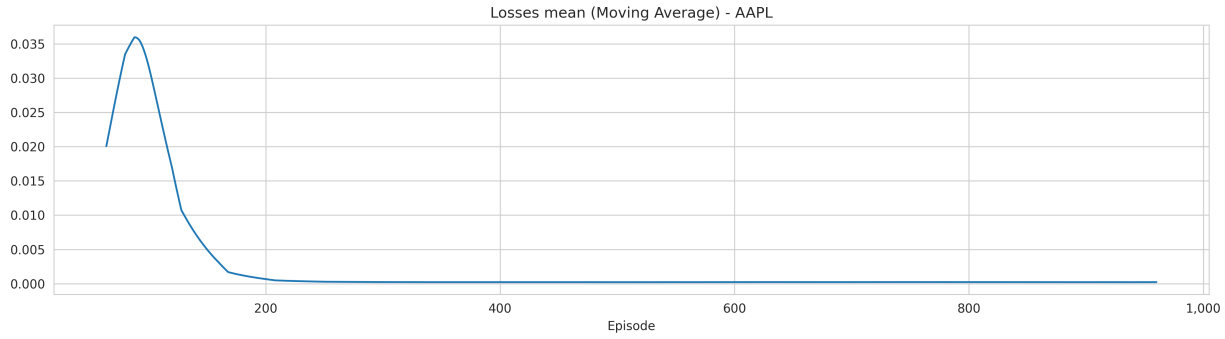


Figura 5.42: Media móvil de promedio de pérdidas acumuladas para Apple Inc

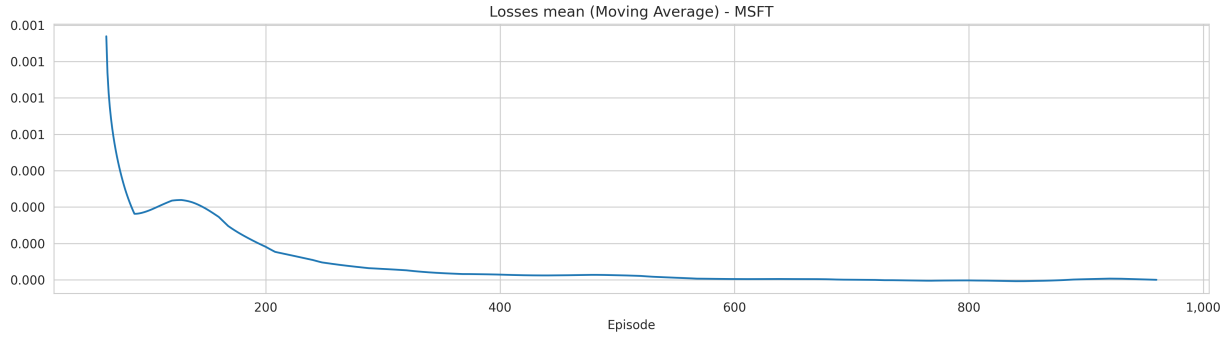


Figura 5.43: Media móvil de promedio de pérdidas acumuladas para Microsoft

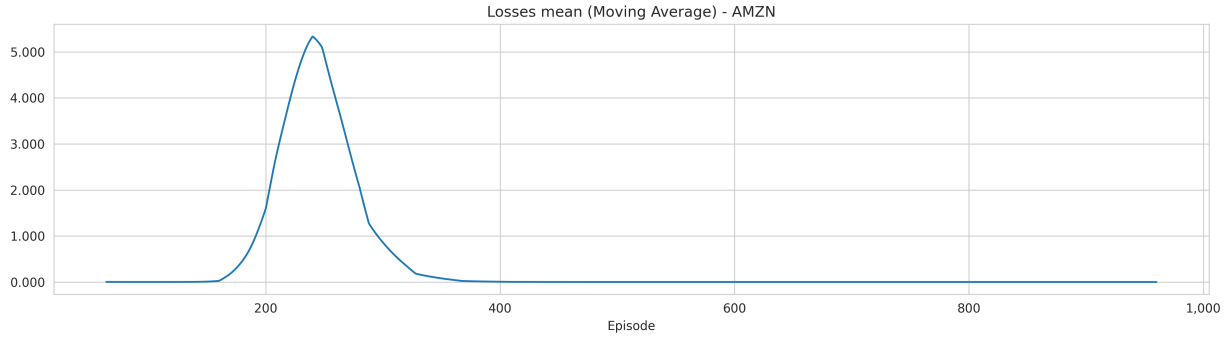


Figura 5.44: Media móvil de promedio de pérdidas acumuladas para Amazon Inc

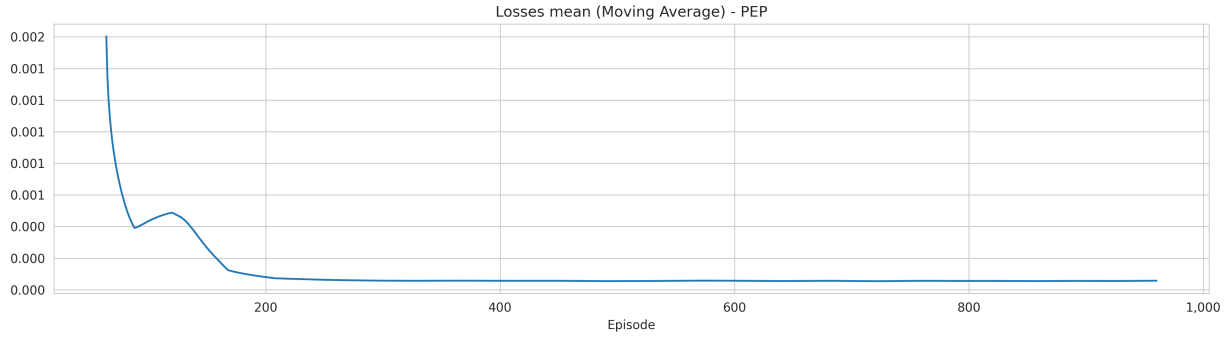


Figura 5.45: Media móvil de promedio de pérdidas acumuladas para Pepsico Inc

Capítulo 6

Conclusiones

6.1. Conclusiones

En este proyecto logramos desarrollar un agente de trading utilizando un marco de aprendizaje por refuerzo, específicamente implementando Double-Deep Q Learning. Nuestro objetivo principal fue evaluar su capacidad para lograr resultados competitivos en comparación con una estrategia de *buy-and-hold* en el contexto de compra y venta de activos. A lo largo de este trabajo surgieron varias observaciones e ideas clave.

El ambiente de trading fue diseñado para responder a algunas complejidades del trading del mundo real, aunque con ciertas simplificaciones. El agente tenía la capacidad de tomar decisiones en cada paso de tiempo, incluyendo comprar, vender en corto o mantener su capital. El espacio de estados se enriqueció con indicadores técnicos y retribuciones en diferentes ventanas de tiempo. Sin embargo, es importante reconocer que nuestro ambiente tenía algunas limitaciones, como la necesidad de utilizar el capital completo para comprar o vender en corto. Además, nuestra función de recompensa solo consideraba los retornos generados por las acciones del agente, sin tener en cuenta los riesgos inherentes asociados con las ventas en corto.

A lo largo del proyecto, realizamos una evaluación exhaustiva del rendimiento de nuestro agente. Inicialmente, los resultados eran consistentes con el desafió de superar una estrategia de *buy-and-hold*. Esto destacó la resistencia de este enfoque, que a menudo es preferido para inversiones a largo plazo por su capacidad de generar rendimientos constantes durante periodos prolongados. Sin embargo, a medida que el agente recibió un amplio entrenamiento a través 1000 episodios (en el último experimento), comenzó a demostrar un progreso notable. Importante destacar que logró resultados competitivos e incluso superó, en algunos casos, los rendimientos de la estrategia *buy-and-hold*. Esto demuestra el potencial de nuestro enfoque de aprendizaje por refuerzo.

Las implicaciones de este logro son significativas. La estrategia de *buy-and-hold*, particularmente en el contexto de la inversión a largo plazo, ha demostrado consistentemente su efectividad al capitalizar el crecimiento inherente de los mercados financieros con el tiempo. La capacidad de nuestro agente de aprendizaje por refuerzo para lograr resultados competitivos destaca su adaptabilidad y capacidad de aprendizaje en respuesta a las dinámicas del mercado, incluso frente a una estrategia a largo plazo establecida y exitosa. Sin embargo, es crucial reconocer que el modelo actual no replica completamente las complejidades de la inversión a largo plazo, como las consideraciones de dividendos e impuestos. Estos factores deben incorporarse en futuras iteraciones para proporcionar una evaluación más completa del rendimiento.

En el futuro, los próximos pasos del proyecto deben centrarse en refinar el modelo e incorporar estas complejidades del mundo real. Las estrategias de gestión de riesgos deben integrarse para

tener en cuenta los desafíos únicos asociados con las ventas en corto. Además, el ajuste de hiperparámetros y la exploración de algoritmos alternativos de aprendizaje por refuerzo pueden mejorar aún más el rendimiento del agente. Realizar análisis de sensibilidad en diferentes condiciones del mercado y evaluar su robustez con datos fuera de muestra será fundamental para medir su eficacia en la aplicación práctica en los mercados financieros reales. En conclusión, nuestro proyecto demuestra el potencial del aprendizaje por refuerzo en el contexto de trading, pero destaca la importancia de la investigación y el desarrollo continuo para hacer que el modelo sea más robusto y aplicable a escenarios del mundo real.

6.2. Trabajo futuro

6.2.1. Incorporación de dimensionamiento de posiciones

En futuras iteraciones de esta investigación, una mejora fundamental a considerar es la incorporación de dimensionamiento de posiciones dentro del entorno de trading. Actualmente, las acciones del agente de comprar y vender en corto se limitan a utilizar todo el capital. Para proporcionar estrategias de trading más realistas y flexibles, podemos modificar el ambiente para permitir que el agente asigne una parte del capital, en lugar de la totalidad, al tomar estas acciones. Este ajuste permitiría al agente emplear tamaños de posición diversos, reflejando la práctica del mundo real de asignación parcial de capital. La implementación de dimensionamiento de posiciones no solo mejoraría el realismo del entorno, sino que también permitiría al agente explorar un espectro más amplio de estrategias de trading, potencialmente conduciendo a un mejor rendimiento y gestión del riesgo.

6.2.2. Integración del riesgo en la función de recompensa

Vender en corto conlleva un gran riesgo porque no existe un límite superior para cuánto puede aumentar el precio de un activo, lo que podría resultar en pérdidas ilimitadas si el precio sube significativamente.

Con el fin de crear una función de recompensa más completa, el trabajo futuro debe centrarse en incluir el riesgo asociado con la venta en corto. La función de recompensa actual depende principalmente de los rendimientos generados por las acciones tomadas en el paso anterior, pasando por alto los riesgos que se puedan generar por los cambios abruptos que puedan ocurrir en el precio de un activo. Para abordar esta limitación, podemos introducir medidas de riesgo como el Valor en Riesgo (VaR) o el Valor en Riesgo Condicional (CVaR) en la función de recompensa. Al cuantificar el riesgo potencial de las posiciones cortas, podemos alentar al agente a adoptar estrategias conscientes del riesgo, fomentando así un enfoque de trading más prudente y equilibrado.

6.2.3. Registrar la cantidad real de posiciones

Actualmente, el entorno rastrea el NAV del capital, lo que proporciona información valiosa sobre el rendimiento general de la cartera. Sin embargo, una ampliación valiosa del entorno implicaría el seguimiento de la cantidad real de activos en posesión, lo que permitiría una gestión de posiciones

más precisa. Para lograr esto, podemos implementar un sistema de registro que supervise la cantidad de cada activo en posesión del agente, además del NAV. Esta expansión del entorno facilitaría un análisis más detallado del comportamiento de trading, incluida la capacidad de medir asignaciones de activos y liquidez. Además, permitiría al agente ejecutar decisiones de trading más detalladas, como ajustar el tamaño de posiciones individuales, lo cual es un elemento crucial en la gestión de carteras.

Incorporar estas mejoras en el entorno de trading y el marco de aprendizaje por refuerzo representa vías prometedoras para investigaciones futuras. Al abordar el dimensionamiento de posiciones, las consideraciones de riesgo y el seguimiento de las cantidades reales de posiciones, podemos elevar aún más el realismo y la eficacia del agente de trading, convirtiéndolo en una herramienta más valiosa tanto para la exploración académica como para la aplicación práctica en los mercados financieros reales.

Bibliografía

[1] T. C, "What is a portfolio?," 2021.

[2] B. N, D. R, and L. T, "Strategic asset allocation: Determining the optimal portfolio with ten asset classes," *Institucional Investor Journal*, 2009.

[3] DALBAR, "Quantitative analysis of investor behaviour 2017," 2017.

[4] A. L, "A concise guide to asset allocation," 2021.

[5] DALBAR, "Dalbar's 20th annual quantitative analysis of investor behavior 2014," 2014.

[6] F. B, "Why people lose money in the market," 2021.

[7] S. L and J. J, "What percentage of americans owns stock?," 2021.

[8] L. J, "Análisis bursátil," 2021.

[9] H. A, "Technical analysis," 2021.

[10] C. J, "Reinforcement learning," 2021.

[11] C. J, "Stock market," 2021.

[12] H. A, "Position," 2021.

[13] F. T, "Reinforcement learning in financial markets - a survey," 2018.

[14] K. J, "Financial time series forecasting using support vector machines," *Neurocomputing*, vol. 55, no. 1-2, pp. 307–319, 2003.

[15] H. W, N. Y, and W. S, "Forecasting stock market movement direction with support vector machine," *Computers & Operations Research*, vol. 32, no. 10, p. 2513–2522, 2005.

[16] K. M and T. M, "Forecasting stock index movement: A comparison of support vector machines and random forest," *Indian Institute of Capital Markets 9th Capital Markets Conference Paper*, 2006.

[17] B. A, G. E, and M. F, "Automated trading with performance weighted random forests and seasonality," *Expert Systems with Applications*, vol. 41, no. 8, pp. 3651–3661, 2014.

[18] M. B and Z. T, "Deep conditional portfolio sorts: The relation between past and future stock returns," *LMU Munich and Harvard University*, 2014.

[19] K. C, D. X, and H. N, "Deep neural networks, gradient-boosted trees, random forests: Statistical arbitrage on the s&p 500," *European Journal of Operational Research*, vol. 259, no. 2, pp. 689–702, 2017.

[20] T. L and O. A, "A method for automatic stock trading combining technical analysis and nearest neighbor classification," *Expert Systems with Applications*, vol. 37, no. 10, p. 6885–6890, 2010.

[21] M. J, W. L, L. Y, and S. M, "Performance functions and reinforcement learning for trading systems and portfolios," *Journal of Forecasting*, vol. 17, no. 56, pp. 441–470, 1998.

[22] Investopedia, "Liquidity crisis," 2020.

[23] S. R and B. A, *Reinforcement learning: An introduction*. MIT Press, Cambridge, 1998.

[24] Y. H, L. X, Z. S, and W. A, "Deep reinforcement learning for automated stock trading: An ensemble strategy," *ACM International Conference on AI in Finance*, 2020.

[25] J. S, *Machine Learning for Algorithmic Trading*. Packt Publishing, 2020.

[26] W. C, "Learning from delayed rewards," 1989.

[27] M. S, "N-step bootstrapping in reinforcement learning," 2023.

[28] D. S and G. S, *Learn Algorithmic Trading*. Packt Publishing, 2019.

[29] Fidelity, "Relative strength index," 2020.

[30] Fidelity, "Moving average convergence/divergence," 2020.

[31] Fidelity, "Average true range," 2020.

[32] Fidelity, "Slow stochastic," 2020.

[33] Fidelity, "Ultimate oscillator," 2020.

[34] Fidelity, "Bollinger bands," 2020.

[35] Fidelity, "On balance volume," 2020.

[36] M. V, K. K, and S. D, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[37] L. T, H. J, and P. A, "Continuous control with deep reinforcement learning," *In Proceedings of the International Conference on Learning Representations (ICLR)*, 2016.

[38] H. T, Z. A, A. P, and L. S, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," *arXiv preprint arXiv:1801.01290*.

[39] J. Z, L. M, Z. W, and S. D, "Market efficiency versus behavioral biases: Evidence from the chinese a-share market," *Journal of Economic Behavior and Organization*, vol. 135, pp. 109–129, 2017.

[40] P. J, M. A, and K. A, "Can machine learning algos learn to trade? a market microstructure and ai study," *arXiv preprint arXiv:1804.00358*, 2018.

[41] G. R, Z. R, and S. S, "Algorithmic trading with machine learning and deep learning: Applications and risks," *In Proceedings of the IEEE Conference on Computational Intelligence for Financial Engineering and Economics (CIFEr)*, 2017.

[42] K. M, N. Y, and V. J, "Machine learning for market microstructure and high frequency trading," *IEEE Intelligent Systems*, vol. 28, no. 2, pp. 62–77, 2013.

[43] L. J, K. E, K. Y, and K. J, "Global stock market prediction based on stock chart images using deep q-network," *IEEE Access*, vol. 4, 2016.

Appendices

Apéndice A

Código de implementación

A.1. Almacenamiento de datos en formato fast HDF

```
df = (pd.read_csv('wiki_prices.csv',
                  parse_dates=['date'],
                  index_col=['date', 'ticker'],
                  infer_datetime_format=True)
     .sort_index())
with pd.HDFStore('wiki_prices.h5') as store:
    store.put('quandl/wiki/prices', df)
```

A.2. Agente DDQN

A.2.1. Constructor

```
def __init__(self, state_dim,
             num_actions,
             learning_rate,
             gamma,
             epsilon_start,
             epsilon_end,
             epsilon_decay_steps,
             epsilon_exponential_decay,
             replay_capacity,
             architecture,
             l2_reg,
             tau,
             batch_size,
             train,
             ticker):

    self.state_dim = state_dim
    self.num_actions = num_actions
    self.experience = deque([], maxlen=replay_capacity)
    self.learning_rate = learning_rate
    self.gamma = gamma
    self.architecture = architecture
    self.l2_reg = l2_reg
    self.ticker = ticker
```

```
25
26      self.online_network = self.build_model()
27      self.target_network = self.build_model(trainable=False)
28      self.update_target()
29
30      if(not train):
31          self.epsilon = 0.0
32      else:
33          self.epsilon = epsilon_start
34      self.epsilon_decay_steps = epsilon_decay_steps
35      self.epsilon_decay = (epsilon_start - epsilon_end) / epsilon_decay_steps
36      self.epsilon_exponential_decay = epsilon_exponential_decay
37
38      self.total_steps = 0
39      self.episodes = 0
40
41      self.batch_size = batch_size
42      self.tau = tau
43      self.losses = []
44      self.idx = tf.range(batch_size)
45      self.train = train
```

A.2.2. Construcción del modelo

```
1  def build_model(self, trainable=True):
2      layers = []
3      n = len(self.architecture)
4      for i, units in enumerate(self.architecture, 1):
5          layers.append(Dense(units=units,
6                              input_dim=self.state_dim if i == 1 else None,
7                              activation='relu',
8                              kernel_regularizer=l2(self.l2_reg),
9                              name=f'Dense_{i}',
10                             trainable=trainable))
11     layers.append(Dropout(.1))
12     layers.append(Dense(units=self.num_actions,
13                         trainable=trainable,
14                         name='Output'))
15     model = Sequential(layers)
16     model.compile(loss='mean_squared_error',
17                   optimizer=Adam(learning_rate=self.learning_rate))
18     return model
```

A.2.3. Predicción Epsilon-Greedy

```
def epsilon_greedy_policy(self, state):
    self.total_steps += 1
    if np.random.rand() <= self.epsilon:
        return np.random.choice(self.num_actions)
    q = self.online_network.predict(state, verbose=0)
    return np.argmax(q, axis=1).squeeze()
```

A.2.4. Memorizar transición

```
def memorize_transition(self, s, a, r, s_prime, done):
    if done:
        if self.episodes < self.epsilon_decay_steps:
            self.epsilon -= self.epsilon_decay
        else:
            self.epsilon *= self.epsilon_exponential_decay
        self.episodes += 1
    self.experience.append((s, a, r, s_prime, 0.0 if done else 1.0))
```

A.2.5. Entrenar con experiencia

```
def experience_replay(self):
    if self.batch_size > len(self.experience):
        return
    minibatch = map(np.array, zip(*sample(self.experience, self.batch_size)))
    states, actions, rewards, next_states, not_done = minibatch

    next_q_values = self.online_network.predict_on_batch(next_states)
    best_actions = tf.argmax(next_q_values, axis=1)

    next_q_values_target = self.target_network.predict_on_batch(next_states)
    target_q_values = tf.gather_nd(next_q_values_target,
                                   tf.stack((self.idx, tf.cast(best_actions, tf.int32)
    ), axis=1))

    targets = rewards + not_done * self.gamma * target_q_values

    q_values = self.online_network.predict_on_batch(states)
    q_values[(self.idx, actions)] = targets

    loss = self.online_network.train_on_batch(x=states, y=q_values)
    self.losses.append(loss)

    if self.total_steps % self.tau == 0:
        self.update_target()
```

A.2.6. Manejo de pesos de red en linea

```
def load_model_weights(self, model_weights_filepath):
    self.online_network.load_weights(model_weights_filepath)
```

```
def save_model_weights(self, folder_id):
    model_weights_filepath = 'weights/{0}/model_{1}.h5'.format(
        folder_id, self.ticker)
    if (not Path('weights/{0}'.format(folder_id)).exists()):
        Path('weights/{0}'.format(folder_id)).mkdir(parents=True)
    self.online_network.save_weights(model_weights_filepath)
```

A.3. Ambiente de trading

A.3.1. DataSource

```
class DataSource:
    """
    Data source for TradingEnvironment

    Loads & preprocesses daily price & volume data
    Provides data for each new episode.
    """

    def __init__(self, trading_days=252, tickers=['AAPL'], normalize=True, start_date=
    '1995-01-01', end_date='2018-03-20'):
        self.tickers = tickers
        self.trading_days = trading_days
        self.normalize = normalize
        self.start_date = start_date
        self.end_date = end_date
        self.data = self.load_data()
        self.min_values = self.data.min()
        self.max_values = self.data.max()
        self.step = 0
        self.offset = None

    def load_data(self):
        ticker_dfs = []
        for ticker in self.tickers:
            log.info('loading data for {}...'.format(ticker))
            idx = pd.IndexSlice
            with pd.HDFStore('/content/drive/MyDrive/deep-reinforcement-learning-for-
    trading/data/assets.h5') as store:
                ticker_df = (store['quandl/wiki/prices']
                             .loc[idx[:, ticker],
                                  ['adj_close', 'adj_volume', 'adj_low', 'adj_high']]
                             .dropna()
```

```
                                .sort_index())
            ticker_df.columns = ['close', 'volume', 'low', 'high']
            log.info('got data for {}...'.format(ticker))

            log.info('preprocessing data for {}...'.format(ticker))
            ticker_df = self.preprocess_data(ticker_df, ticker)
            log.info('finished preprocessing for {}...'.format(ticker))

            ticker_dfs.append(ticker_df.copy())
        df = pd.concat(ticker_dfs)
        return df

    def preprocess_data(self, ticker_df, ticker):
        """
        Calculates returns and percentiles, then removes missing values
        """

        items = list(map(lambda date: (date, ticker), pd.date_range(
            start=self.start_date, end=self.end_date)))
        data = ticker_df.filter(items=items, axis=0)

        data['returns'] = data.close.pct_change()
        data['ret_2'] = data.close.pct_change(2)
        data['ret_5'] = data.close.pct_change(5)
        data['ret_10'] = data.close.pct_change(10)
        data['ret_21'] = data.close.pct_change(21)
        data['rsi'] = talib.STOCHRSI(data.close)[1]
        data['macd'] = talib.MACD(data.close)[1]
        data['atr'] = talib.ATR(data.high, data.low, data.close)
        slowk, slowd = talib.STOCH(data.high, data.low, data.close)
        data['stoch'] = slowd - slowk
        data['ultosc'] = talib.ULTOSC(data.high, data.low, data.close)
        up, mid, low = talib.BBANDS(data.close)
        data['bbp'] = (data.close - low) / (up - low)
        data['obv'] = talib.OBV(data.close, data.volume)
        data['adx'] = talib.ADX(data.high, data.low, data.close)

        data = (data.replace((np.inf, -np.inf), np.nan)
                .drop(['high', 'low', 'close', 'volume'], axis=1)
                .dropna())

        r = data.returns.copy()
        if self.normalize:
            data = pd.DataFrame(scale(data),
                                columns=data.columns,
                                index=data.index)
        features = data.columns.drop('returns')
        data['returns'] = r  # don't scale returns
        data = data.loc[:, ['returns'] + list(features)]
        return data

```

```
    def reset(self):
        """
        Provides starting index for time series and resets step
        """
        high = {ticker: len(self.data.loc[(slice(None), ticker), :].index) - self.
    trading_days for ticker in self.tickers}
        self.offset = {}
        for ticker in self.tickers:
          self.offset[ticker] = np.random.randint(low=0, high=high[ticker])
        self.step = 0

    def take_step(self):
        """
        Returns data for current trading day and done signal
        """
        observations = {}
        for ticker in self.tickers:
            observations[ticker] = self.data.loc[(
                slice(None), ticker), :].iloc[self.offset[ticker] + self.step].values
        self.step += 1
        done = self.step > self.trading_days
        return observations, done
```

A.3.2. TradingSimulator

```
class TradingSimulator:
    """
    Implements core trading simulator for multiple tickers
    """

    def __init__(self, steps, trading_cost_bps, time_cost_bps, tickers=['AAPL']):
        self.trading_cost_bps = trading_cost_bps
        self.time_cost_bps = time_cost_bps
        self.steps = steps
        self.tickers = tickers

        # change every step
        self.step = 0
        self.navs = {ticker: np.ones(self.steps) for ticker in self.tickers}
        self.market_navs = {ticker: np.ones(
            self.steps) for ticker in self.tickers}
        self.strategy_returns = {ticker: np.zeros(
            self.steps) for ticker in self.tickers}
        self.market_returns = {ticker: np.zeros(
            self.steps) for ticker in self.tickers}
        self.actions = {ticker: np.zeros(self.steps)
                        for ticker in self.tickers}
        self.positions = {ticker: np.zeros(self.steps)
                          for ticker in self.tickers}
```

```
        self.costs = {ticker: np.zeros(self.steps) for ticker in self.tickers}
        self.trades = {ticker: np.zeros(self.steps) for ticker in self.tickers}

    def reset(self):
        self.step = 0
        for ticker in self.tickers:
            self.navs[ticker].fill(1)
            self.market_navs[ticker].fill(1)
            self.strategy_returns[ticker].fill(0)
            self.market_returns[ticker].fill(0)
            self.actions[ticker].fill(0)
            self.positions[ticker].fill(0)
            self.costs[ticker].fill(0)
            self.trades[ticker].fill(0)

    def take_step(self, actions, market_returns):
        """
        Calculates NAVs, trading costs and reward
        based on an action and latest market return
        and returns the reward and a summary of the day's
        activity.
        """
        ticker_rewards = {}
        ticker_navs = {}
        ticker_costs = {}
        for ticker in self.tickers:
            start_position = self.positions[ticker][max(0, self.step - 1)]
            start_nav = self.navs[ticker][max(0, self.step - 1)]
            start_market_nav = self.market_navs[ticker][max(0, self.step - 1)]
            self.market_returns[ticker][self.step] = market_returns[ticker]
            self.actions[ticker][self.step] = actions[ticker]

            end_position = actions[ticker] - 1  # short, neutral, long
            n_trades = end_position - start_position
            self.positions[ticker][self.step] = end_position
            self.trades[ticker][self.step] = n_trades

            trade_costs = abs(n_trades) * self.trading_cost_bps
            time_cost = 0 if n_trades else self.time_cost_bps
            self.costs[ticker][self.step] = trade_costs + time_cost
            reward = start_position * \
                market_returns[ticker] - \
                self.costs[ticker][max(0, self.step-1)]
            self.strategy_returns[ticker][self.step] = reward

            if self.step != 0:
                self.navs[ticker][self.step] = start_nav * \
                    (1 + self.strategy_returns[ticker][self.step])
                self.market_navs[ticker][self.step] = start_market_nav * \
                    (1 + self.market_returns[ticker][self.step])
            ticker_rewards[ticker] = reward
```

```
            ticker_navs[ticker] = self.navs[ticker][self.step]
            ticker_costs[ticker] = self.costs[ticker][self.step]

        info = {'rewards': ticker_rewards,
                'navs': ticker_navs,
                'costs': ticker_costs}

        self.step += 1
        return ticker_rewards, info

    def result(self):
        """
        Returns current state as pd.DataFrame
        """
        return pd.DataFrame({'action': self.actions,
                             'nav': self.navs,
                             'market_nav': self.market_navs,
                             'market_return': self.market_returns,
                             'strategy_return': self.strategy_returns,
                             'position': self.positions,
                             'cost': self.costs,
                             'trade': self.trades})
```

A.3.3. TradingEnvironment

```
class TradingEnvironment(gym.Env):
    """
    A trading environment for reinforcement learning.
    """

    def __init__(self,
                 trading_days=252,
                 trading_cost_bps=1e-3,
                 time_cost_bps=1e-4,
                 tickers=['AAPL'],
                 start_date='1995-01-01',
                 end_date='2018-03-20'):
        self.trading_days = trading_days
        self.trading_cost_bps = trading_cost_bps
        self.tickers = tickers
        self.time_cost_bps = time_cost_bps
        self.start_date = start_date
        self.end_date = end_date
        self.data_source = DataSource(trading_days=self.trading_days,
                                      tickers=self.tickers,
                                      start_date=self.start_date,
                                      end_date=self.end_date)
        self.simulator = TradingSimulator(steps=self.trading_days,
                                          trading_cost_bps=self.trading_cost_bps,
```

```
                                          time_cost_bps=self.time_cost_bps,
                                          tickers=self.tickers)
        self.action_space = spaces.Discrete(3)
        self.observation_space = spaces.Box(self.data_source.min_values,
                                            self.data_source.max_values)
        self.reset()

    def seed(self, seed=None):
        self.np_random, seed = seeding.np_random(seed)
        return [seed]

    def step(self, actions):
        """
        Returns state observation, reward, done and info
        """
        observations, done = self.data_source.take_step()
        rewards, info = self.simulator.take_step(actions=actions,
                                                 market_returns={ticker: observations[
    ticker][0] for ticker in self.tickers})
        return observations, rewards, done, info

    def reset(self):
        """
        Resets DataSource and TradingSimulator; returns first observation
        """
        self.data_source.reset()
        self.simulator.reset()
        return self.data_source.take_step()[0]
```

A.4. Entrenamiento

```
for episode in range(1, max_episodes + 1):
    current_states = trading_environment.reset()
    for episode_step in range(max_episode_steps):
        actions = {}
        for ticker in tickers:
            action = int(ddqn[ticker].epsilon_greedy_policy(current_states[ticker].
    reshape(-1, state_dim)))
            actions[ticker] = action

        next_states, rewards, done, _ = trading_environment.step(actions)

        if train:
            for ticker in tickers:
                ddqn[ticker].memorize_transition(current_states[ticker],
                                                 actions[ticker],
                                                 rewards[ticker],
                                                 next_states[ticker],
```

```
                                        done)
                ddqn[ticker].experience_replay()
        if done:
            break
        current_states = next_states.copy()
```

Apéndice B

Resultados

En este apéndice se presentan los resultados no mostrados para los experimentos 1, 2 y 3.

B.1. Experimento 1

En esta sección se presentarán los resultados para el experimento descrito en la sección 4.3.3.1.

B.1.1. Rendimientos

B.1.1.1. Apple Inc (AAPL)

La figura B.1 muestra la media móvil sobre los últimos 50 episodios de los valores de retorno acumulativo para los 480 periodos de entrenamiento, así como también la media móvil de la proporción de los últimos 50 episodios en los que el agente superó la estrategia *buy-and-hold*. La figura B.2 muestra estos mismos resultados sin una media móvil sobre los episodios. De manera similar, en las figuras B.3 y B.4 se muestra, bajo esta misma configuración, el promedio de los valores de retorno acumulativo obtenidos cada día durante un episodio.

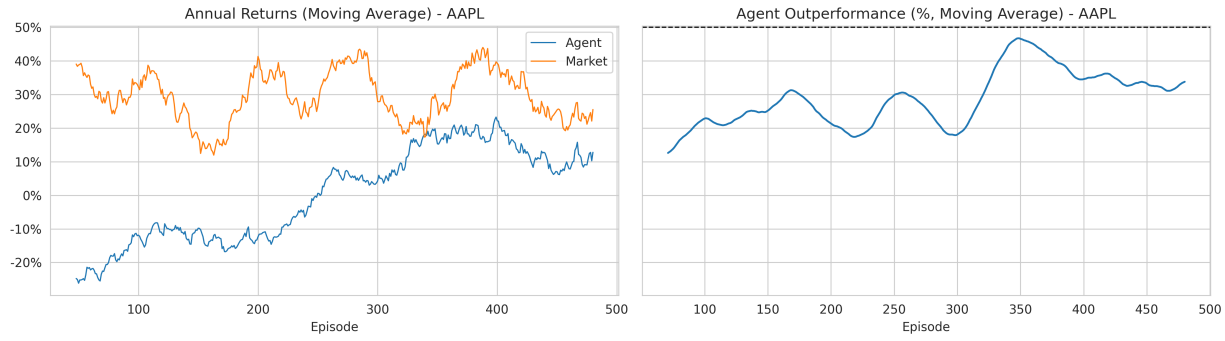


Figura B.1: Media móvil de rendimientos anuales

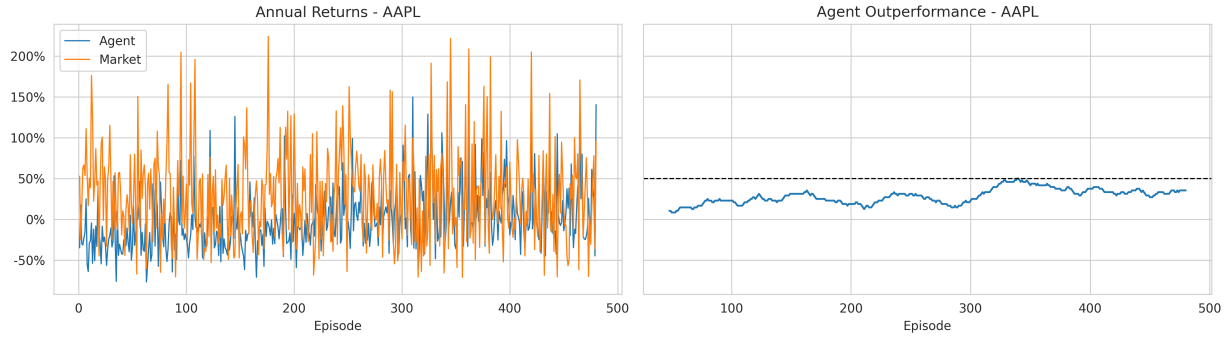


Figura B.2: Rendimientos anuales

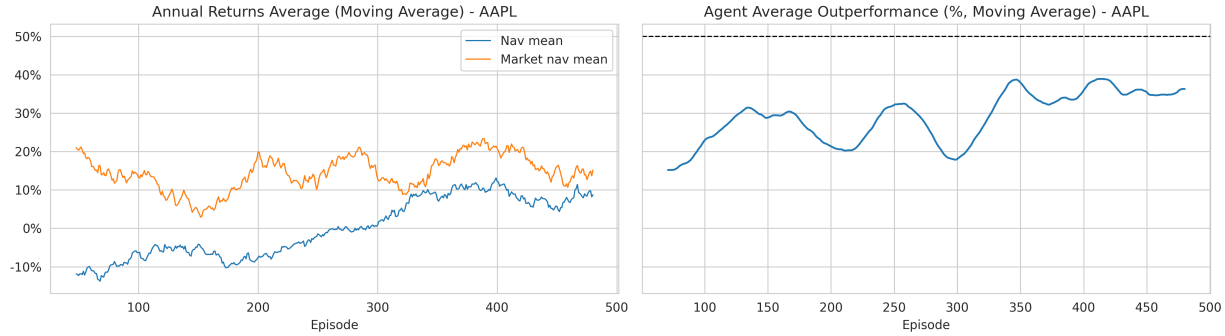


Figura B.3: Media móvil de rendimientos anuales promedio

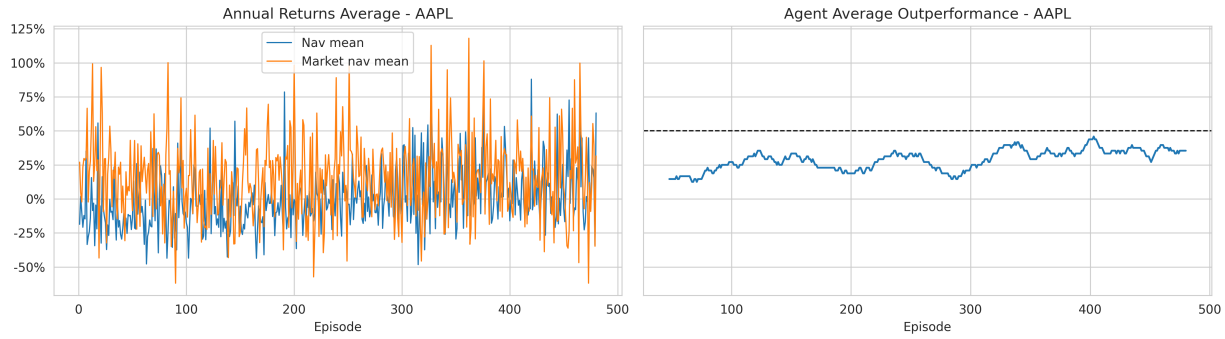


Figura B.4: Rendimientos anuales promedio

Finalmente, en la figura B.5 se muestra la media móvil sobre los últimos 50 episodios de la proporción de días en los que el agente superó la estrategia *buy-and-hold* durante un episodio. La figura B.6 muestra estos mismos resultados sin una media móvil sobre los episodios.

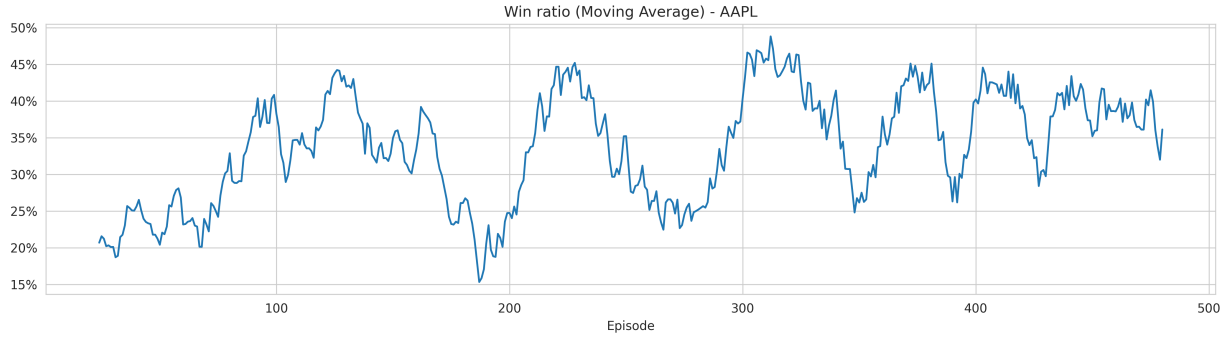


Figura B.5: Media móvil de proporción de ganancias

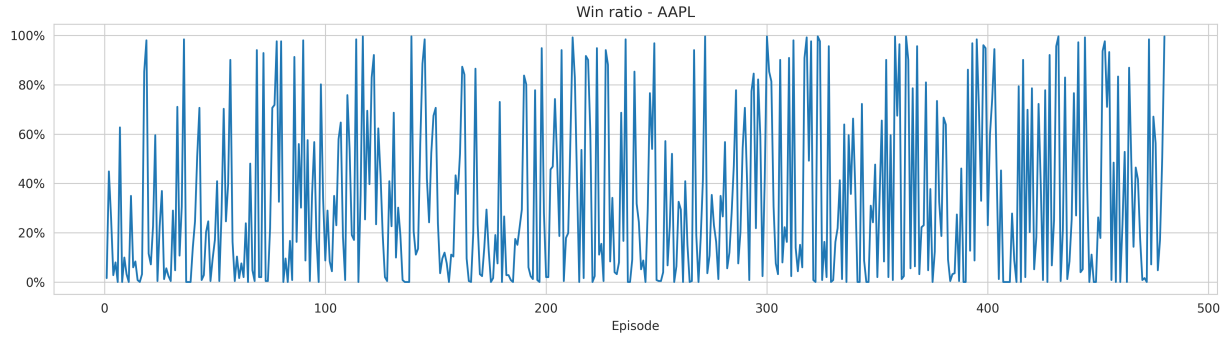


Figura B.6: Proporción de ganancias

B.1.1.2. Microsoft (MSFT)

La figura B.7 muestra la media móvil sobre los últimos 50 episodios de los valores de retorno acumulativo para los 480 periodos de entrenamiento, así como también la media móvil de la proporción de los últimos 50 episodios en los que el agente superó la estrategia *buy-and-hold*. La figura B.8 muestra estos mismos resultados sin una media móvil sobre los episodios. De manera similar, en las figuras B.9 y B.10 se muestra, bajo esta misma configuración, el promedio de los valores de retorno acumulativo obtenidos cada día durante un episodio.

Figura B.7: Media móvil de rendimientos anuales

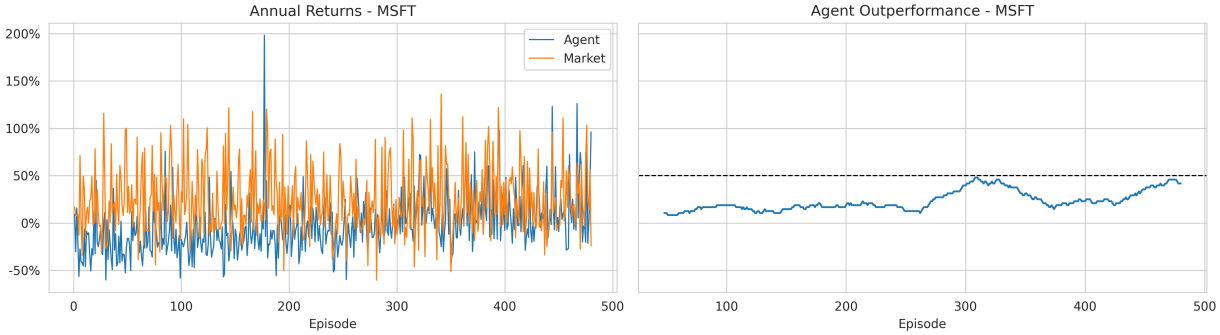


Figura B.8: Rendimientos anuales

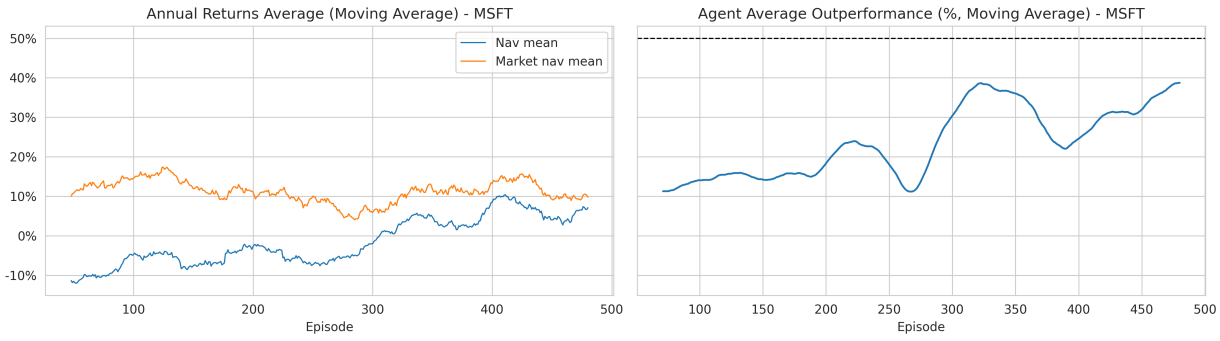


Figura B.9: Media móvil de rendimientos anuales promedio

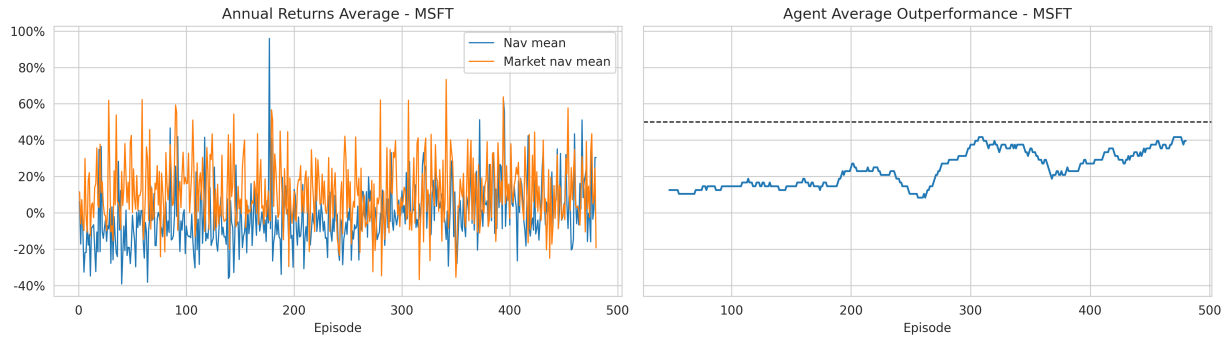


Figura B.10: Rendimientos anuales promedio

Finalmente, en la figura B.11 se muestra la media móvil sobre los últimos 50 episodios de la proporción de días en los que el agente superó la estrategia *buy-and-hold* durante un episodio. La figura B.12 muestra estos mismos resultados sin una media móvil sobre los episodios.

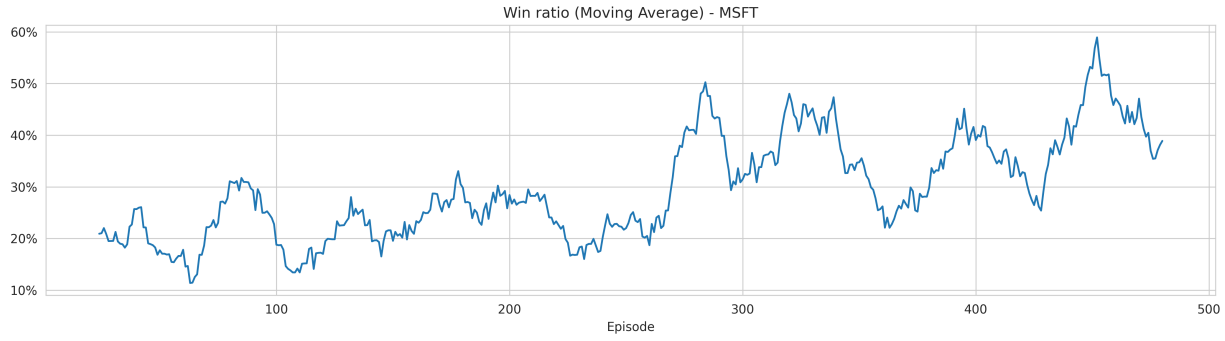


Figura B.11: Media móvil de proporción de ganancias

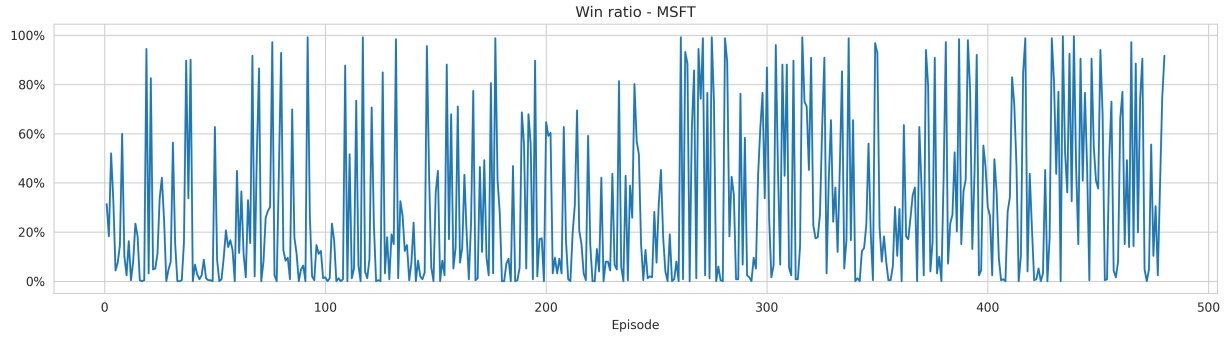


Figura B.12: Proporción de ganancias

B.1.1.3. Amazon Inc (AMZN)

La figura B.13 muestra la media móvil sobre los últimos 50 episodios de los valores de retorno acumulativo para los 480 periodos de entrenamiento, así como también la media móvil de la proporción de los últimos 50 episodios en los que el agente superó la estrategia *buy-and-hold*. La figura B.14 muestra estos mismos resultados sin una media móvil sobre los episodios. De manera similar, en las figuras B.15 y B.16 se muestra, bajo esta misma configuración, el promedio de los valores de retorno acumulativo obtenidos cada día durante un episodio.

