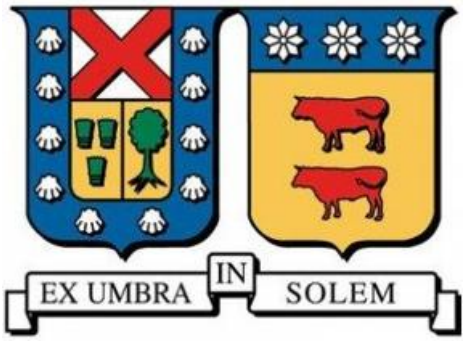


UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA

DEPARTAMENTO DE INDUSTRIAS

ESTRATEGIA DE INVERSIÓN MEDIANTE LA COMBINACIÓN DE REDES NEURONALES Y EL ALGORITMO DE ACTOR-CRÍTICO CON VENTAJA (A2C).

MEMORIA PARA OPTAR AL TÍTULO DE INGENIERIA CIVIL INDUSTRIAL.

AUTOR

RODOLFO ALARCÓN

PROFESOR GUÍA

WERNER KRISJANPOLLER

PROFESOR CORREFERENTE

DIEGO CANESSA

VALPARAÍSO, 11 DE DICIEMBRE DEL 2024

Índice

1. Problema de investigación

En el último tiempo, Deep Learning ha sido implementado para distintos usos en las industrias a nivel mundial, desde el procesamiento de imágenes, lenguaje y audio, hasta en la realización de pronósticos. Es por ello por lo que su utilización en el contexto de los Mercados de Capitales es de vital importancia para así estar a la vanguardia de las tecnologías actuales, por lo que su manejo y desarrollo no puede ser inverosímil.

Deep Learning (DL) se califica como una técnica avanzada de Machine Learning basada en el uso de Redes Neuronales (Neuronal Networks – NN), siendo muchas veces comparado con modelos de Machine Learning tradicionales como 'Support Vector Machine' y 'K-Nearest Vector', en donde DL posee la ventaja del aprendizaje no supervisado y en poder ser usado para entrenamiento robusto en Big Data, lo que ha generado su rápida implementación en medicina, neurociencia, física, astronomía y finanzas, demostrando excelentes resultados al ser aplicados para la gestión de inversiones dado a su capacidad de encontrar patrones en series de tiempo no lineales de distintas acciones en los Mercados de Capitales (Jian Huang, 2020).

Hoy en día DL ha expandido su aplicación llegando a ser implementado para algoritmos de Aprendizaje por refuerzo, mejor conocido como Reinforcement Learning (RL), los cuales combinados con las Redes Neuronales los convierten en una metodología efectiva en el contexto de inversiones, impulsado la necesidad de los inversores en implementar dichos modelos en sus estrategias de inversión buscando mejores retornos en sus portafolios de inversión mejorando la mitigación del riesgo (Shuo Sun, 2023).

Diversos estudios señalan la eficacia de la incorporación de algoritmos de RL por sí solos incrementa los retornos en portafolios de inversión, así lo deja claro ejemplos en la

incorporación de modelos como: Deep Deterministic Policy Gradient (DDPG), Proximal Policy Optimization (PPO), Soft Actor-Critic (SAC), Twin Delayed Deep Deterministic Policy Gradient (TD3), entre otros, en donde también se señala que combinando estos algoritmos con el Modelo de Markowitz pueden incrementar aún más la eficiencia de los portafolios (Santos, 2023). Por otro lado, conocer el contexto de los mercados mediante indicadores Macroeconómicas es fundamental, por ello se recomienda que el control y la decisión final recaiga siempre sobre el inversor de dicho portafolio al instante de tomar la decisión final (Santos, 2023).

Para la gestión de portafolios de inversión, la combinación de los modelos de DL como Redes Neuronales más el modelo de RL como el algoritmo de Actor-Crítico con Ventaja (A2C), ¿Es posible incrementar el retorno de los portafolios implementando dichos modelos?, ¿Cómo se podría administrar un portafolio de inversión con dichos modelos ante los Mercados de Capitales con alta volatilidad?, ¿Cómo funcionan dichos modelos y que variables es posible controlar en ellas para obtener mejores resultados de pronóstico?

2. Objetivos

2.1 Objetivo General

Combinar pronósticos hechas por Redes Neuronales con el Algoritmo de Actor-Crítico con Ventaja (A2C) en portafolios de inversión para la toma de decisiones óptimas de inversión en la compra y venta de acciones en el corto y largo plazo.

2.2 Objetivos Específicos

Entender el cómo funcionan los Mercados de Capitales y del cómo son regulados para brindar seguridad de Inversión.

Identificar las propiedades de las Redes Neuronales (NN), desde su construcción hasta la modificación de sus parámetros.

Analizar la efectividad del optimizador Adam para el entrenamiento de las Redes Neuronales (NN) con los datos entrenados y así obtener resultados cercanos a la realidad comparándolos con los datos de testeo pertenecientes al horizonte evaluado.