Figura B.13: Media móvil de rendimientos anuales

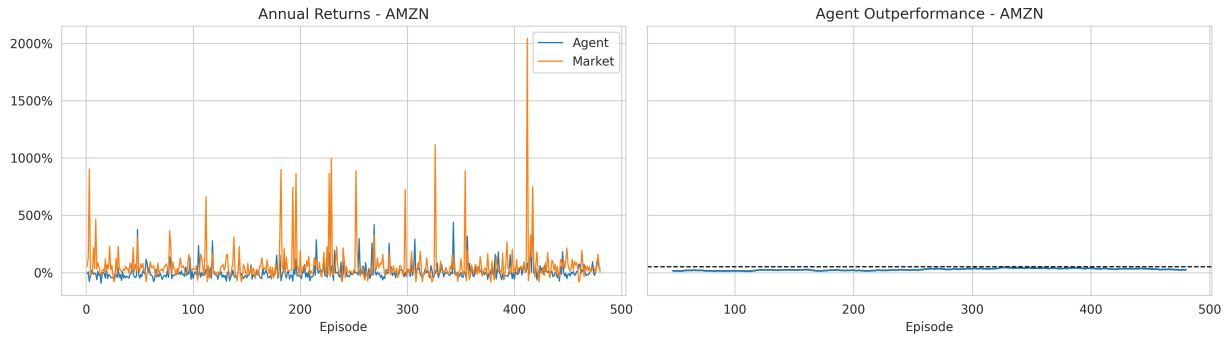


Figura B.14: Rendimientos anuales

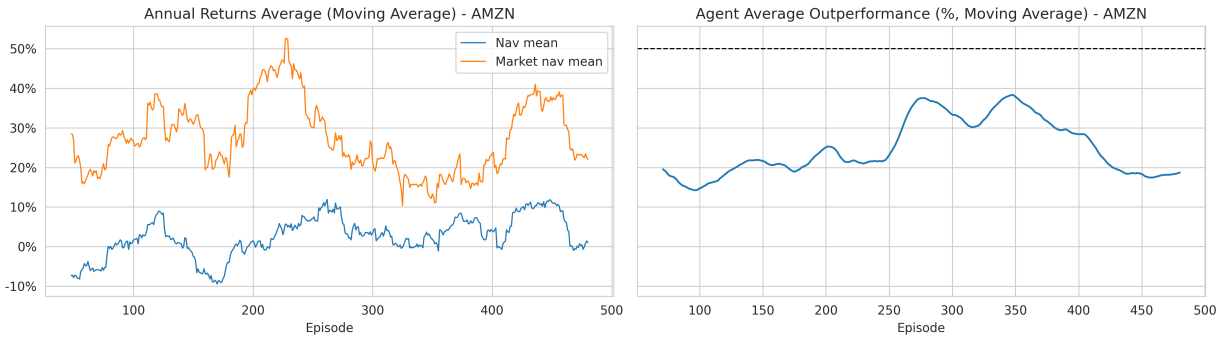


Figura B.15: Media móvil de rendimientos anuales promedio

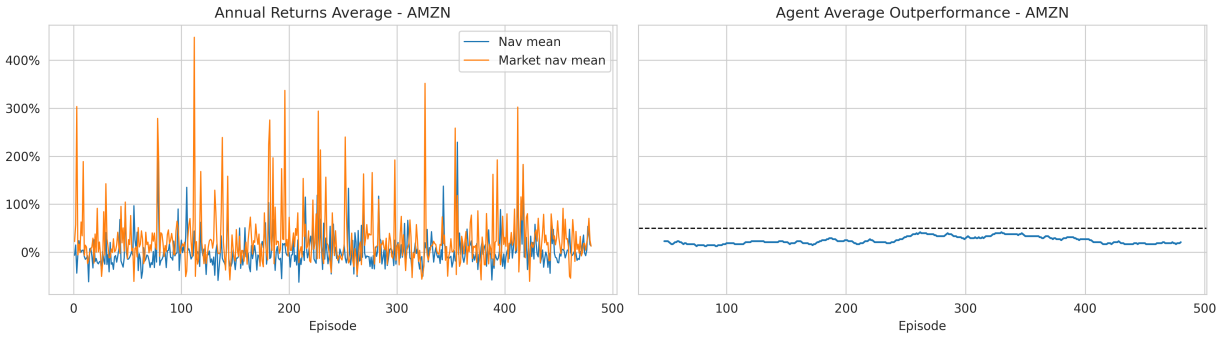


Figura B.16: Rendimientos anuales promedio

Finalmente, en la figura B.17 se muestra la media móvil sobre los últimos 50 episodios de la proporción de días en los que el agente superó la estrategia *buy-and-hold* durante un episodio. La figura B.18 muestra estos mismos resultados sin una media móvil sobre los episodios.

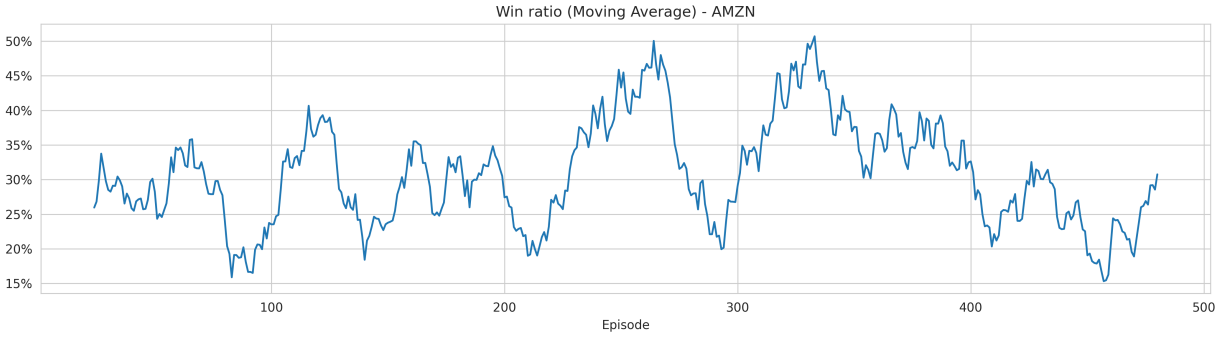


Figura B.17: Media móvil de proporción de ganancias

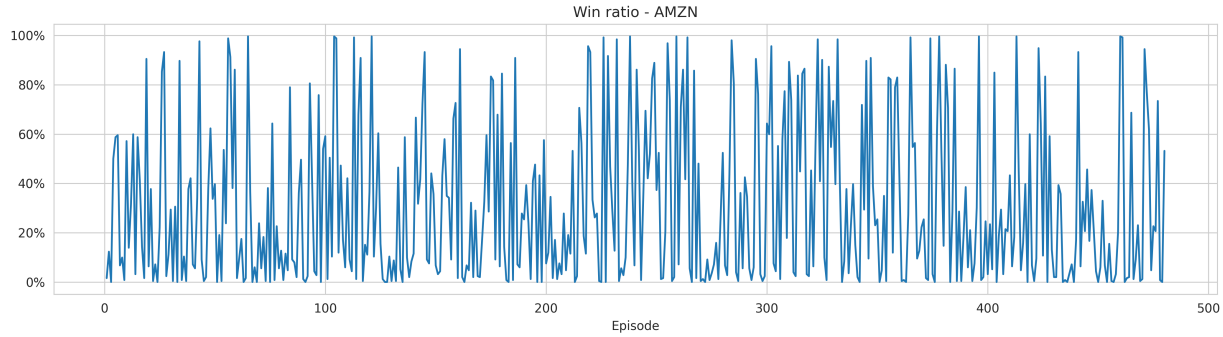


Figura B.18: Proporción de ganancias

B.1.1.4. Pepsico Inc (PEP)

La figura B.19 muestra la media móvil sobre los últimos 50 episodios de los valores de retorno acumulativo para los 480 periodos de entrenamiento, así como también la media móvil de la proporción de los últimos 50 episodios en los que el agente superó la estrategia *buy-and-hold*. La figura B.20 muestra estos mismos resultados sin una media móvil sobre los episodios. De manera similar, en las figuras B.21 y B.22 se muestra, bajo esta misma configuración, el promedio de los valores de retorno acumulativo obtenidos cada día durante un episodio.

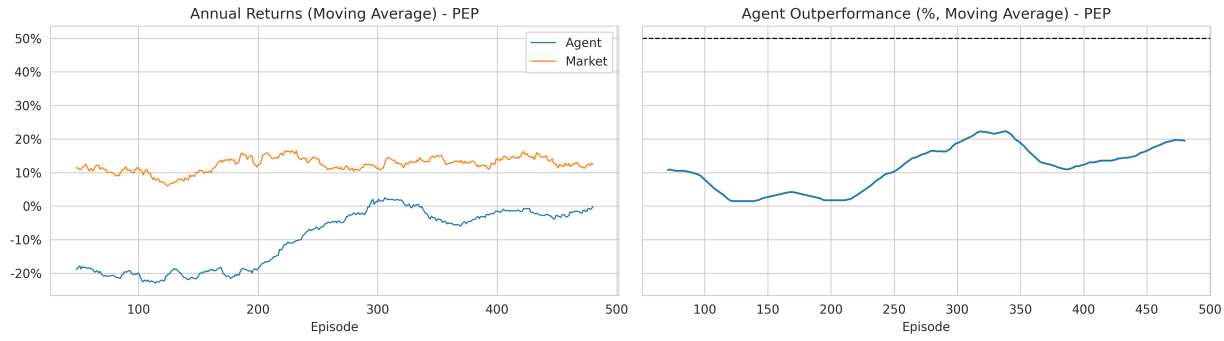


Figura B.19: Media móvil de rendimientos anuales

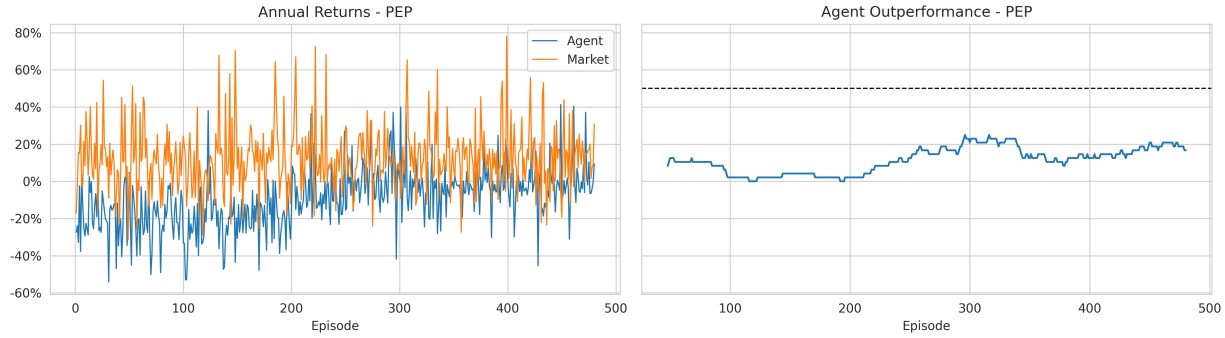


Figura B.20: Rendimientos anuales

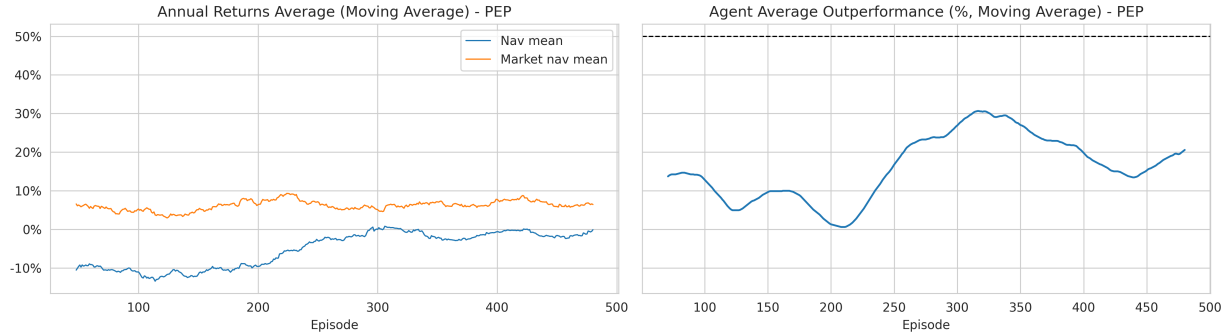


Figura B.21: Media móvil de rendimientos anuales promedio

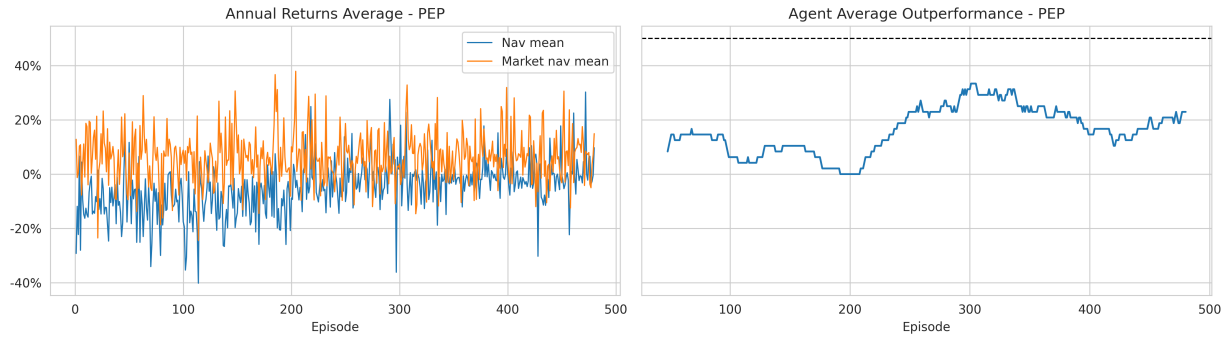


Figura B.22: Rendimientos anuales promedio

Finalmente, en la figura B.23 se muestra la media móvil sobre los últimos 50 episodios de la proporción de días en los que el agente superó la estrategia *buy-and-hold* durante un episodio. La figura B.24 muestra estos mismos resultados sin una media móvil sobre los episodios.

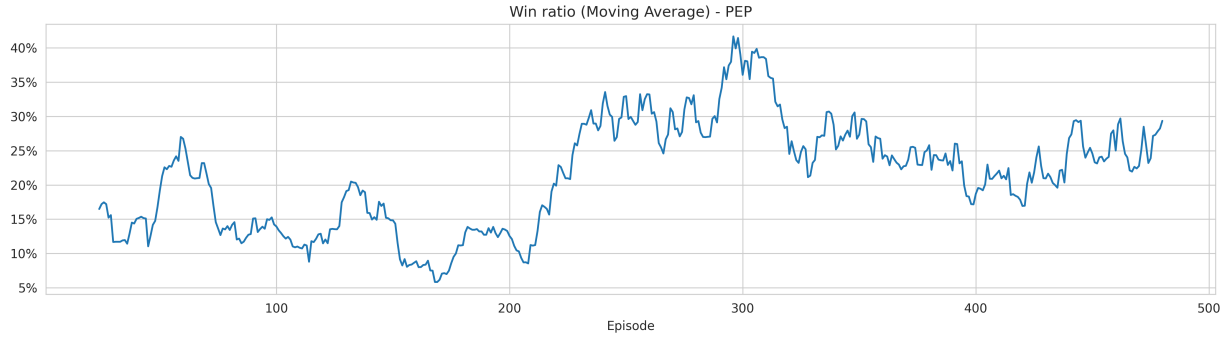


Figura B.23: Media móvil de proporción de ganancias

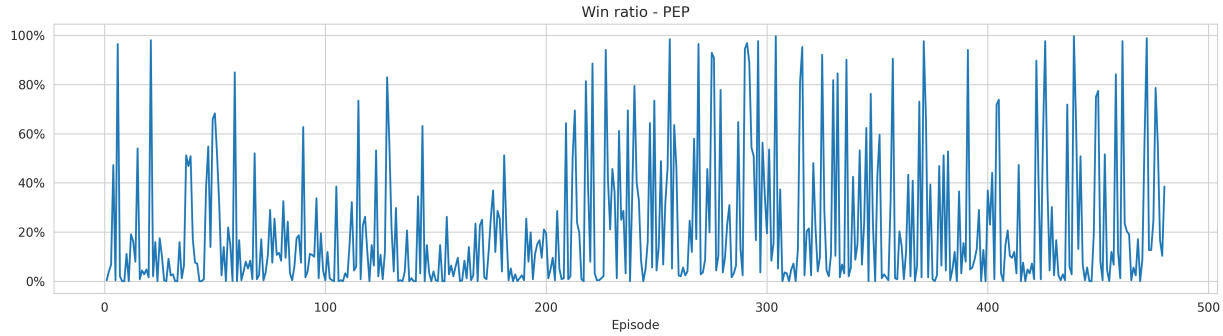


Figura B.24: Proporción de ganancias

B.1.2. Recompensas

B.1.2.1. Apple Inc (AAPL)

La figura B.25 muestra la media móvil sobre los últimos 25 episodios de la suma acumulada de recompensas obtenidas durante un episodio para los 480 periodos de entrenamiento. La figura B.26 muestra estos mismos resultados sin una media móvil sobre los episodios. De manera similar, en las figuras B.27 y B.28 se muestra, bajo esta misma configuración, el promedio de recompensas obtenidas durante un episodio.

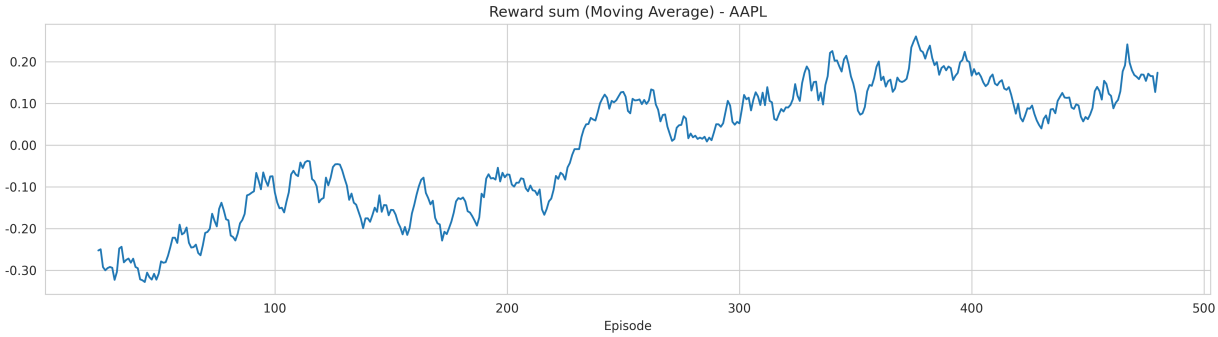


Figura B.25: Media móvil de suma acumulada de recompensas

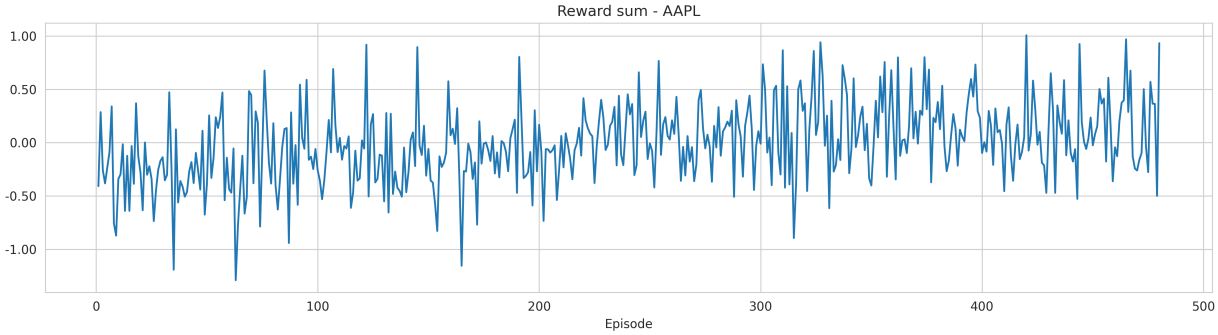


Figura B.26: Suma acumulada de recompensas

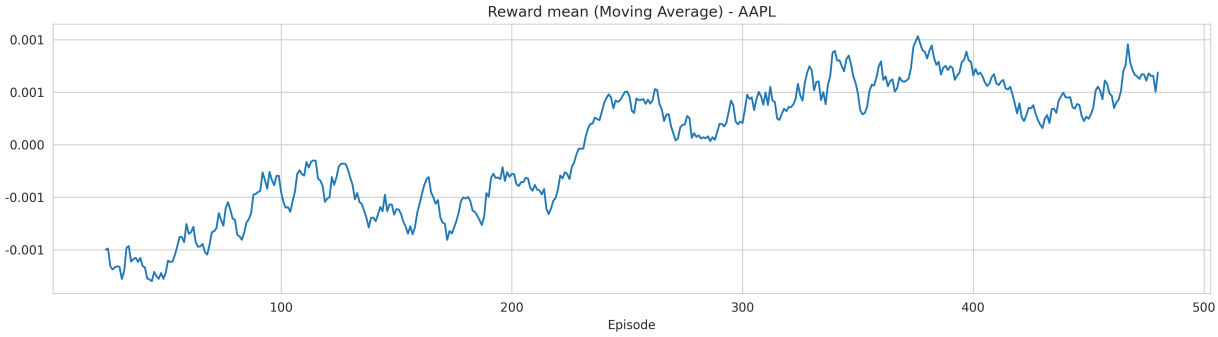


Figura B.27: Media móvil de promedio de recompensas

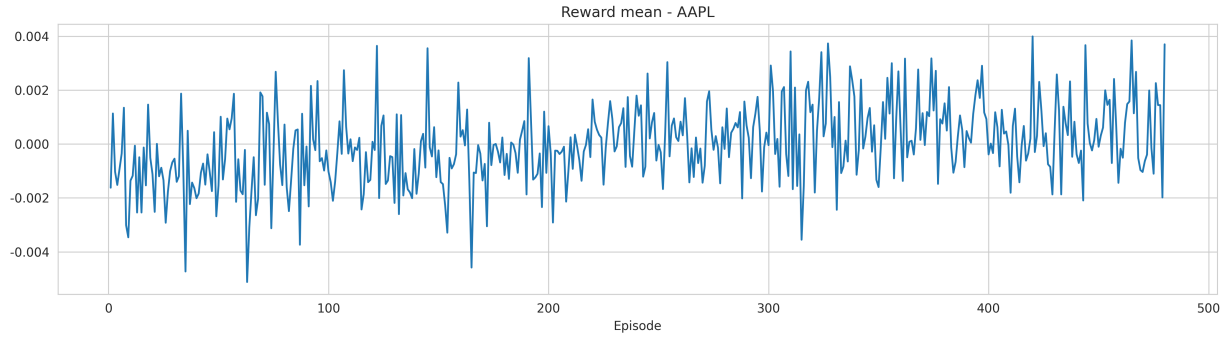


Figura B.28: Promedio de recompensas

Finalmente, en la figura B.29 se muestra la media móvil sobre los últimos 25 episodios de la diferencia entre la mayor y menor recompensa obtenidas durante un episodio. La figura B.30 muestra estos mismos resultados sin una media móvil sobre los episodios.

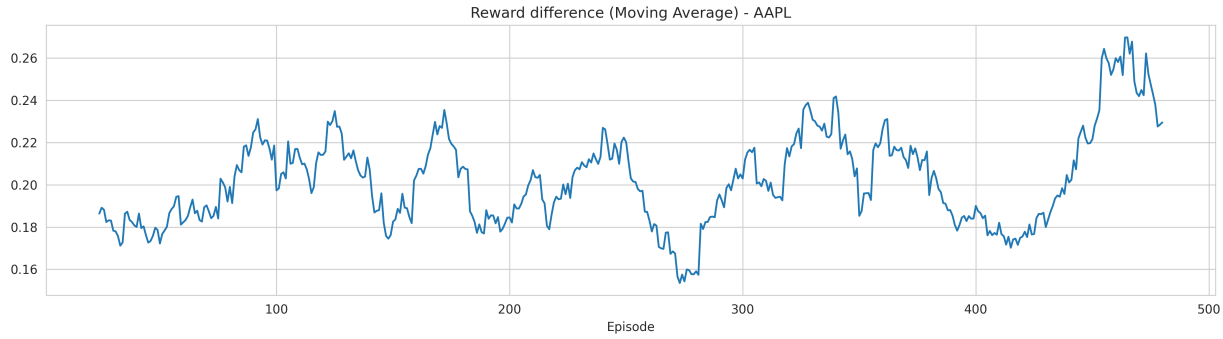


Figura B.29: Media móvil de diferencia de recompensa

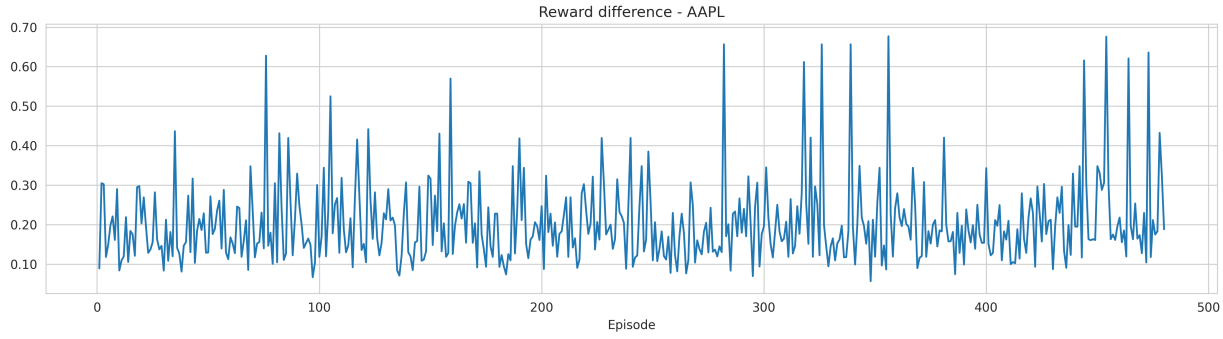


Figura B.30: Diferencia de recompensa

B.1.2.2. Microsoft (MSFT)

La figura B.31 muestra la media móvil sobre los últimos 25 episodios de la suma acumulada de recompensas obtenidas durante un episodio para los 480 periodos de entrenamiento. La figura B.32 muestra estos mismos resultados sin una media móvil sobre los episodios. De manera similar, en las figuras B.33 y B.34 se muestra, bajo esta misma configuración, el promedio de recompensas obtenidas durante un episodio.

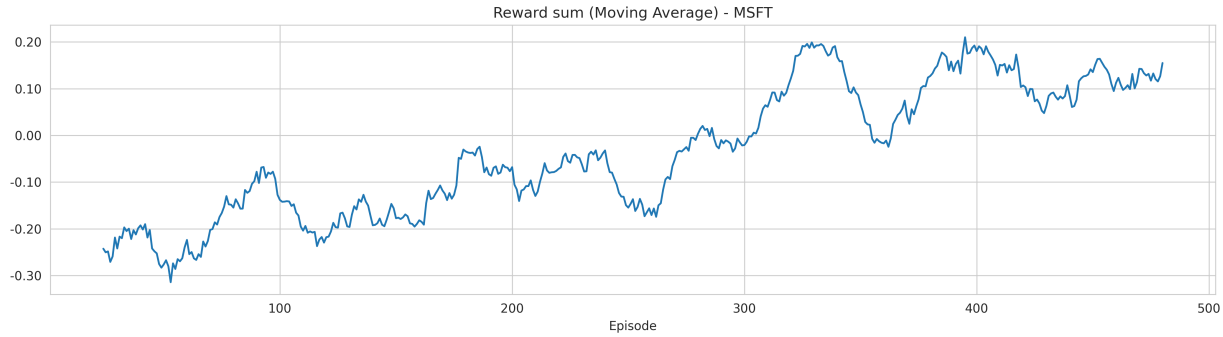


Figura B.31: Media móvil de suma acumulada de recompensas

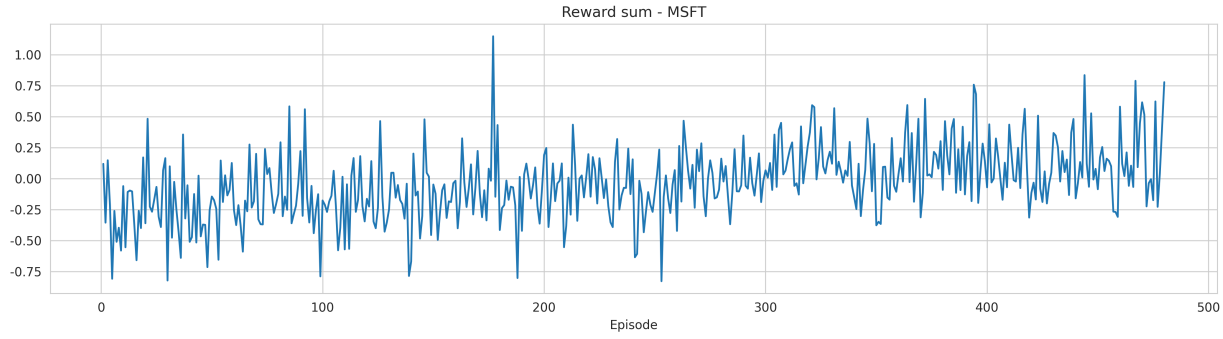


Figura B.32: Suma acumulada de recompensas

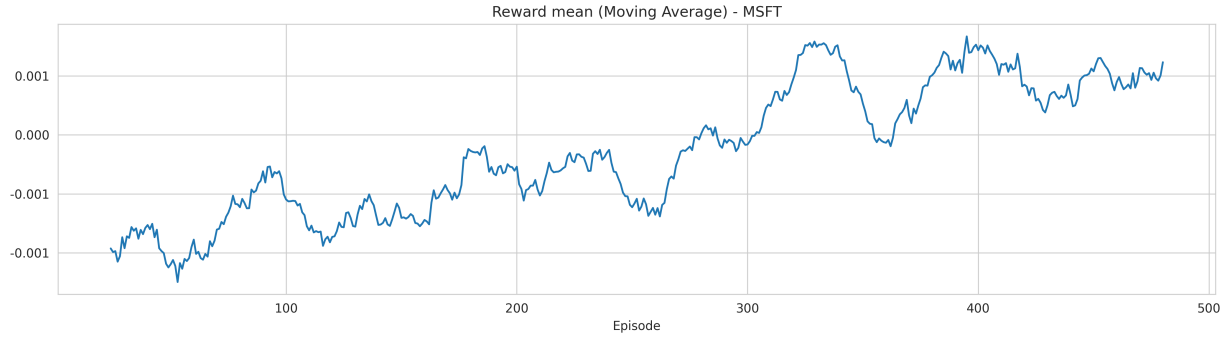


Figura B.33: Media móvil de promedio de recompensas

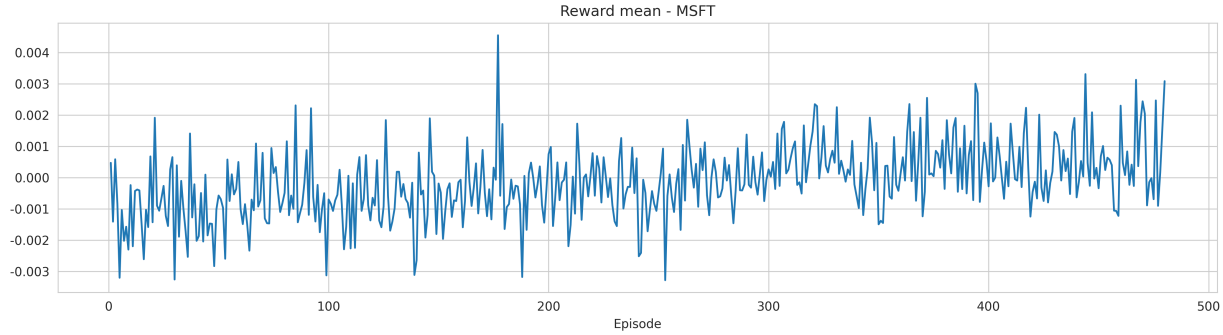


Figura B.34: Promedio de recompensas

Finalmente, en la figura B.35 se muestra la media móvil sobre los últimos 25 episodios de la diferencia entre la mayor y menor recompensa obtenidas durante un episodio. La figura B.36 muestra estos mismos resultados sin una media móvil sobre los episodios.

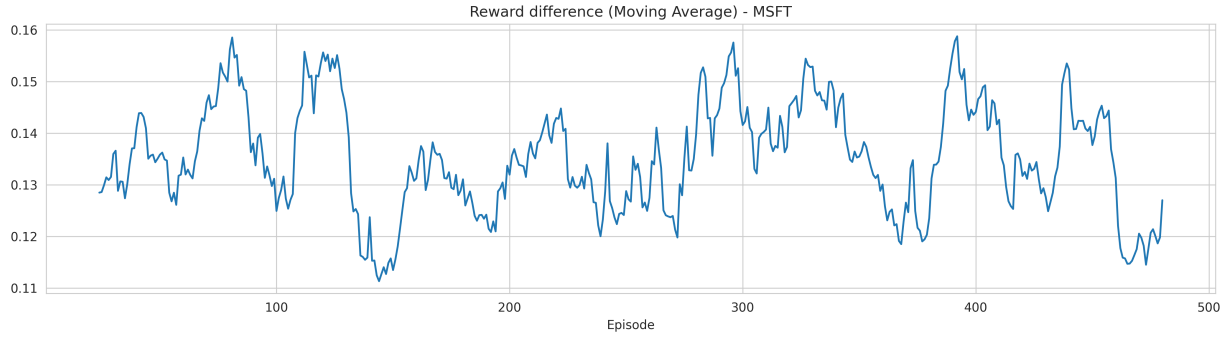


Figura B.35: Media móvil de diferencia de recompensa

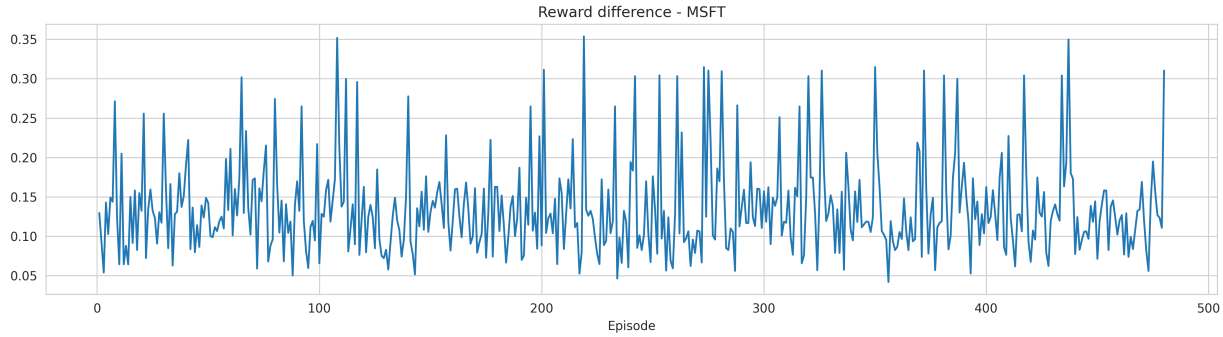


Figura B.36: Diferencia de recompensa

B.1.2.3. Amazon Inc (AMZN)

La figura B.37 muestra la media móvil sobre los últimos 25 episodios de la suma acumulada de recompensas obtenidas durante un episodio para los 480 periodos de entrenamiento. La figura B.38 muestra estos mismos resultados sin una media móvil sobre los episodios. De manera similar, en las figuras B.39 y B.40 se muestra, bajo esta misma configuración, el promedio de recompensas obtenidas durante un episodio.

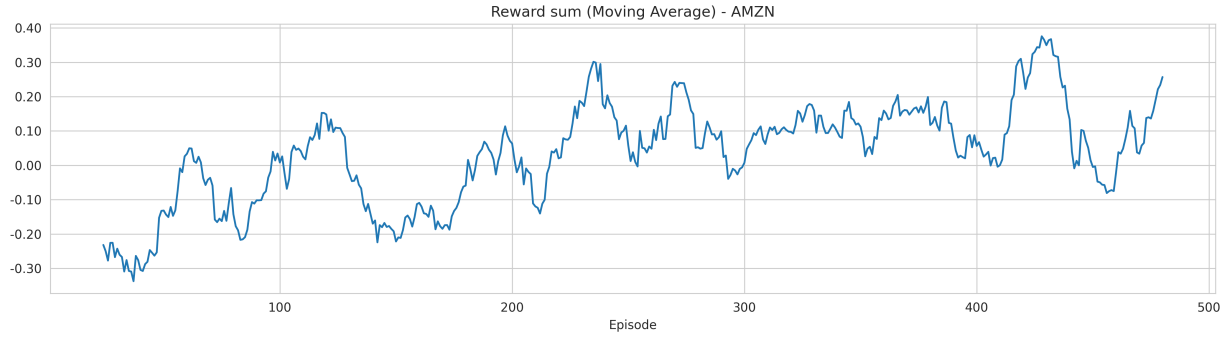


Figura B.37: Media móvil de suma acumulada de recompensas

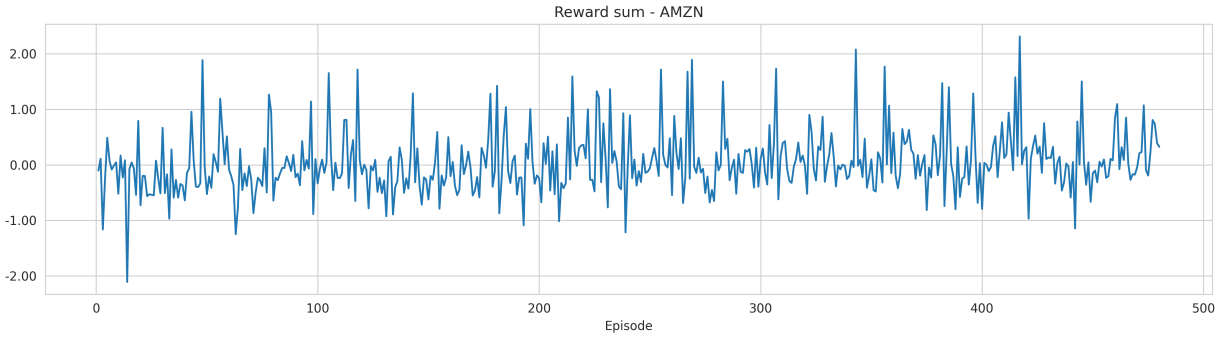


Figura B.38: Suma acumulada de recompensas

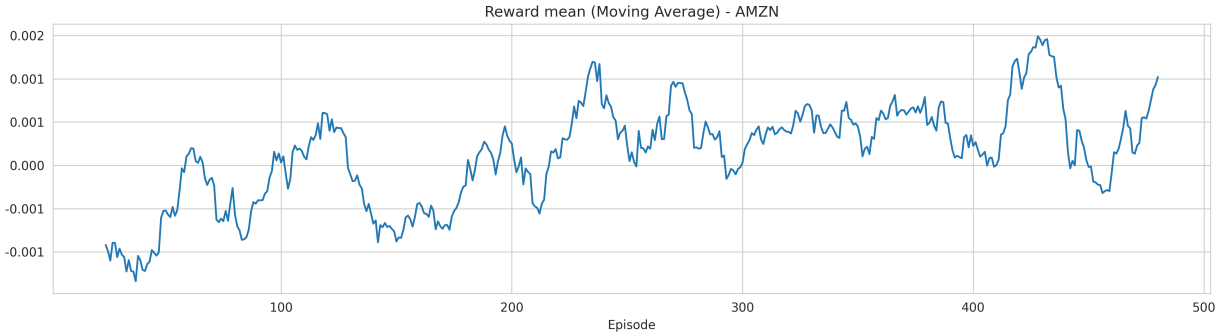


Figura B.39: Media móvil de promedio de recompensas

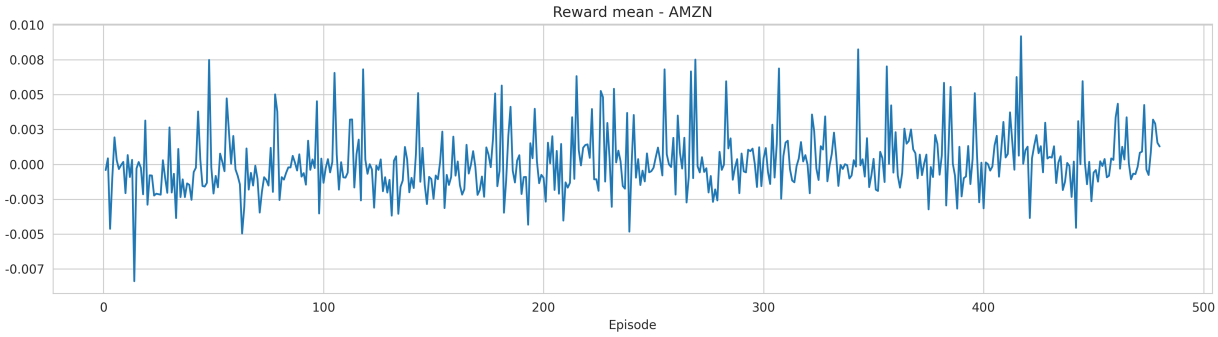


Figura B.40: Promedio de recompensas

Finalmente, en la figura B.41 se muestra la media móvil sobre los últimos 25 episodios de la diferencia entre la mayor y menor recompensa obtenidas durante un episodio. La figura B.42 muestra estos mismos resultados sin una media móvil sobre los episodios.

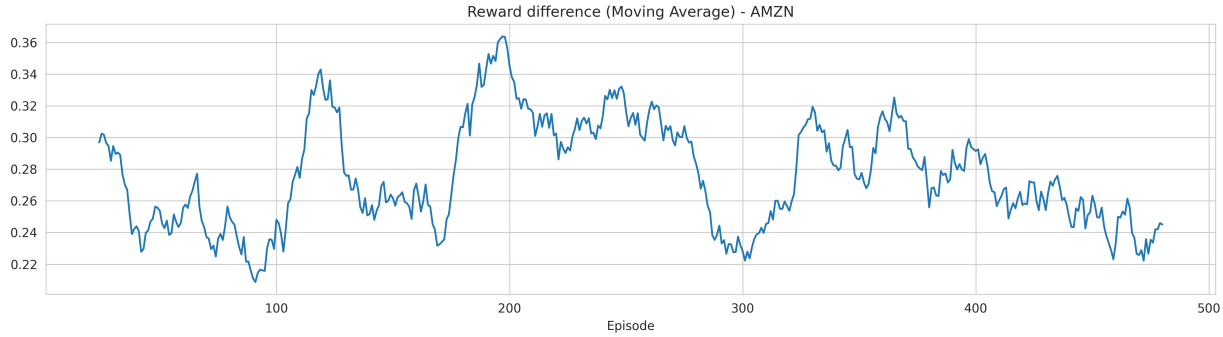


Figura B.41: Media móvil de diferencia de recompensa

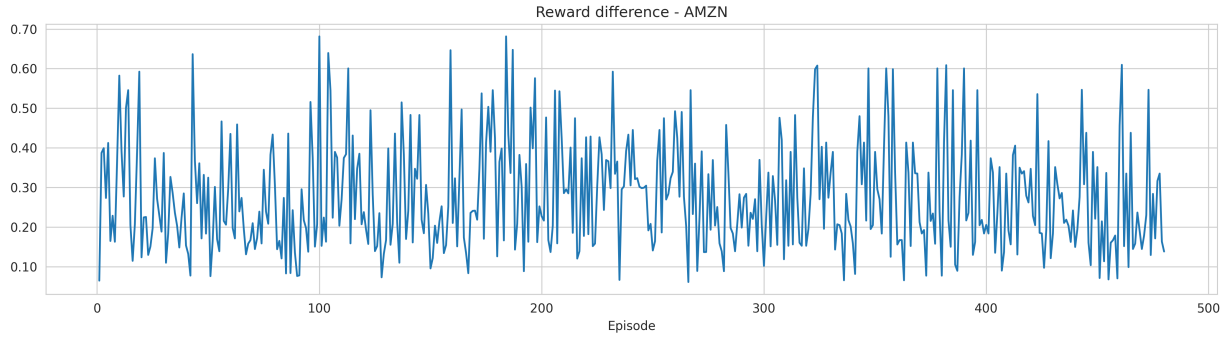


Figura B.42: Diferencia de recompensa

B.1.2.4. Pepsico Inc (PEP)

La figura B.43 muestra la media móvil sobre los últimos 25 episodios de la suma acumulada de recompensas obtenidas durante un episodio para los 480 periodos de entrenamiento. La figura B.44 muestra estos mismos resultados sin una media móvil sobre los episodios. De manera similar, en las figuras B.45 y B.46 se muestra, bajo esta misma configuración, el promedio de recompensas obtenidas durante un episodio.

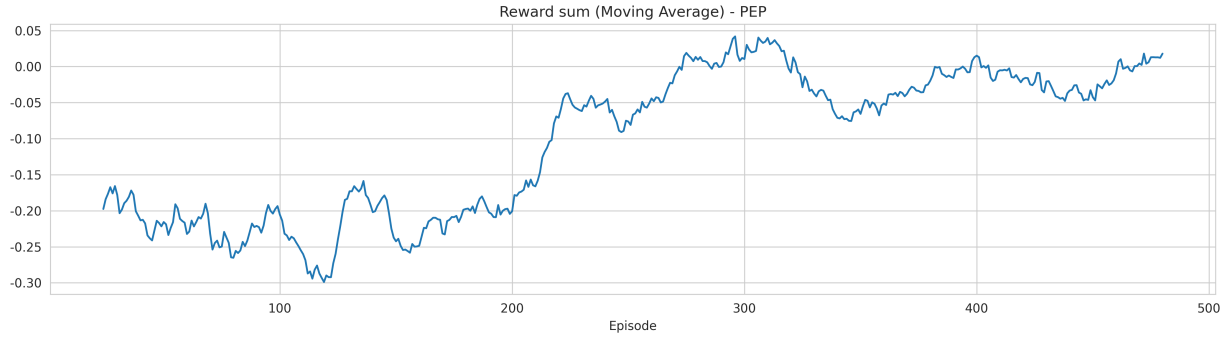


Figura B.43: Media móvil de suma acumulada de recompensas

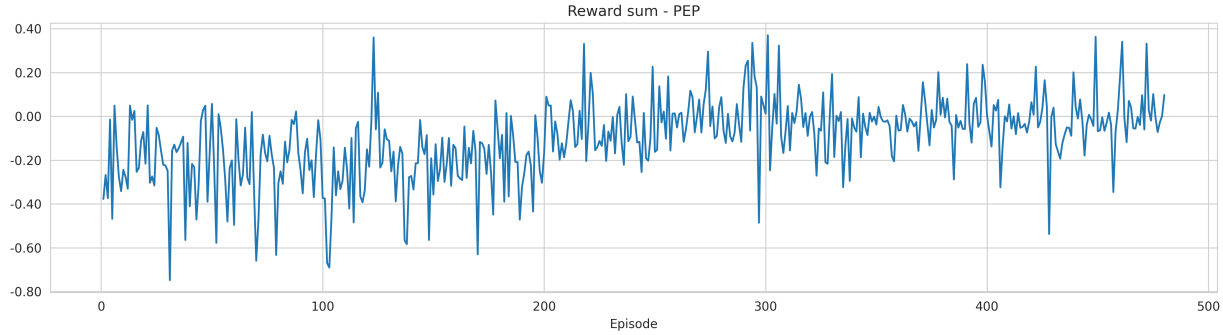


Figura B.44: Suma acumulada de recompensas

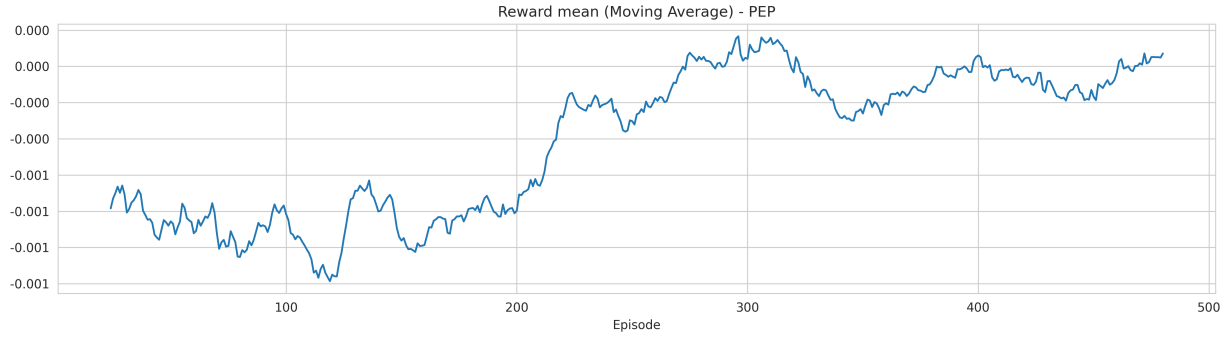


Figura B.45: Media móvil de promedio de recompensas

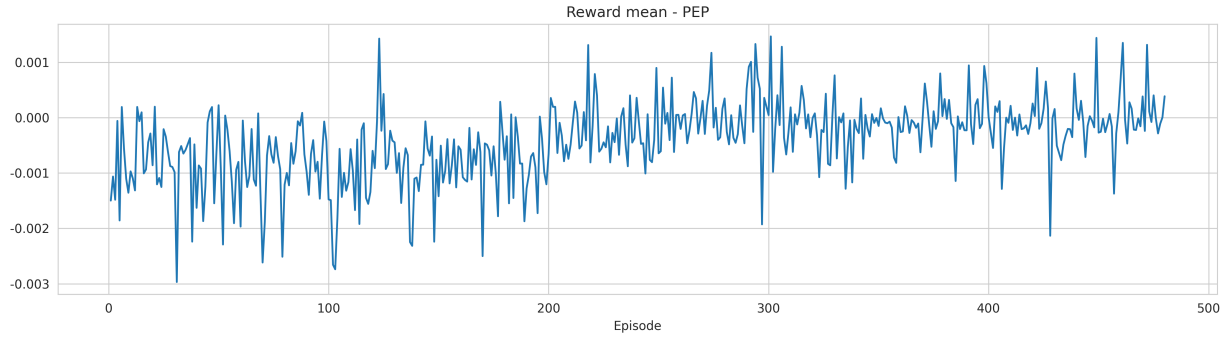


Figura B.46: Promedio de recompensas

Finalmente, en la figura B.47 se muestra la media móvil sobre los últimos 25 episodios de la diferencia entre la mayor y menor recompensa obtenidas durante un episodio. La figura B.48 muestra estos mismos resultados sin una media móvil sobre los episodios.

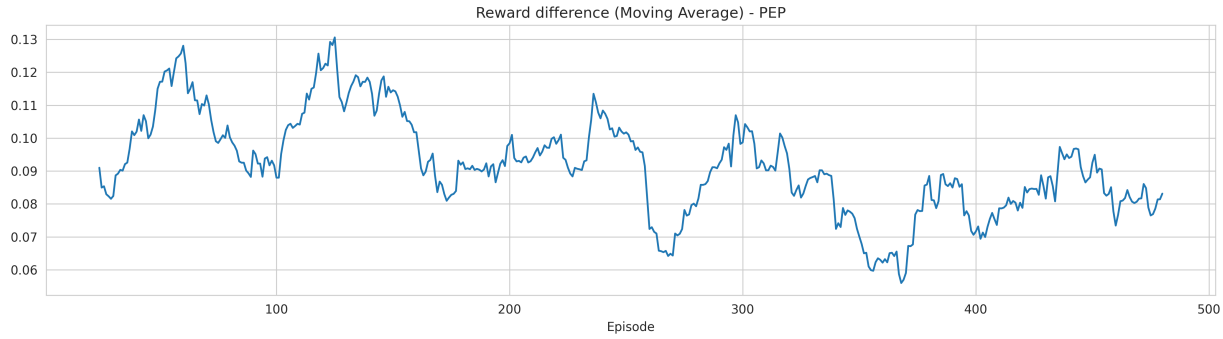


Figura B.47: Media móvil de diferencia de recompensa

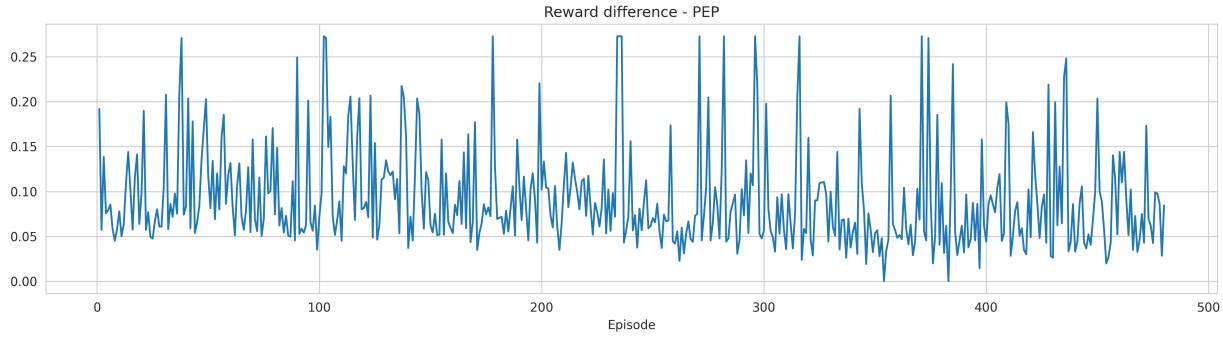


Figura B.48: Diferencia de recompensa

B.1.3. Costos

B.1.3.1. Apple Inc (AAPL)

La figura B.49 muestra la media móvil sobre los último 25 episodios de la suma acumulada de costos obtenidos durante un episodio para los 480 periodos de entrenamiento. La figura B.50 muestra estos mismos resultados sin una media móvil sobre los episodios. De manera similar, en las figuras B.51 y B.52 se muestra, bajo esta misma configuración, el promedio de costos obtenidos durante un episodio.

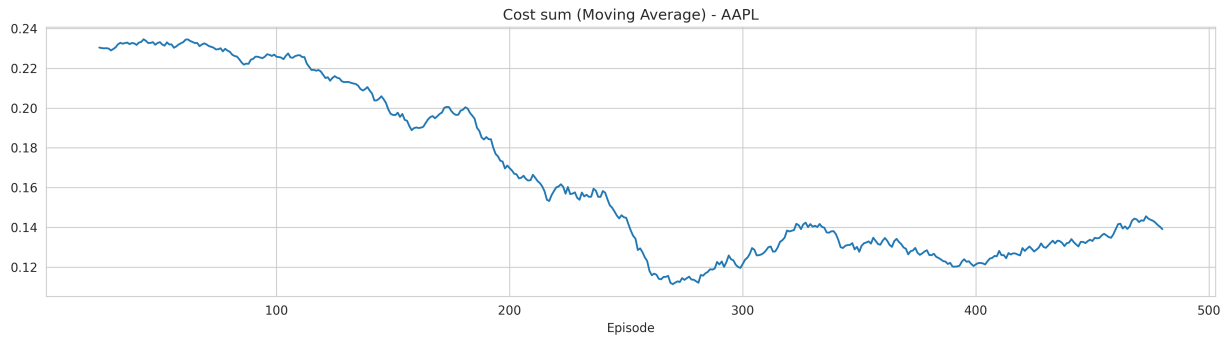


Figura B.49: Media móvil de suma acumulada de costos

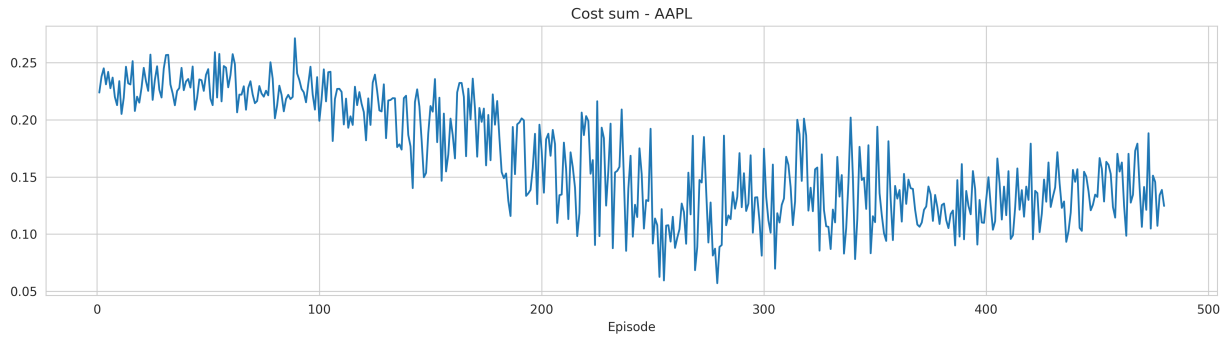


Figura B.50: Suma acumulada de costos

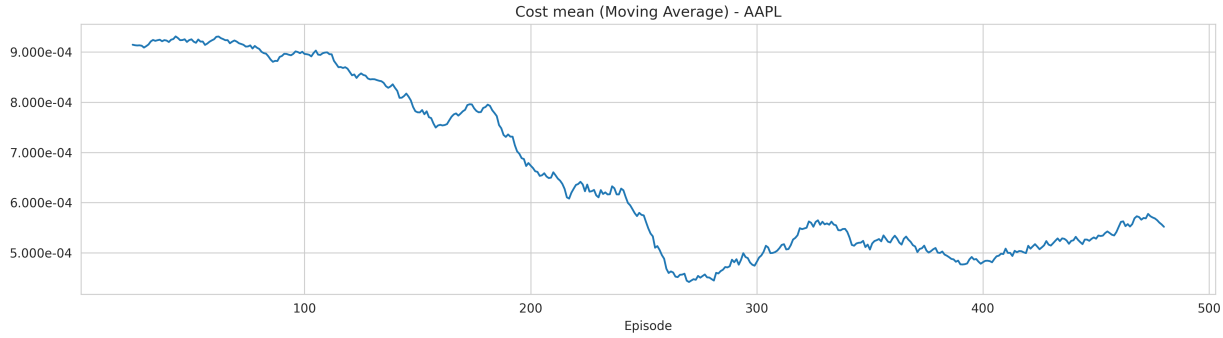


Figura B.51: Media móvil de promedio de costos

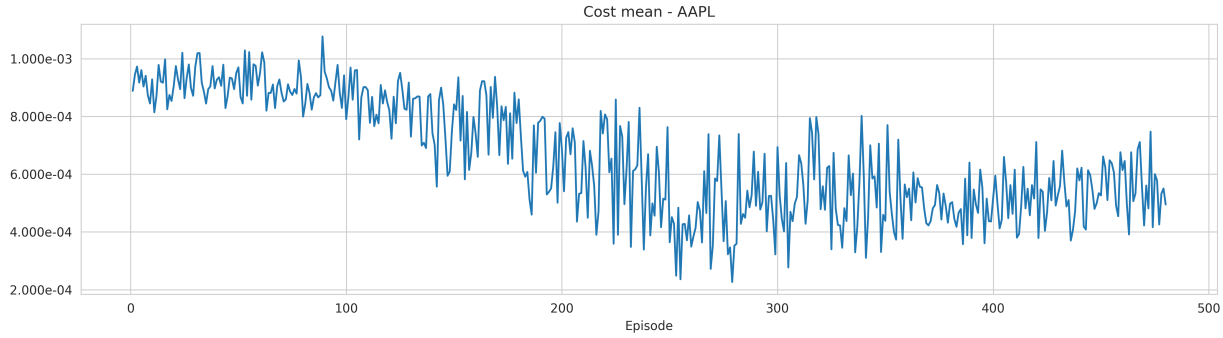


Figura B.52: Promedio de costos

B.1.3.2. Microsoft (MSFT)

La figura B.53 muestra la media móvil sobre los último 25 episodios de la suma acumulada de costos obtenidos durante un episodio para los 480 periodos de entrenamiento. La figura B.54 muestra estos mismos resultados sin una media móvil sobre los episodios. De manera similar, en las figuras B.55 y B.56 se muestra, bajo esta misma configuración, el promedio de costos obtenidos durante un episodio.

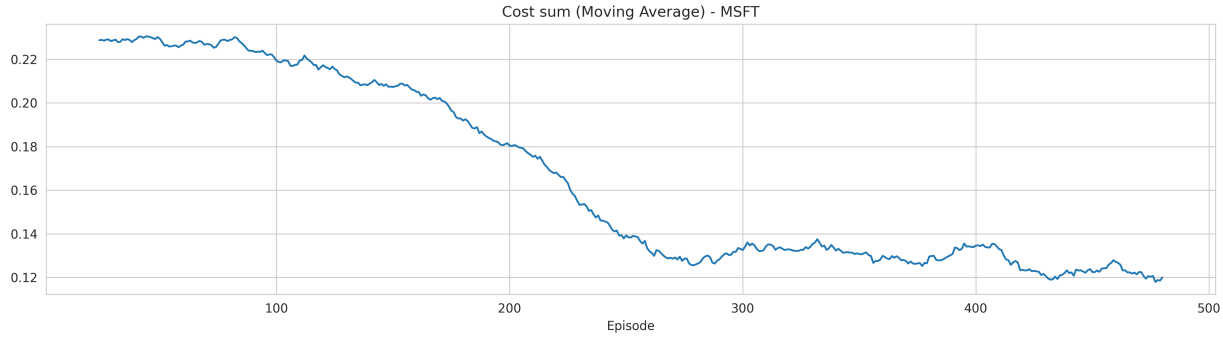


Figura B.53: Media móvil de suma acumulada de costos

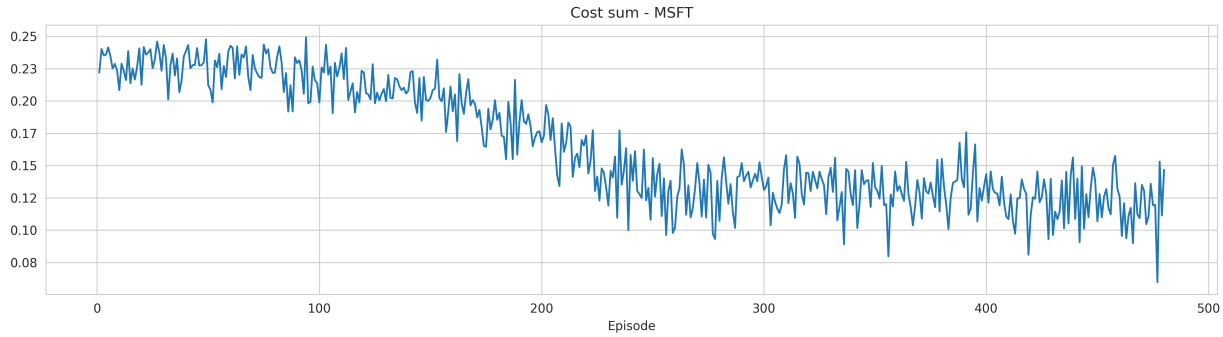


Figura B.54: Suma acumulada de costos

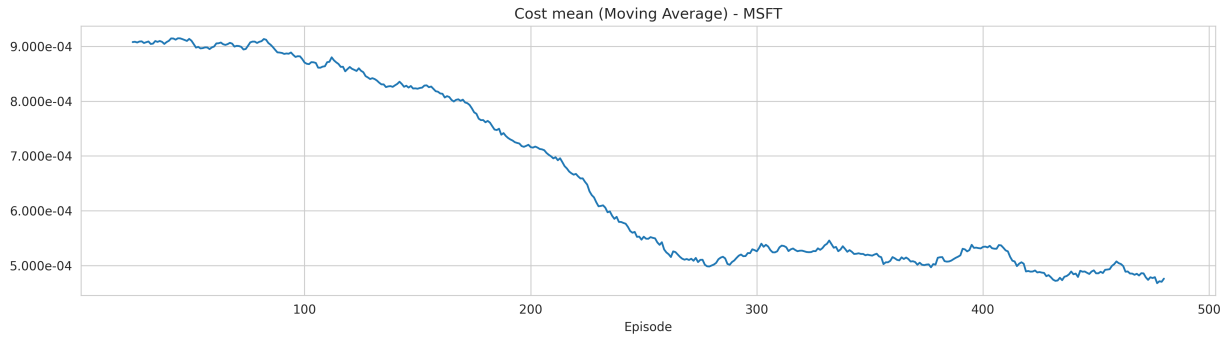


Figura B.55: Media móvil de promedio de costos

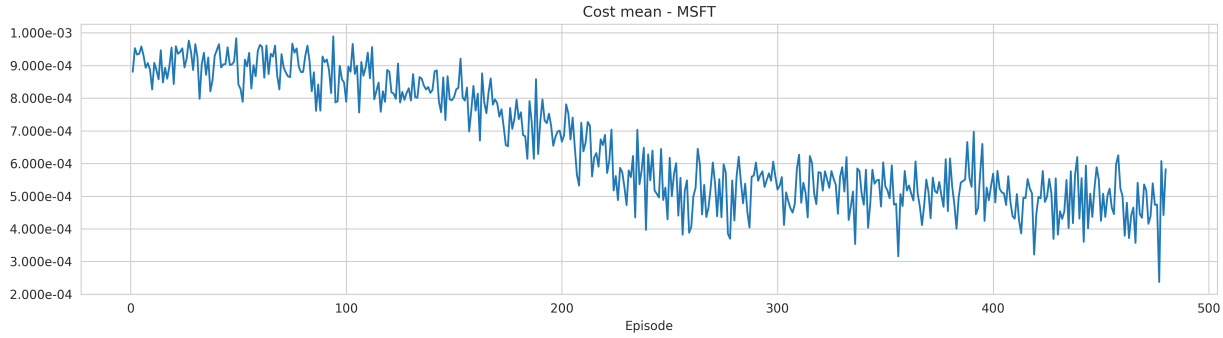


Figura B.56: Promedio de costos

B.1.3.3. Amazon Inc (AMZN)

La figura B.57 muestra la media móvil sobre los último 25 episodios de la suma acumulada de costos obtenidos durante un episodio para los 480 periodos de entrenamiento. La figura B.58 muestra estos mismos resultados sin una media móvil sobre los episodios. De manera similar, en las figuras B.59 y B.60 se muestra, bajo esta misma configuración, el promedio de costos obtenidos durante un episodio.

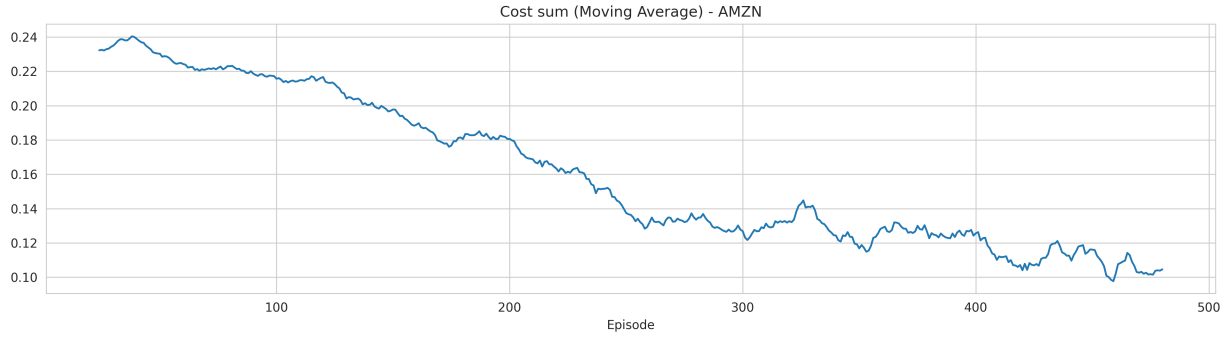


Figura B.57: Media móvil de suma acumulada de costos

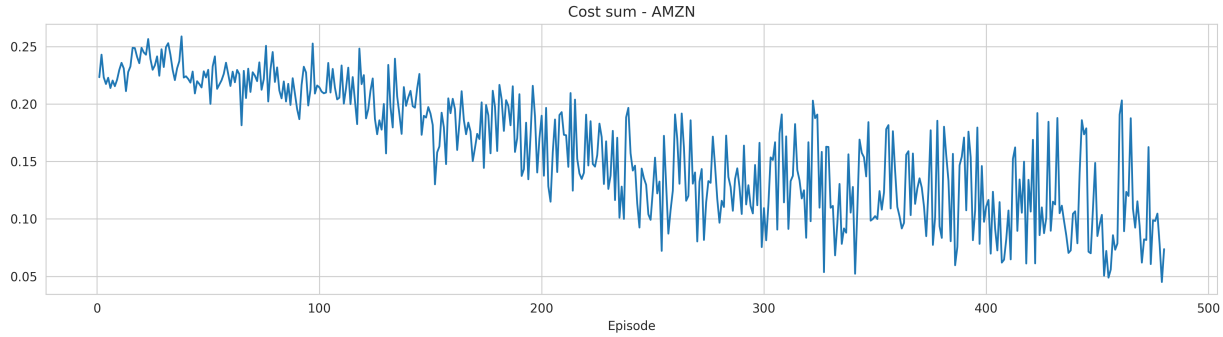


Figura B.58: Suma acumulada de costos

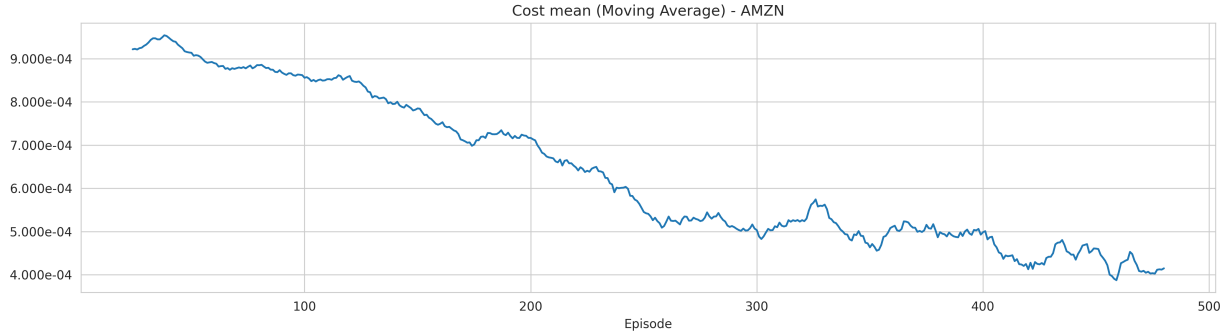


Figura B.59: Media móvil de promedio de costos

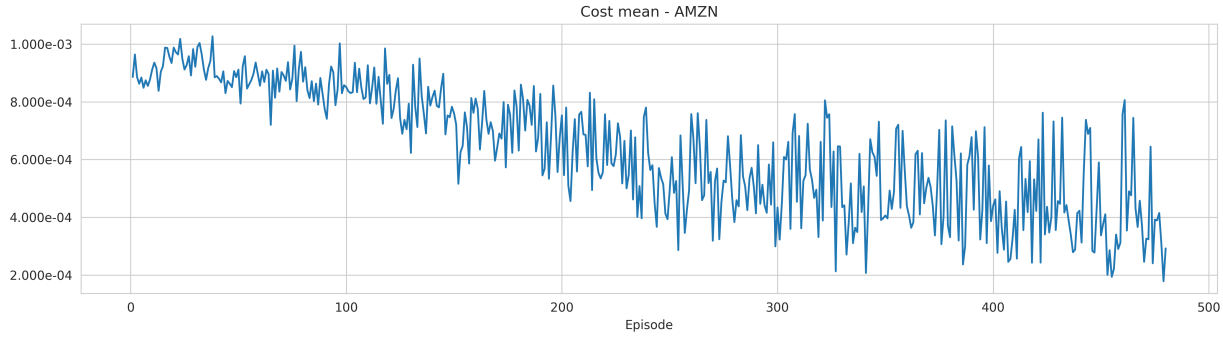


Figura B.60: Promedio de costos

B.1.3.4. Pepsico Inc (PEP)

La figura B.61 muestra la media móvil sobre los último 25 episodios de la suma acumulada de costos obtenidos durante un episodio para los 480 periodos de entrenamiento. La figura B.62 muestra estos mismos resultados sin una media móvil sobre los episodios. De manera similar, en las figuras B.63 y B.64 se muestra, bajo esta misma configuración, el promedio de costos obtenidos durante un episodio.

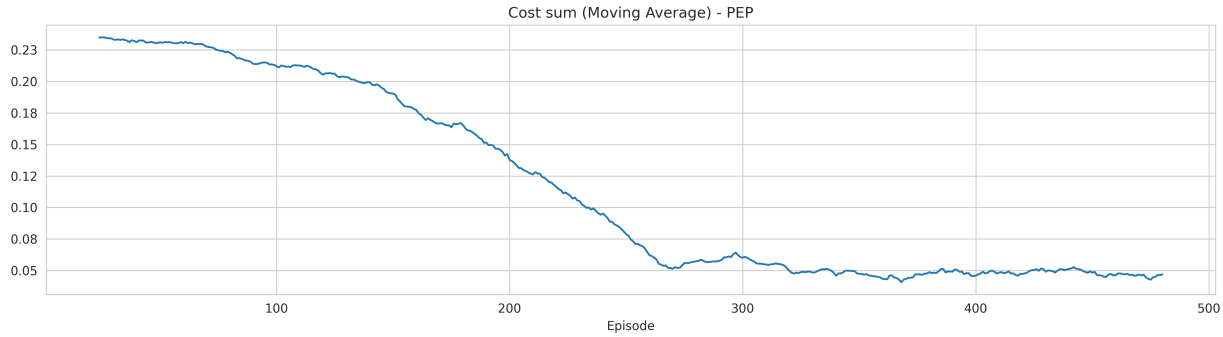


Figura B.61: Media móvil de suma acumulada de costos

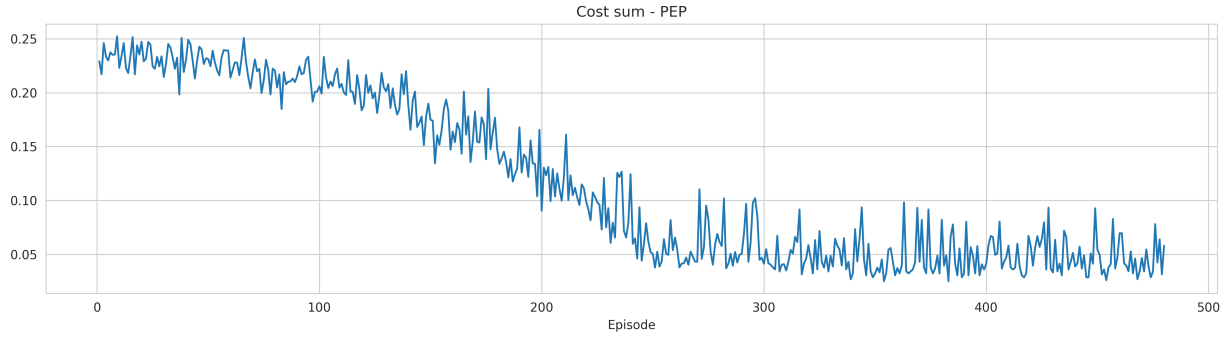


Figura B.62: Suma acumulada de costos

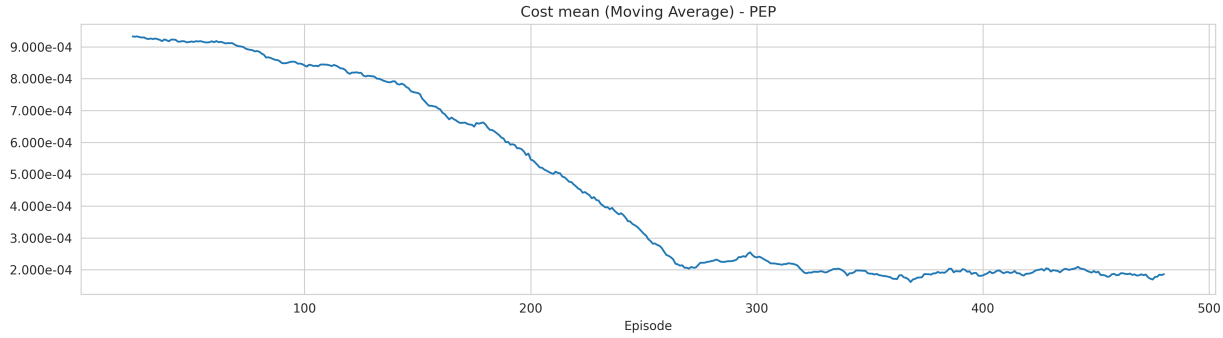


Figura B.63: Media móvil de promedio de costos

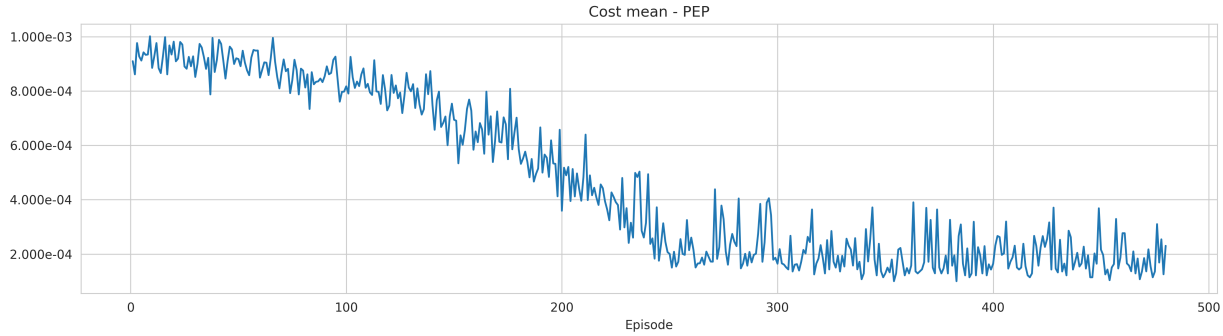


Figura B.64: Promedio de costos

B.1.4. Operaciones de trading

B.1.4.1. Apple Inc (AAPL)

La figura B.65 muestra la media móvil sobre los últimos 25 episodios de la suma acumulada de operaciones de trading durante un episodio para los 480 periodos de entrenamiento. La figura B.66 muestra estos mismos resultados sin una media móvil sobre los episodios. De manera similar, en las figuras B.67 y B.68 se muestra, bajo esta misma configuración, el promedio de operaciones de trading realizadas cada día durante un episodio.

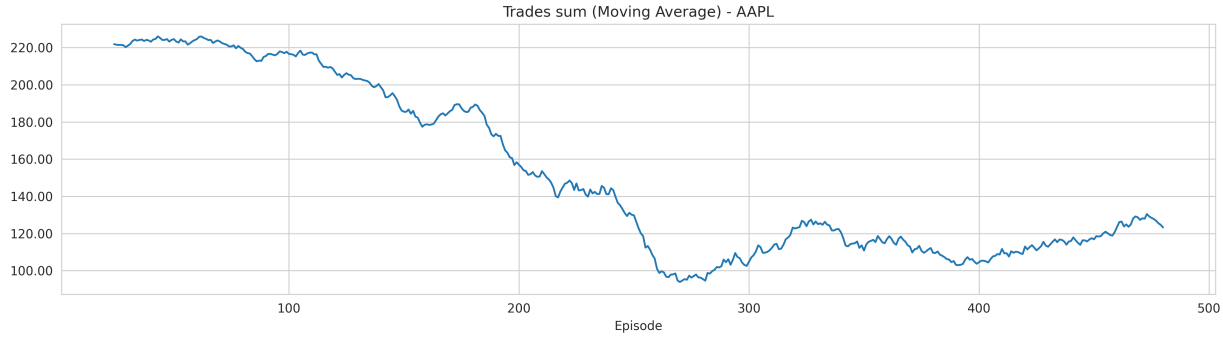


Figura B.65: Media móvil de suma acumulada de operaciones

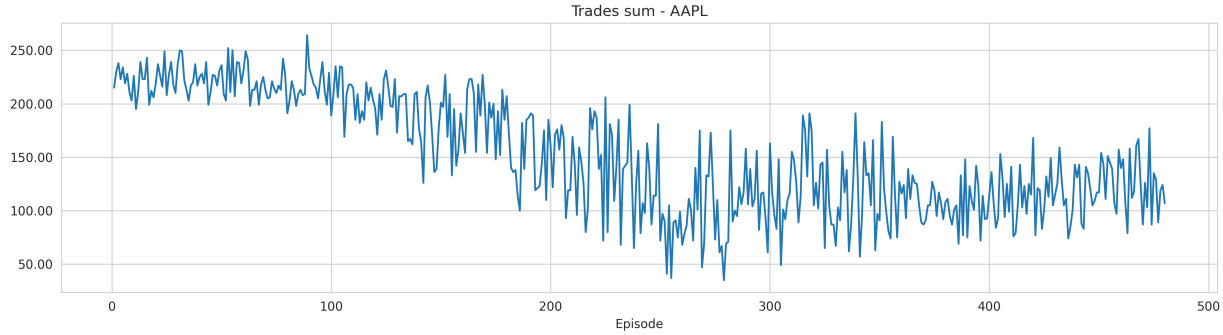


Figura B.66: Suma acumulada de operaciones

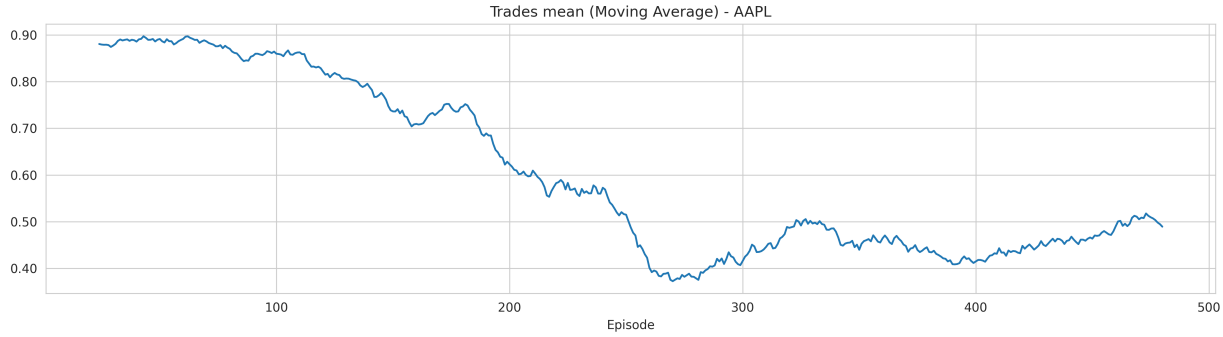


Figura B.67: Media móvil de promedio de operaciones

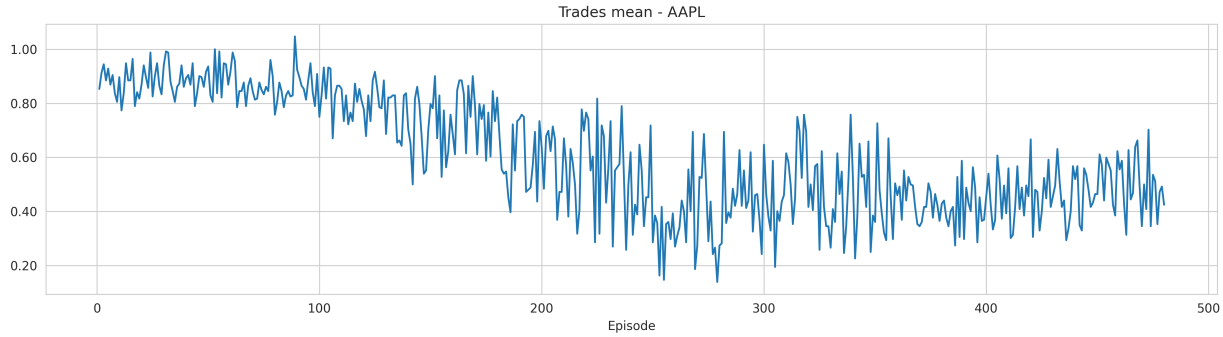


Figura B.68: Promedio de operaciones

B.1.4.2. Microsoft (MSFT)

La figura B.69 muestra la media móvil sobre los últimos 25 episodios de la suma acumulada de operaciones de trading durante un episodio para los 480 periodos de entrenamiento. La figura B.70 muestra estos mismos resultados sin una media móvil sobre los episodios. De manera similar, en las figuras B.71 y B.72 se muestra, bajo esta misma configuración, el promedio de operaciones de trading realizadas cada día durante un episodio.

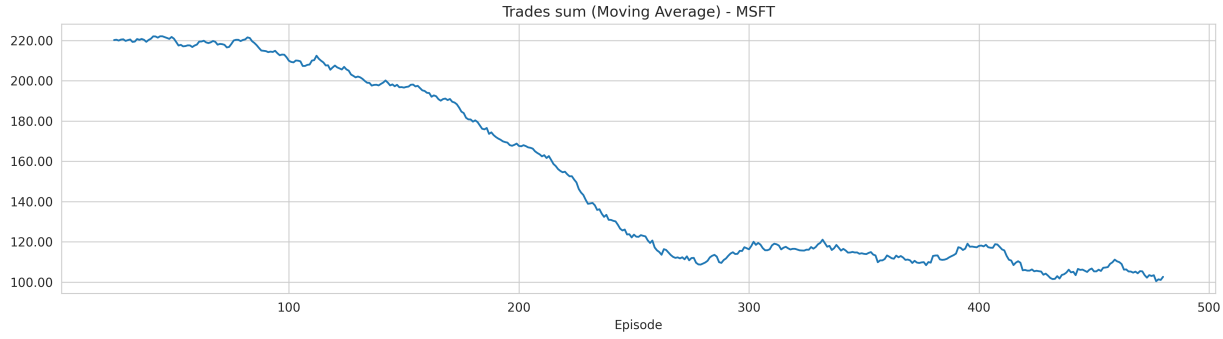


Figura B.69: Media móvil de suma acumulada de operaciones

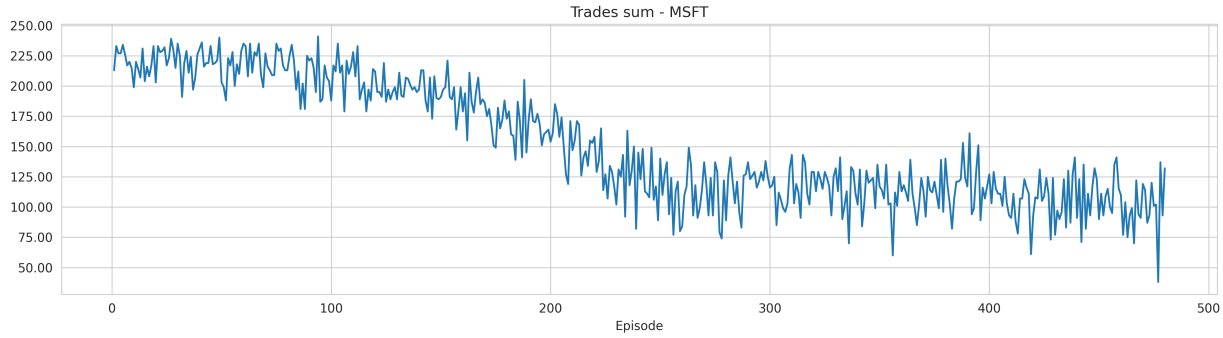


Figura B.70: Suma acumulada de operaciones

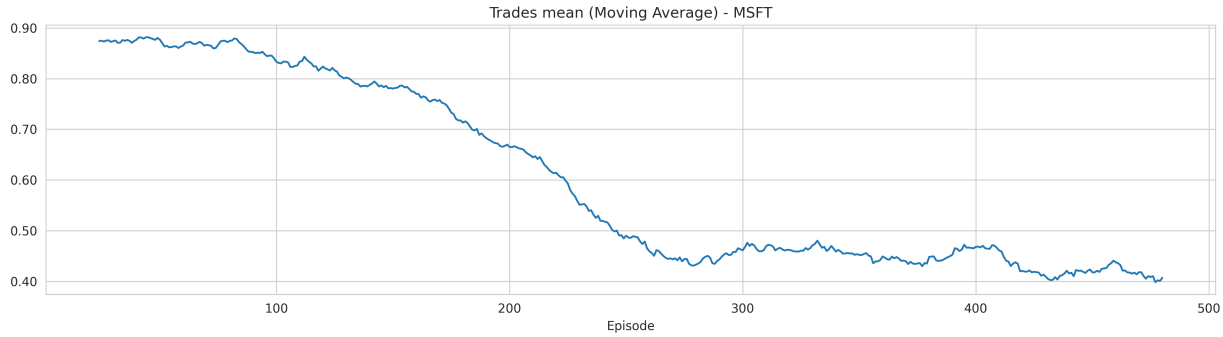


Figura B.71: Media móvil de promedio de operaciones

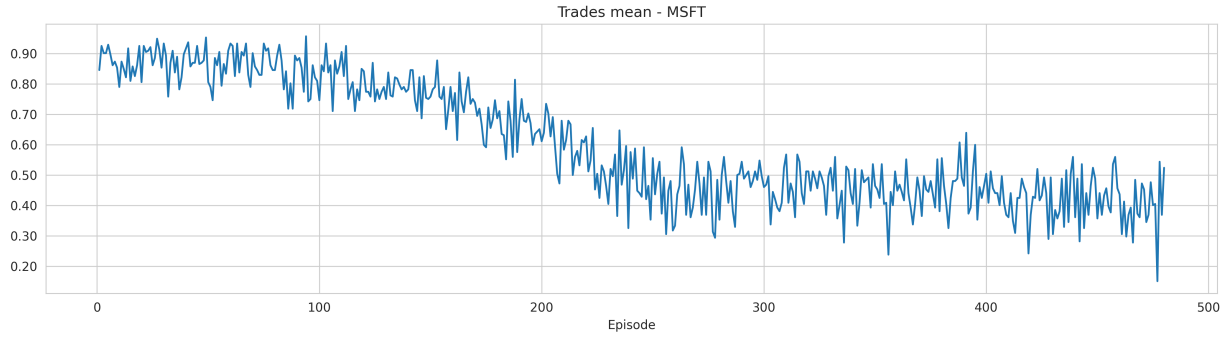


Figura B.72: Promedio de operaciones

B.1.4.3. Amazon Inc (AMZN)

La figura B.73 muestra la media móvil sobre los últimos 25 episodios de la suma acumulada de operaciones de trading durante un episodio para los 480 periodos de entrenamiento. La figura B.74 muestra estos mismos resultados sin una media móvil sobre los episodios. De manera similar, en las figuras B.75 y B.76 se muestra, bajo esta misma configuración, el promedio de operaciones de trading realizadas cada día durante un episodio.

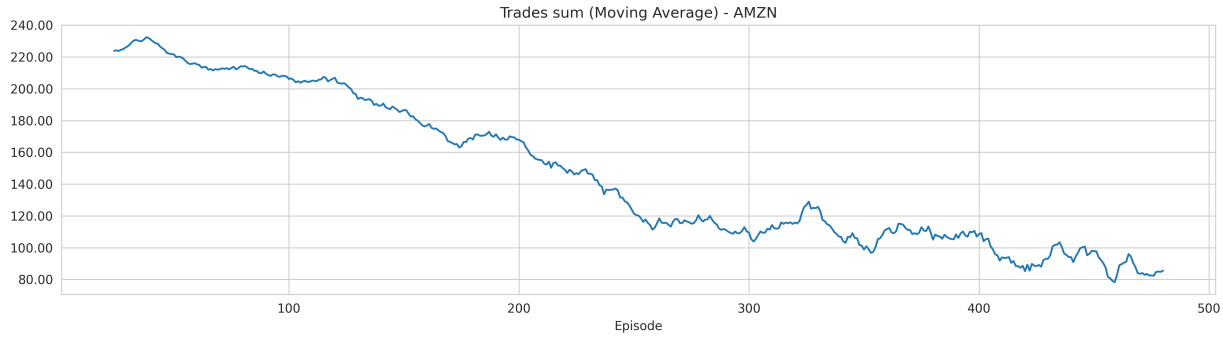


Figura B.73: Media móvil de suma acumulada de operaciones

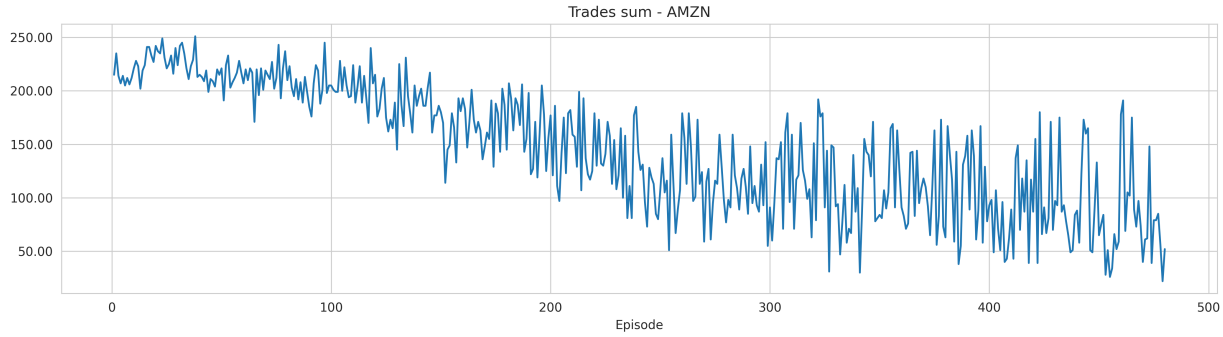


Figura B.74: Suma acumulada de operaciones

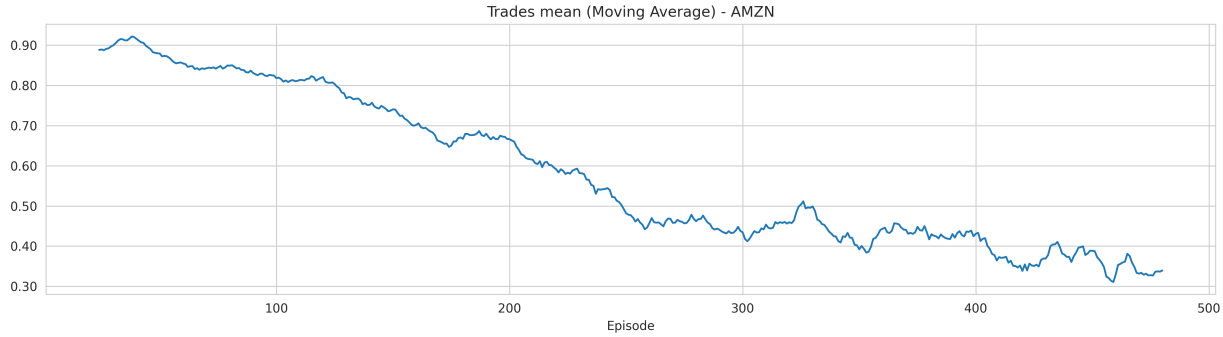


Figura B.75: Media móvil de promedio de operaciones

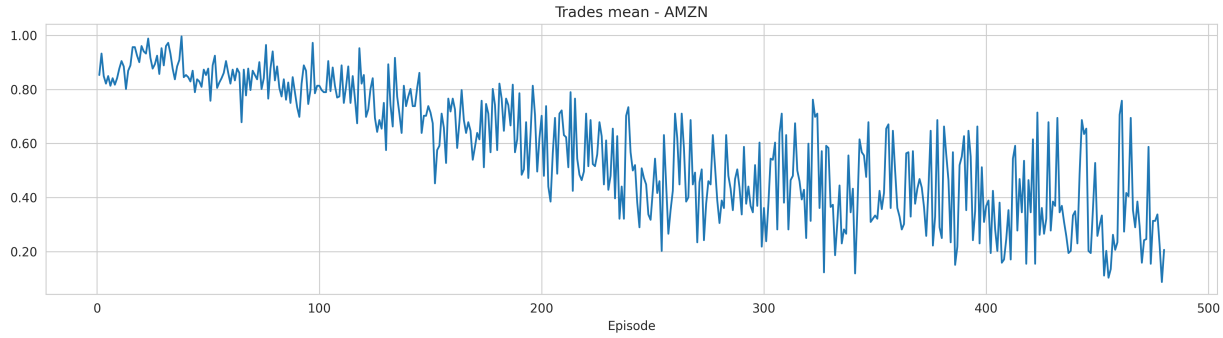


Figura B.76: Promedio de operaciones

B.1.4.4. Pepsico Inc (PEP)

La figura B.77 muestra la media móvil sobre los últimos 25 episodios de la suma acumulada de operaciones de trading durante un episodio para los 480 periodos de entrenamiento. La figura B.78 muestra estos mismos resultados sin una media móvil sobre los episodios. De manera similar, en las figuras B.79 y B.80 se muestra, bajo esta misma configuración, el promedio de operaciones de trading realizadas cada día durante un episodio.

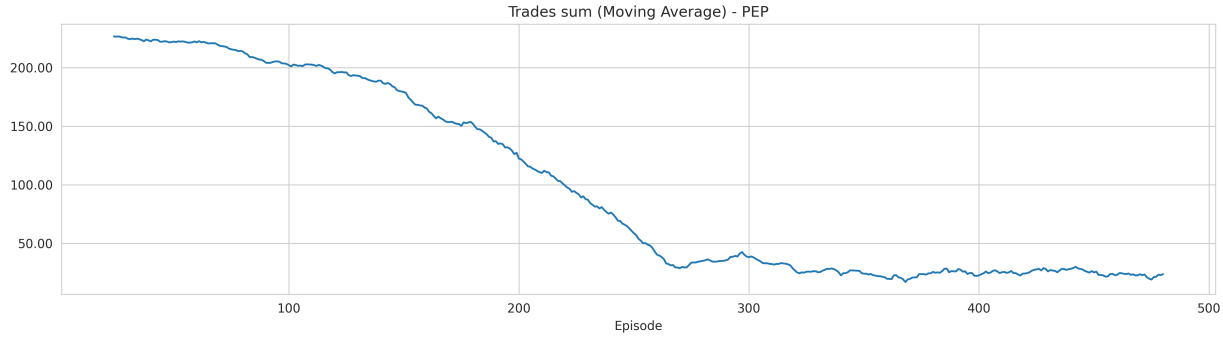


Figura B.77: Media móvil de suma acumulada de operaciones

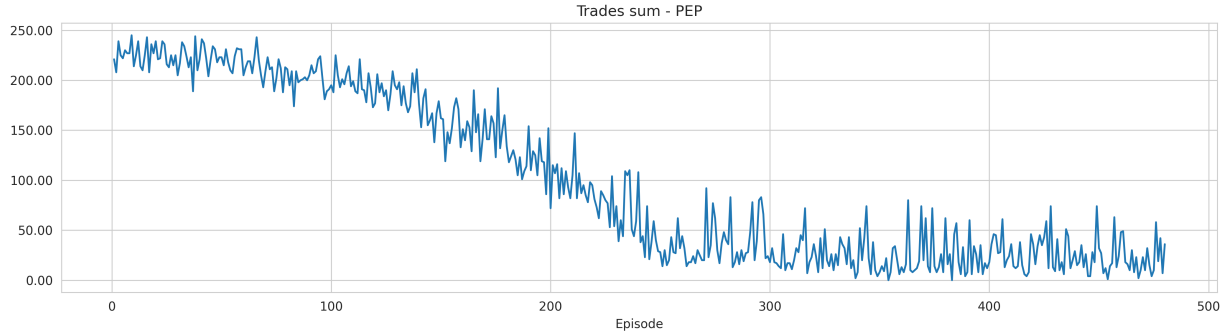


Figura B.78: Suma acumulada de operaciones

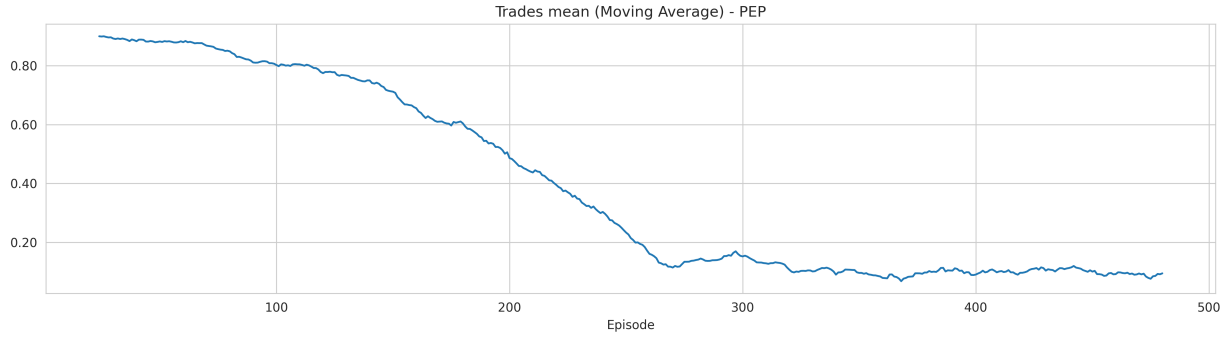


Figura B.79: Media móvil de promedio de operaciones

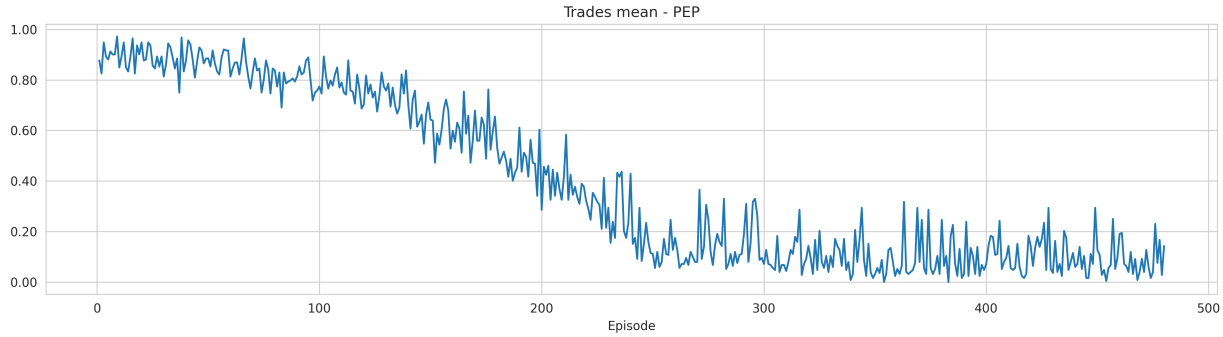


Figura B.80: Promedio de operaciones

B.1.5. Función de pérdida

B.1.5.1. Apple Inc (AAPL)

La figura B.81 muestra la media móvil sobre los últimos 25 episodios del promedio de pérdidas acumuladas para los 480 periodos de entrenamiento.

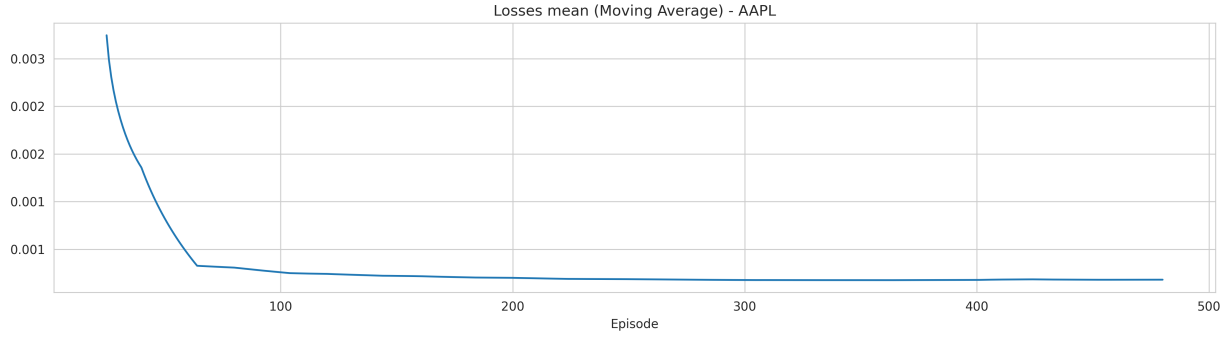


Figura B.81: Media móvil de promedio de pérdidas acumuladas

B.1.5.2. Microsoft (MSFT)

La figura B.82 muestra la media móvil sobre los últimos 25 episodios del promedio de pérdidas acumuladas para los 480 periodos de entrenamiento.

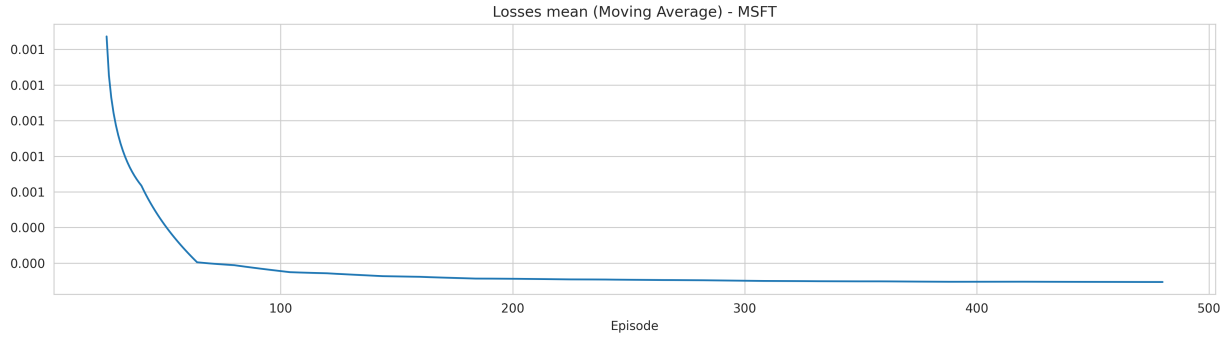


Figura B.82: Media móvil de promedio de pérdidas acumuladas

B.1.5.3. Amazon Inc (AMZN)

La figura B.83 muestra la media móvil sobre los últimos 25 episodios del promedio de pérdidas acumuladas para los 480 periodos de entrenamiento.

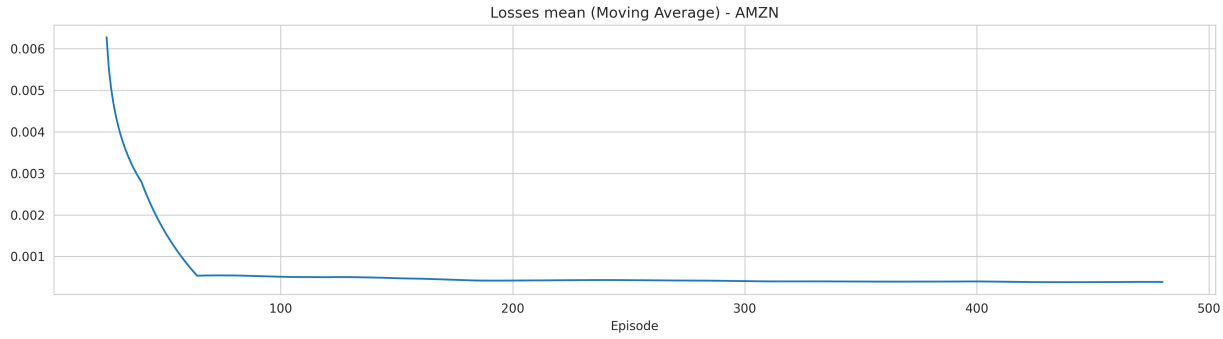


Figura B.83: Media móvil de promedio de pérdidas acumuladas

B.1.5.4. Pepsico Inc (PEP)

La figura B.84 muestra la media móvil sobre los últimos 25 episodios del promedio de pérdidas acumuladas para los 480 periodos de entrenamiento.

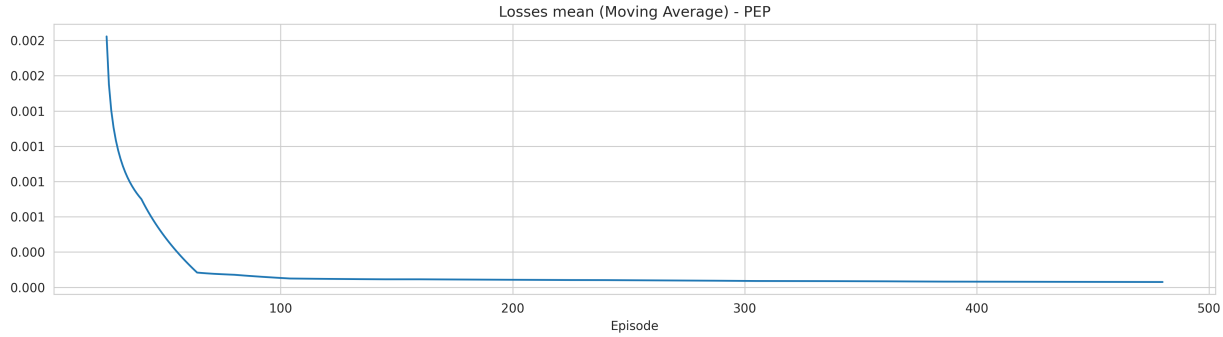


Figura B.84: Media móvil de promedio de pérdidas acumuladas

B.2. Experimento 2

En esta sección se presentarán los resultados para el experimento descrito en la sección 4.3.3.2.

B.2.1. Rendimientos

B.2.1.1. Apple Inc (AAPL)

La figura B.85 muestra la media móvil sobre los últimos 50 episodios de los valores de retorno acumulativo para los 480 periodos de entrenamiento, así como también la media móvil de la proporción de los últimos 50 episodios en los que el agente superó la estrategia *buy-and-hold*. La figura B.86 muestra estos mismos resultados sin una media móvil sobre los episodios. De manera similar, en las figuras B.87 y B.88 se muestra, bajo esta misma configuración, el promedio de los valores de retorno acumulativo obtenidos cada día durante un episodio.

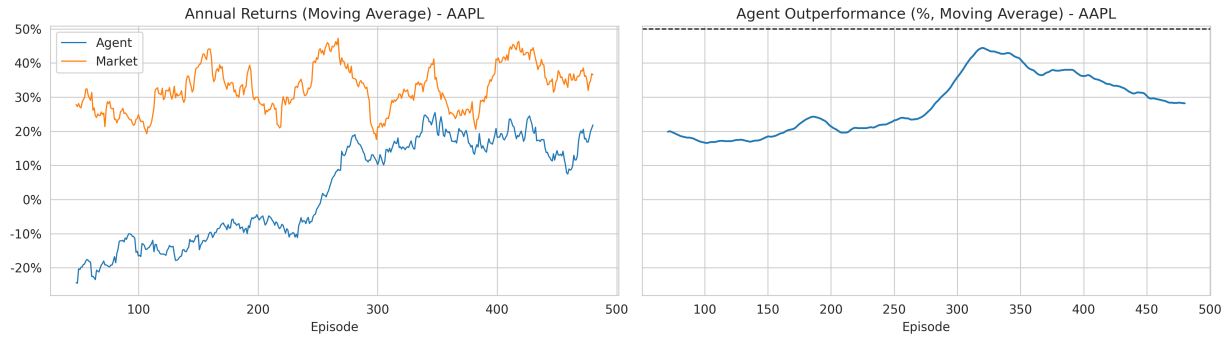


Figura B.85: Media móvil de rendimientos anuales

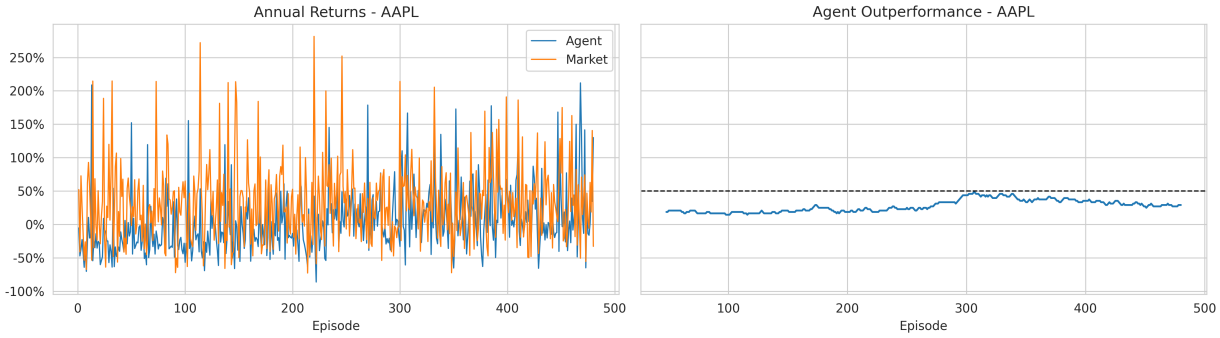


Figura B.86: Rendimientos anuales

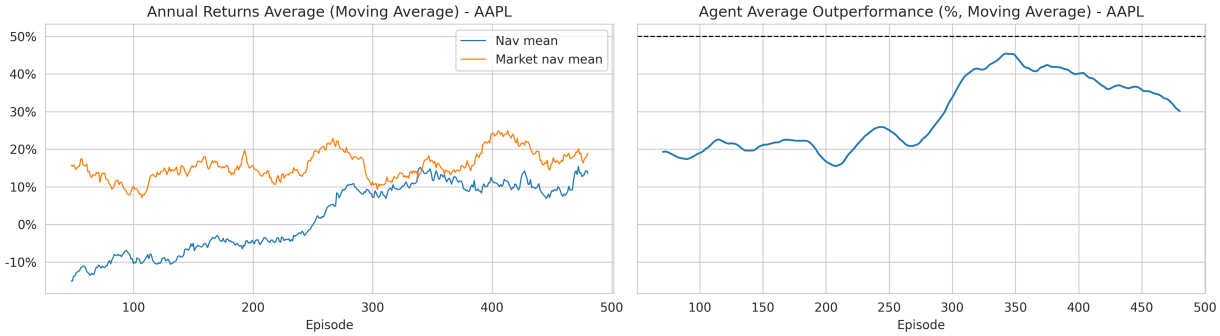


Figura B.87: Media móvil de rendimientos anuales promedio

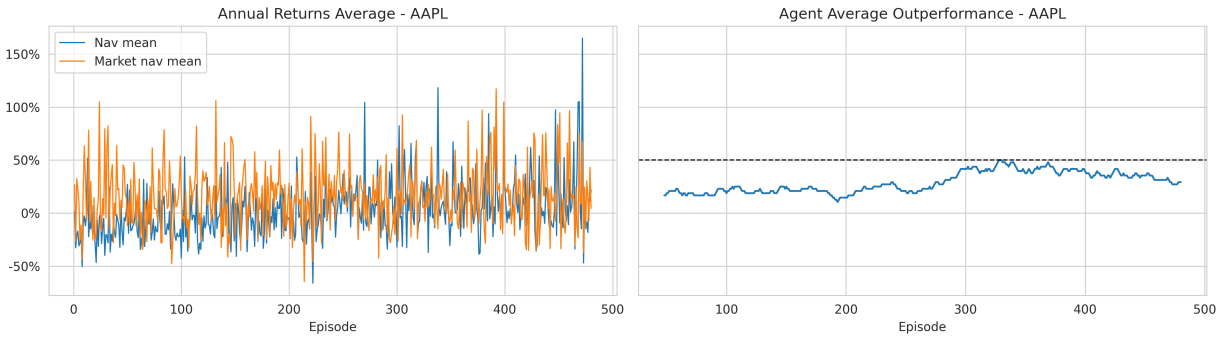


Figura B.88: Rendimientos anuales promedio

Finalmente, en la figura B.89 se muestra la media móvil sobre los últimos 50 episodios de la proporción de días en los que el agente superó la estrategia *buy-and-hold* durante un episodio. La figura B.90 muestra estos mismos resultados sin una media móvil sobre los episodios.

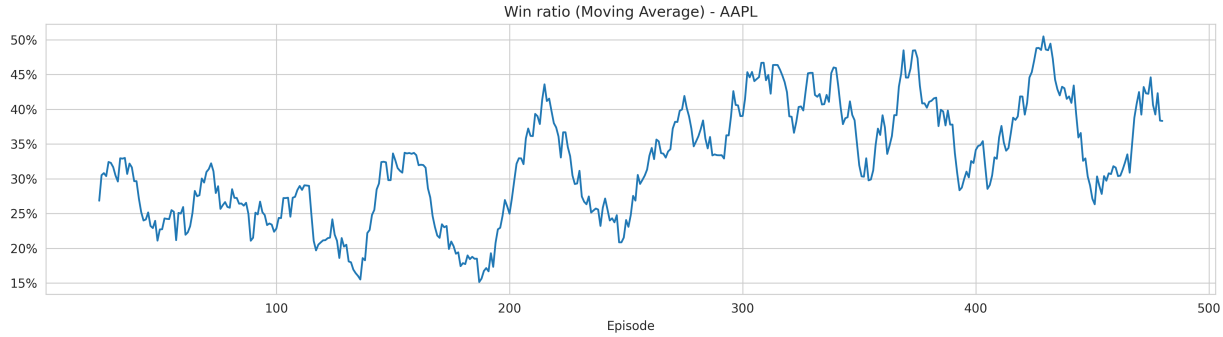


Figura B.89: Media móvil de proporción de ganancias

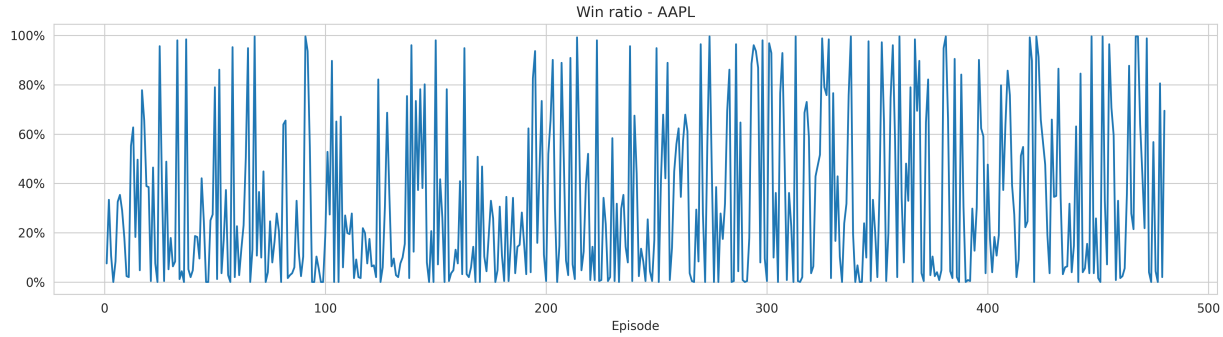


Figura B.90: Proporción de ganancias

B.2.1.2. Microsoft (MSFT)

La figura B.91 muestra la media móvil sobre los últimos 50 episodios de los valores de retorno acumulativo para los 480 periodos de entrenamiento, así como también la media móvil de la proporción de los últimos 50 episodios en los que el agente superó la estrategia *buy-and-hold*. La figura B.92 muestra estos mismos resultados sin una media móvil sobre los episodios. De manera similar, en las figuras B.93 y B.94 se muestra, bajo esta misma configuración, el promedio de los valores de retorno acumulativo obtenidos cada día durante un episodio.

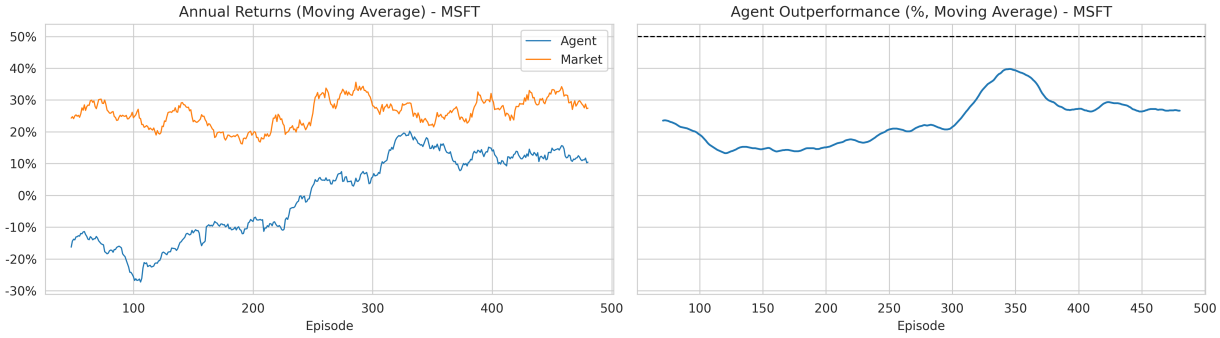


Figura B.91: Media móvil de rendimientos anuales

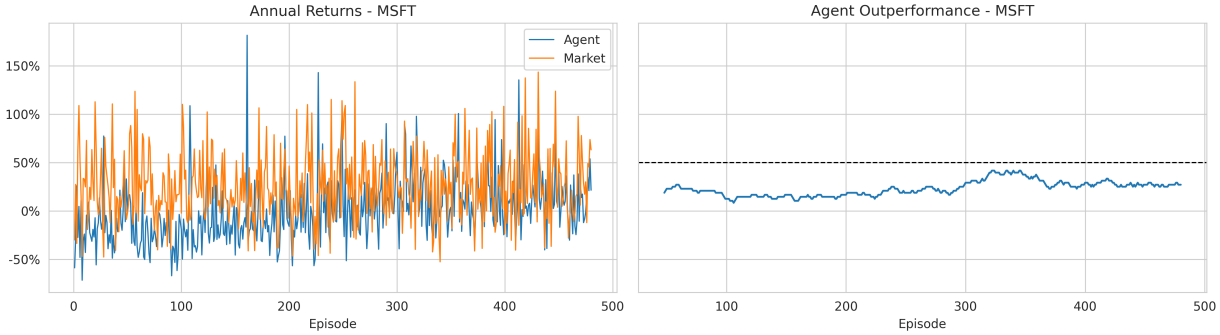


Figura B.92: Rendimientos anuales

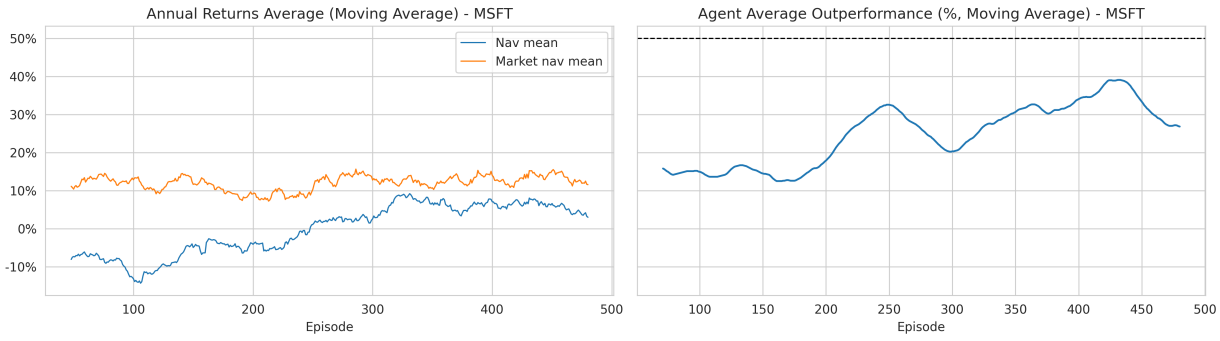


Figura B.93: Media móvil de rendimientos anuales promedio

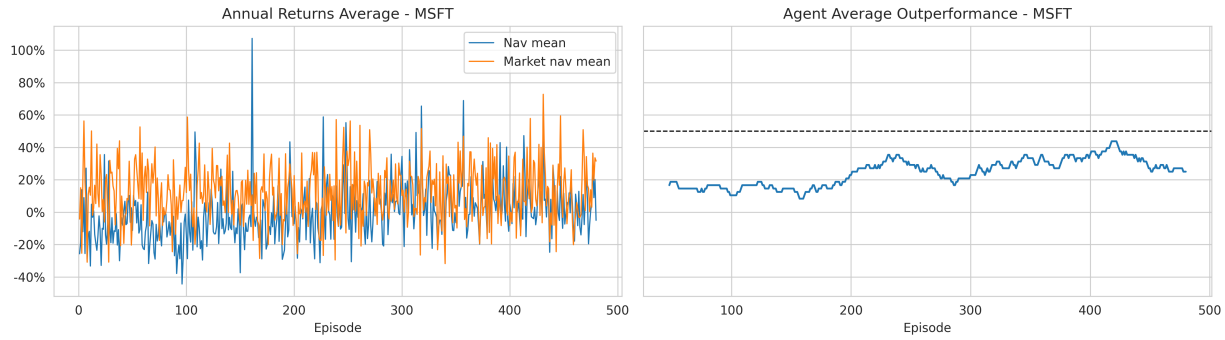


Figura B.94: Rendimientos anuales promedio

Finalmente, en la figura B.95 se muestra la media móvil sobre los últimos 50 episodios de la proporción de días en los que el agente superó la estrategia *buy-and-hold* durante un episodio. La figura B.96 muestra estos mismos resultados sin una media móvil sobre los episodios.

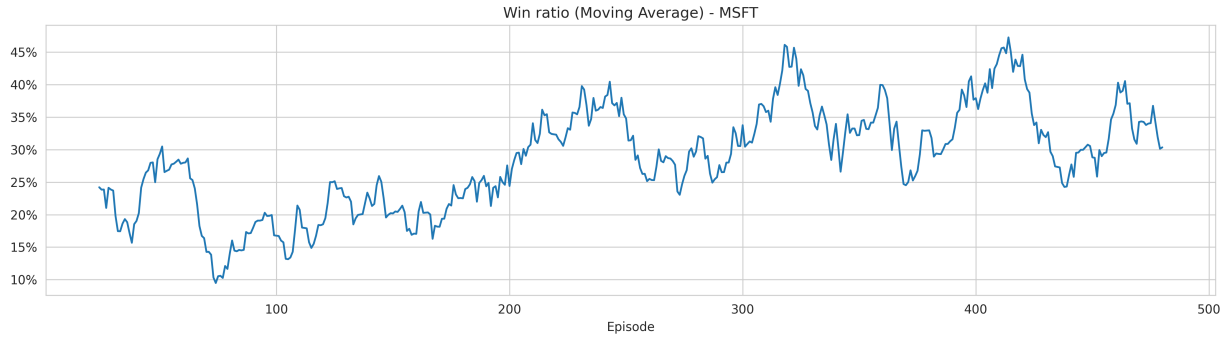


Figura B.95: Media móvil de proporción de ganancias

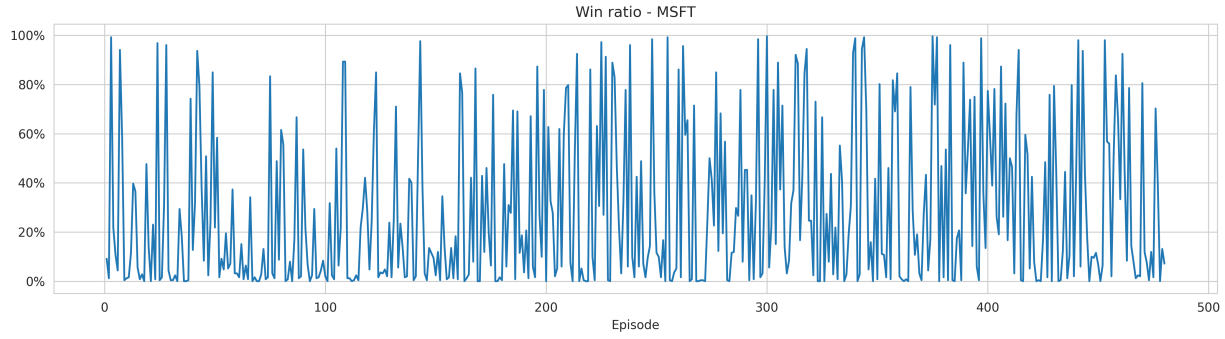


Figura B.96: Proporción de ganancias

B.2.1.3. Amazon Inc (AMZN)

La figura B.97 muestra la media móvil sobre los últimos 50 episodios de los valores de retorno acumulativo para los 480 periodos de entrenamiento, así como también la media móvil de la proporción de los últimos 50 episodios en los que el agente superó la estrategia *buy-and-hold*. La figura B.98 muestra estos mismos resultados sin una media móvil sobre los episodios. De manera similar, en las figuras B.99 y B.100 se muestra, bajo esta misma configuración, el promedio de los valores de retorno acumulativo obtenidos cada día durante un episodio.

Figura B.97: Media móvil de rendimientos anuales

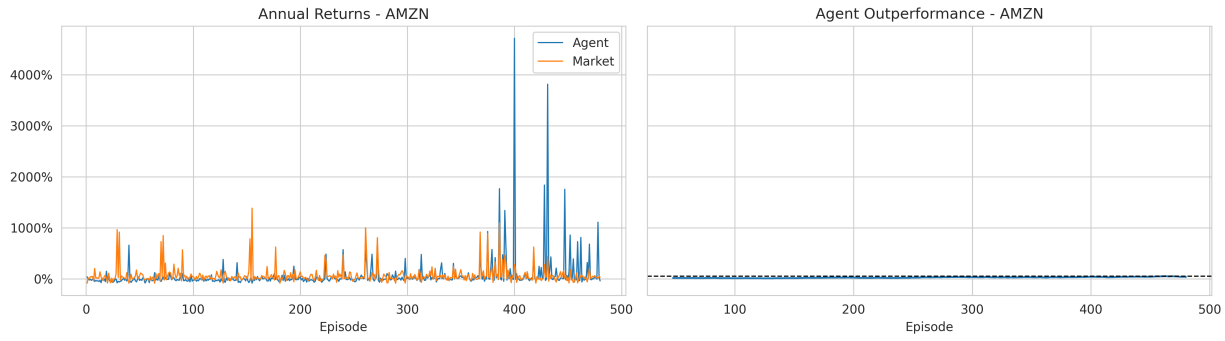


Figura B.98: Rendimientos anuales

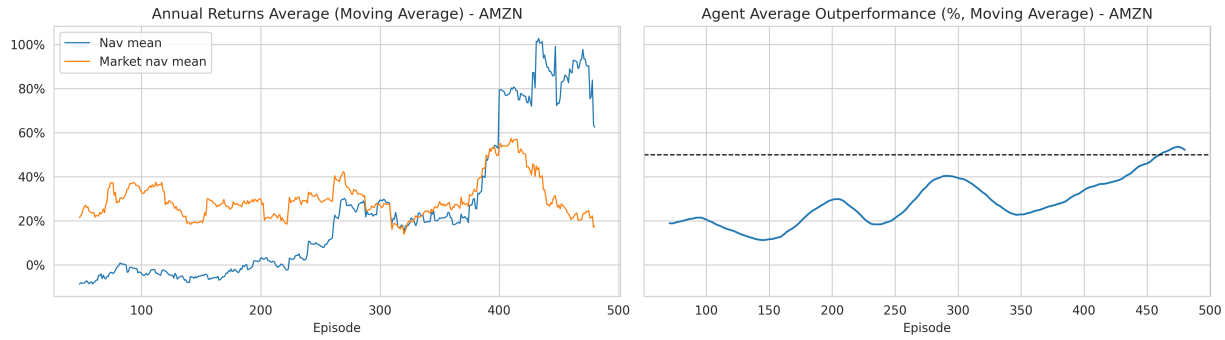


Figura B.99: Media móvil de rendimientos anuales promedio

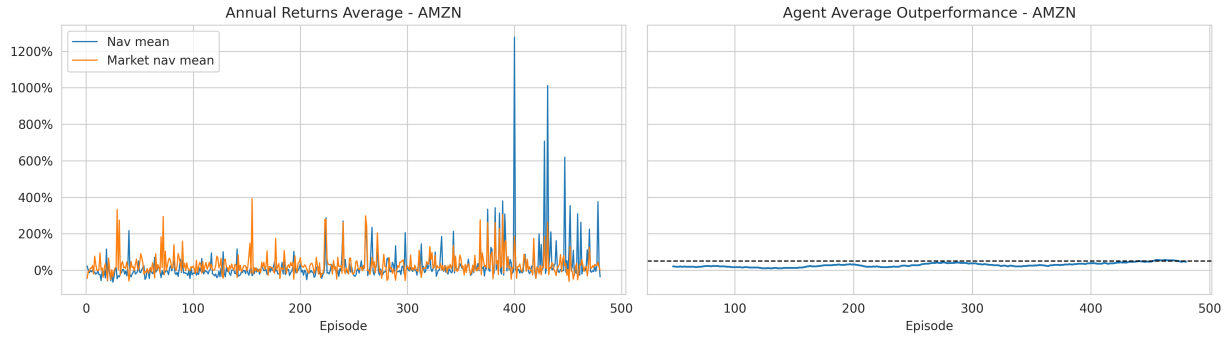


Figura B.100: Rendimientos anuales promedio

Finalmente, en la figura B.101 se muestra la media móvil sobre los últimos 50 episodios de la proporción de días en los que el agente superó la estrategia *buy-and-hold* durante un episodio. La figura B.102 muestra estos mismos resultados sin una media móvil sobre los episodios.

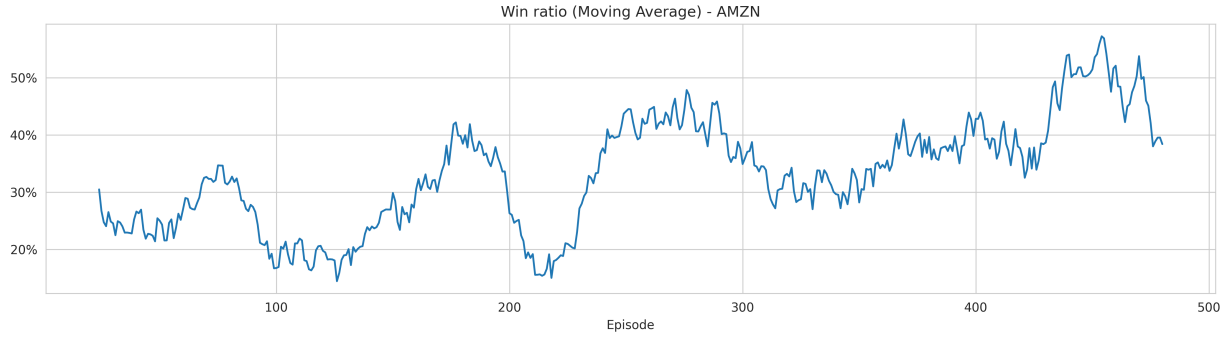


Figura B.101: Media móvil de proporción de ganancias

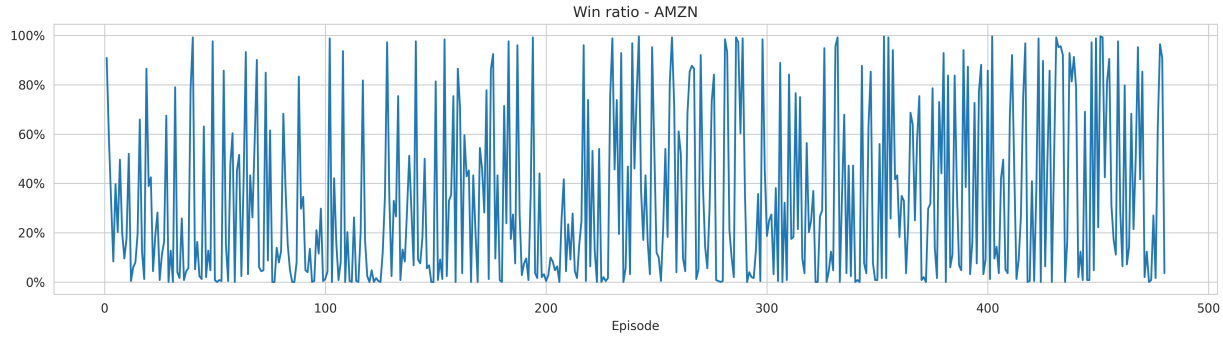


Figura B.102: Proporción de ganancias

B.2.1.4. Pepsico Inc (PEP)

La figura B.103 muestra la media móvil sobre los últimos 50 episodios de los valores de retorno acumulativo para los 480 periodos de entrenamiento, así como también la media móvil de la proporción de los últimos 50 episodios en los que el agente superó la estrategia *buy-and-hold*. La figura B.104 muestra estos mismos resultados sin una media móvil sobre los episodios. De manera similar, en las figuras B.105 y B.106 se muestra, bajo esta misma configuración, el promedio de los valores de retorno acumulativo obtenidos cada día durante un episodio.

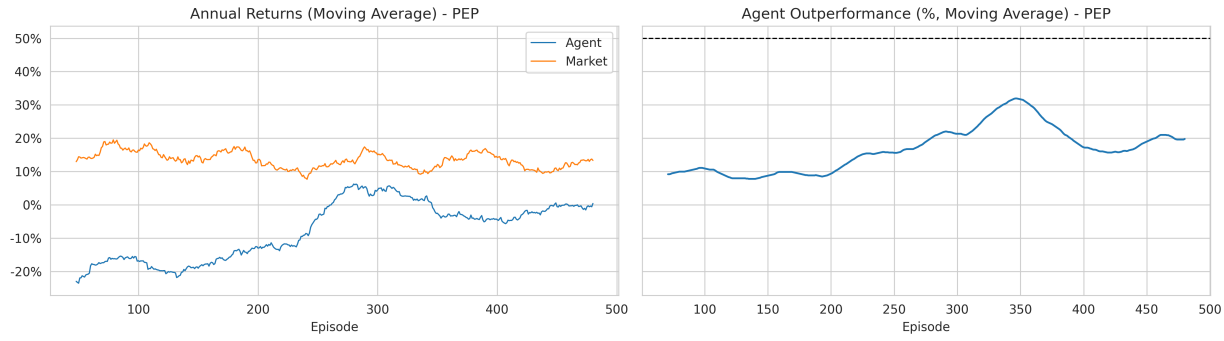


Figura B.103: Media móvil de rendimientos anuales

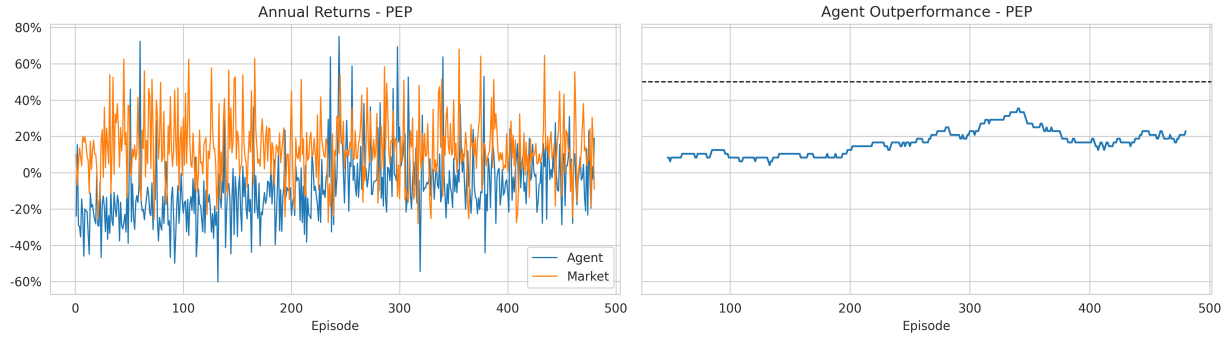


Figura B.104: Rendimientos anuales

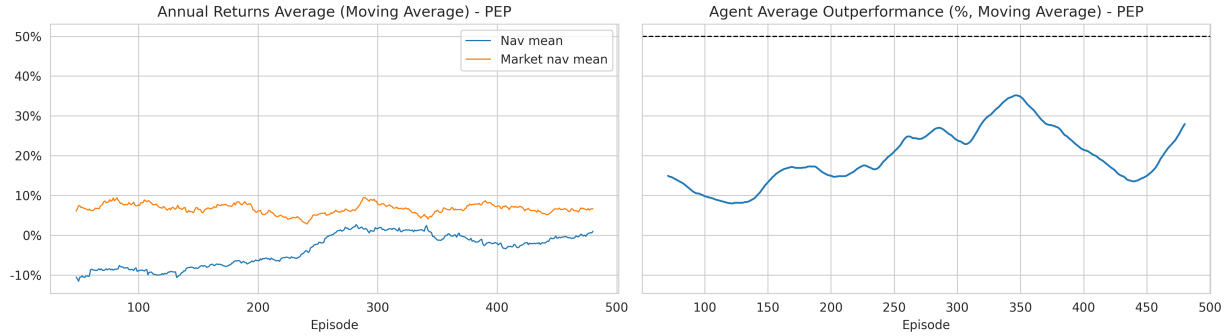


Figura B.105: Media móvil de rendimientos anuales promedio

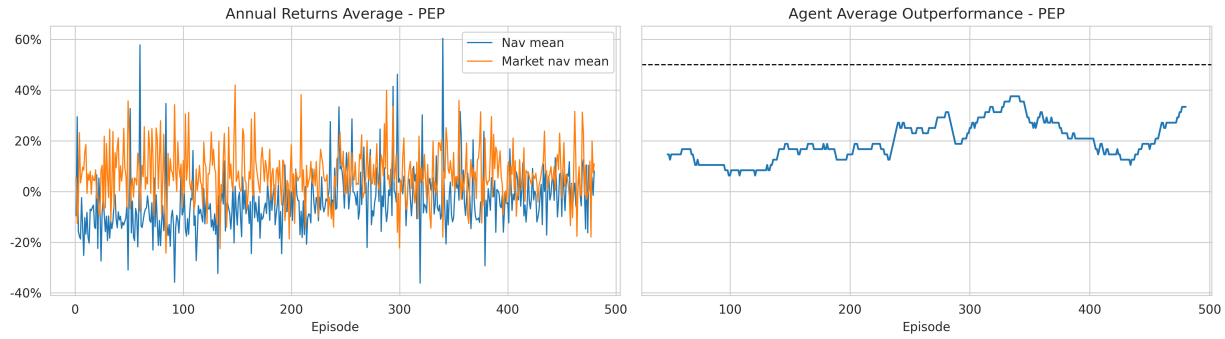


Figura B.106: Rendimientos anuales promedio

Finalmente, en la figura B.107 se muestra la media móvil sobre los últimos 50 episodios de la proporción de días en los que el agente superó la estrategia *buy-and-hold* durante un episodio. La figura B.108 muestra estos mismos resultados sin una media móvil sobre los episodios.

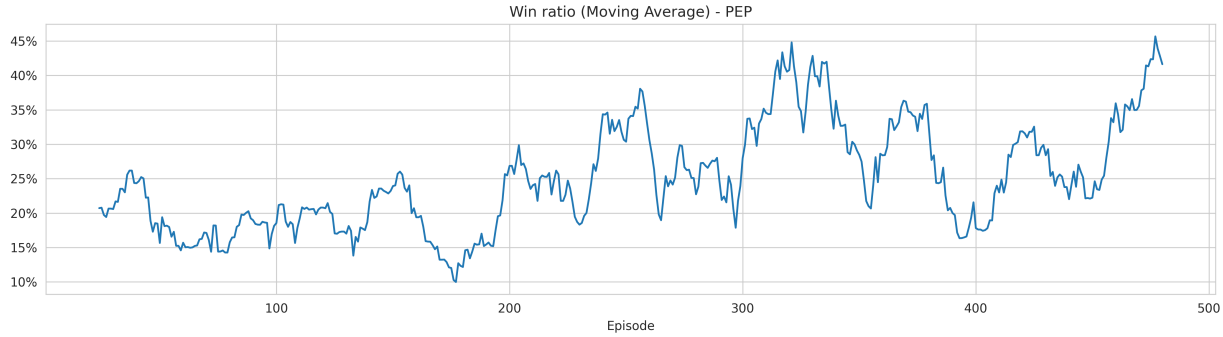


Figura B.107: Media móvil de proporción de ganancias

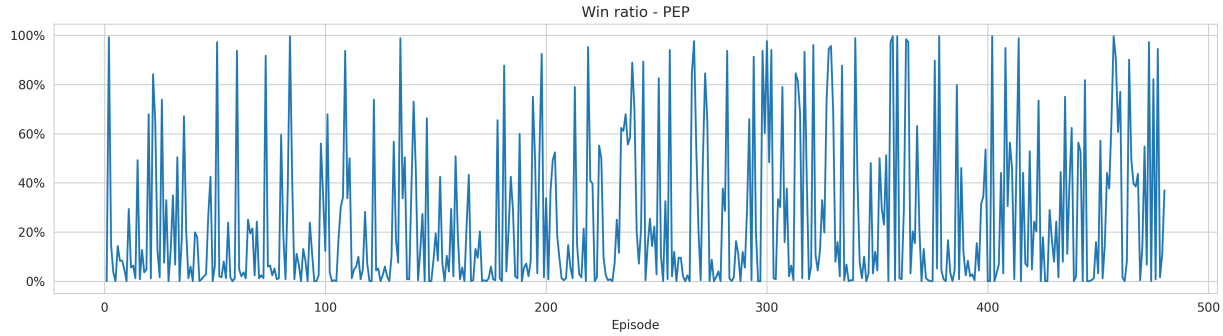


Figura B.108: Proporción de ganancias

B.2.2. Recompensas

B.2.2.1. Apple Inc (AAPL)

La figura B.109 muestra la media móvil sobre los últimos 25 episodios de la suma acumulada de recompensas obtenidas durante un episodio para los 480 periodos de entrenamiento. La figura B.110 muestra estos mismos resultados sin una media móvil sobre los episodios. De manera similar, en las figuras B.111 y B.112 se muestra, bajo esta misma configuración, el promedio de recompensas obtenidas durante un episodio.

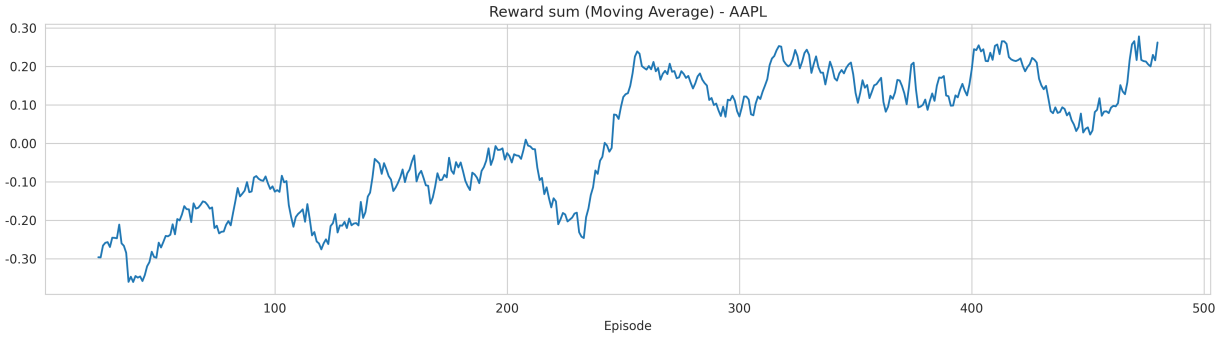


Figura B.109: Media móvil de suma acumulada de recompensas

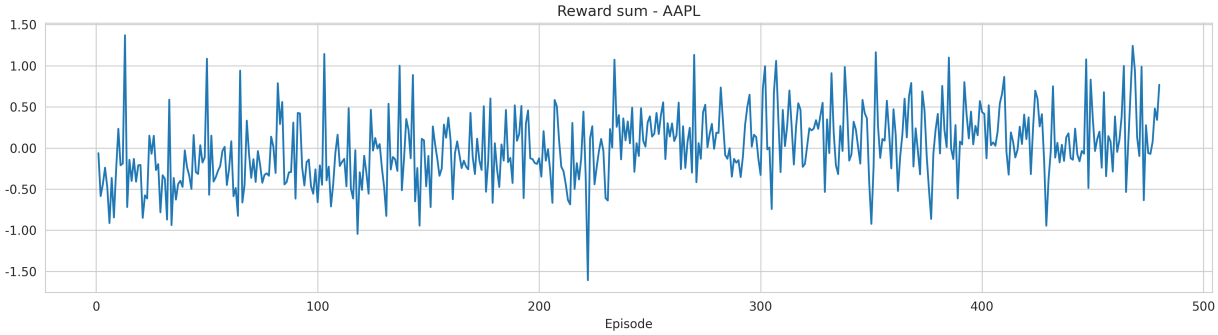


Figura B.110: Suma acumulada de recompensas

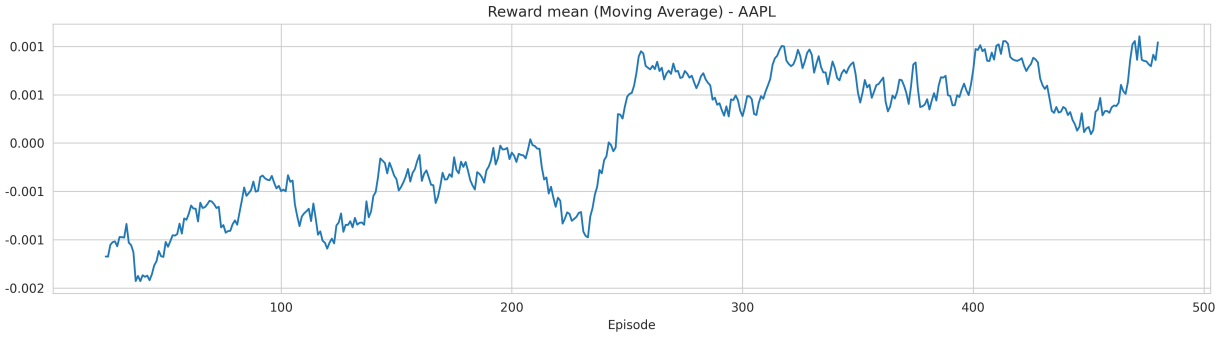


Figura B.111: Media móvil de promedio de recompensas

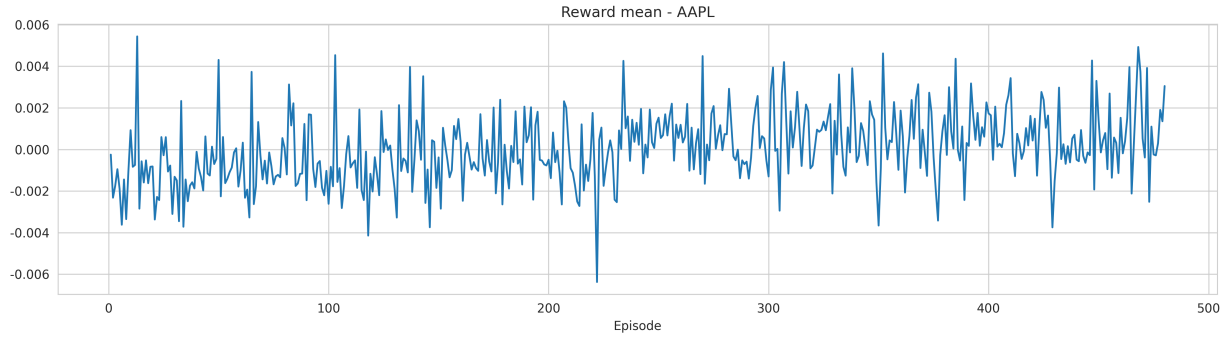


Figura B.112: Promedio de recompensas

Finalmente, en la figura B.113 se muestra la media móvil sobre los últimos 25 episodios de la diferencia entre la mayor y menor recompensa obtenidas durante un episodio. La figura B.114 muestra estos mismos resultados sin una media móvil sobre los episodios.

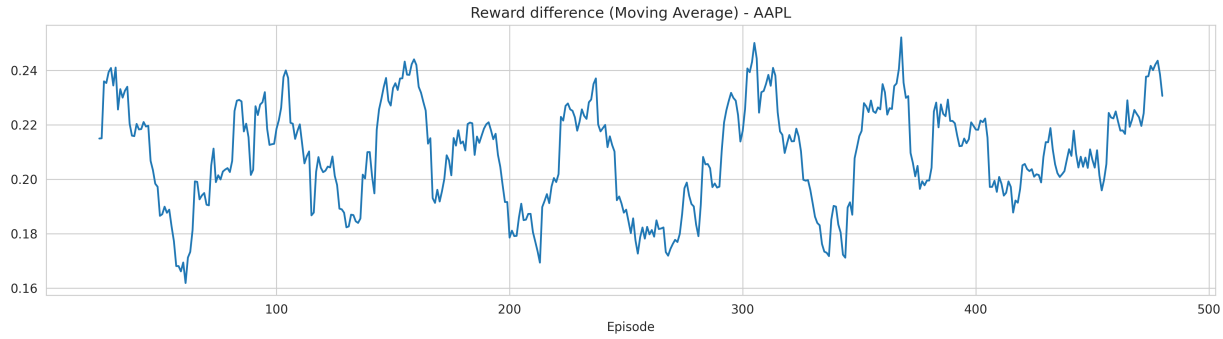


Figura B.113: Media móvil de diferencia de recompensa

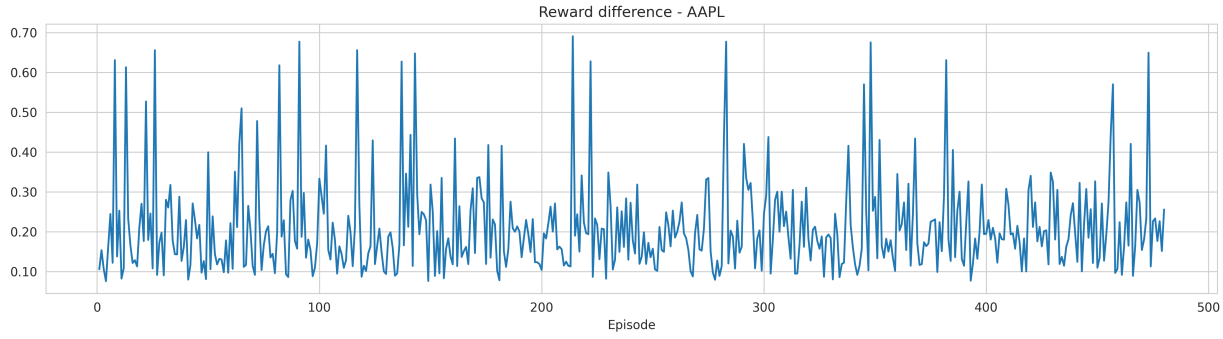


Figura B.114: Diferencia de recompensa

B.2.2.2. Microsoft (MSFT)

La figura B.115 muestra la media móvil sobre los últimos 25 episodios de la suma acumulada de recompensas obtenidas durante un episodio para los 480 periodos de entrenamiento. La figura B.116 muestra estos mismos resultados sin una media móvil sobre los episodios. De manera similar, en las figuras B.117 y B.118 se muestra, bajo esta misma configuración, el promedio de recompensas obtenidas durante un episodio.

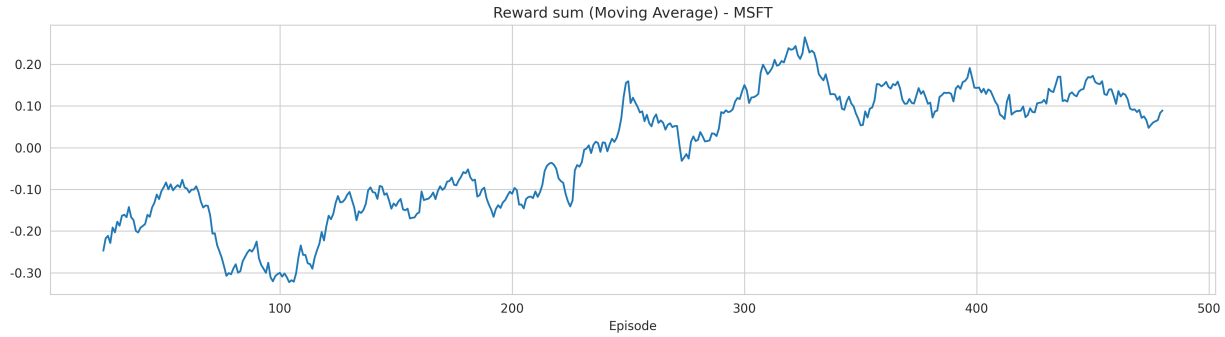


Figura B.115: Media móvil de suma acumulada de recompensas

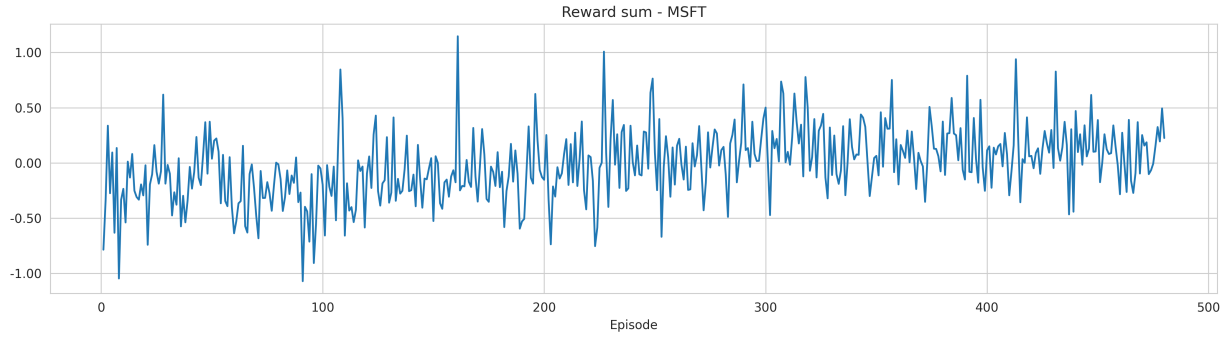


Figura B.116: Suma acumulada de recompensas

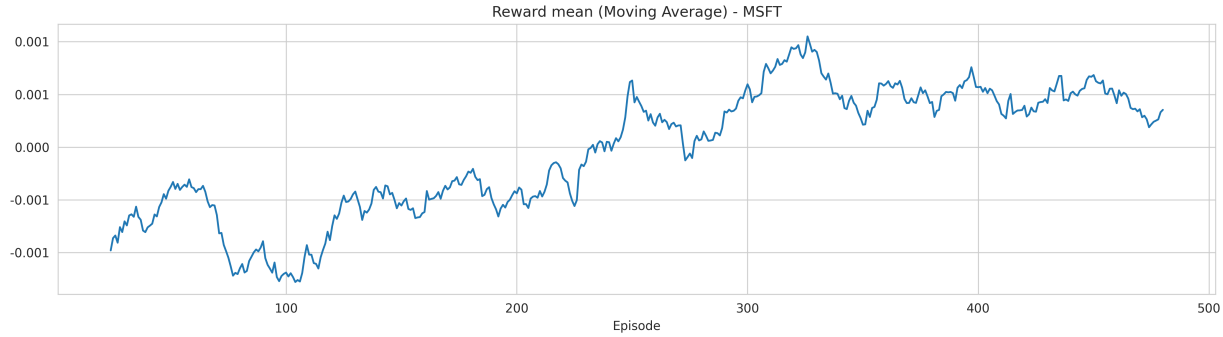


Figura B.117: Media móvil de promedio de recompensas

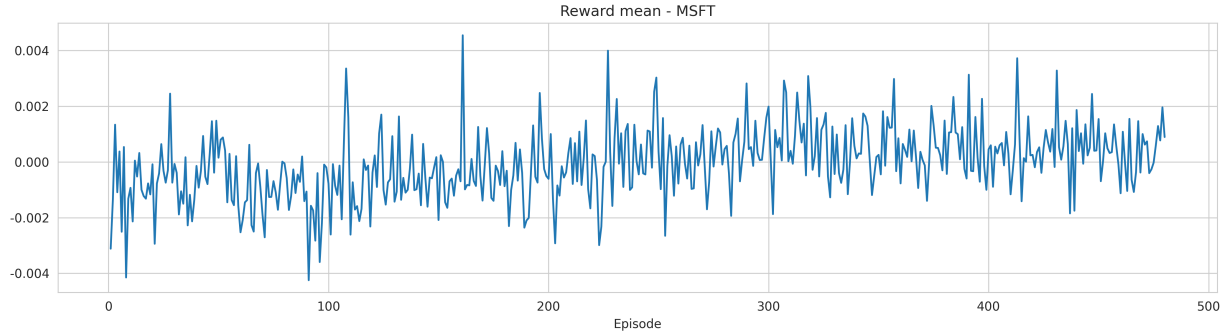


Figura B.118: Promedio de recompensas

Finalmente, en la figura B.119 se muestra la media móvil sobre los últimos 25 episodios de la diferencia entre la mayor y menor recompensa obtenidas durante un episodio. La figura B.120 muestra estos mismos resultados sin una media móvil sobre los episodios.

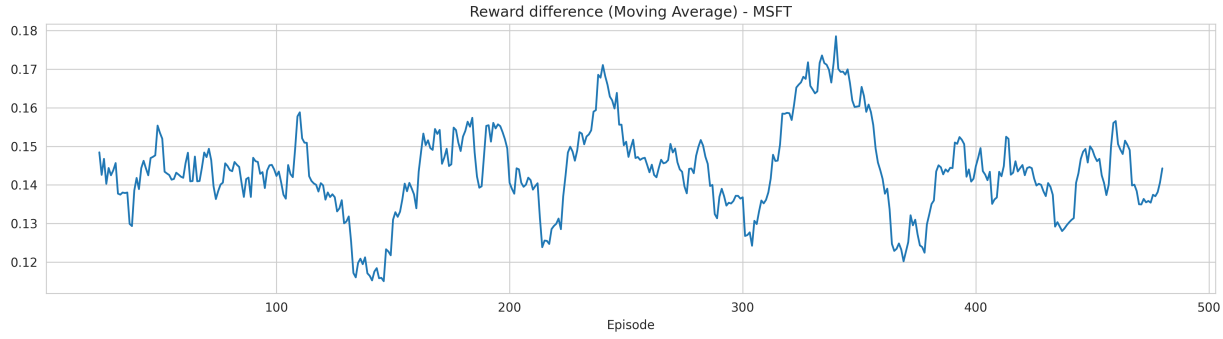


Figura B.119: Media móvil de diferencia de recompensa

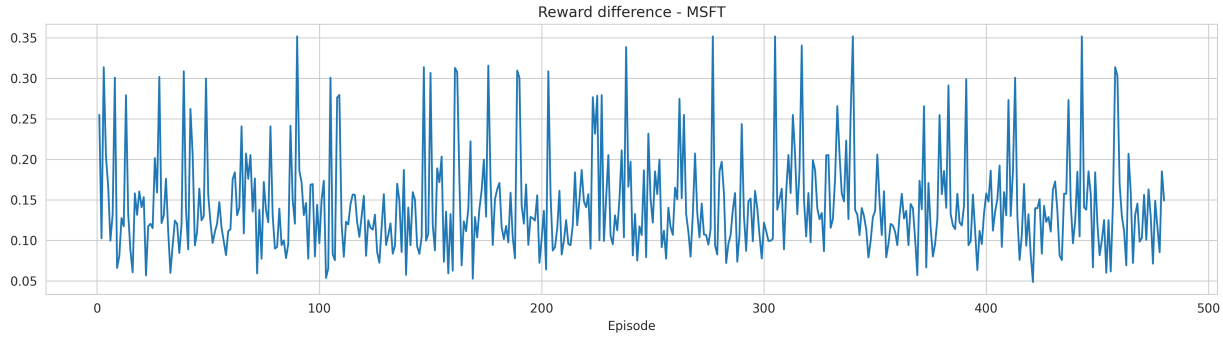


Figura B.120: Diferencia de recompensa

B.2.2.3. Amazon Inc (AMZN)

La figura B.121 muestra la media móvil sobre los últimos 25 episodios de la suma acumulada de recompensas obtenidas durante un episodio para los 480 periodos de entrenamiento. La figura B.122 muestra estos mismos resultados sin una media móvil sobre los episodios. De manera similar, en las figuras B.123 y B.124 se muestra, bajo esta misma configuración, el promedio de recompensas obtenidas durante un episodio.

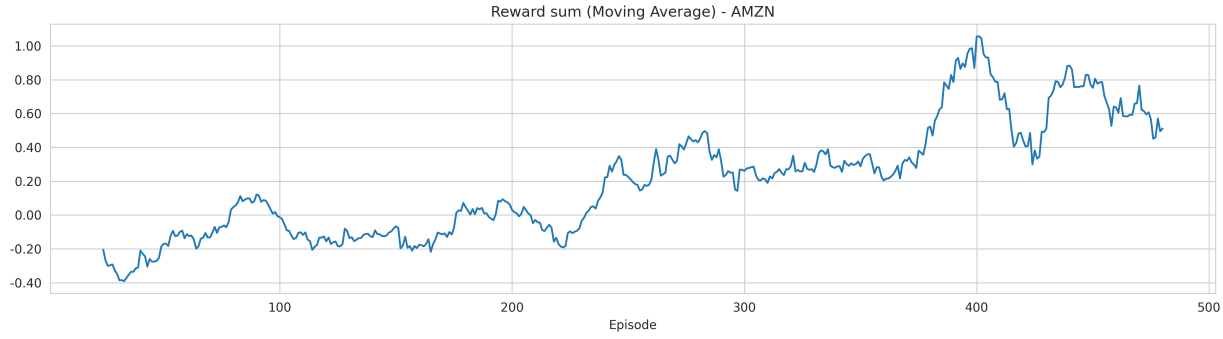


Figura B.121: Media móvil de suma acumulada de recompensas

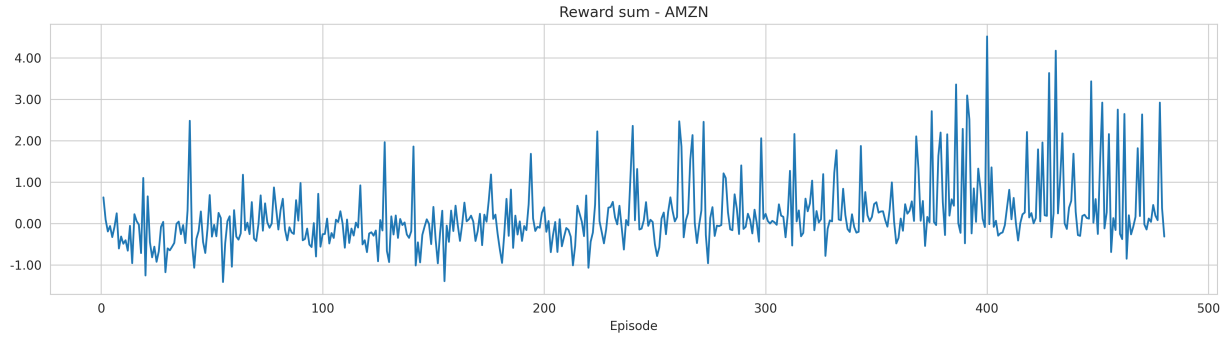


Figura B.122: Suma acumulada de recompensas

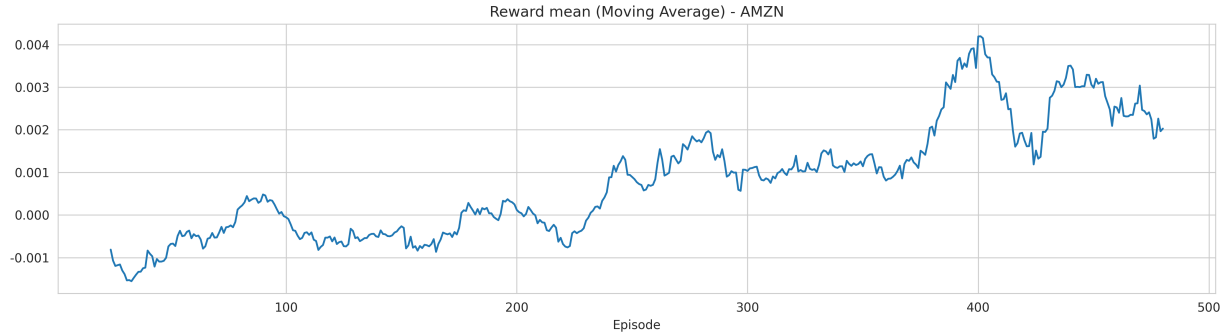


Figura B.123: Media móvil de promedio de recompensas

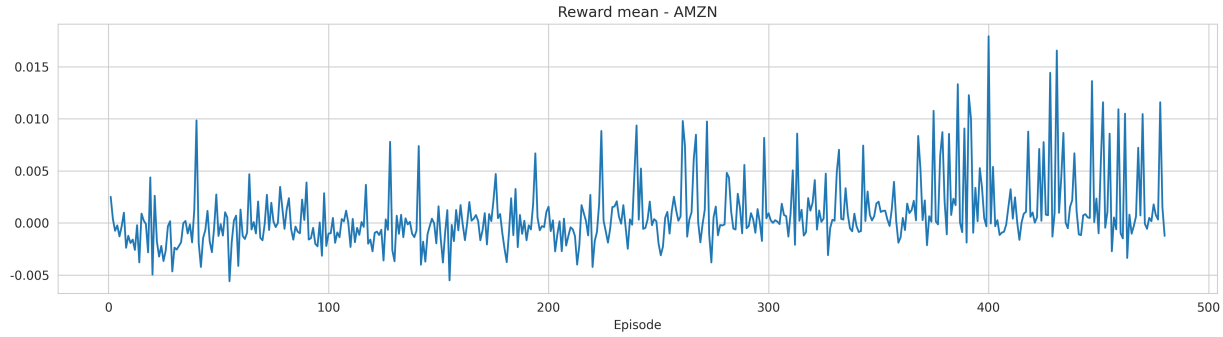


Figura B.124: Promedio de recompensas

Finalmente, en la figura B.125 se muestra la media móvil sobre los últimos 25 episodios de la diferencia entre la mayor y menor recompensa obtenidas durante un episodio. La figura B.126 muestra estos mismos resultados sin una media móvil sobre los episodios.

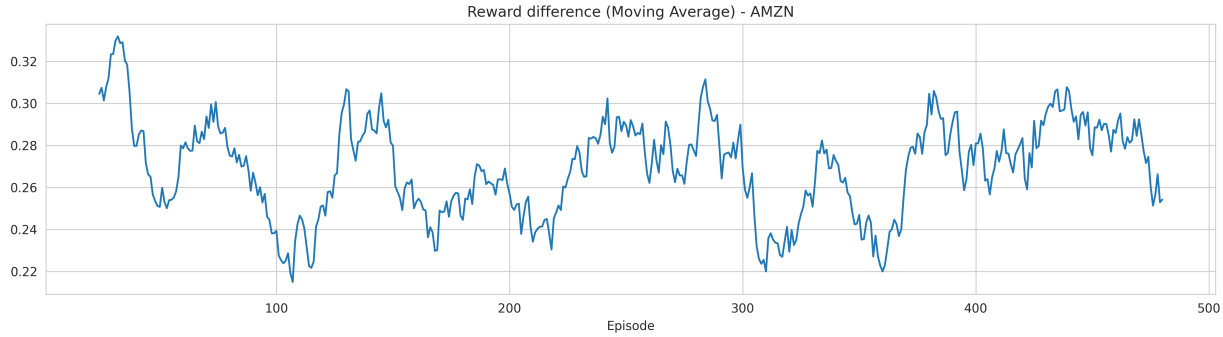


Figura B.125: Media móvil de diferencia de recompensa

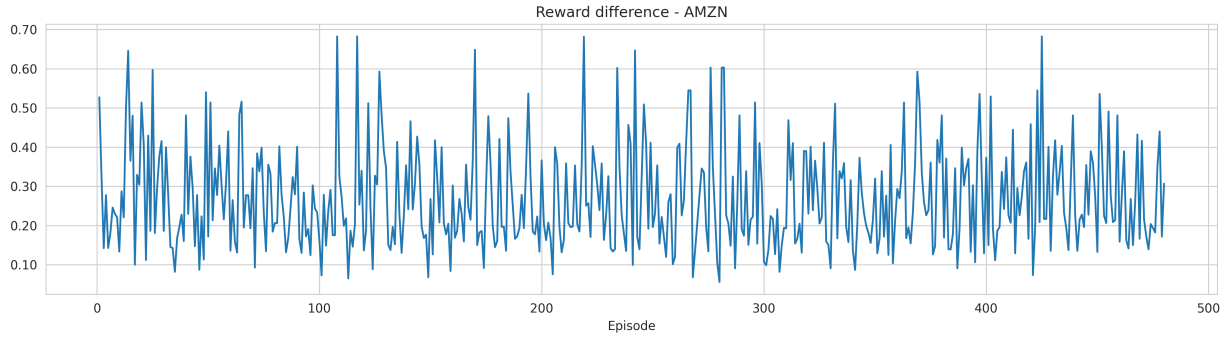


Figura B.126: Diferencia de recompensa

B.2.2.4. Pepsico Inc (PEP)

La figura B.127 muestra la media móvil sobre los últimos 25 episodios de la suma acumulada de recompensas obtenidas durante un episodio para los 480 periodos de entrenamiento. La figura B.128 muestra estos mismos resultados sin una media móvil sobre los episodios. De manera similar, en las figuras B.129 y B.130 se muestra, bajo esta misma configuración, el promedio de recompensas obtenidas durante un episodio.

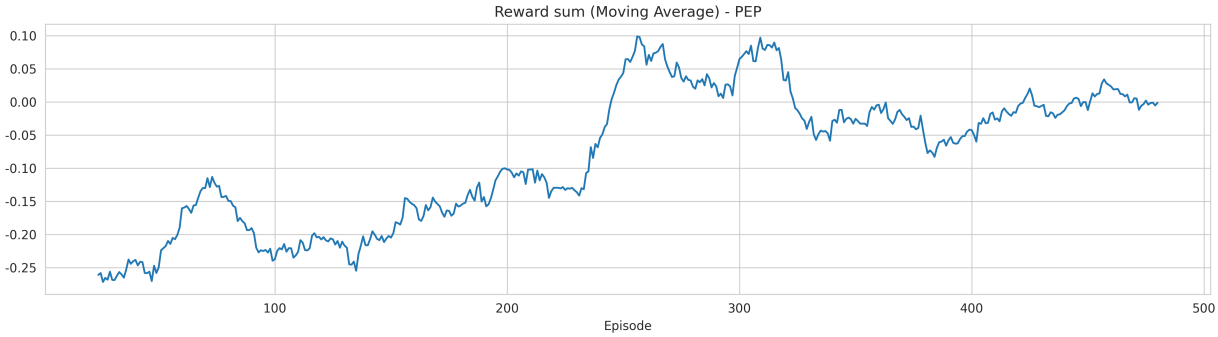


Figura B.127: Media móvil de suma acumulada de recompensas

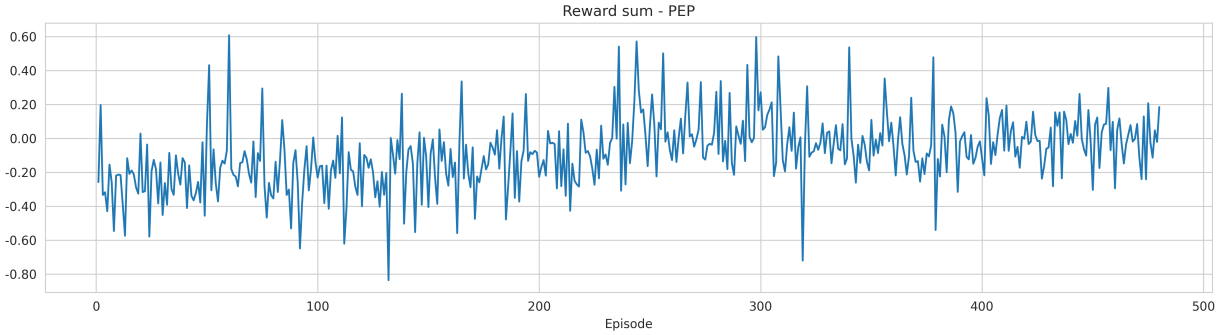


Figura B.128: Suma acumulada de recompensas

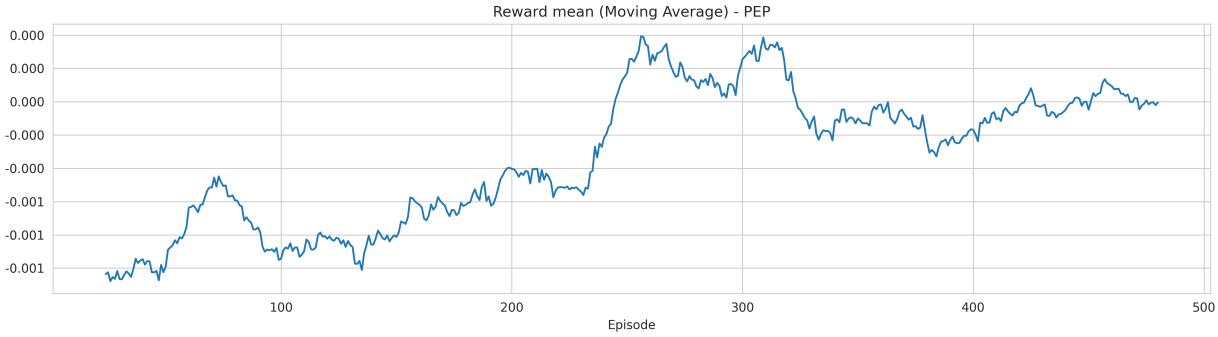


Figura B.129: Media móvil de promedio de recompensas

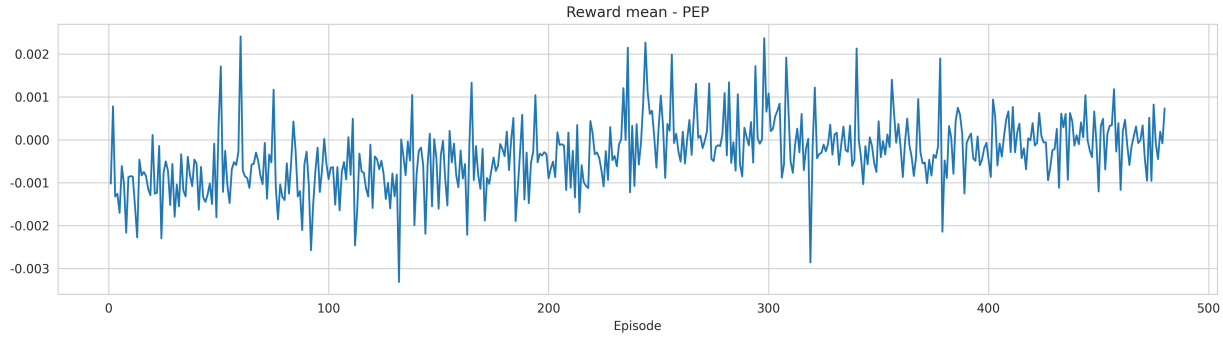


Figura B.130: Promedio de recompensas

Finalmente, en la figura B.131 se muestra la media móvil sobre los últimos 25 episodios de la diferencia entre la mayor y menor recompensa obtenidas durante un episodio. La figura B.132 muestra estos mismos resultados sin una media móvil sobre los episodios.

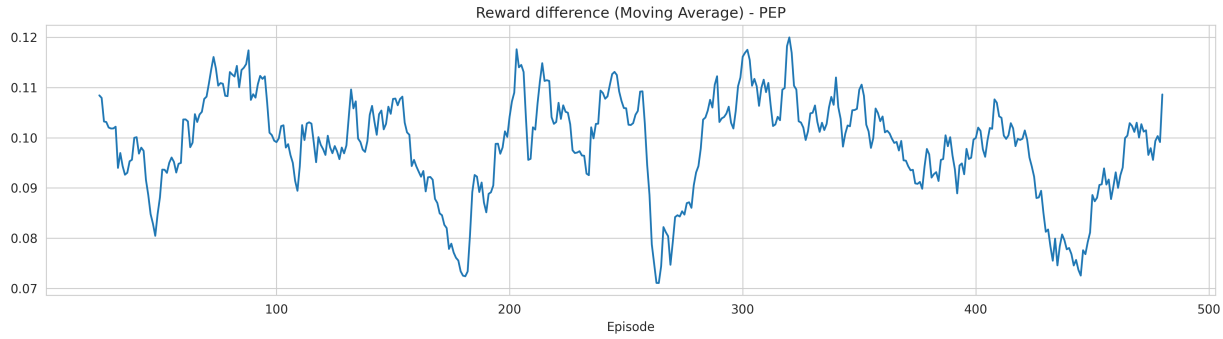


Figura B.131: Media móvil de diferencia de recompensa

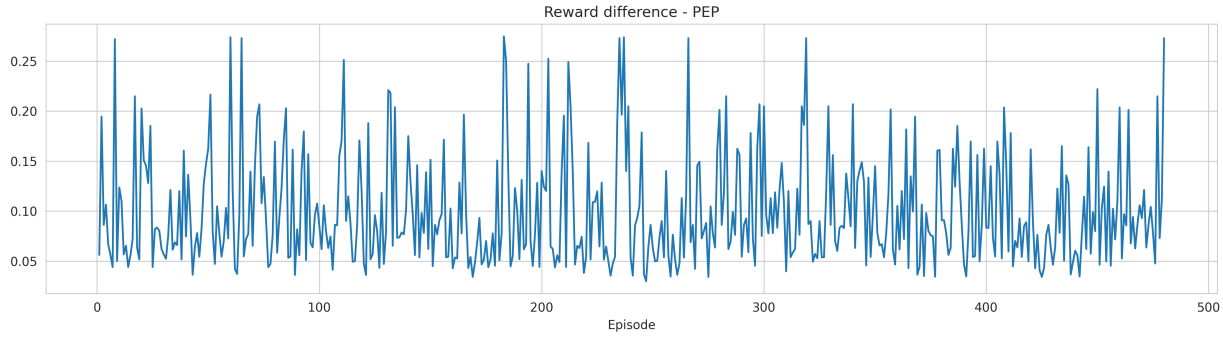


Figura B.132: Diferencia de recompensa

B.2.3. Costos

B.2.3.1. Apple Inc (AAPL)

La figura B.133 muestra la media móvil sobre los último 25 episodios de la suma acumulada de costos obtenidos durante un episodio para los 480 periodos de entrenamiento. La figura B.134 muestra estos mismos resultados sin una media móvil sobre los episodios. De manera similar, en las figuras B.135 y B.136 se muestra, bajo esta misma configuración, el promedio de costos obtenidos durante un episodio.

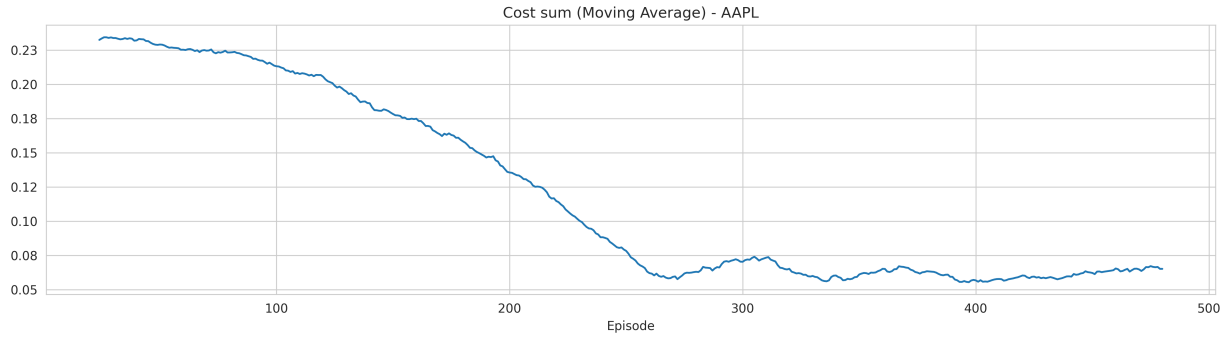


Figura B.133: Media móvil de suma acumulada de costos

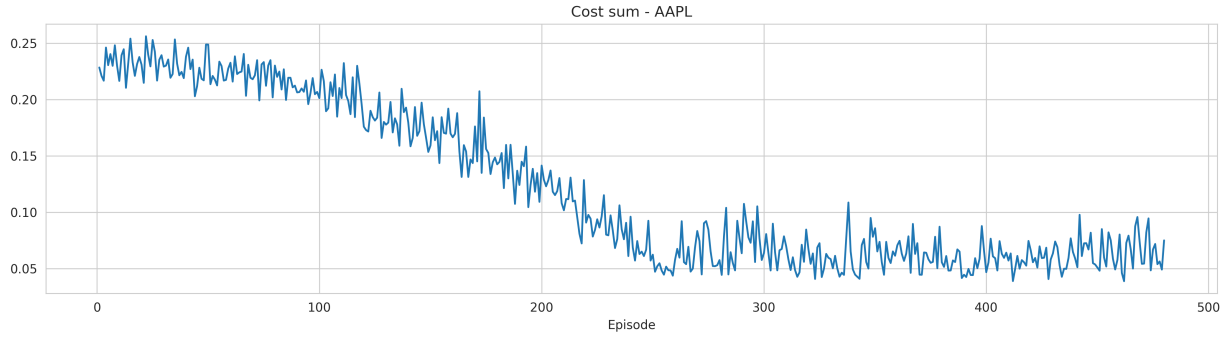


Figura B.134: Suma acumulada de costos

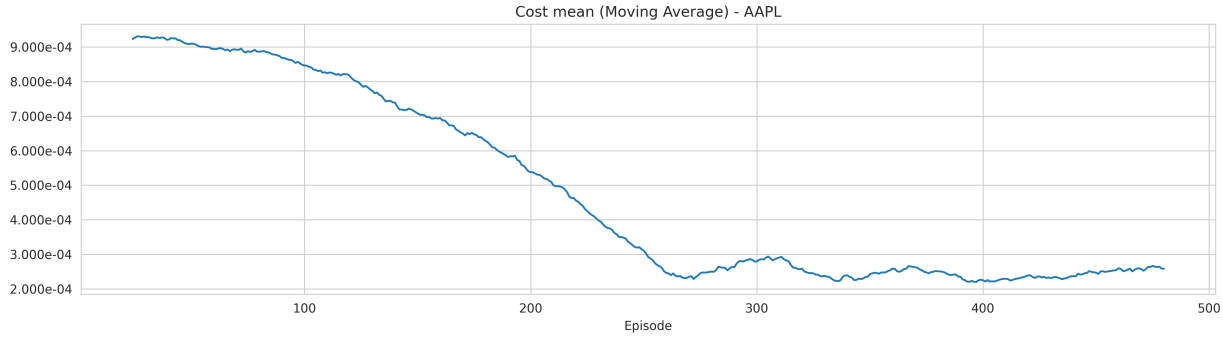


Figura B.135: Media móvil de promedio de costos

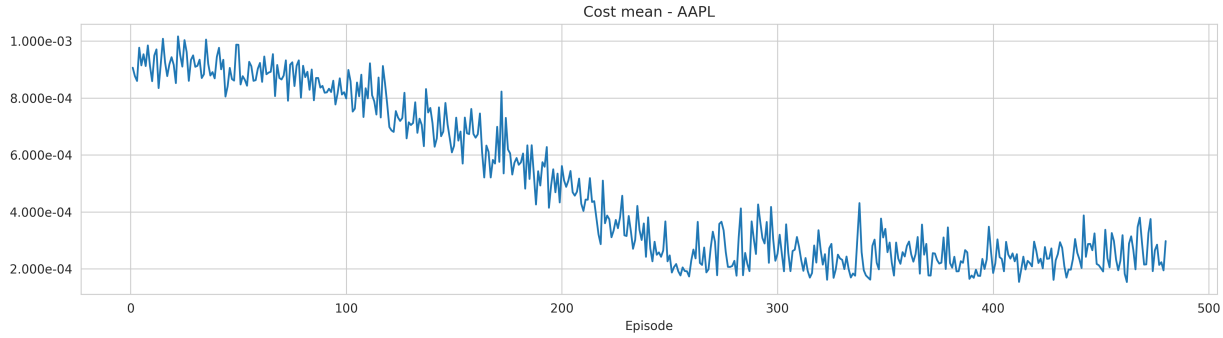


Figura B.136: Promedio de costos

B.2.3.2. Microsoft (MSFT)

La figura B.137 muestra la media móvil sobre los último 25 episodios de la suma acumulada de costos obtenidos durante un episodio para los 480 periodos de entrenamiento. La figura B.138 muestra estos mismos resultados sin una media móvil sobre los episodios. De manera similar, en las figuras B.139 y B.140 se muestra, bajo esta misma configuración, el promedio de costos obtenidos durante un episodio.

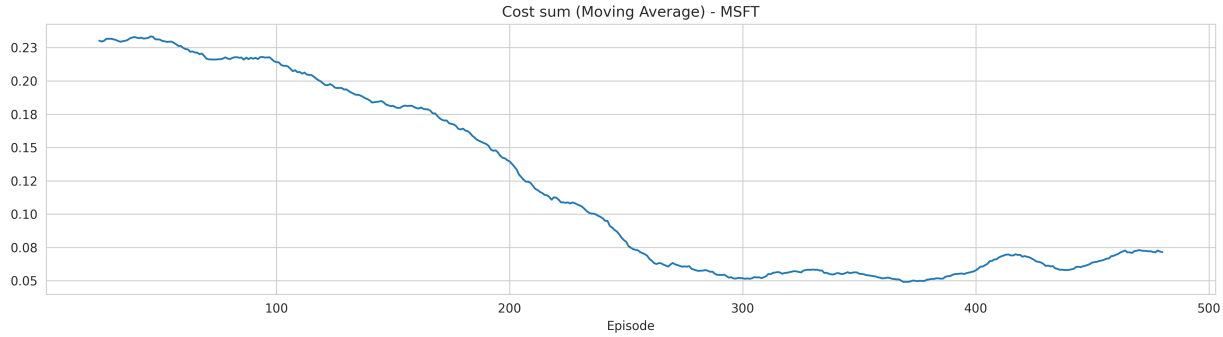


Figura B.137: Media móvil de suma acumulada de costos

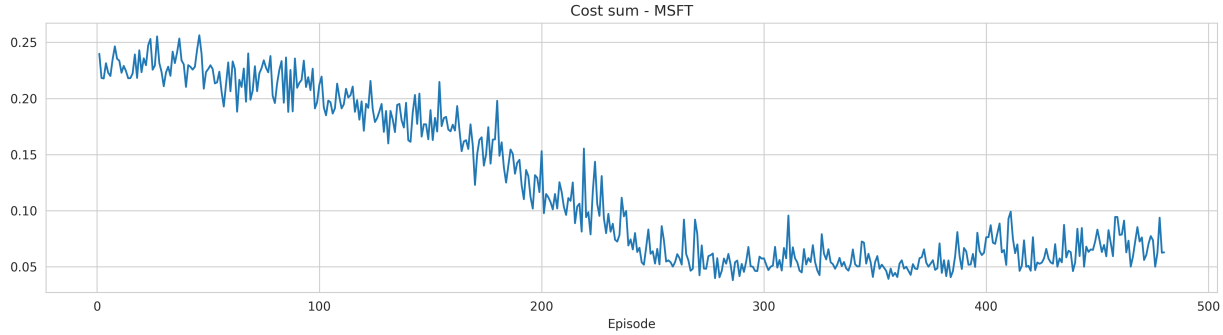


Figura B.138: Suma acumulada de costos

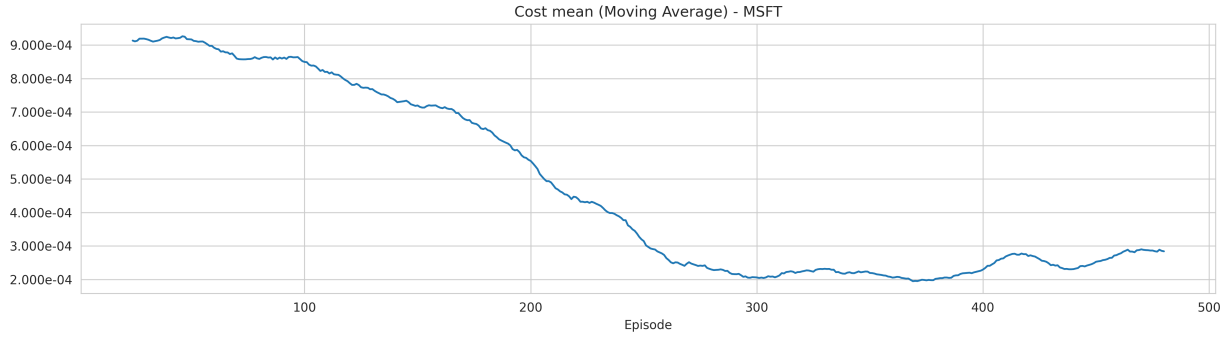


Figura B.139: Media móvil de promedio de costos

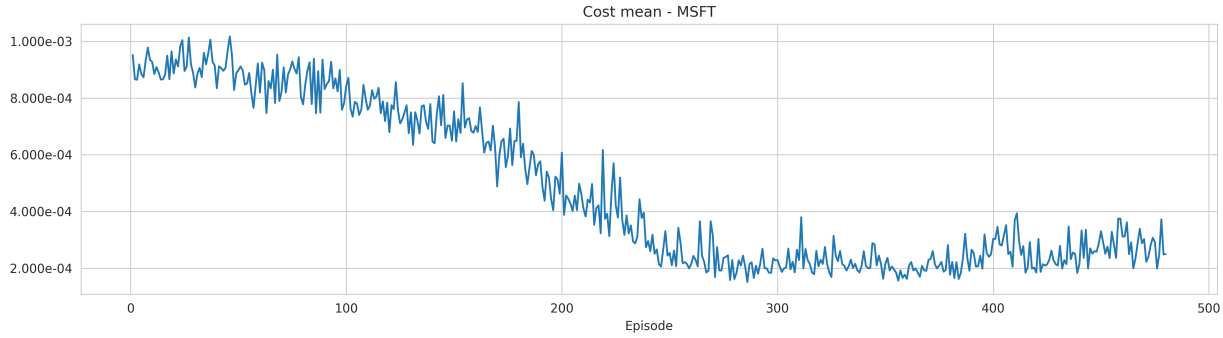


Figura B.140: Promedio de costos

B.2.3.3. Amazon Inc (AMZN)

La figura B.141 muestra la media móvil sobre los último 25 episodios de la suma acumulada de costos obtenidos durante un episodio para los 480 periodos de entrenamiento. La figura B.142 muestra estos mismos resultados sin una media móvil sobre los episodios. De manera similar, en las figuras B.143 y B.144 se muestra, bajo esta misma configuración, el promedio de costos obtenidos durante un episodio.

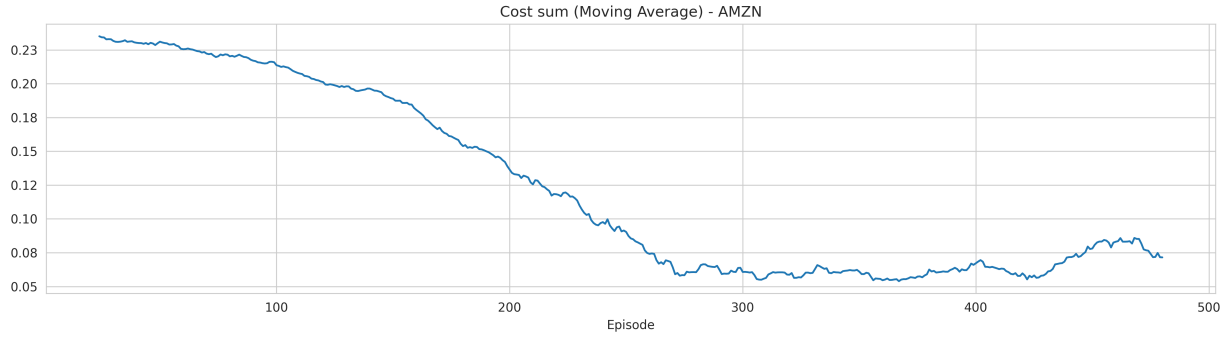


Figura B.141: Media móvil de suma acumulada de costos

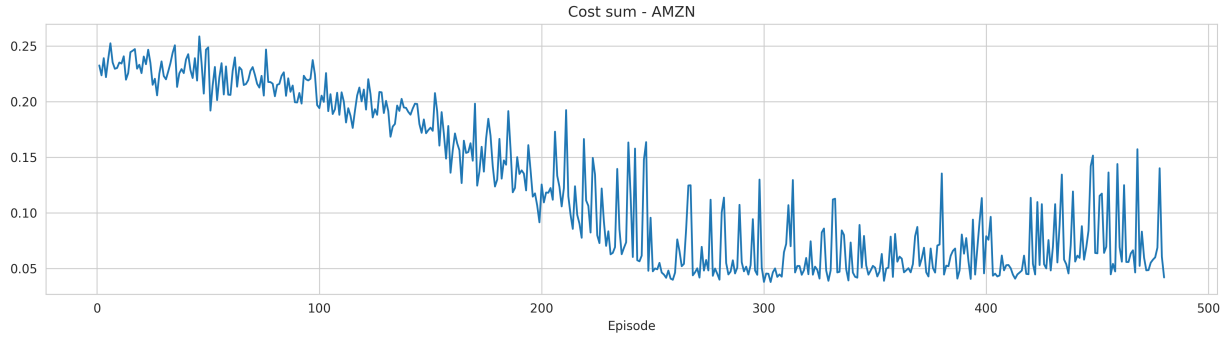


Figura B.142: Suma acumulada de costos

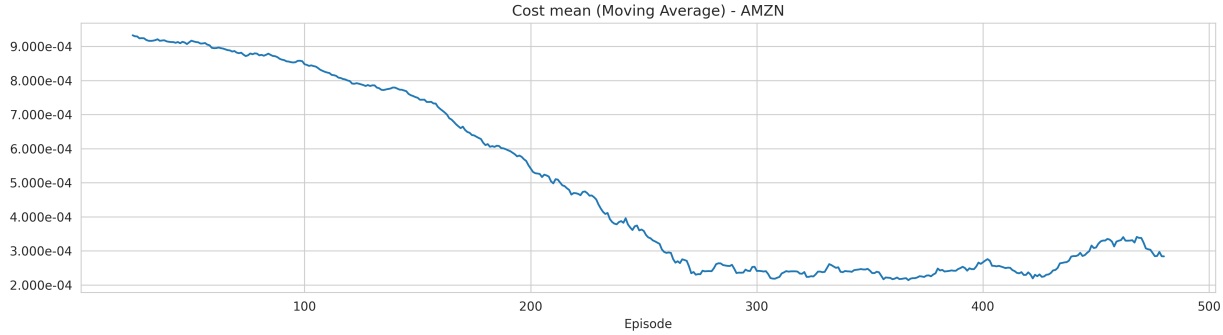


Figura B.143: Media móvil de promedio de costos

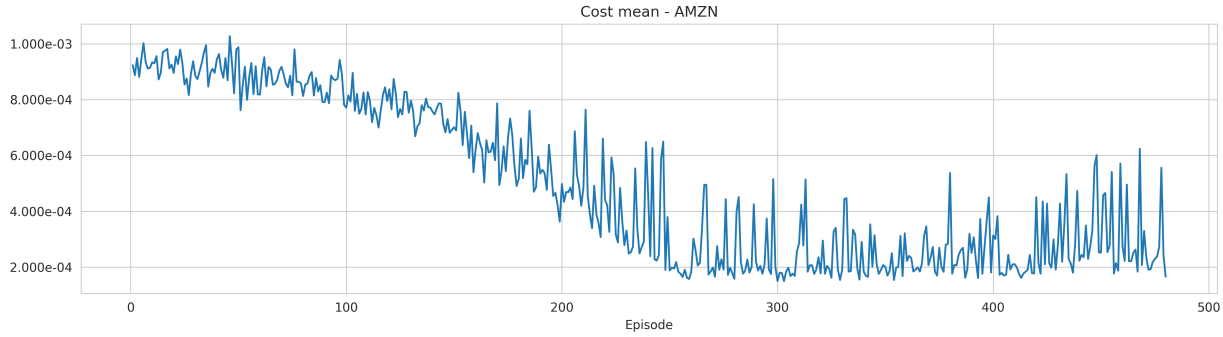


Figura B.144: Promedio de costos

B.2.3.4. Pepsico Inc (PEP)

La figura B.145 muestra la media móvil sobre los último 25 episodios de la suma acumulada de costos obtenidos durante un episodio para los 480 periodos de entrenamiento. La figura B.146 muestra estos mismos resultados sin una media móvil sobre los episodios. De manera similar, en las figuras B.147 y B.148 se muestra, bajo esta misma configuración, el promedio de costos obtenidos durante un episodio.

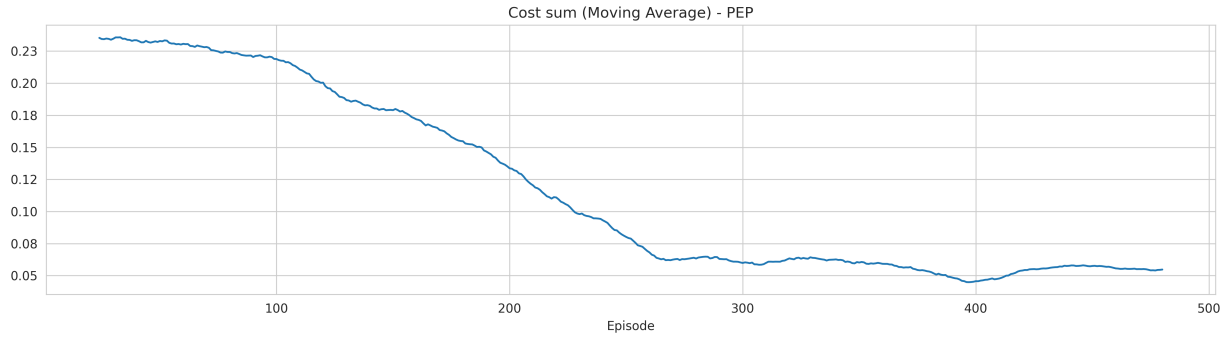


Figura B.145: Media móvil de suma acumulada de costos

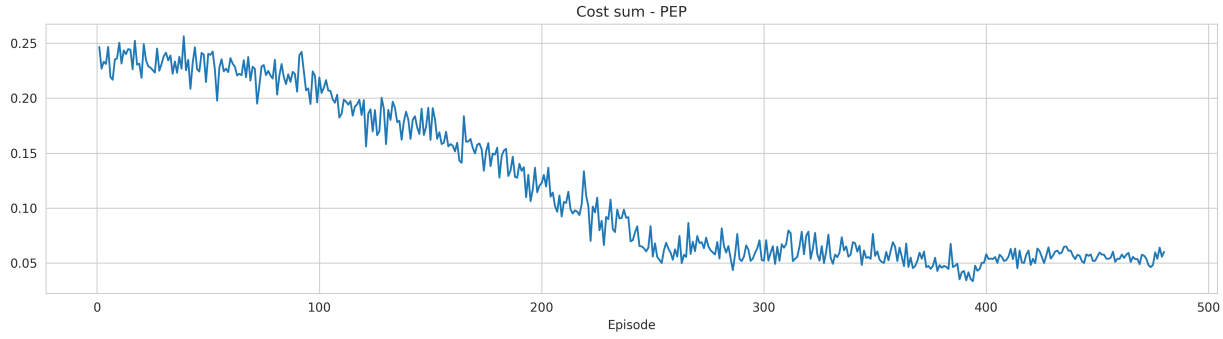


Figura B.146: Suma acumulada de costos

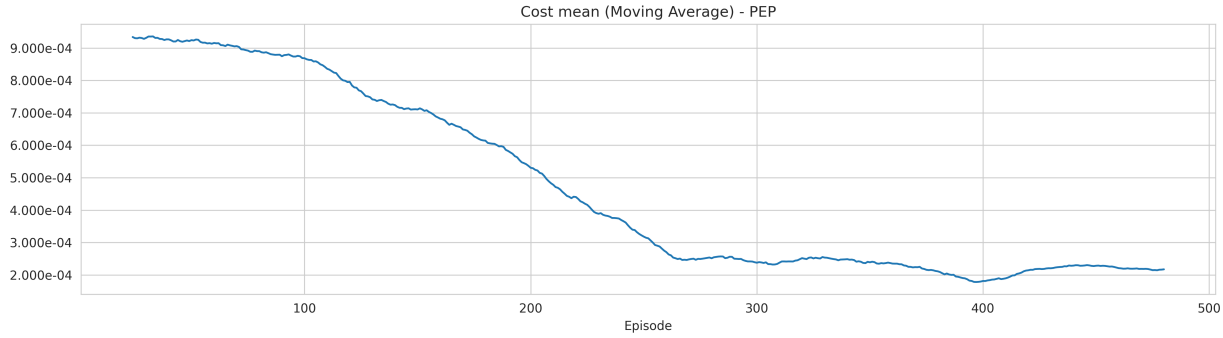


Figura B.147: Media móvil de promedio de costos

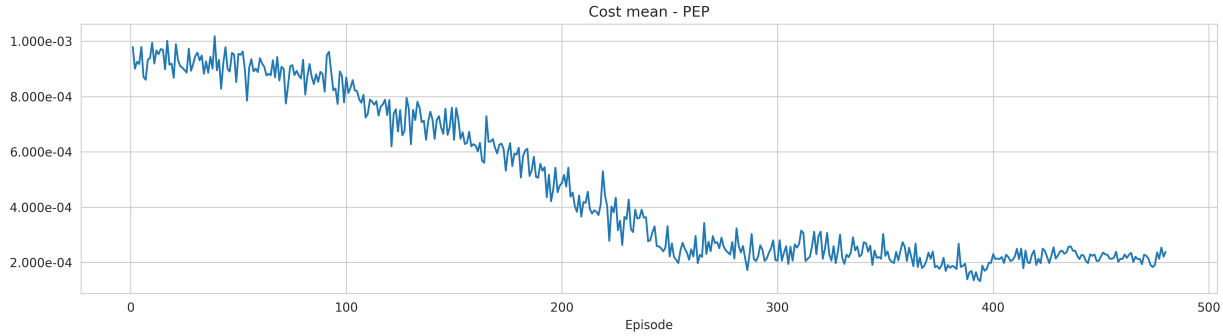


Figura B.148: Promedio de costos

B.2.4. Operaciones de trading

B.2.4.1. Apple Inc (AAPL)

La figura B.149 muestra la media móvil sobre los últimos 25 episodios de la suma acumulada de operaciones de trading durante un episodio para los 480 periodos de entrenamiento. La figura B.150 muestra estos mismos resultados sin una media móvil sobre los episodios. De manera similar, en las figuras B.151 y B.152 se muestra, bajo esta misma configuración, el promedio de operaciones de trading realizadas cada día durante un episodio.

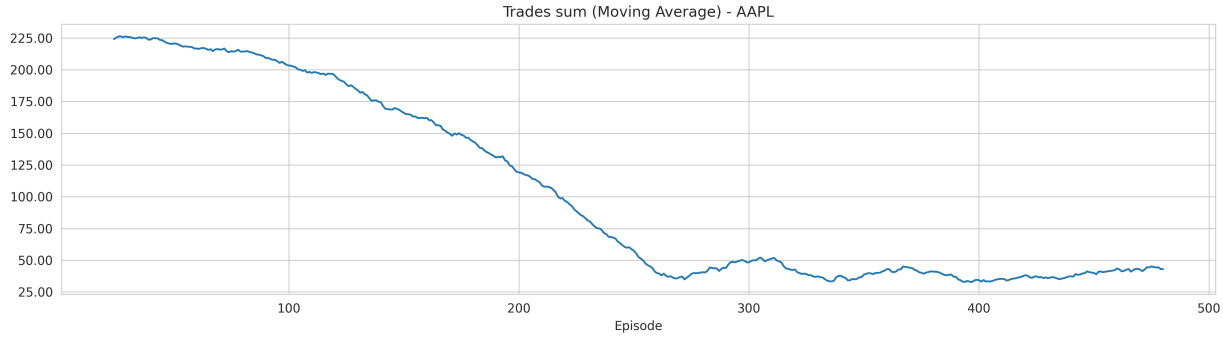


Figura B.149: Media móvil de suma acumulada de operaciones

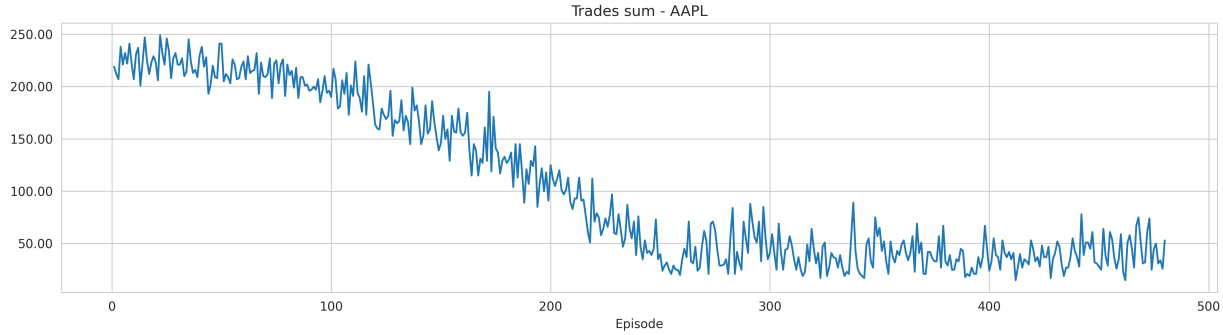


Figura B.150: Suma acumulada de operaciones

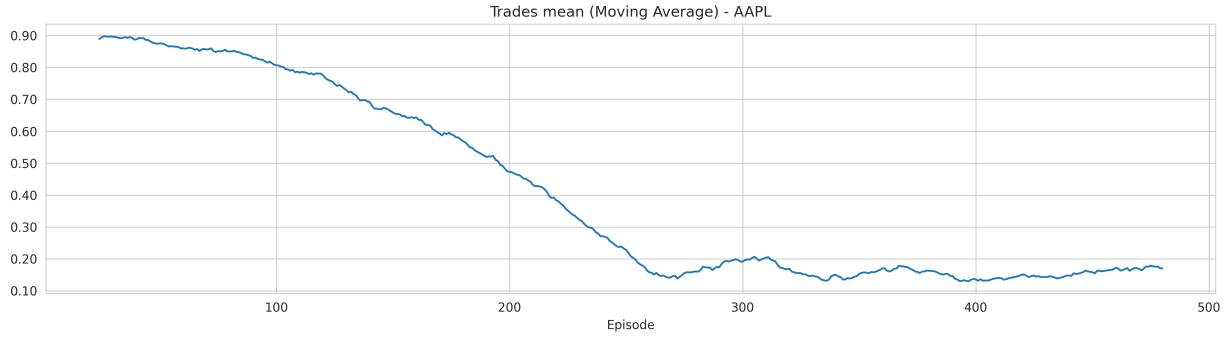


Figura B.151: Media móvil de promedio de operaciones

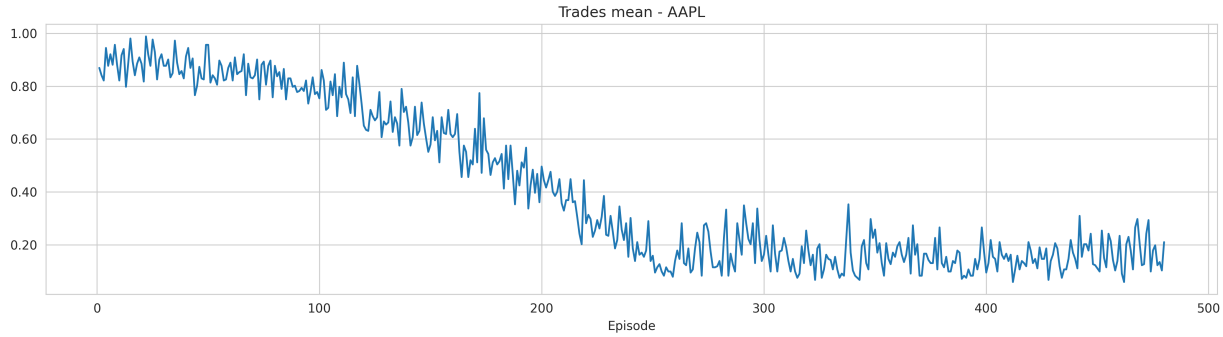


Figura B.152: Promedio de operaciones

B.2.4.2. Microsoft (MSFT)

La figura B.153 muestra la media móvil sobre los últimos 25 episodios de la suma acumulada de operaciones de trading durante un episodio para los 480 periodos de entrenamiento. La figura B.154 muestra estos mismos resultados sin una media móvil sobre los episodios. De manera similar, en las figuras B.155 y B.156 se muestra, bajo esta misma configuración, el promedio de operaciones de trading realizadas cada día durante un episodio.

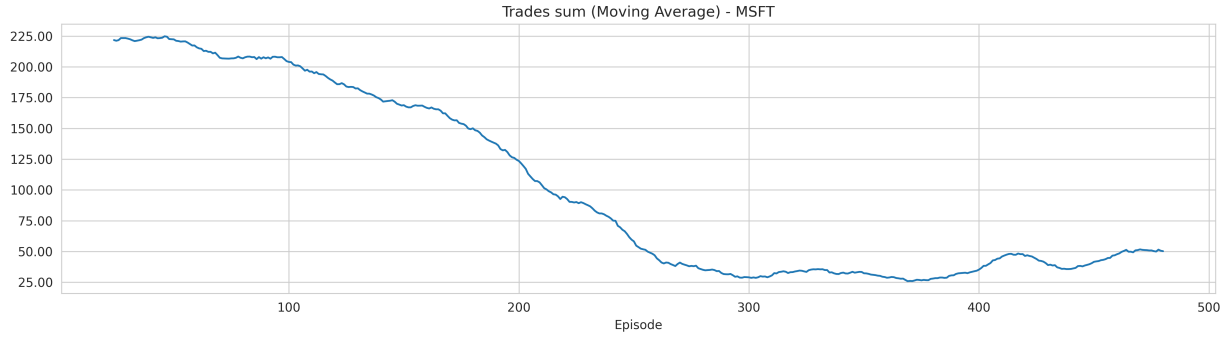


Figura B.153: Media móvil de suma acumulada de operaciones

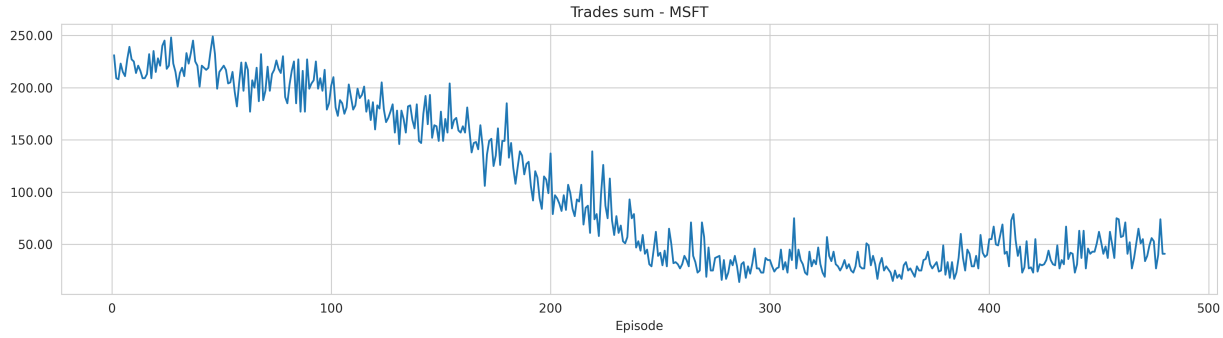


Figura B.154: Suma acumulada de operaciones

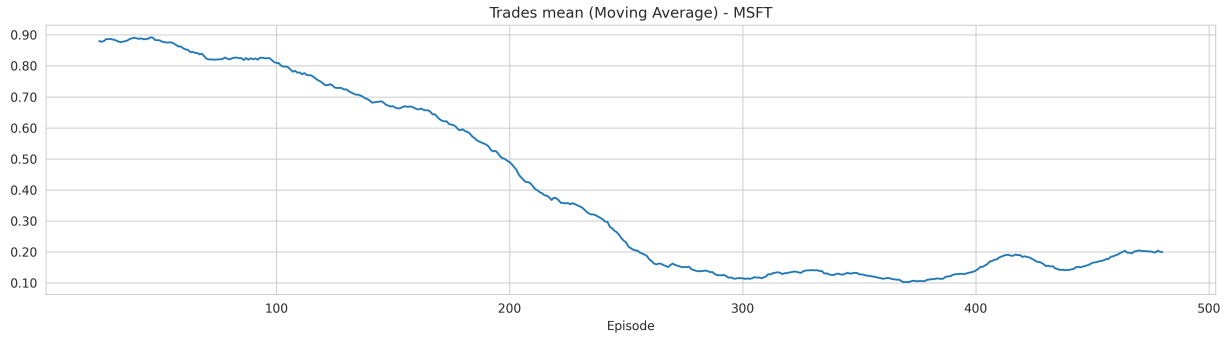


Figura B.155: Media móvil de promedio de operaciones

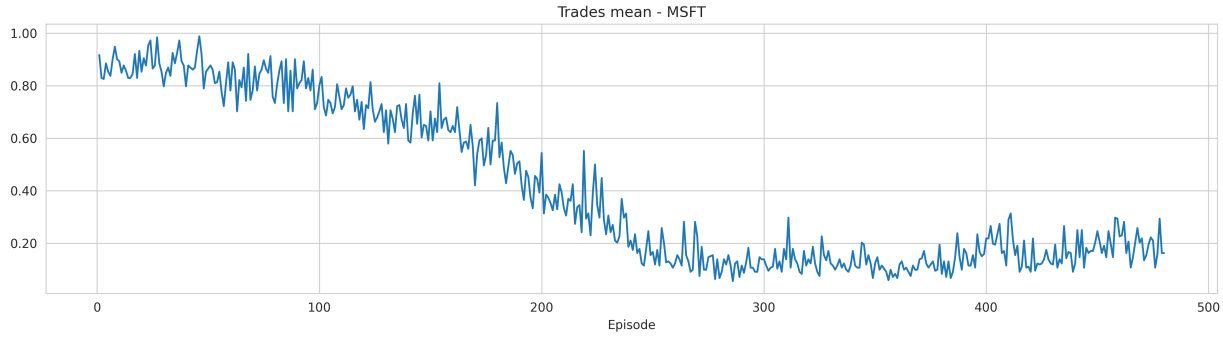


Figura B.156: Promedio de operaciones

B.2.4.3. Amazon Inc (AMZN)

La figura B.157 muestra la media móvil sobre los últimos 25 episodios de la suma acumulada de operaciones de trading durante un episodio para los 480 periodos de entrenamiento. La figura B.158 muestra estos mismos resultados sin una media móvil sobre los episodios. De manera similar, en las figuras B.159 y B.160 se muestra, bajo esta misma configuración, el promedio de operaciones de trading realizadas cada día durante un episodio.

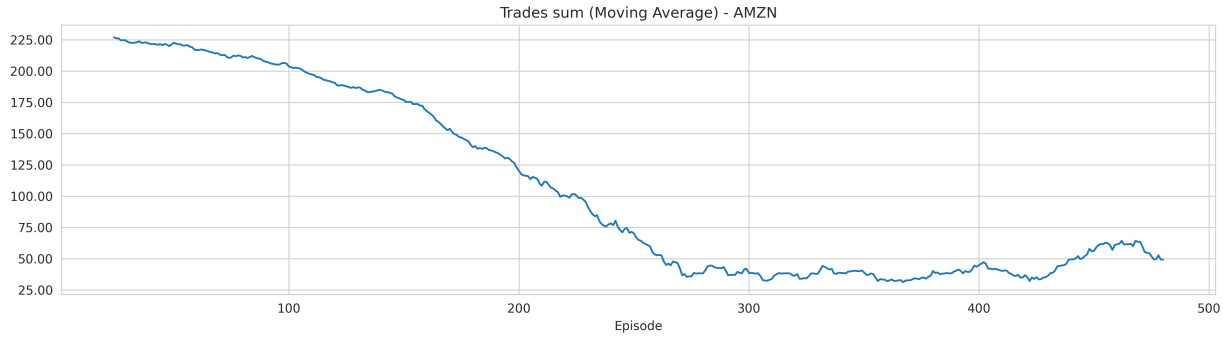


Figura B.157: Media móvil de suma acumulada de operaciones

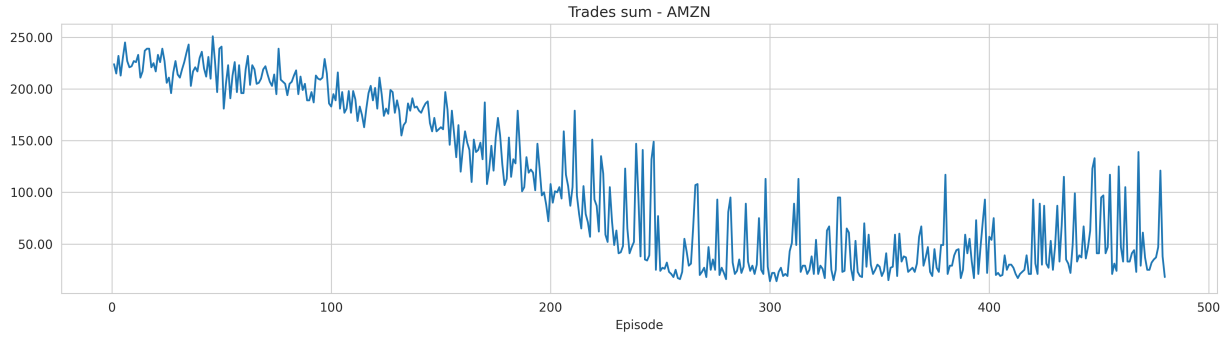


Figura B.158: Suma acumulada de operaciones

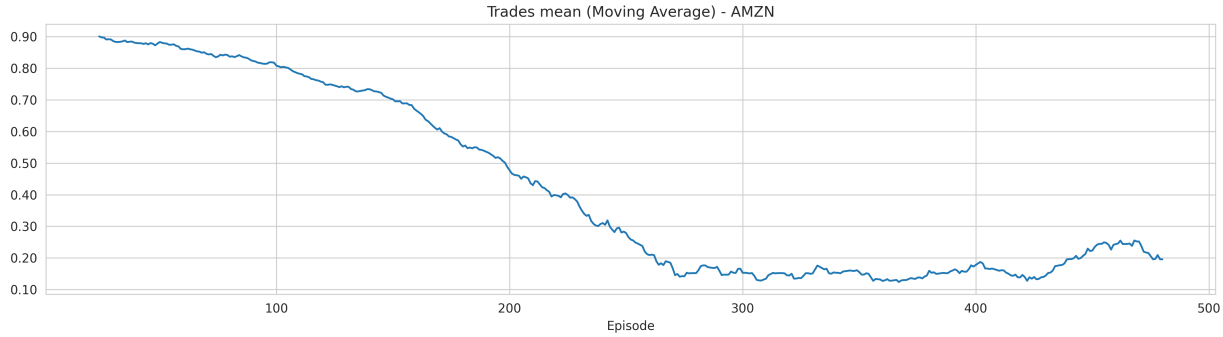


Figura B.159: Media móvil de promedio de operaciones

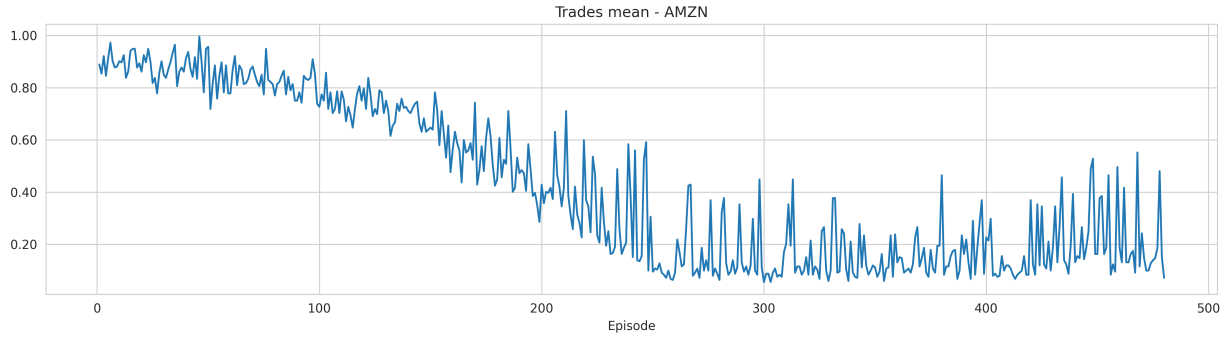


Figura B.160: Promedio de operaciones

B.2.4.4. Pepsico Inc (PEP)

La figura B.161 muestra la media móvil sobre los últimos 25 episodios de la suma acumulada de operaciones de trading durante un episodio para los 480 periodos de entrenamiento. La figura B.162 muestra estos mismos resultados sin una media móvil sobre los episodios. De manera similar, en las figuras B.163 y B.164 se muestra, bajo esta misma configuración, el promedio de operaciones de trading realizadas cada día durante un episodio.

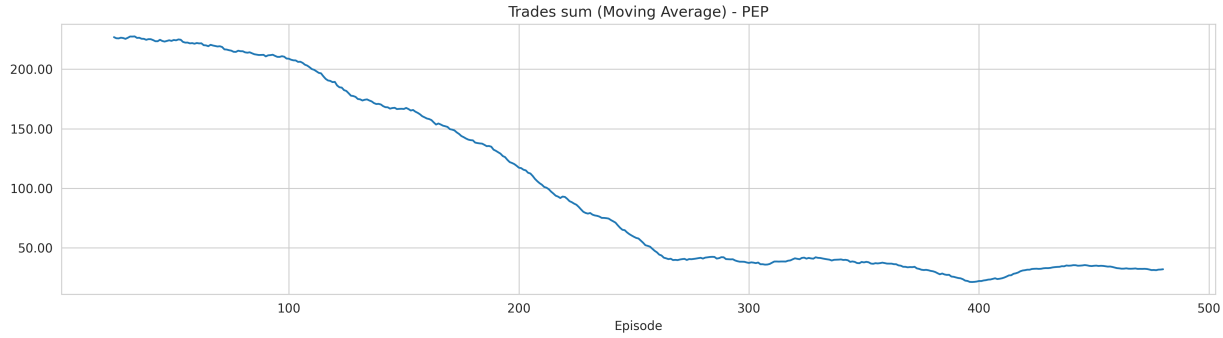


Figura B.161: Media móvil de suma acumulada de operaciones

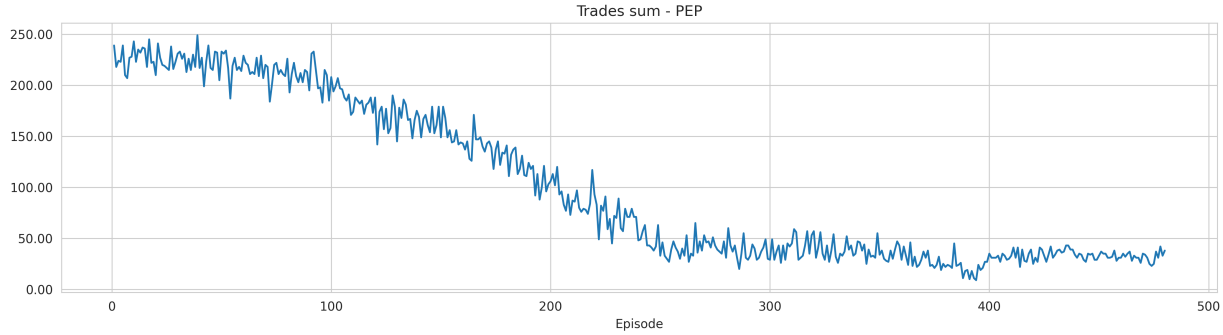


Figura B.162: Suma acumulada de operaciones

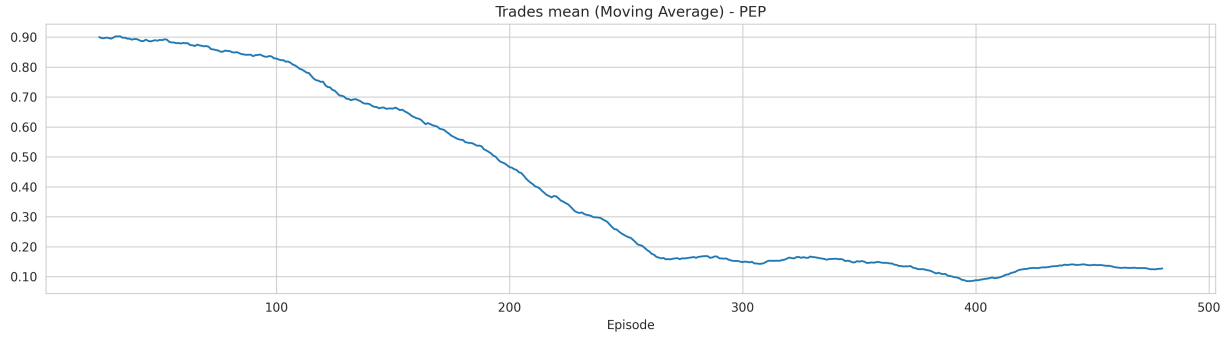


Figura B.163: Media móvil de promedio de operaciones

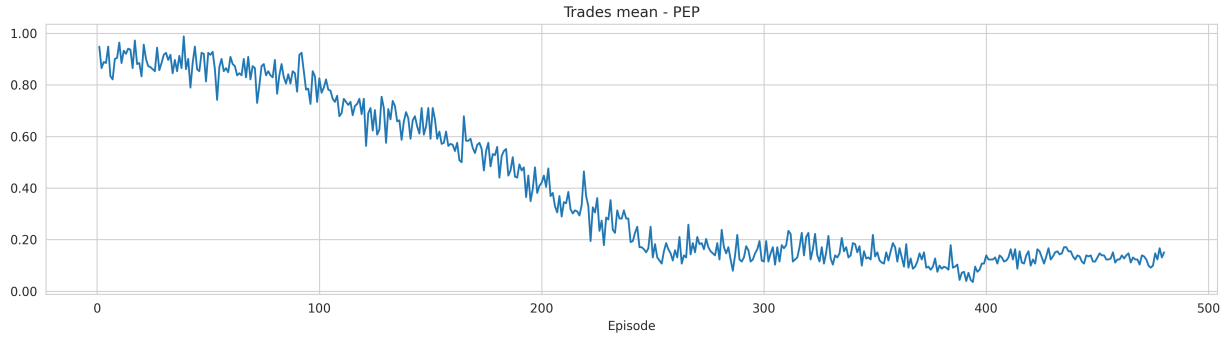


Figura B.164: Promedio de operaciones

B.2.5. Función de pérdida

B.2.5.1. Apple Inc (AAPL)

La figura B.165 muestra la media móvil sobre los últimos 25 episodios del promedio de pérdidas acumuladas para los 480 periodos de entrenamiento.

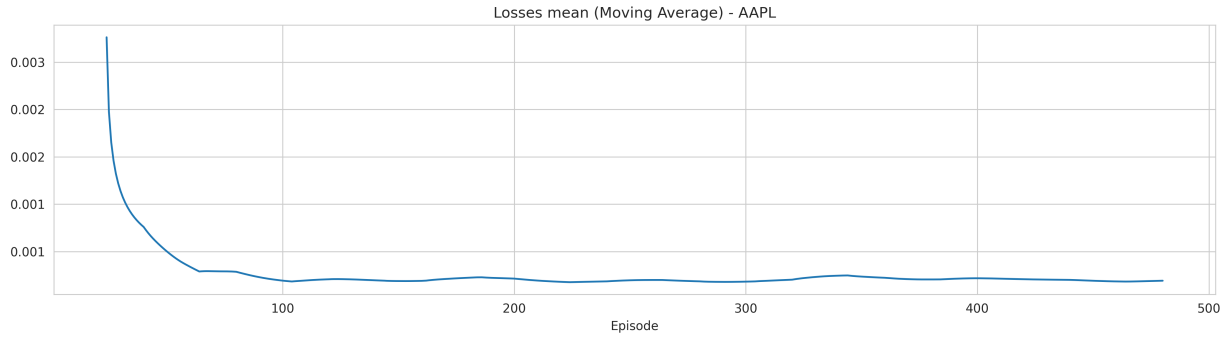


Figura B.165: Media móvil de promedio de pérdidas acumuladas

B.2.5.2. Microsoft (MSFT)

La figura B.166 muestra la media móvil sobre los últimos 25 episodios del promedio de pérdidas acumuladas para los 480 periodos de entrenamiento.

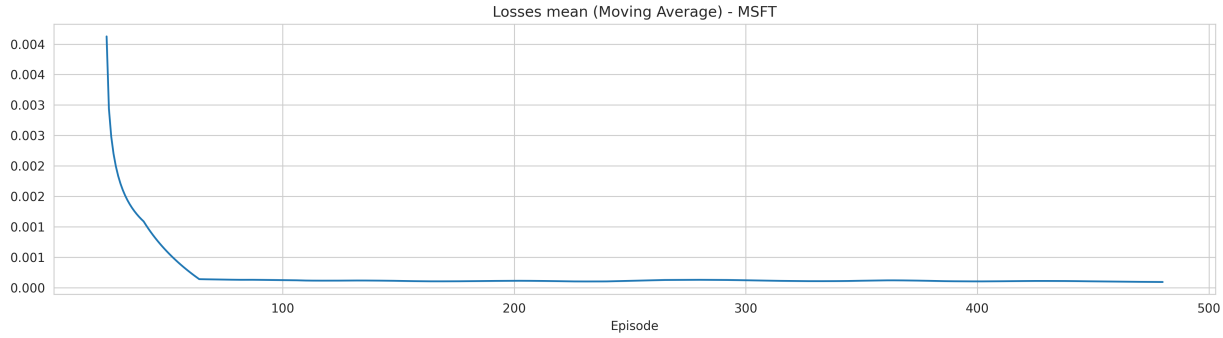


Figura B.166: Media móvil de promedio de pérdidas acumuladas

B.2.5.3. Amazon Inc (AMZN)

La figura B.167 muestra la media móvil sobre los últimos 25 episodios del promedio de pérdidas acumuladas para los 480 periodos de entrenamiento.

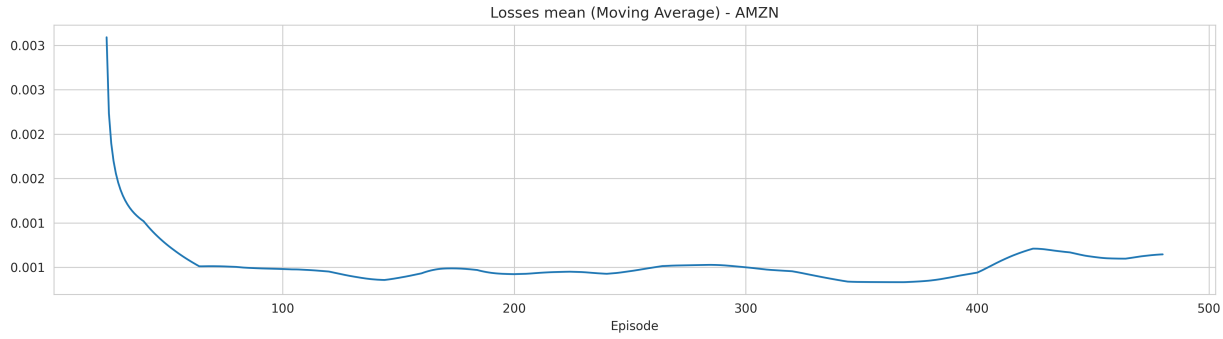


Figura B.167: Media móvil de promedio de pérdidas acumuladas

B.2.5.4. Pepsico Inc (PEP)

La figura B.168 muestra la media móvil sobre los últimos 25 episodios del promedio de pérdidas acumuladas para los 480 periodos de entrenamiento.

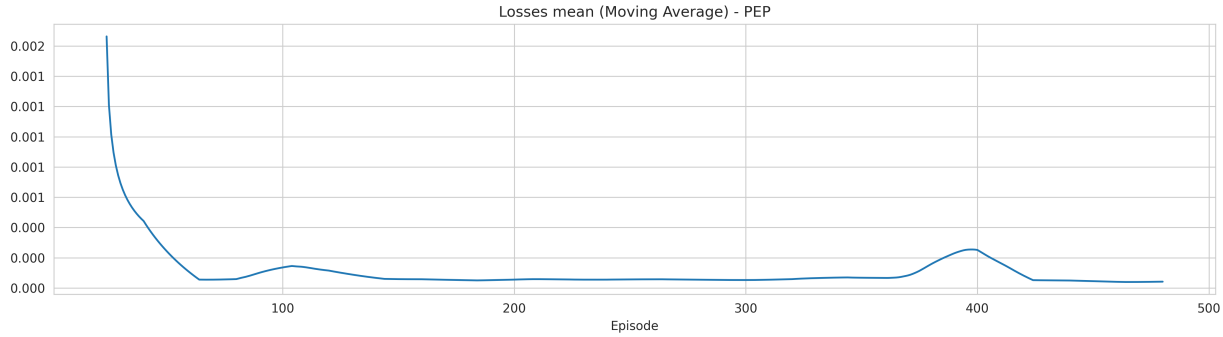


Figura B.168: Media móvil de promedio de pérdidas acumuladas

B.3. Experimento 3

En esta sección se presentarán los resultados para el experimento descrito en la sección 4.3.3.3.

B.3.1. Rendimientos

B.3.1.1. Apple Inc (AAPL)

La figura B.169 muestra la media móvil sobre los últimos 50 episodios de los valores de retorno acumulativo para los 480 periodos de entrenamiento, así como también la media móvil de la proporción de los últimos 50 episodios en los que el agente superó la estrategia *buy-and-hold*. La figura B.170 muestra estos mismos resultados sin una media móvil sobre los episodios. De manera similar, en las figuras B.171 y B.172 se muestra, bajo esta misma configuración, el promedio de los valores de retorno acumulativo obtenidos cada día durante un episodio.

Figura B.169: Media móvil de rendimientos anuales

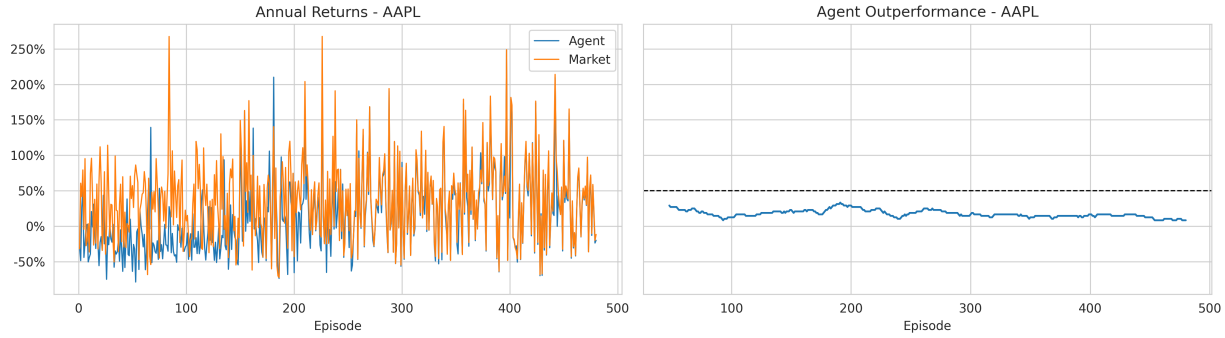


Figura B.170: Rendimientos anuales

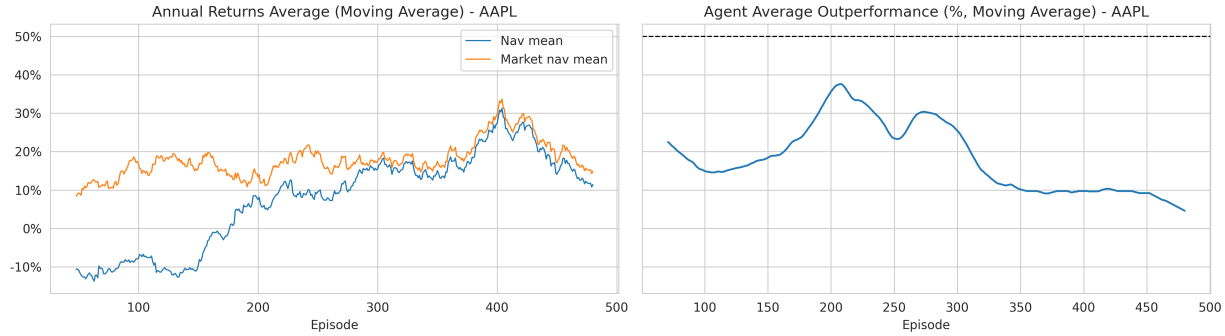


Figura B.171: Media móvil de rendimientos anuales promedio

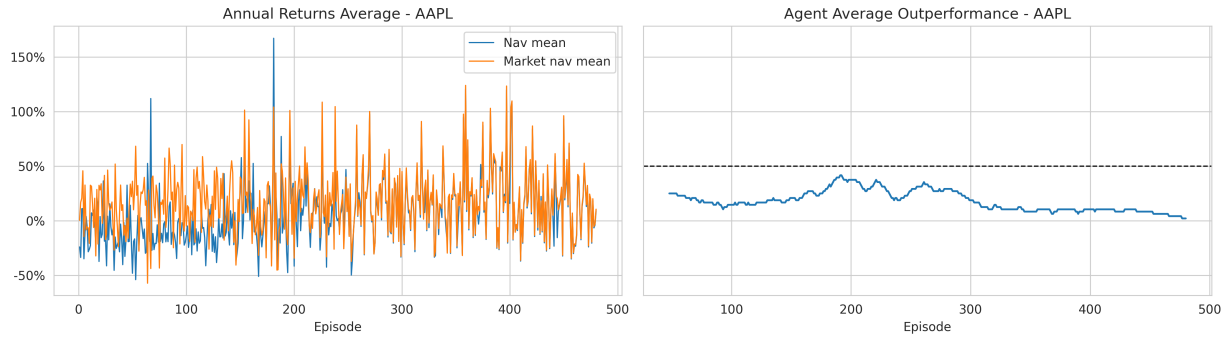


Figura B.172: Rendimientos anuales promedio

Finalmente, en la figura B.173 se muestra la media móvil sobre los últimos 50 episodios de la proporción de días en los que el agente superó la estrategia *buy-and-hold* durante un episodio. La figura B.174 muestra estos mismos resultados sin una media móvil sobre los episodios.

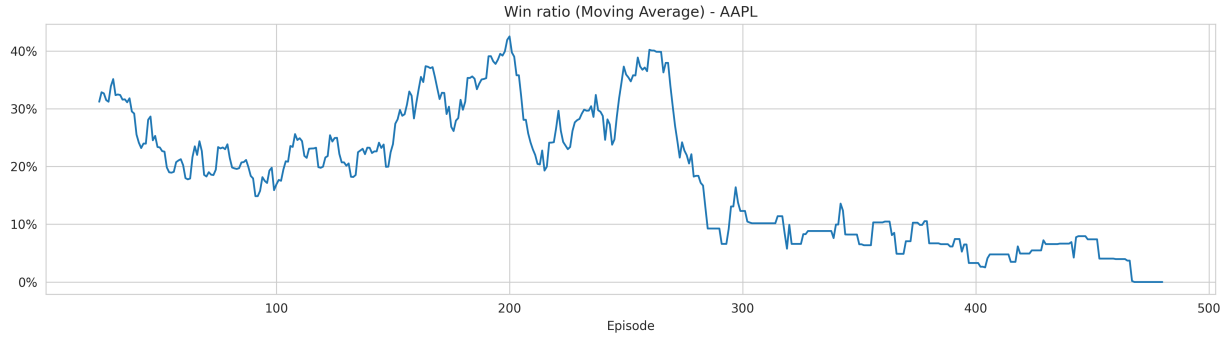


Figura B.173: Media móvil de proporción de ganancias

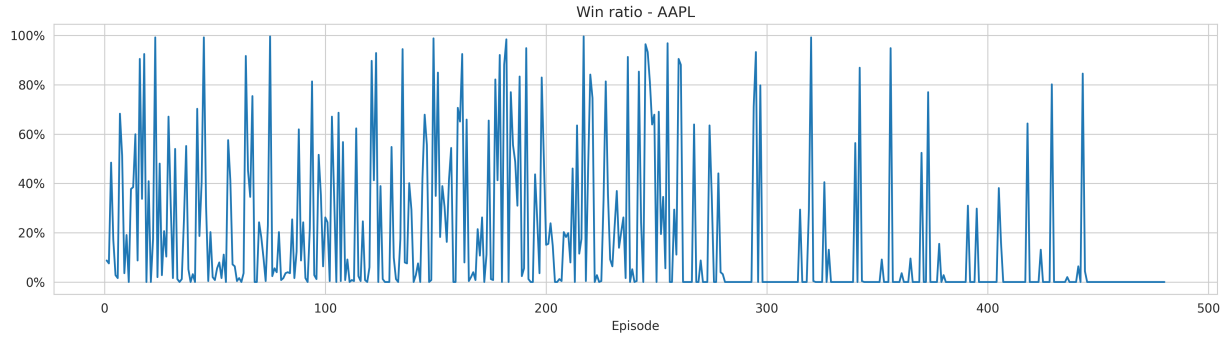


Figura B.174: Proporción de ganancias

B.3.1.2. Microsoft (MSFT)

La figura B.175 muestra la media móvil sobre los últimos 50 episodios de los valores de retorno acumulativo para los 480 periodos de entrenamiento, así como también la media móvil de la proporción de los últimos 50 episodios en los que el agente superó la estrategia *buy-and-hold*. La figura B.176 muestra estos mismos resultados sin una media móvil sobre los episodios. De manera similar, en las figuras B.177 y B.178 se muestra, bajo esta misma configuración, el promedio de los valores de retorno acumulativo obtenidos cada día durante un episodio.

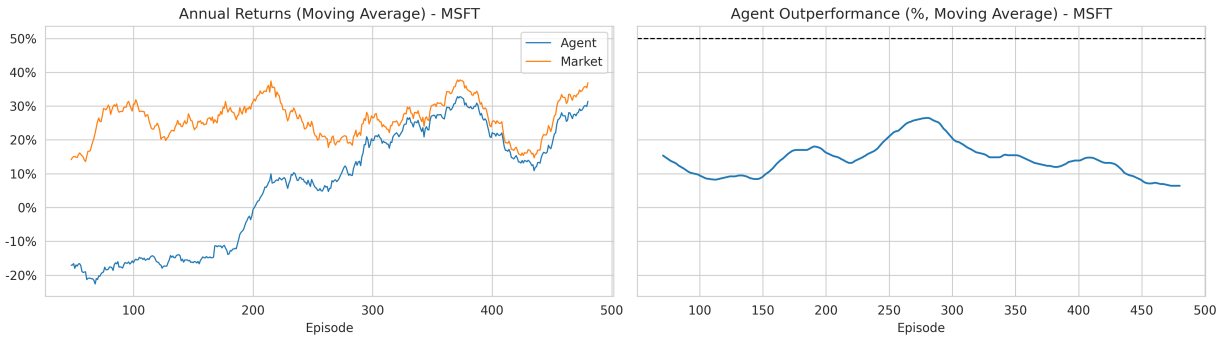


Figura B.175: Media móvil de rendimientos anuales

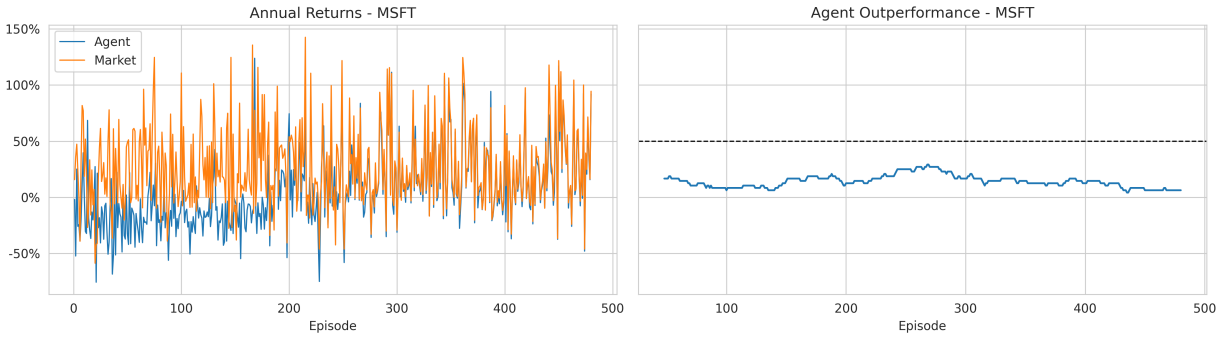


Figura B.176: Rendimientos anuales

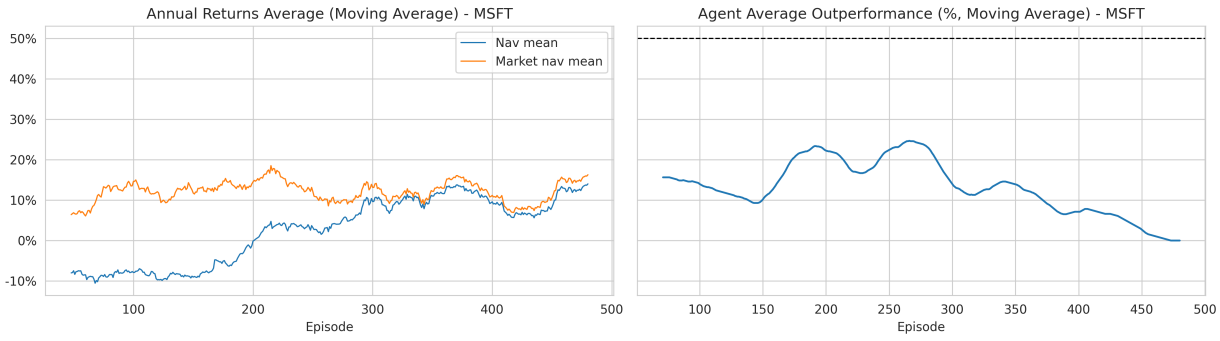


Figura B.177: Media móvil de rendimientos anuales promedio

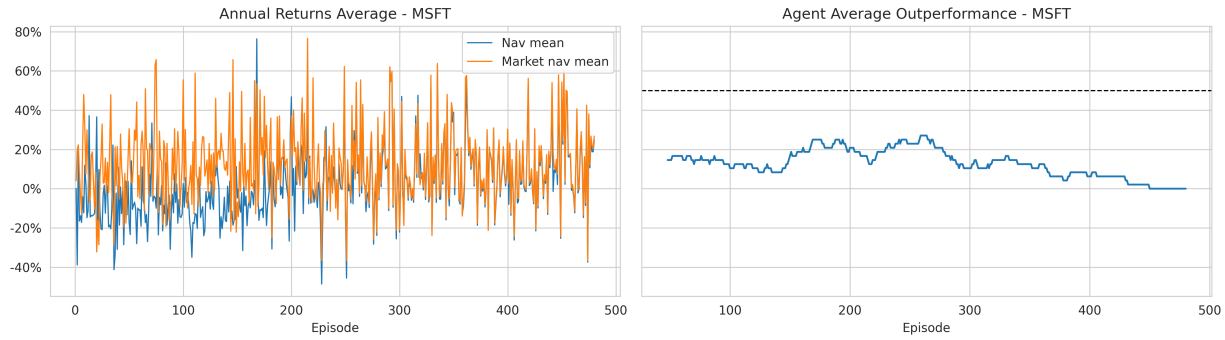


Figura B.178: Rendimientos anuales promedio

Finalmente, en la figura B.179 se muestra la media móvil sobre los últimos 50 episodios de la proporción de días en los que el agente superó la estrategia *buy-and-hold* durante un episodio. La figura B.180 muestra estos mismos resultados sin una media móvil sobre los episodios.

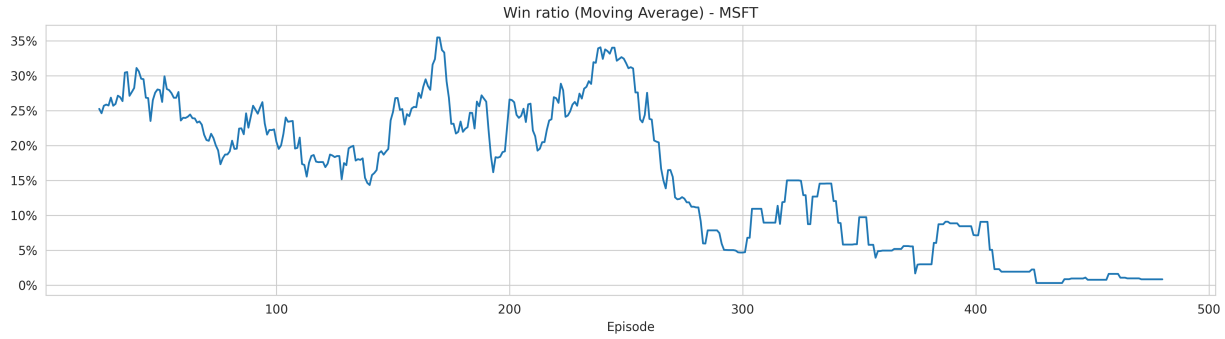


Figura B.179: Media móvil de proporción de ganancias

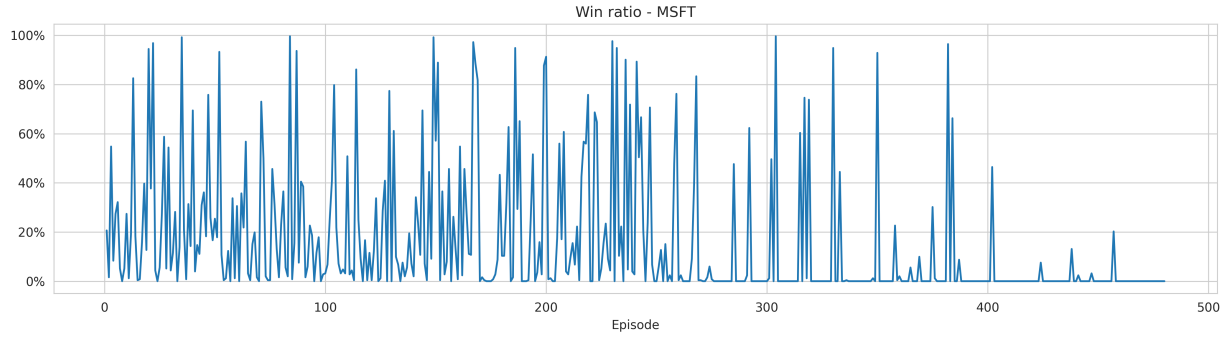


Figura B.180: Proporción de ganancias

B.3.1.3. Amazon Inc (AMZN)

La figura B.181 muestra la media móvil sobre los últimos 50 episodios de los valores de retorno acumulativo para los 480 periodos de entrenamiento, así como también la media móvil de la proporción de los últimos 50 episodios en los que el agente superó la estrategia *buy-and-hold*. La figura B.182 muestra estos mismos resultados sin una media móvil sobre los episodios. De manera similar, en las figuras B.183 y B.184 se muestra, bajo esta misma configuración, el promedio de los valores de retorno acumulativo obtenidos cada día durante un episodio.

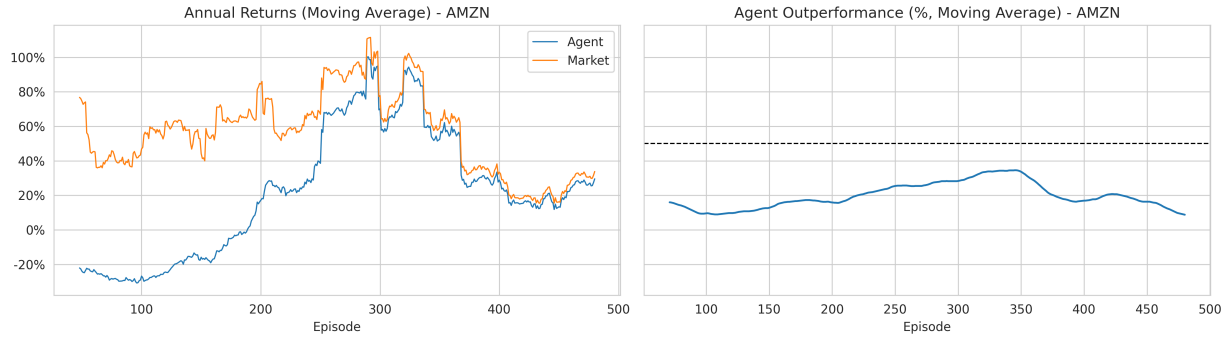


Figura B.181: Media móvil de rendimientos anuales

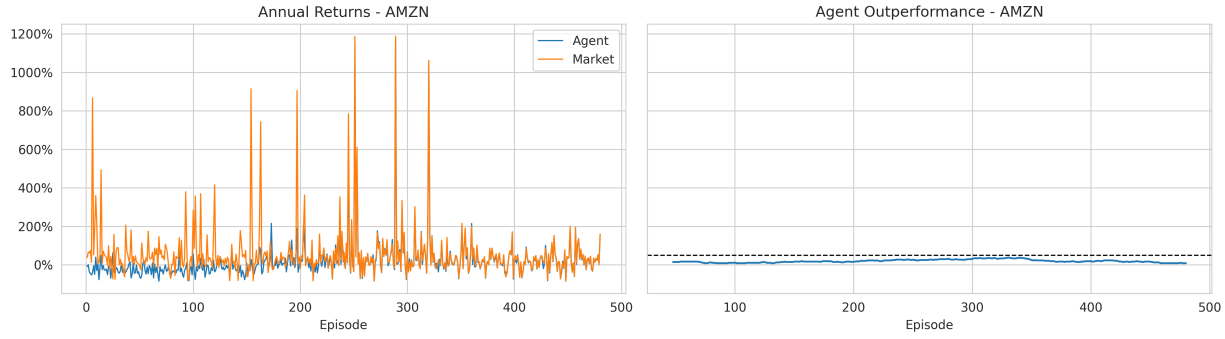


Figura B.182: Rendimientos anuales

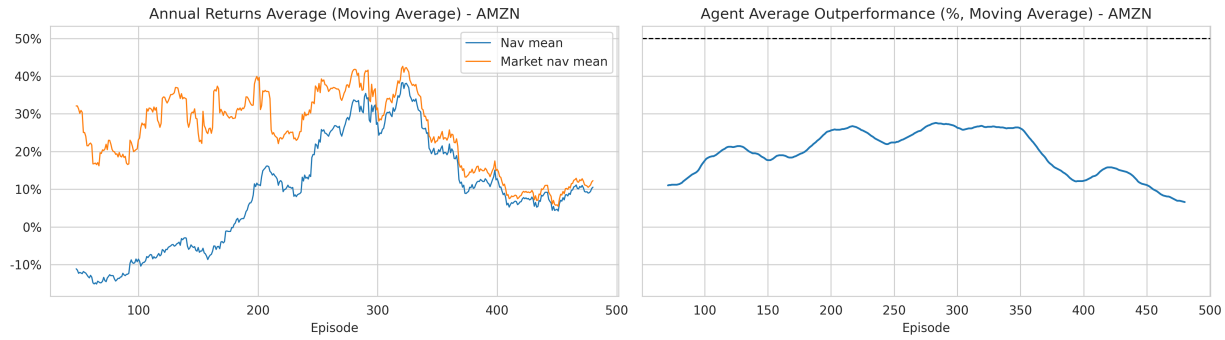


Figura B.183: Media móvil de rendimientos anuales promedio

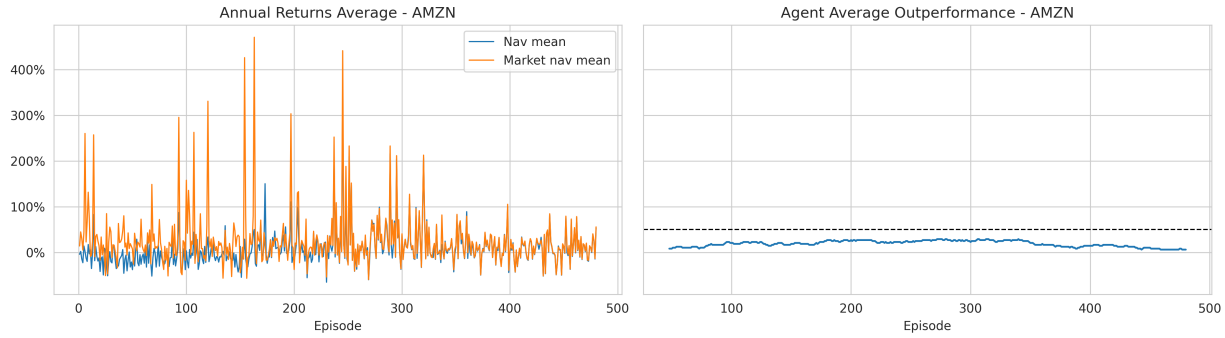


Figura B.184: Rendimientos anuales promedio

Finalmente, en la figura B.185 se muestra la media móvil sobre los últimos 50 episodios de la proporción de días en los que el agente superó la estrategia *buy-and-hold* durante un episodio. La figura B.186 muestra estos mismos resultados sin una media móvil sobre los episodios.

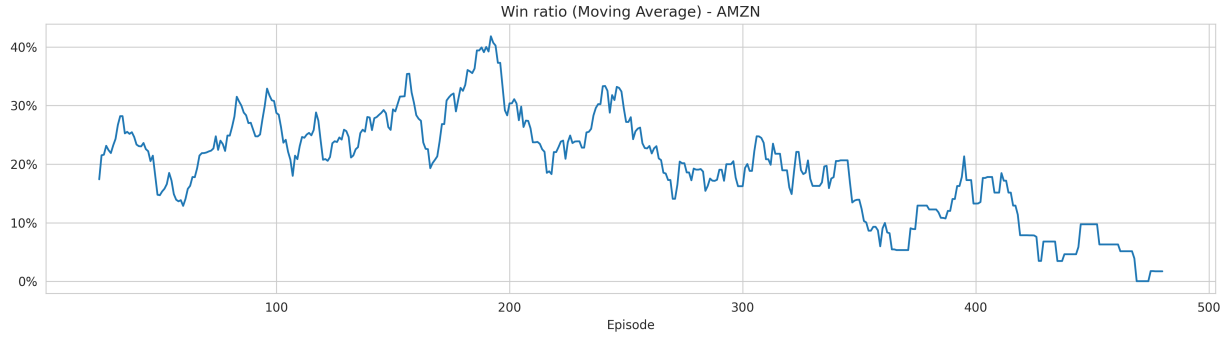


Figura B.185: Media móvil de proporción de ganancias

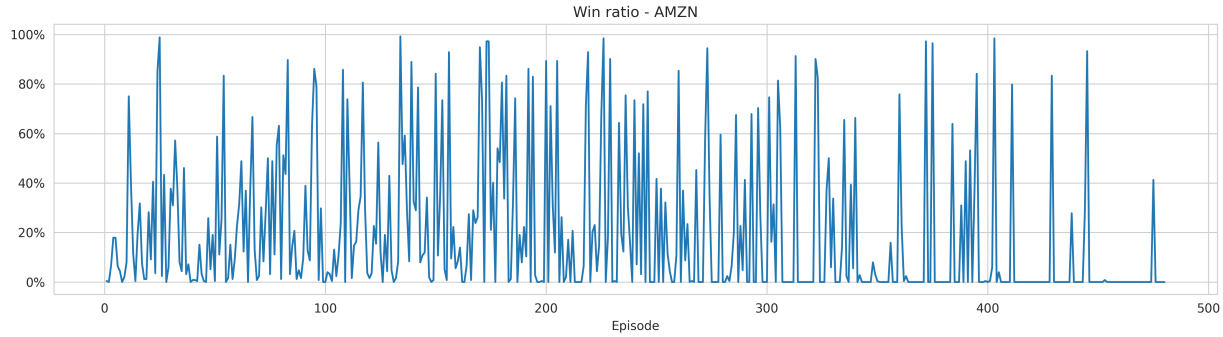


Figura B.186: Proporción de ganancias

B.3.1.4. Pepsico Inc (PEP)

La figura B.187 muestra la media móvil sobre los últimos 50 episodios de los valores de retorno acumulativo para los 480 periodos de entrenamiento, así como también la media móvil de la proporción de los últimos 50 episodios en los que el agente superó la estrategia *buy-and-hold*. La figura B.188 muestra estos mismos resultados sin una media móvil sobre los episodios. De manera similar, en las figuras B.189 y B.190 se muestra, bajo esta misma configuración, el promedio de los valores de retorno acumulativo obtenidos cada día durante un episodio.

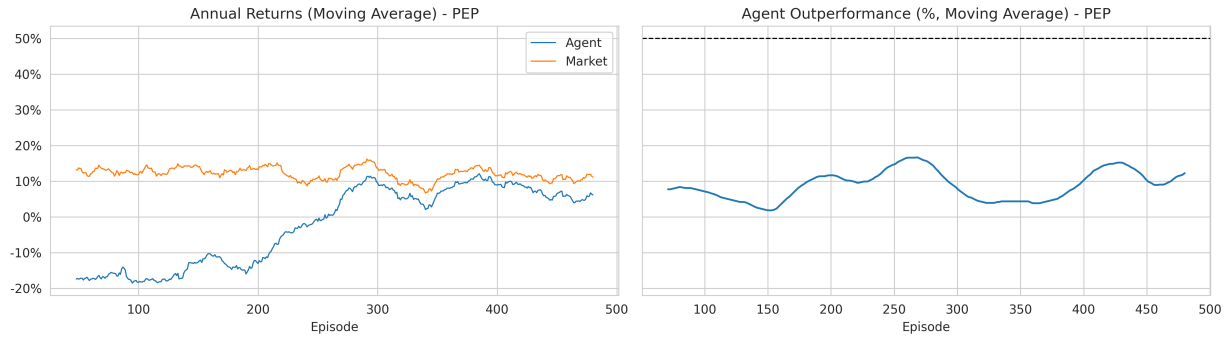


Figura B.187: Media móvil de rendimientos anuales

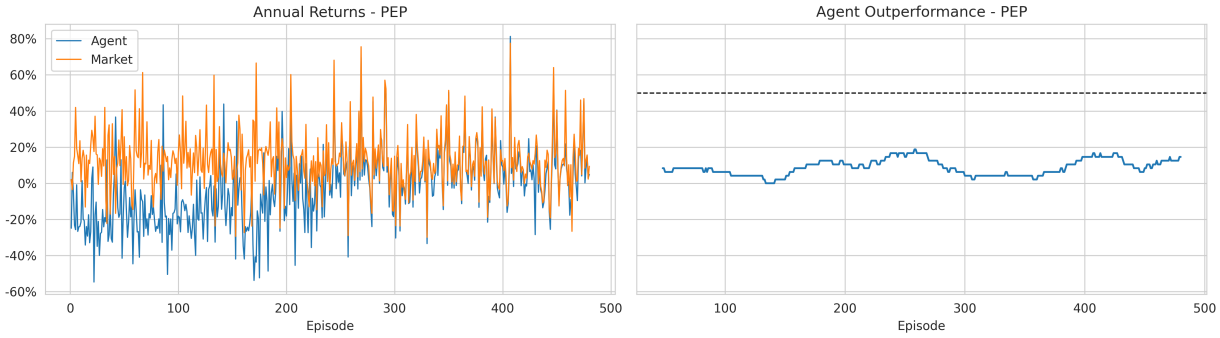


Figura B.188: Rendimientos anuales

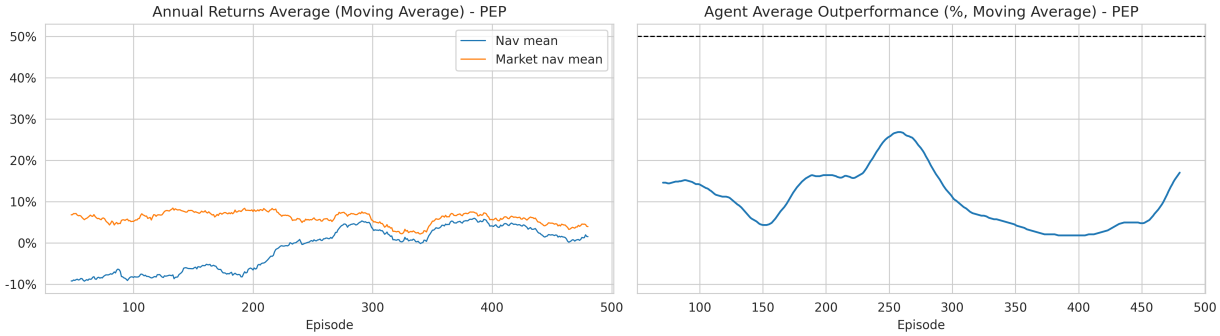


Figura B.189: Media móvil de rendimientos anuales promedio

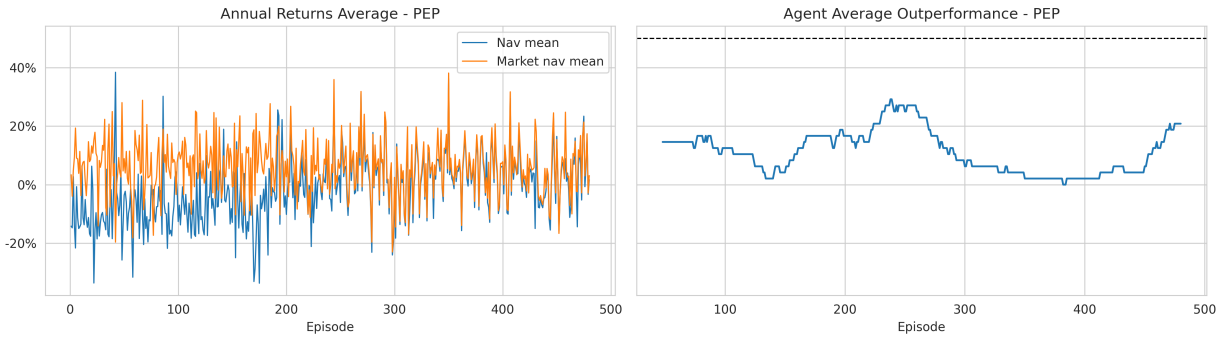


Figura B.190: Rendimientos anuales promedio

Finalmente, en la figura B.191 se muestra la media móvil sobre los últimos 50 episodios de la proporción de días en los que el agente superó la estrategia *buy-and-hold* durante un episodio. La figura B.192 muestra estos mismos resultados sin una media móvil sobre los episodios.

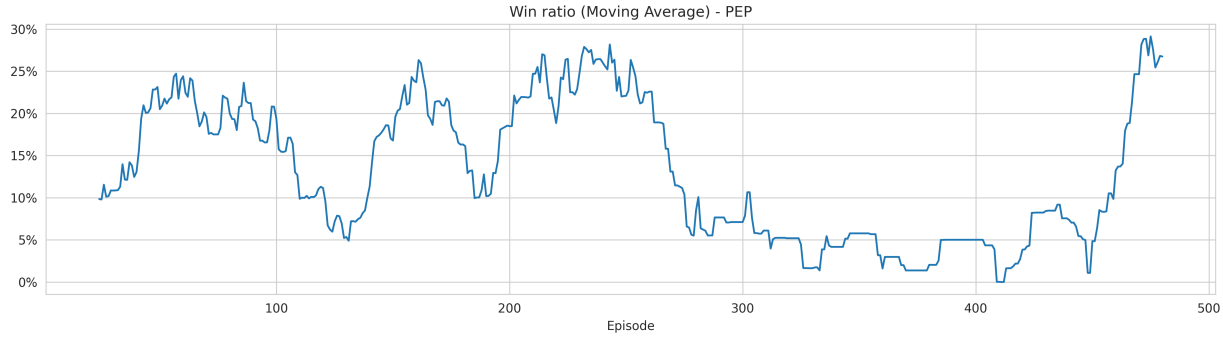


Figura B.191: Media móvil de proporción de ganancias

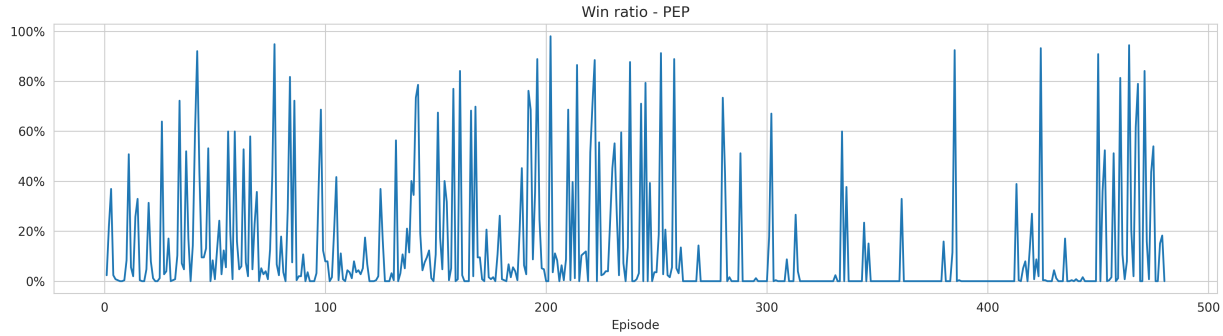


Figura B.192: Proporción de ganancias

B.3.2. Recompensas

B.3.2.1. Apple Inc (AAPL)

La figura B.193 muestra la media móvil sobre los últimos 25 episodios de la suma acumulada de recompensas obtenidas durante un episodio para los 480 periodos de entrenamiento. La figura B.194 muestra estos mismos resultados sin una media móvil sobre los episodios. De manera similar, en las figuras B.195 y B.196 se muestra, bajo esta misma configuración, el promedio de recompensas obtenidas durante un episodio.

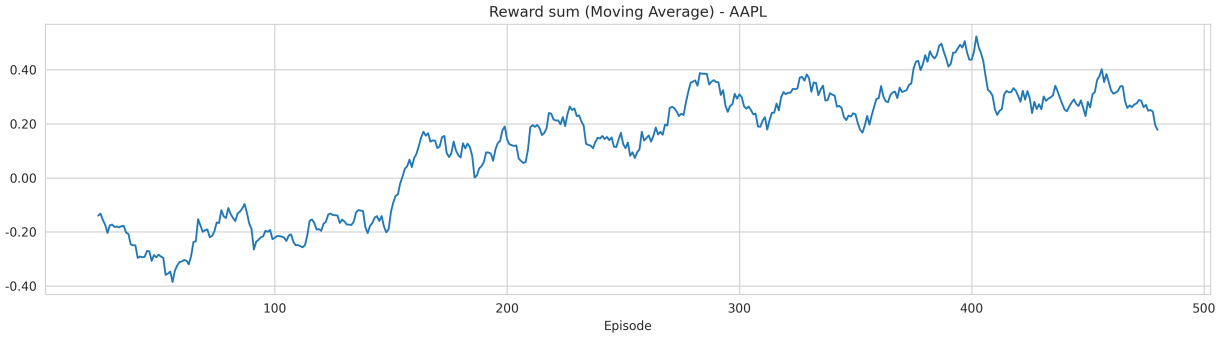


Figura B.193: Media móvil de suma acumulada de recompensas

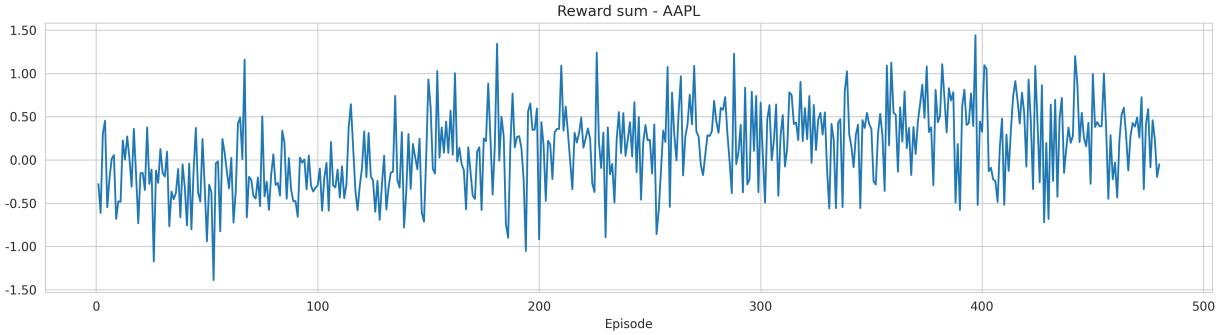


Figura B.194: Suma acumulada de recompensas

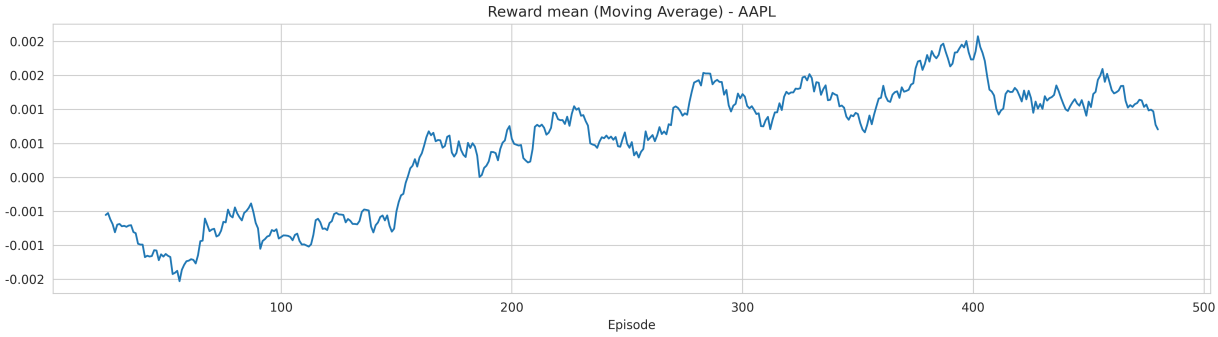


Figura B.195: Media móvil de promedio de recompensas

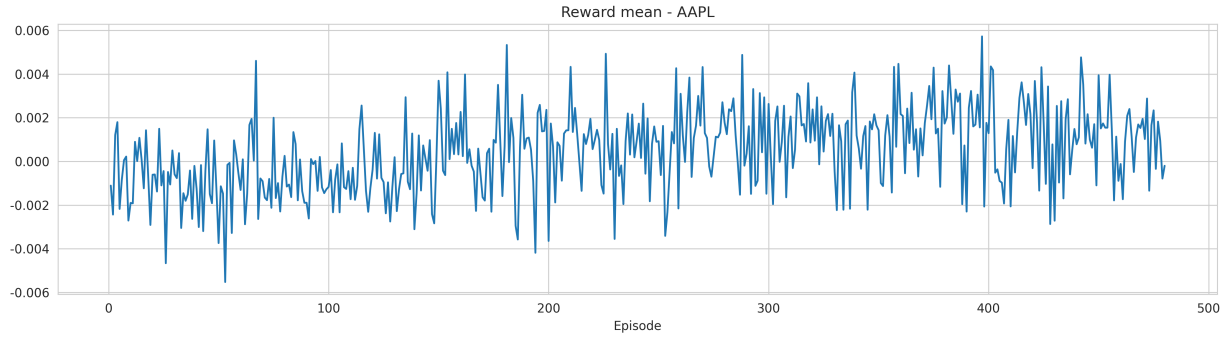


Figura B.196: Promedio de recompensas

Finalmente, en la figura B.197 se muestra la media móvil sobre los últimos 25 episodios de la diferencia entre la mayor y menor recompensa obtenidas durante un episodio. La figura B.198 muestra estos mismos resultados sin una media móvil sobre los episodios.

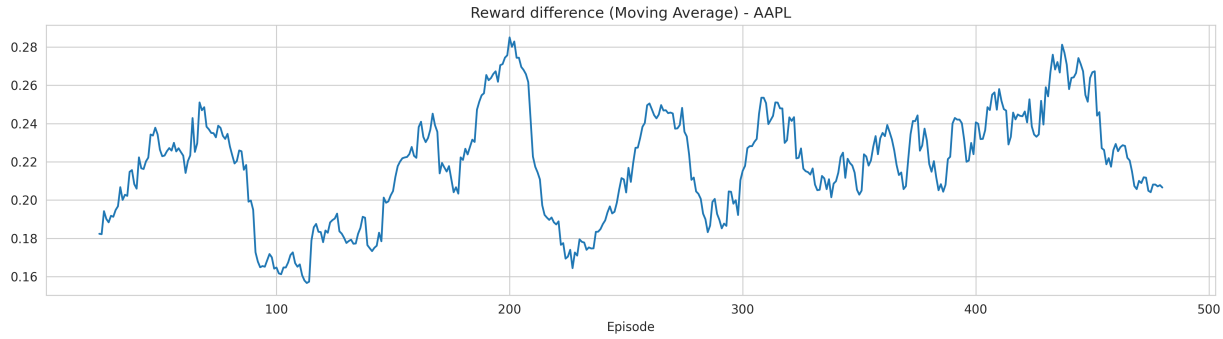


Figura B.197: Media móvil de diferencia de recompensa

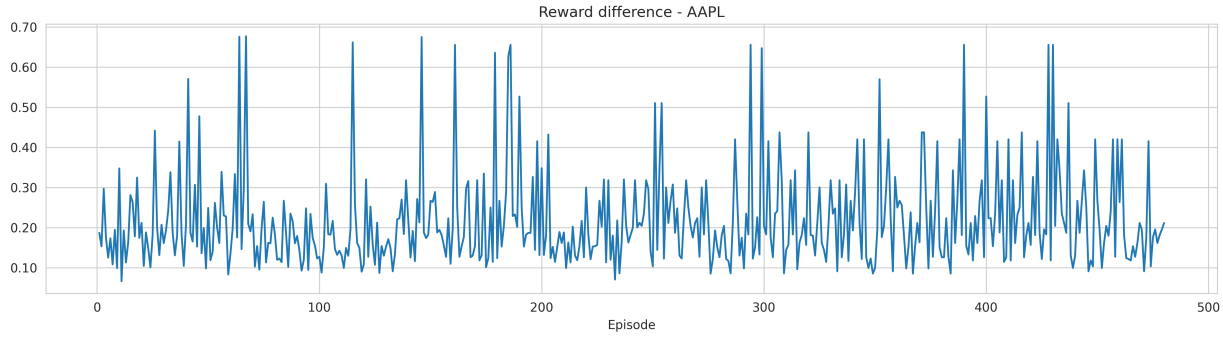


Figura B.198: Diferencia de recompensa

B.3.2.2. Microsoft (MSFT)

La figura B.199 muestra la media móvil sobre los últimos 25 episodios de la suma acumulada de recompensas obtenidas durante un episodio para los 480 periodos de entrenamiento. La figura B.200 muestra estos mismos resultados sin una media móvil sobre los episodios. De manera similar, en las figuras B.201 y B.202 se muestra, bajo esta misma configuración, el promedio de recompensas obtenidas durante un episodio.

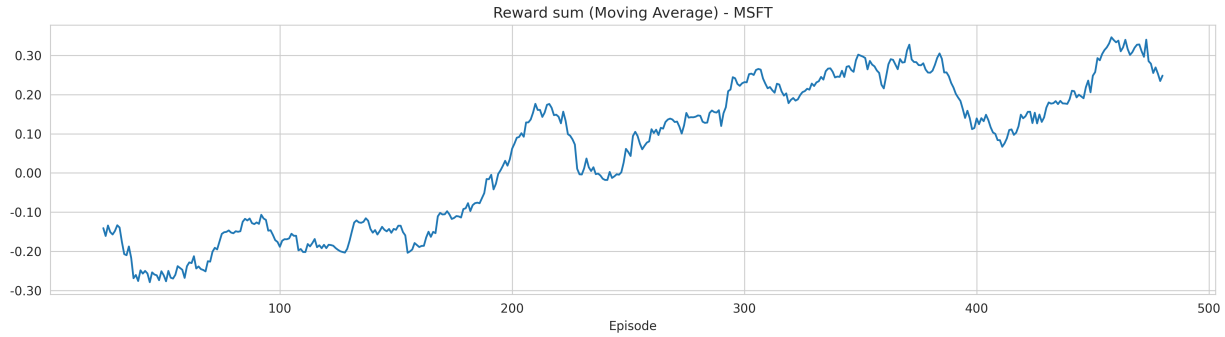


Figura B.199: Media móvil de suma acumulada de recompensas

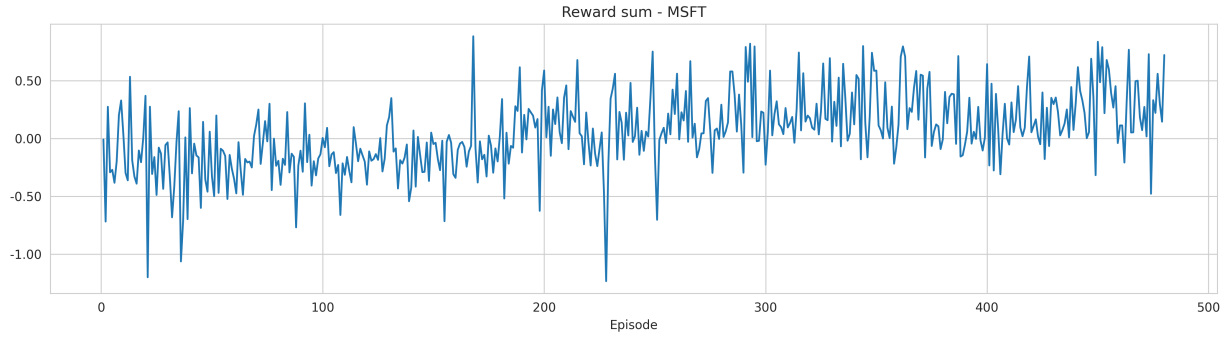


Figura B.200: Suma acumulada de recompensas

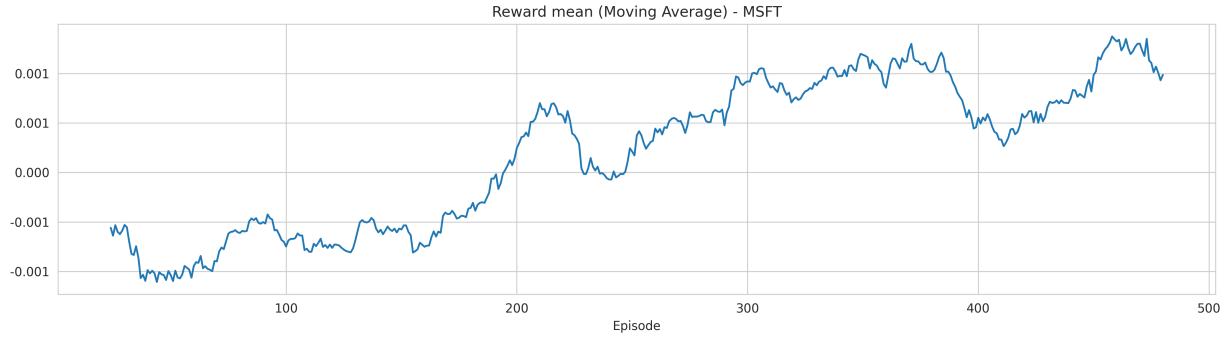


Figura B.201: Media móvil de promedio de recompensas

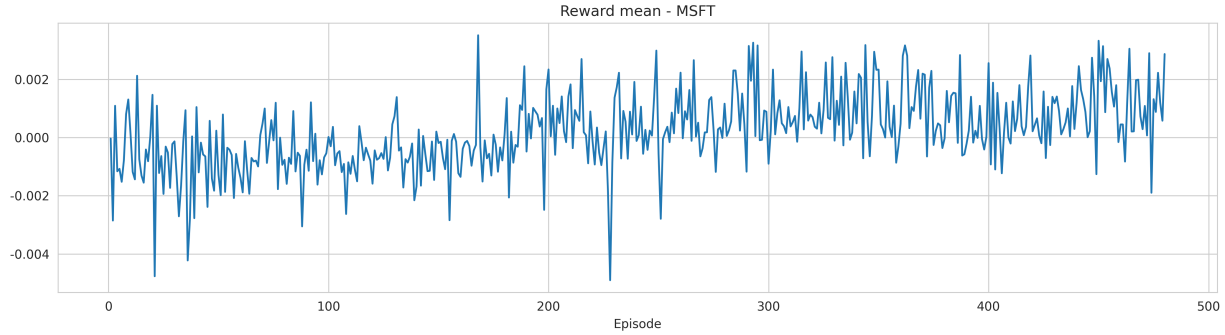


Figura B.202: Promedio de recompensas

Finalmente, en la figura B.203 se muestra la media móvil sobre los últimos 25 episodios de la diferencia entre la mayor y menor recompensa obtenidas durante un episodio. La figura B.204 muestra estos mismos resultados sin una media móvil sobre los episodios.

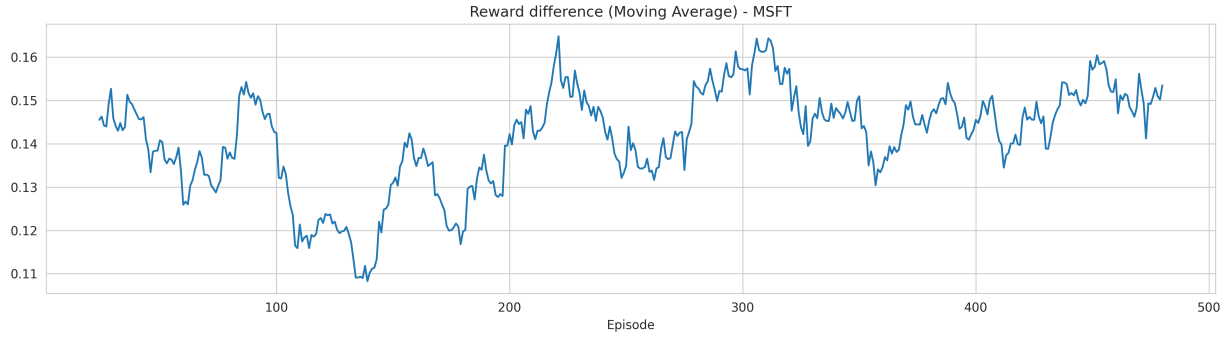


Figura B.203: Media móvil de diferencia de recompensa

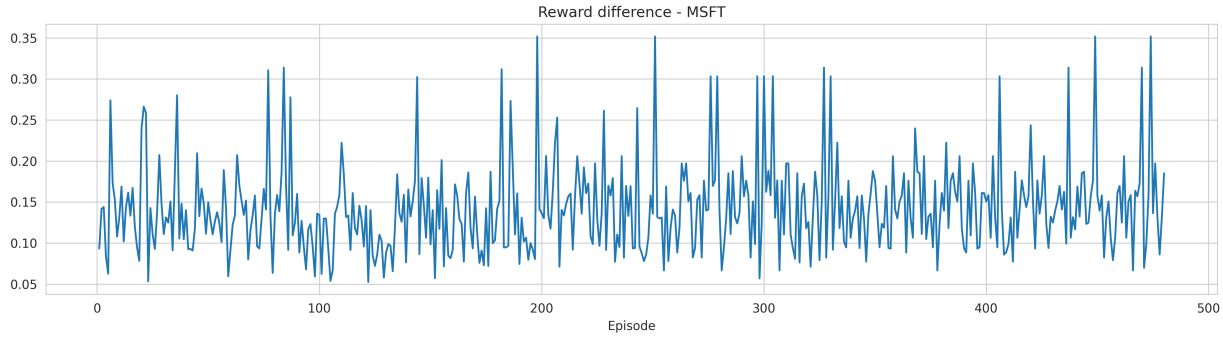


Figura B.204: Diferencia de recompensa

B.3.2.3. Amazon Inc (AMZN)

La figura B.205 muestra la media móvil sobre los últimos 25 episodios de la suma acumulada de recompensas obtenidas durante un episodio para los 480 periodos de entrenamiento. La figura B.206 muestra estos mismos resultados sin una media móvil sobre los episodios. De manera similar, en las figuras B.207 y B.208 se muestra, bajo esta misma configuración, el promedio de recompensas obtenidas durante un episodio.

Figura B.205: Media móvil de suma acumulada de recompensas

Figura B.206: Suma acumulada de recompensas

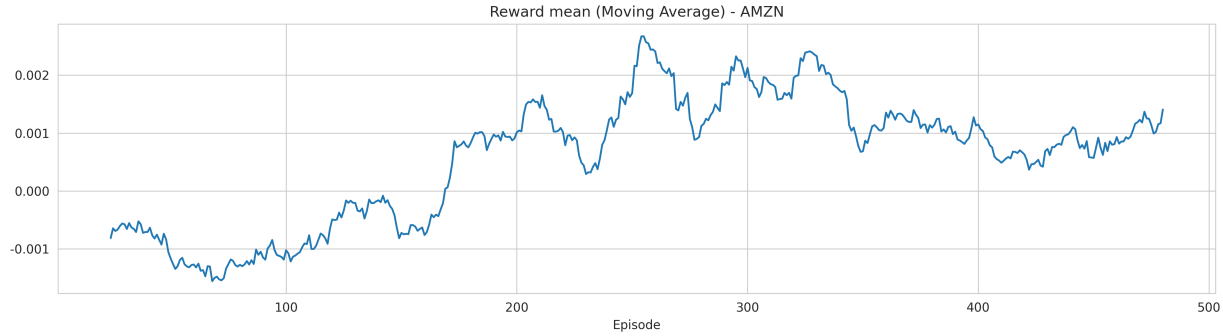


Figura B.207: Media móvil de promedio de recompensas

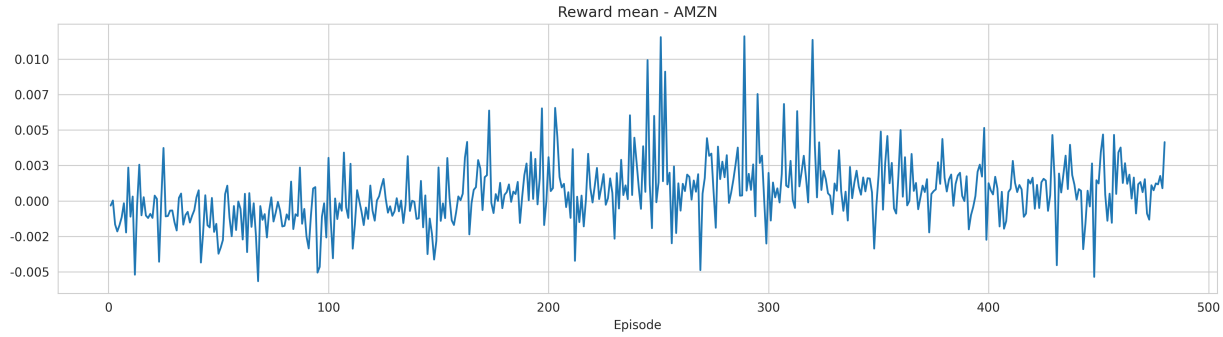


Figura B.208: Promedio de recompensas

Finalmente, en la figura B.209 se muestra la media móvil sobre los últimos 25 episodios de la diferencia entre la mayor y menor recompensa obtenidas durante un episodio. La figura B.210 muestra estos mismos resultados sin una media móvil sobre los episodios.

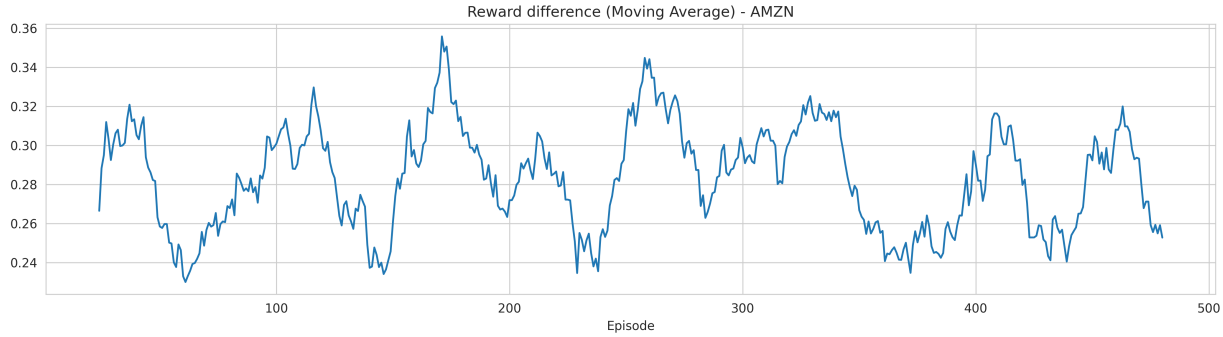


Figura B.209: Media móvil de diferencia de recompensa

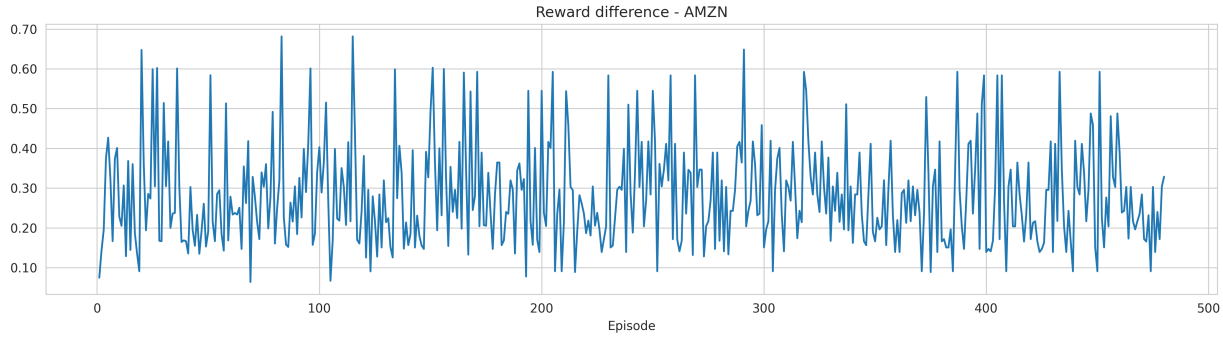


Figura B.210: Diferencia de recompensa

B.3.2.4. Pepsico Inc (PEP)

La figura B.211 muestra la media móvil sobre los últimos 25 episodios de la suma acumulada de recompensas obtenidas durante un episodio para los 480 periodos de entrenamiento. La figura B.212 muestra estos mismos resultados sin una media móvil sobre los episodios. De manera similar, en las figuras B.213 y B.214 se muestra, bajo esta misma configuración, el promedio de recompensas obtenidas durante un episodio.

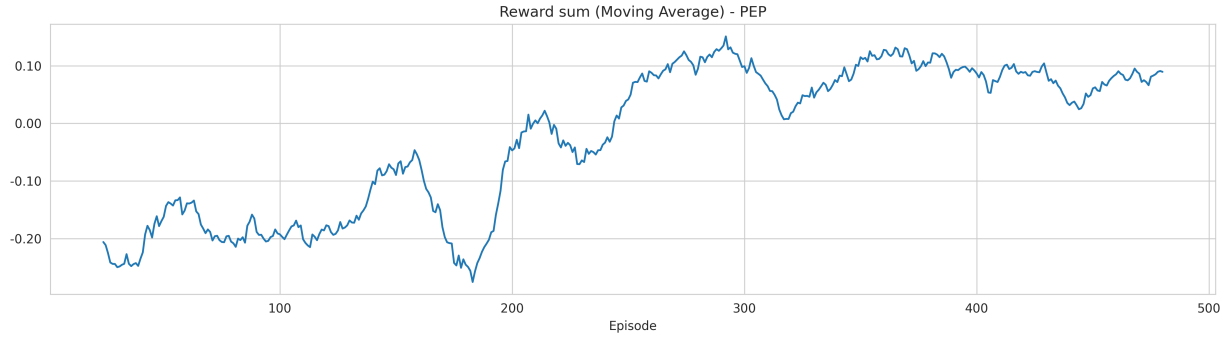


Figura B.211: Media móvil de suma acumulada de recompensas

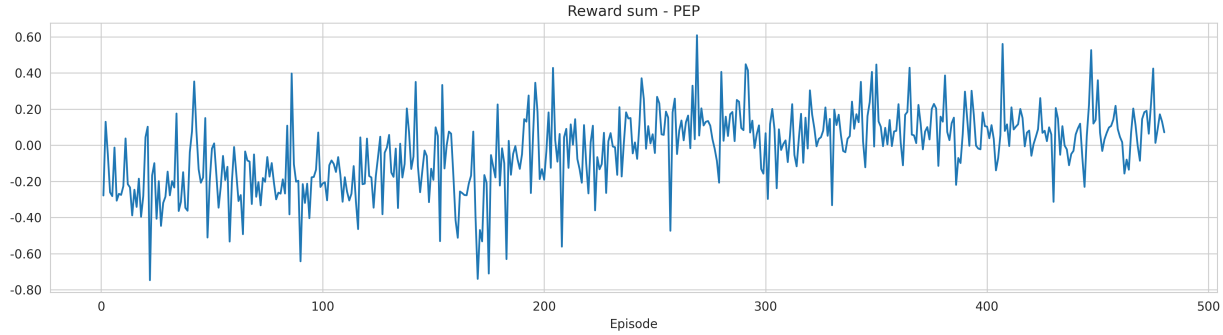


Figura B.212: Suma acumulada de recompensas

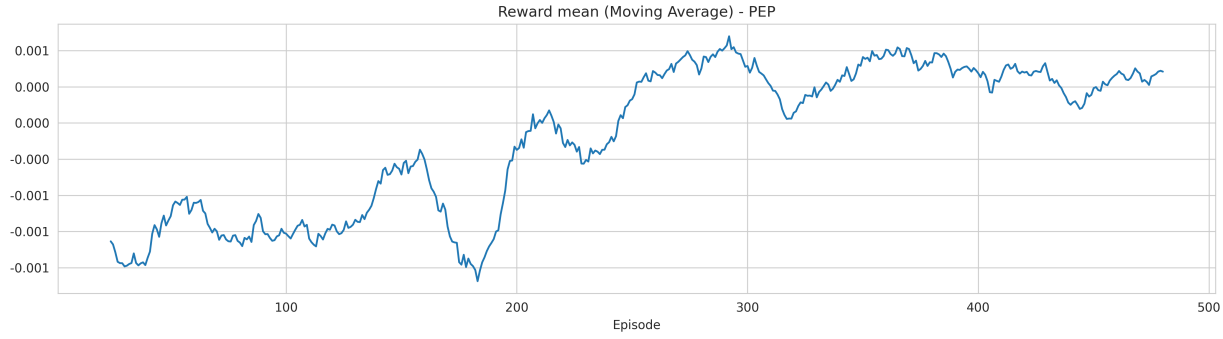


Figura B.213: Media móvil de promedio de recompensas

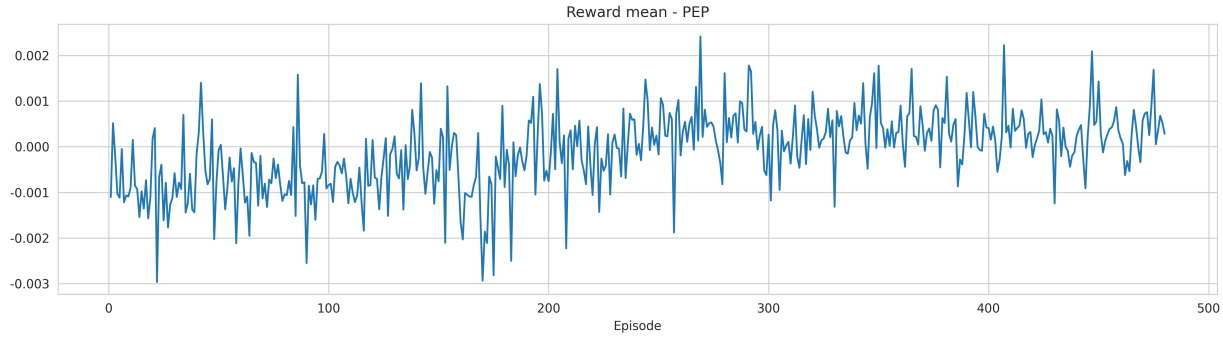


Figura B.214: Promedio de recompensas

Finalmente, en la figura B.215 se muestra la media móvil sobre los últimos 25 episodios de la diferencia entre la mayor y menor recompensa obtenidas durante un episodio. La figura B.216 muestra estos mismos resultados sin una media móvil sobre los episodios.

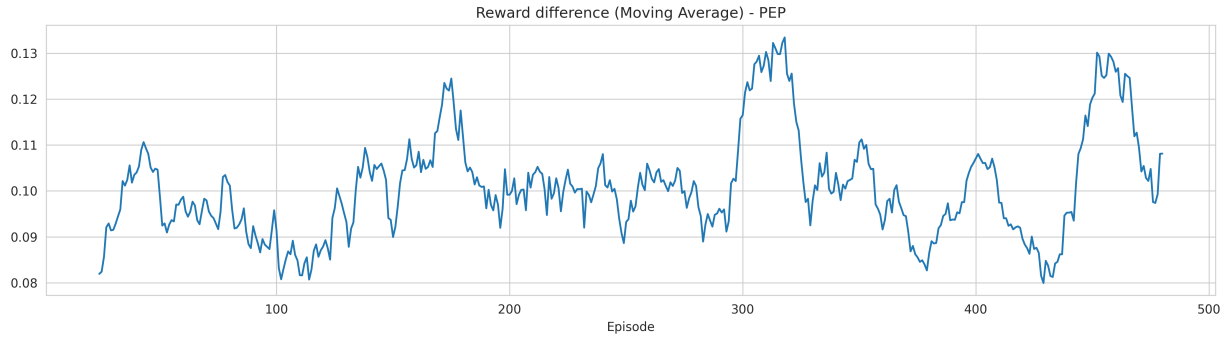


Figura B.215: Media móvil de diferencia de recompensa

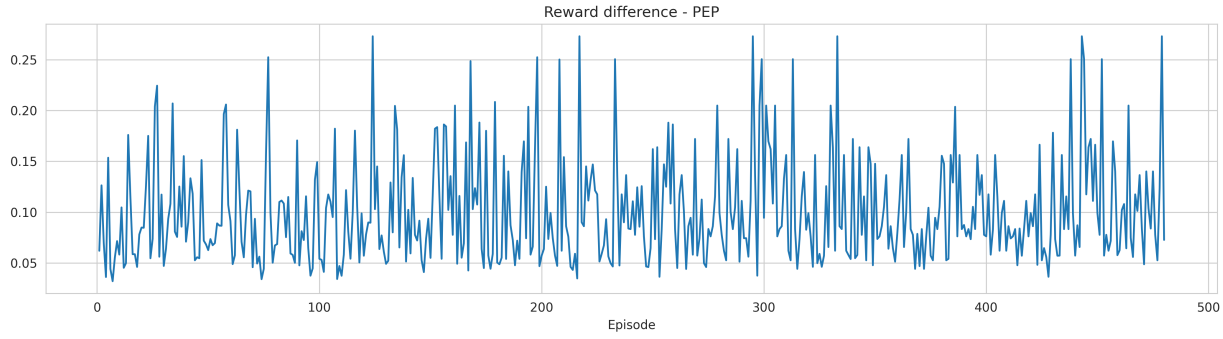


Figura B.216: Diferencia de recompensa

B.3.3. Costos

B.3.3.1. Apple Inc (AAPL)

La figura B.217 muestra la media móvil sobre los último 25 episodios de la suma acumulada de costos obtenidos durante un episodio para los 480 periodos de entrenamiento. La figura B.218 muestra estos mismos resultados sin una media móvil sobre los episodios. De manera similar, en las figuras B.219 y B.220 se muestra, bajo esta misma configuración, el promedio de costos obtenidos durante un episodio.

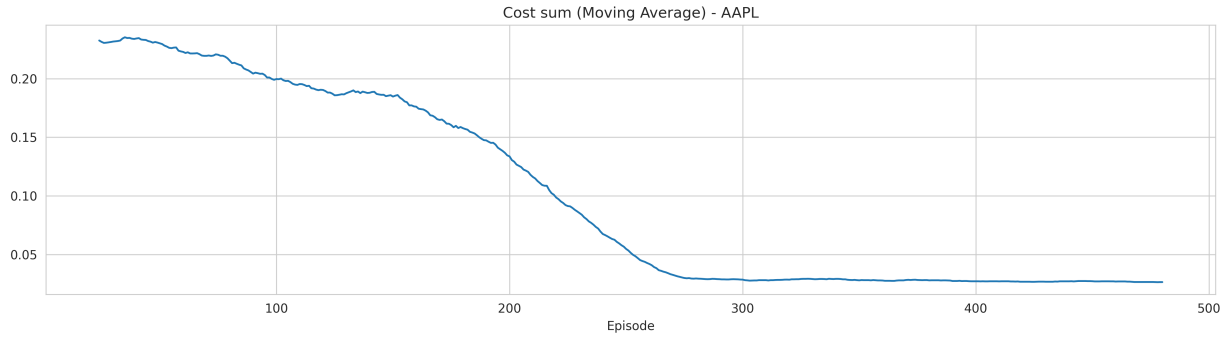


Figura B.217: Media móvil de suma acumulada de costos

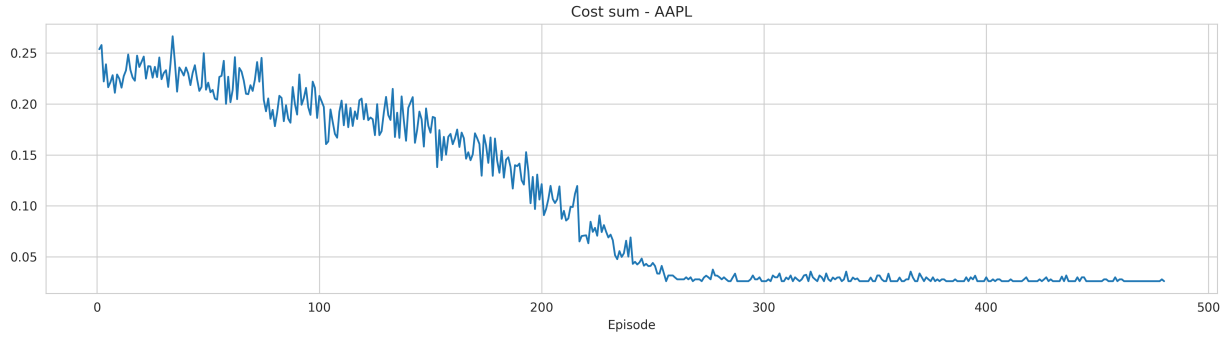


Figura B.218: Suma acumulada de costos

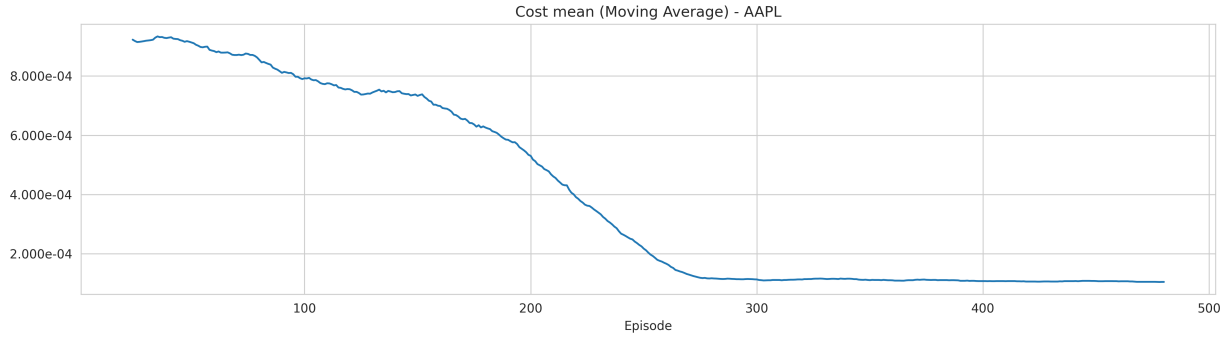


Figura B.219: Media móvil de promedio de costos

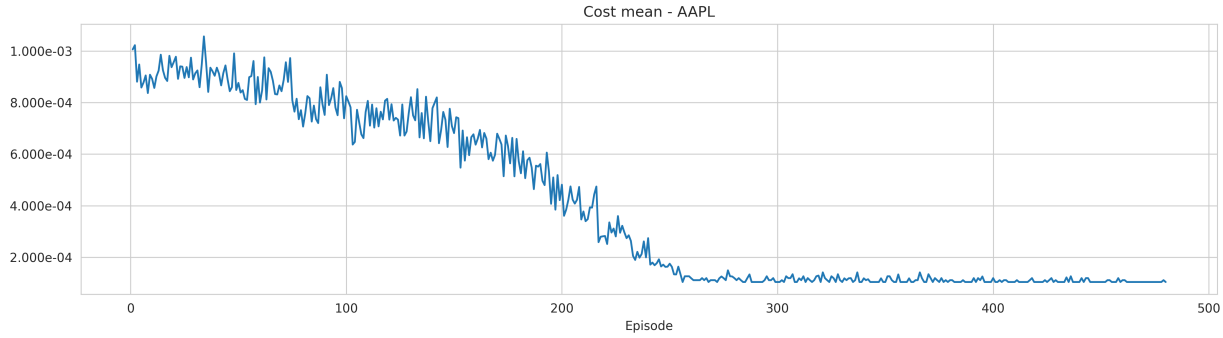


Figura B.220: Promedio de costos

B.3.3.2. Microsoft (MSFT)

La figura B.221 muestra la media móvil sobre los último 25 episodios de la suma acumulada de costos obtenidos durante un episodio para los 480 periodos de entrenamiento. La figura B.222 muestra estos mismos resultados sin una media móvil sobre los episodios. De manera similar, en las figuras B.223 y B.224 se muestra, bajo esta misma configuración, el promedio de costos obtenidos durante un episodio.

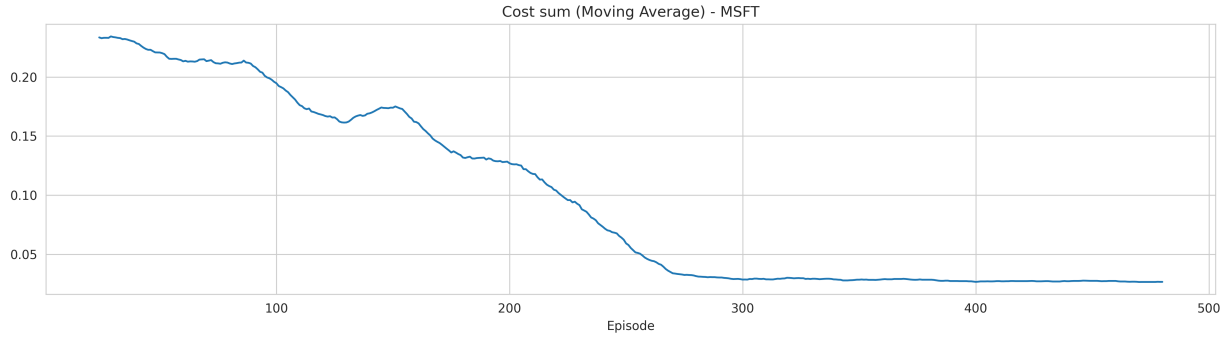


Figura B.221: Media móvil de suma acumulada de costos

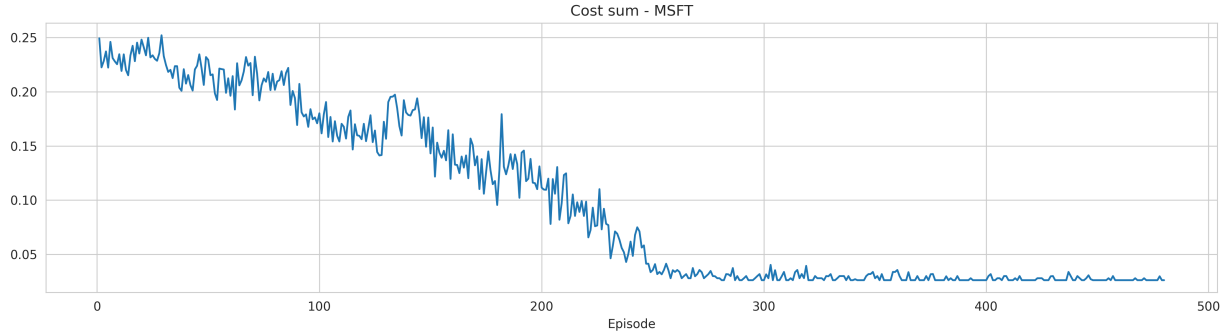


Figura B.222: Suma acumulada de costos

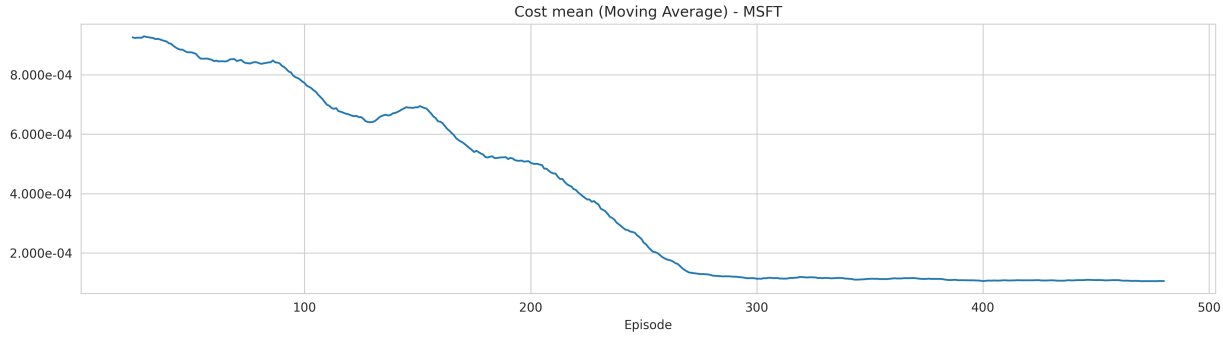


Figura B.223: Media móvil de promedio de costos

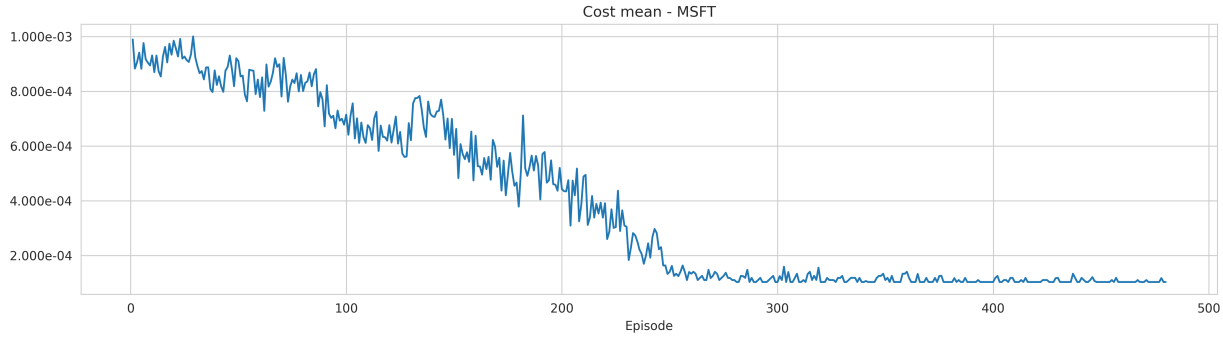


Figura B.224: Promedio de costos

B.3.3.3. Amazon Inc (AMZN)

La figura B.225 muestra la media móvil sobre los último 25 episodios de la suma acumulada de costos obtenidos durante un episodio para los 480 periodos de entrenamiento. La figura B.226 muestra estos mismos resultados sin una media móvil sobre los episodios. De manera similar, en las figuras B.227 y B.228 se muestra, bajo esta misma configuración, el promedio de costos obtenidos durante un episodio.

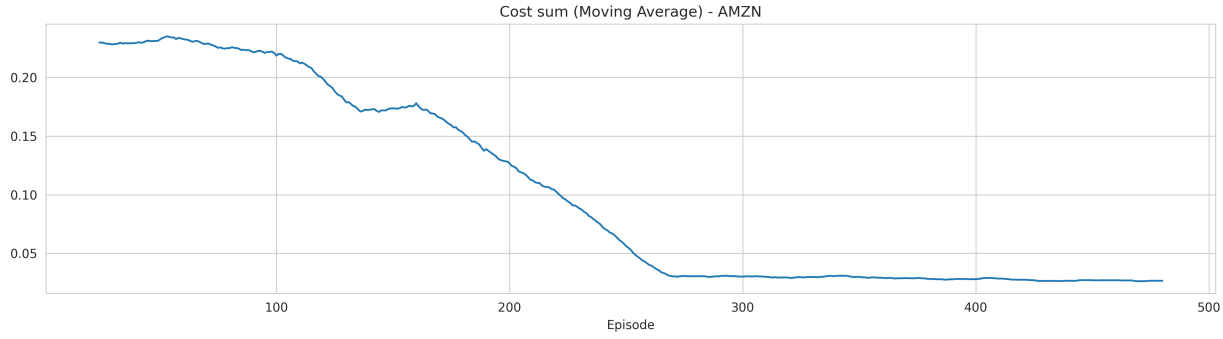


Figura B.225: Media móvil de suma acumulada de costos

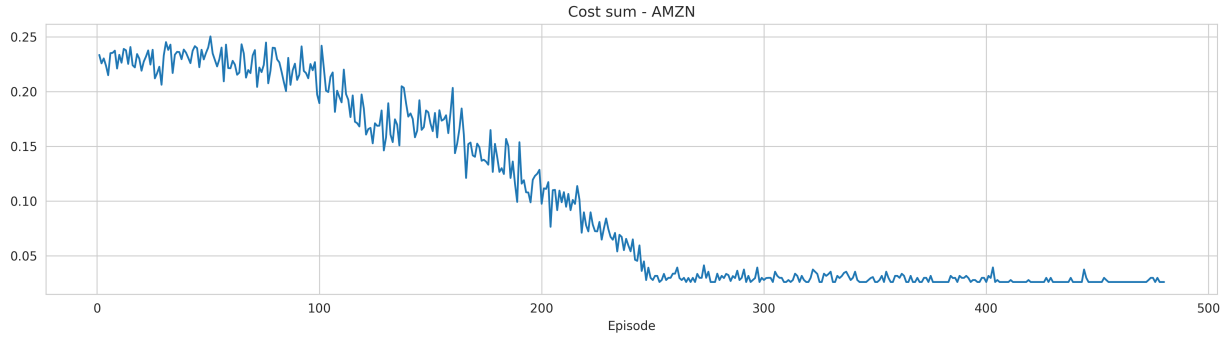


Figura B.226: Suma acumulada de costos

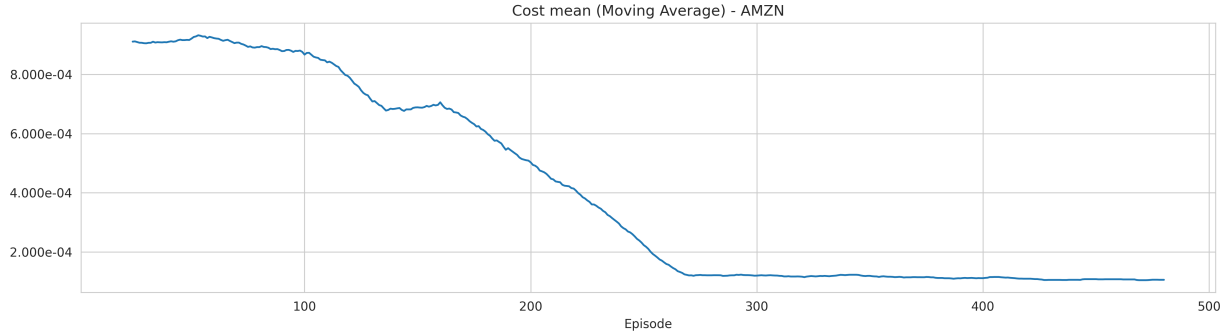


Figura B.227: Media móvil de promedio de costos

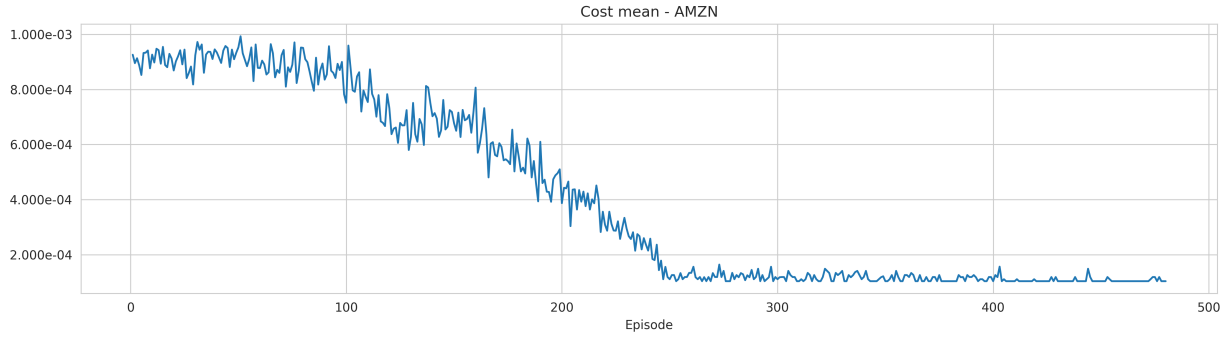


Figura B.228: Promedio de costos

B.3.3.4. Pepsico Inc (PEP)

La figura B.229 muestra la media móvil sobre los último 25 episodios de la suma acumulada de costos obtenidos durante un episodio para los 480 periodos de entrenamiento. La figura B.230 muestra estos mismos resultados sin una media móvil sobre los episodios. De manera similar, en las figuras B.231 y B.232 se muestra, bajo esta misma configuración, el promedio de costos obtenidos durante un episodio.

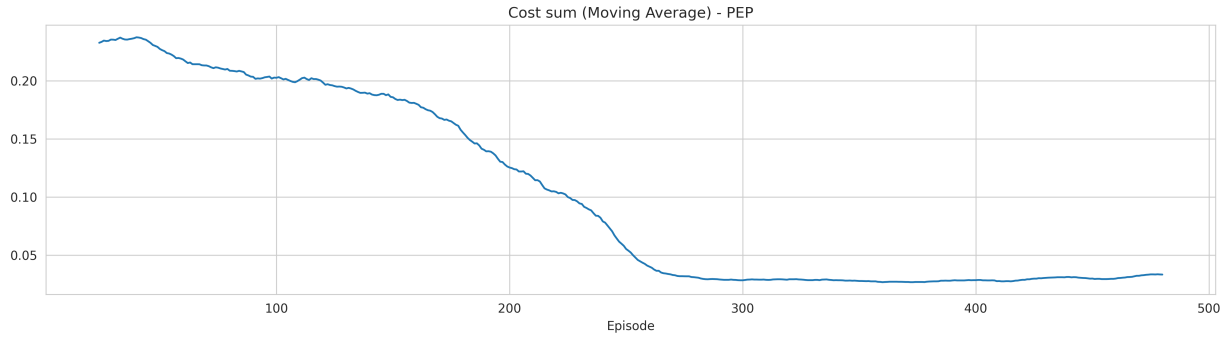


Figura B.229: Media móvil de suma acumulada de costos

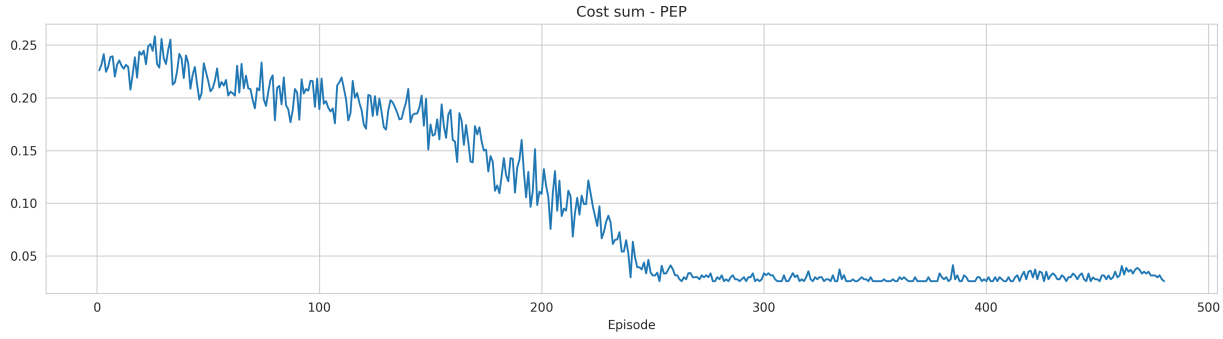


Figura B.230: Suma acumulada de costos

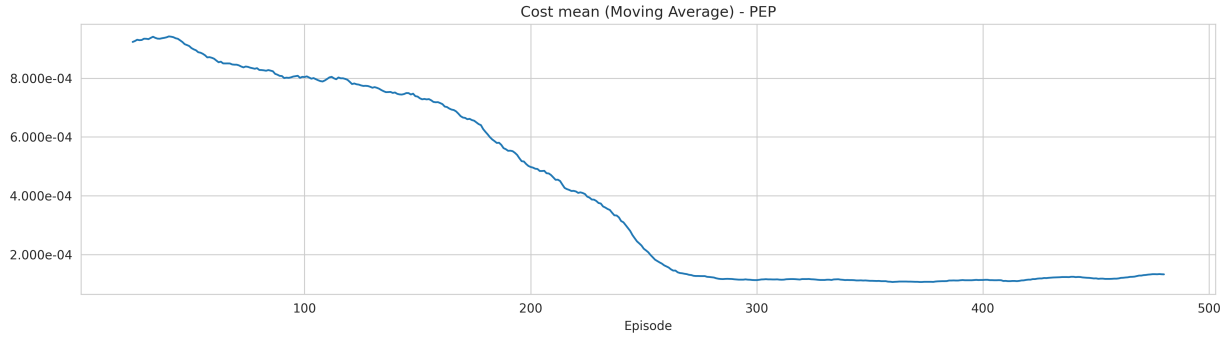


Figura B.231: Media móvil de promedio de costos

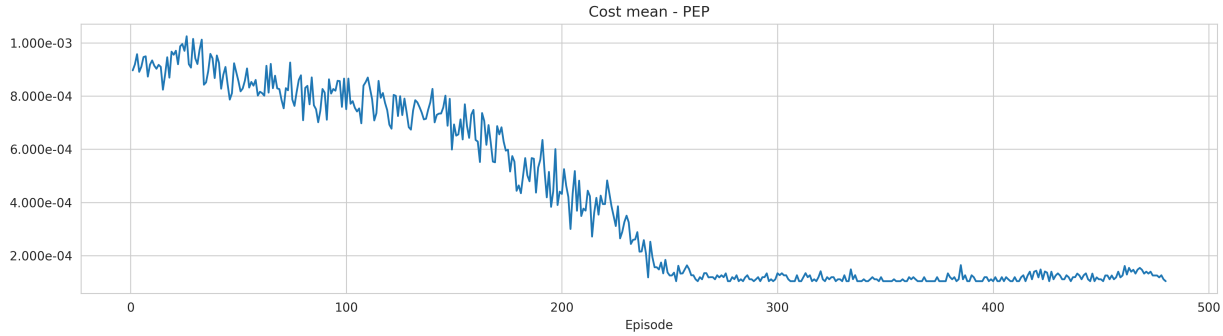


Figura B.232: Promedio de costos

B.3.4. Operaciones de trading

B.3.4.1. Apple Inc (AAPL)

La figura B.233 muestra la media móvil sobre los últimos 25 episodios de la suma acumulada de operaciones de trading durante un episodio para los 480 periodos de entrenamiento. La figura B.234 muestra estos mismos resultados sin una media móvil sobre los episodios. De manera similar, en las figuras B.235 y B.236 se muestra, bajo esta misma configuración, el promedio de operaciones de trading realizadas cada día durante un episodio.

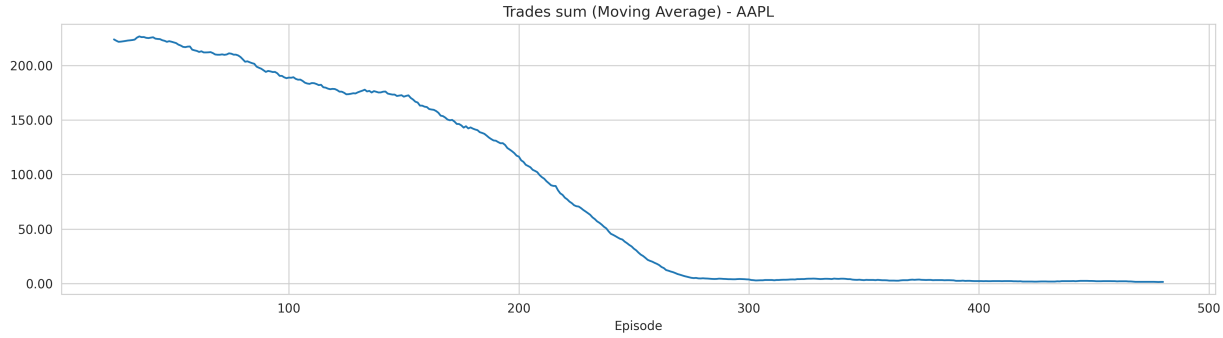


Figura B.233: Media móvil de suma acumulada de operaciones

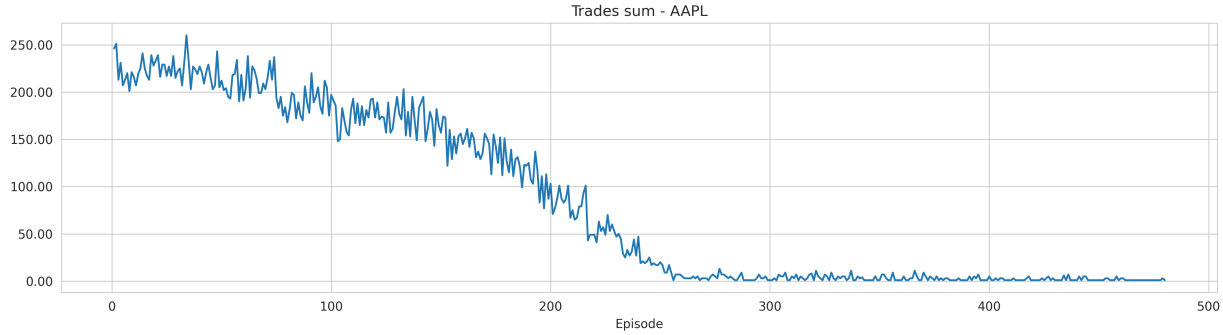


Figura B.234: Suma acumulada de operaciones

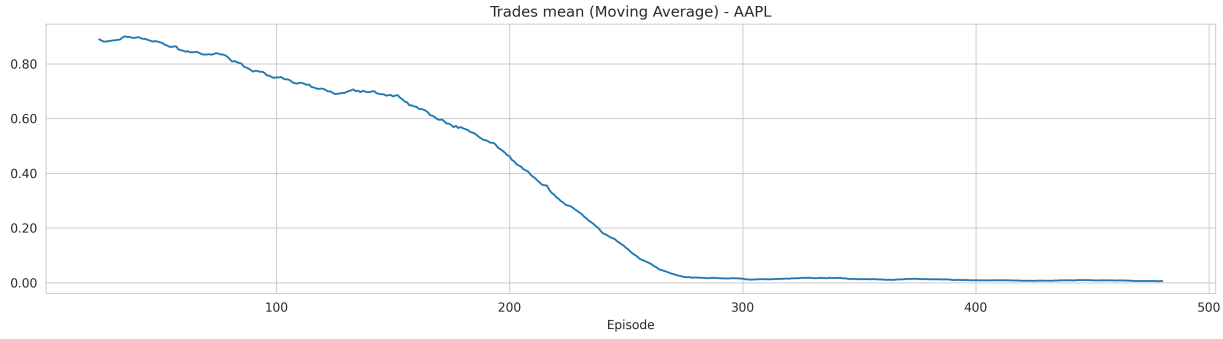


Figura B.235: Media móvil de promedio de operaciones

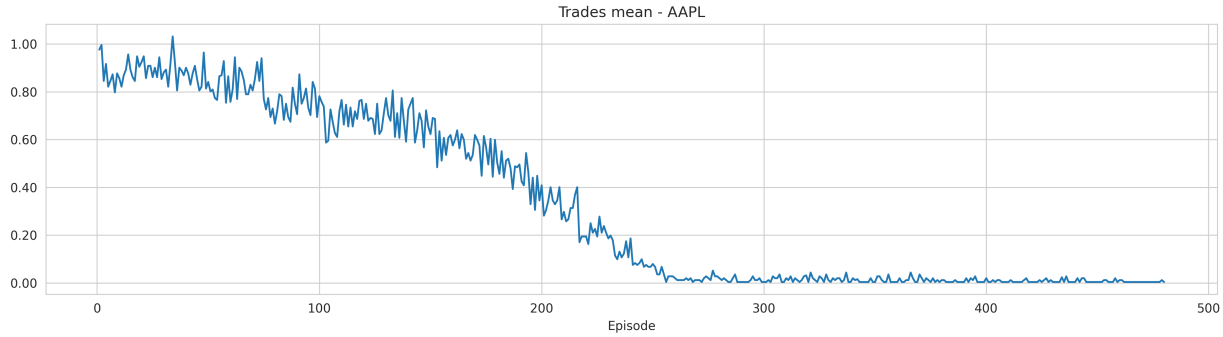


Figura B.236: Promedio de operaciones

B.3.4.2. Microsoft (MSFT)

La figura B.237 muestra la media móvil sobre los últimos 25 episodios de la suma acumulada de operaciones de trading durante un episodio para los 480 periodos de entrenamiento. La figura B.238 muestra estos mismos resultados sin una media móvil sobre los episodios. De manera similar, en las figuras B.239 y B.240 se muestra, bajo esta misma configuración, el promedio de operaciones de trading realizadas cada día durante un episodio.

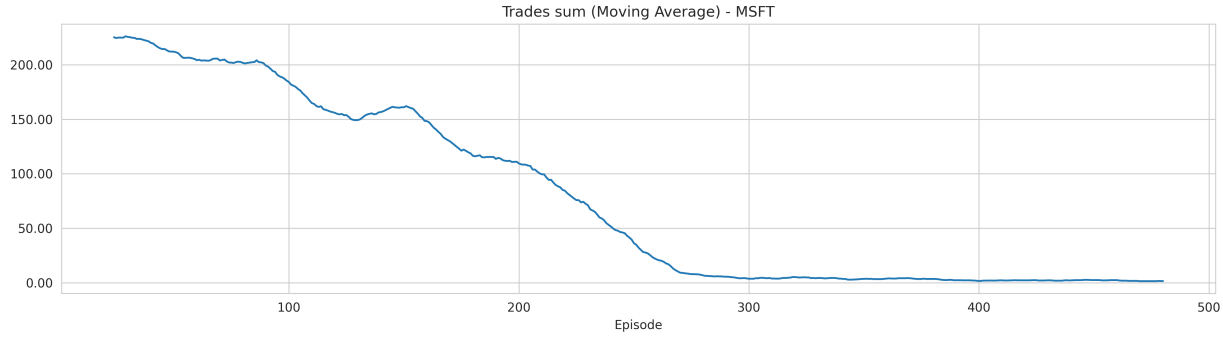


Figura B.237: Media móvil de suma acumulada de operaciones

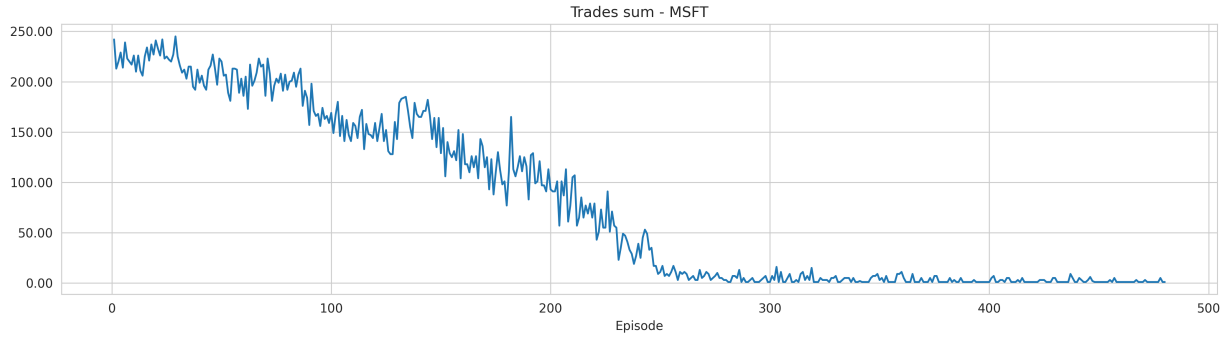


Figura B.238: Suma acumulada de operaciones

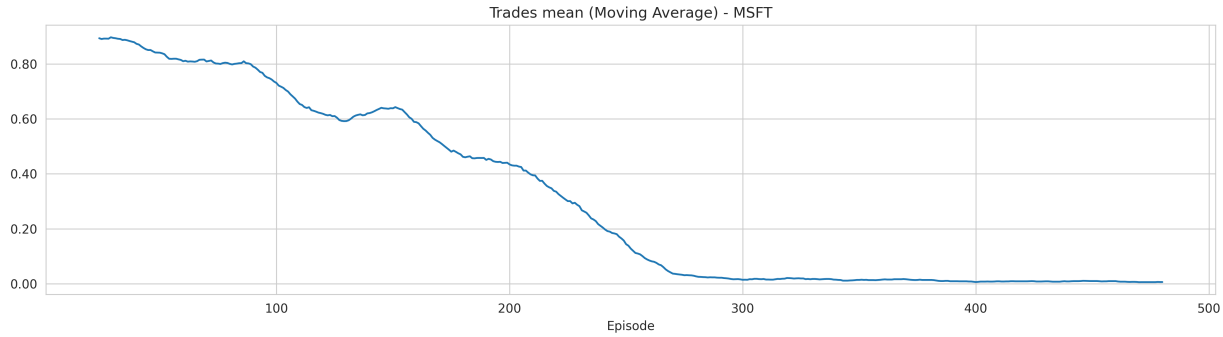


Figura B.239: Media móvil de promedio de operaciones

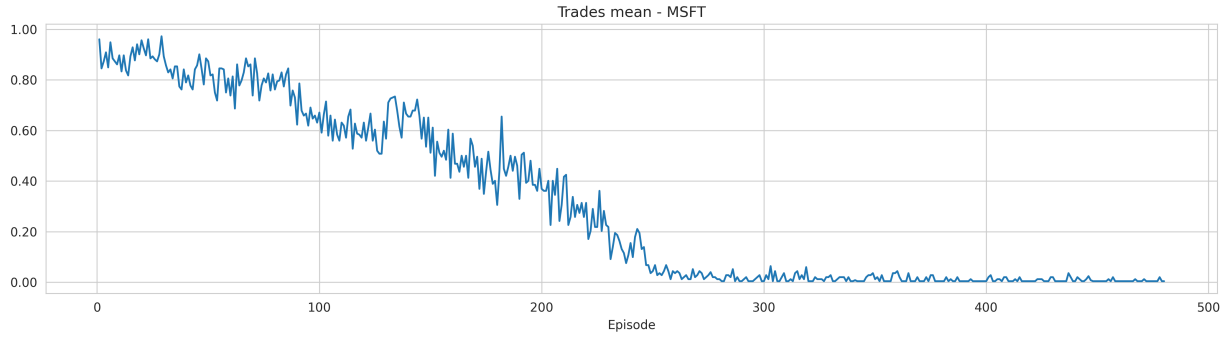


Figura B.240: Promedio de operaciones

B.3.4.3. Amazon Inc (AMZN)

La figura B.241 muestra la media móvil sobre los últimos 25 episodios de la suma acumulada de operaciones de trading durante un episodio para los 480 periodos de entrenamiento. La figura B.242 muestra estos mismos resultados sin una media móvil sobre los episodios. De manera similar, en las figuras B.243 y B.244 se muestra, bajo esta misma configuración, el promedio de operaciones de trading realizadas cada día durante un episodio.

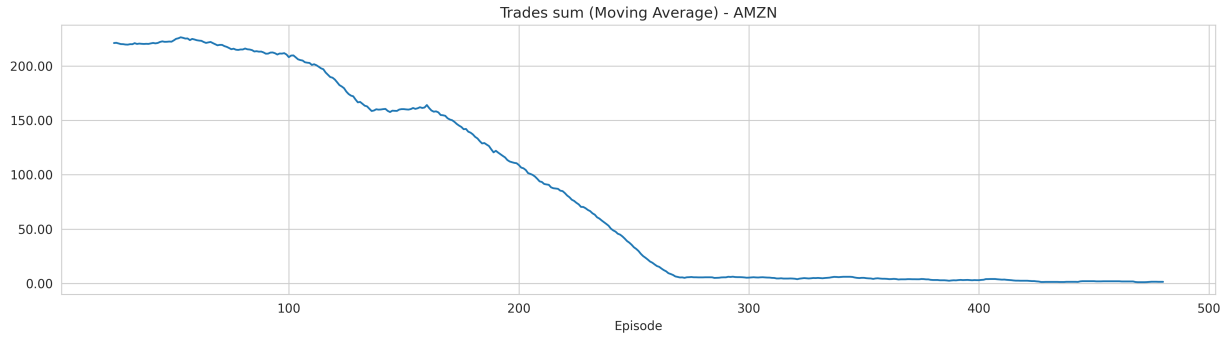


Figura B.241: Media móvil de suma acumulada de operaciones

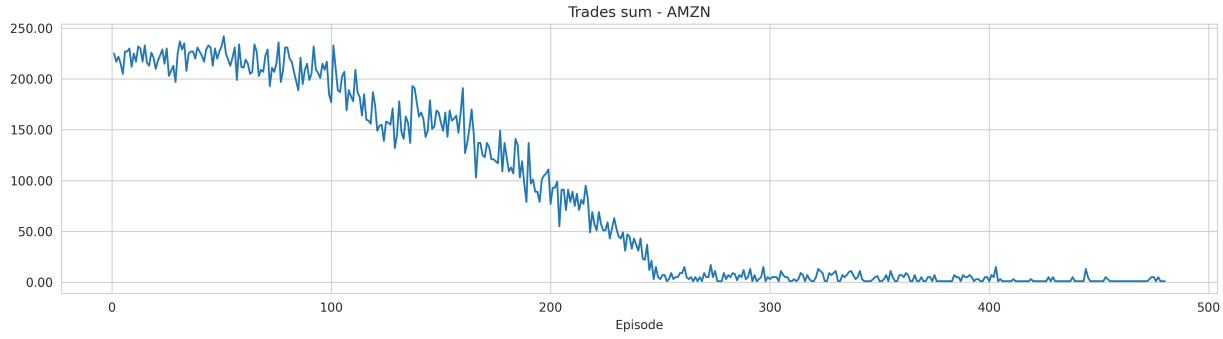


Figura B.242: Suma acumulada de operaciones

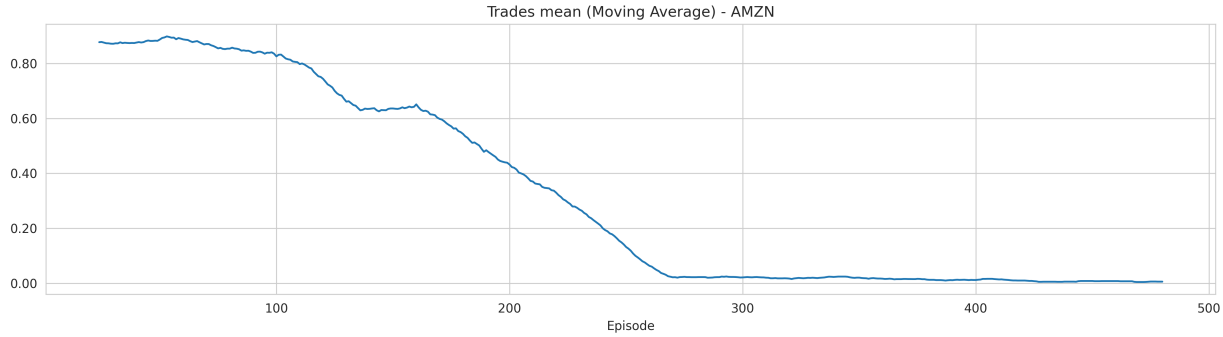


Figura B.243: Media móvil de promedio de operaciones

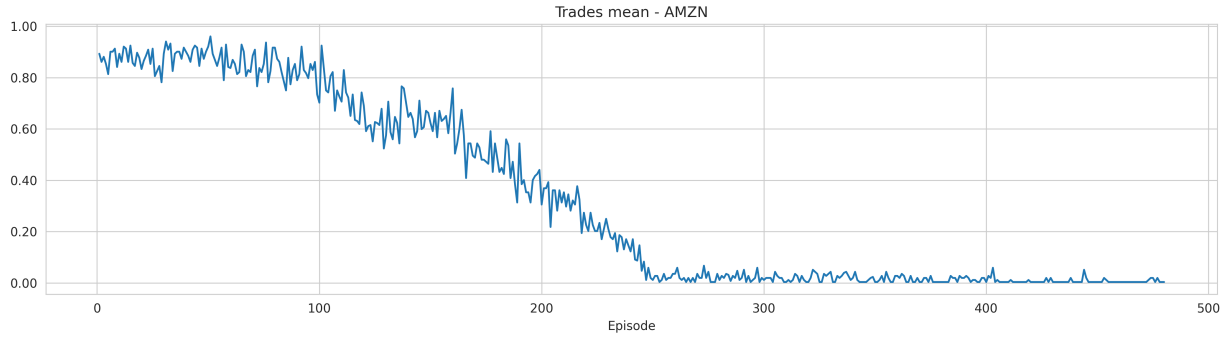


Figura B.244: Promedio de operaciones

B.3.4.4. Pepsico Inc (PEP)

La figura B.245 muestra la media móvil sobre los últimos 25 episodios de la suma acumulada de operaciones de trading durante un episodio para los 480 periodos de entrenamiento. La figura B.246 muestra estos mismos resultados sin una media móvil sobre los episodios. De manera similar, en las figuras B.247 y B.248 se muestra, bajo esta misma configuración, el promedio de operaciones de trading realizadas cada día durante un episodio.

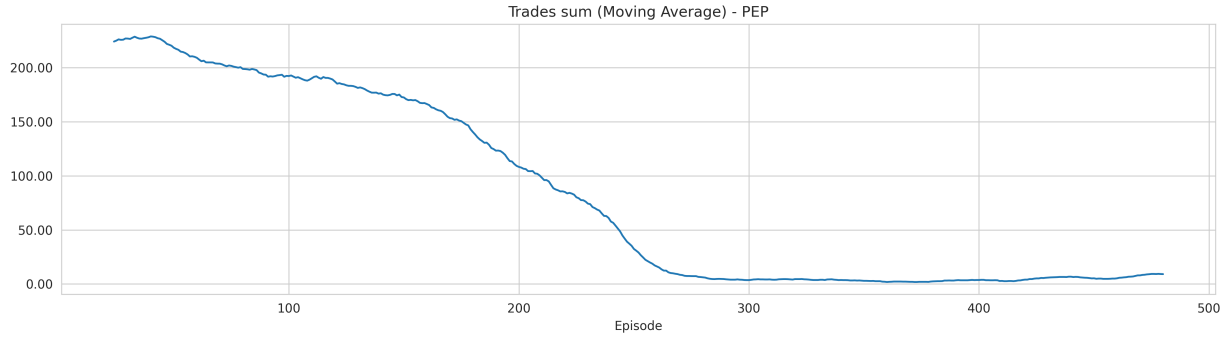


Figura B.245: Media móvil de suma acumulada de operaciones

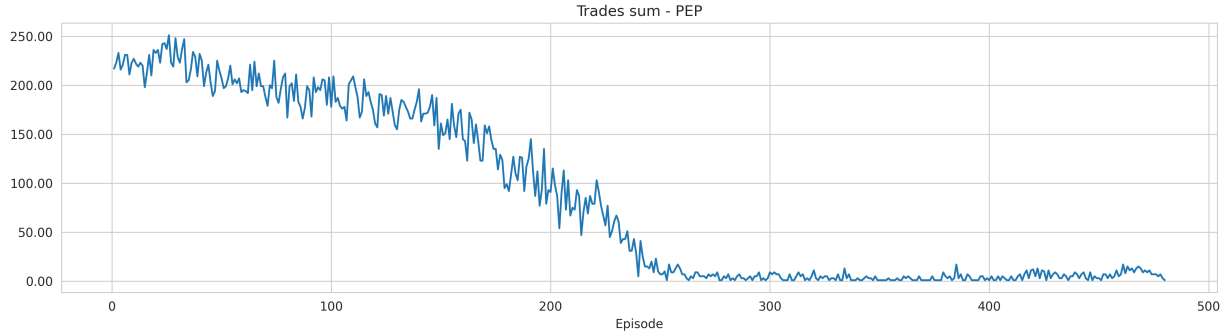


Figura B.246: Suma acumulada de operaciones

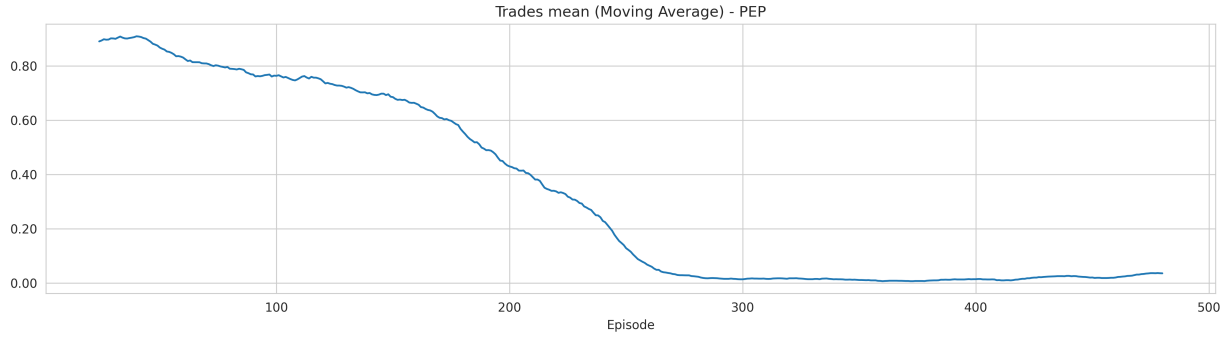


Figura B.247: Media móvil de promedio de operaciones

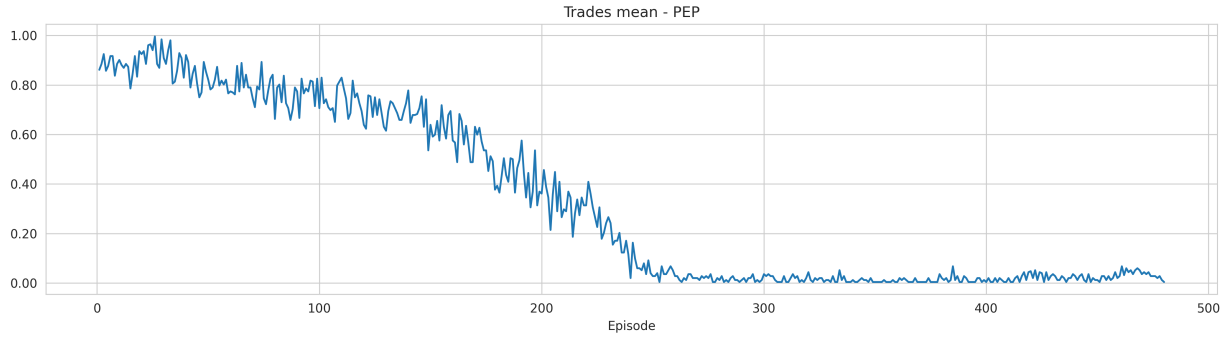


Figura B.248: Promedio de operaciones

B.3.5. Función de pérdida

B.3.5.1. Apple Inc (AAPL)

La figura B.249 muestra la media móvil sobre los últimos 25 episodios del promedio de pérdidas acumuladas para los 480 periodos de entrenamiento.

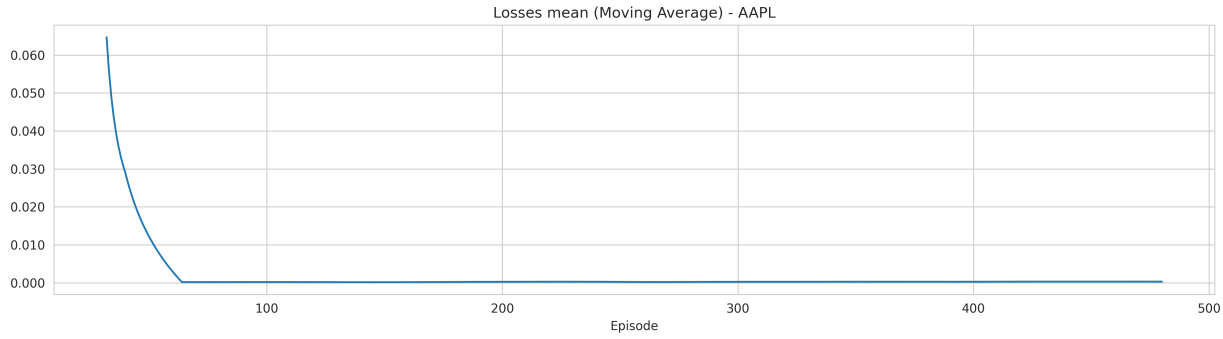


Figura B.249: Media móvil de promedio de pérdidas acumuladas

B.3.5.2. Microsoft (MSFT)

La figura B.250 muestra la media móvil sobre los últimos 25 episodios del promedio de pérdidas acumuladas para los 480 periodos de entrenamiento.

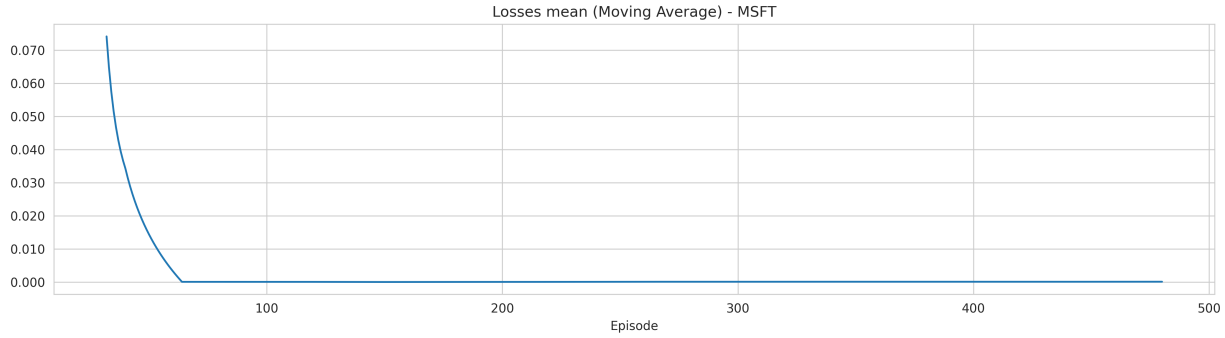


Figura B.250: Media móvil de promedio de pérdidas acumuladas

B.3.5.3. Amazon Inc (AMZN)

La figura B.251 muestra la media móvil sobre los últimos 25 episodios del promedio de pérdidas acumuladas para los 480 periodos de entrenamiento.

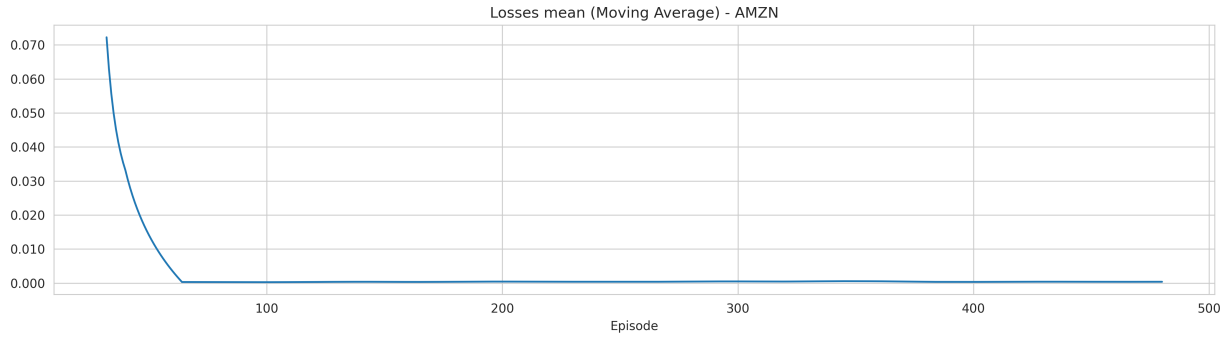


Figura B.251: Media móvil de promedio de pérdidas acumuladas

B.3.5.4. Pepsico Inc (PEP)

La figura B.252 muestra la media móvil sobre los últimos 25 episodios del promedio de pérdidas acumuladas para los 480 periodos de entrenamiento.

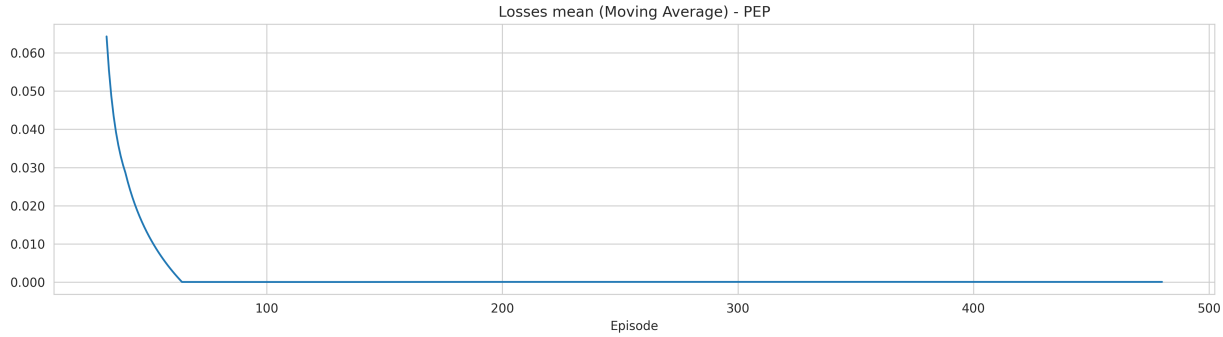


Figura B.252: Media móvil de promedio de pérdidas acumuladas