Obtener el índice de Sharpe para medir la eficacia de los resultados en la estrategia de inversión propuesta y compararlo con el método tradicional de inversión Buy & Hold.

Concluir sobre la fiabilidad de la implementación de las Redes Neuronales (NN) más el Algoritmo de Actor Crítico con Ventaja (A2C) para la obtención de retornos acumulables positivos a lo largo del tiempo.

3. Marco Teórico

3.1 Escenario Mundial con la Inteligencia Artificial Generativa.

En la actualidad la Inteligencia Artificial se ha convertido en el principal actor de la economía moderna dado a su capacidad de reemplazar el trabajo humano, pero a su vez, dicho escenario se convierte en un fiel reflejo de cómo será el impulso económico para los próximos años. Para tal progreso las grandes compañías relacionadas a esta industria han invertido grandes sumas en inversión para el desarrollo de Software, Capital Humano, Equipamiento y Fábricas que puedan complementar su producción (The Economist, 2024).

En otros sectores, sin embargo, los planes son más modestos, ya que excluyendo a las empresas que impulsan la revolución de la inteligencia artificial, como Microsoft y Nvidia, las empresas del S&P 500 prevén un aumento de las inversiones de alrededor del 2,5% en 2024, es decir, una cantidad acorde a las proyecciones de inflación (The Economist, 2024).

En el conjunto de la economía, la situación es aún más sombría. Un «rastreador» del Capex estadounidense elaborado por el banco Goldman Sachs ofrece una imagen de los desembolsos de las empresas, así como una pista de las intenciones futuras. Actualmente está cayendo un 4% interanual, opinión no muy diferente a los pronósticos por parte de JP Morgan (The Economist, 2024).

El empeoramiento de las perspectivas económicas en Europa dificulta las cosas. Las intenciones de inversión de las empresas de servicios de la Unión Europea son menos de la mitad de las ambiciosas que a principios de 2022. Las empresas británicas prevén aumentar su inversión en capital fijo en tan solo un 3% durante el próximo año, frente al 10% que

preveían a principios de 2022 (The Economist, 2024).

Estas tendencias sugieren que las grandes empresas tecnológicas les encantan el desarrollo e inserción de la IA, pero también se prevé que tendrán dificultades para encontrar clientes que requieran de productos o servicios relacionados con esta tecnología y que estén dispuestos a invertir docenas de miles de millones de dólares dado a la complejidad en su funcionamiento. No sería la primera vez en la historia reciente que los tecnólogos sobrestiman la demanda de nuevas innovaciones, siendo un claro ejemplo de esto los resultados demostrados por empresas relacionadas con el metaverso (The Economist, 2024).

3.2 Mercados de Capitales.

El término ‘mercado de capitales’ es un término amplio que se utiliza para describir los espacios presenciales y digitales en donde diversas entidades negocian tipos de instrumentos financieros. Estos espacios pueden incluir el mercado de valores, el mercado de bonos y el mercado de divisas (Forex). La mayoría de los mercados se concentran en grandes centros financieros como Nueva York, Londres, Singapur y Hong Kong (Hayes, 2024).

Los mercados de capitales se componen de proveedores y usuarios de fondos. Entre los proveedores figuran los hogares a través de las cuentas de ahorro y los productos que mantienen en los bancos, así como instituciones de fondos de pensiones y jubilación, compañías de seguros de vida, fundaciones benéficas y empresas no financieras que generan excedentes de tesorería. Por otro lado, usuarios de los fondos distribuidos en los mercados de capitales figuran los compradores de viviendas o vehículos de motor, empresas no financieras y los gobiernos que financian inversiones en infraestructuras y gastos de funcionamiento (Hayes, 2024).

Los mercados de capitales se utilizan principalmente para vender productos financieros como ‘acciones’ y ‘títulos de deuda’. Las acciones son títulos que representan participaciones en la propiedad de una empresa. Los títulos de deuda, como los bonos, son pagarés que devengan intereses (Hayes, 2024).

3.2.1 Mercados Primarios y Secundarios

Los mercados de capitales se componen de mercados primarios y secundarios. Una empresa participa en el mercado primario de capitales cuando vende públicamente nuevas

acciones u obligaciones por primera vez, como en una oferta pública inicial (OPI). Este mercado se denomina a veces mercado de nuevas emisiones. La empresa que ofrece los valores contrata a una empresa de suscripción cuando los inversores compran valores en el mercado primario de capitales. La empresa lo revisa y elabora un folleto en el que se describen el precio y otros detalles de los valores que se van a emitir (Clara Fabiola, 2020).

Todas las emisiones en el mercado primario están sujetas a una regulación estricta. Las empresas en EEUU deben presentar declaraciones ante la Comisión del Mercado de Valores de EEUU (SEC) y otras agencias de valores, en donde deben esperar a que sus declaraciones sean aprobadas antes de poder salir a bolsa (U.S. Securities and Exchange Commision , 2005).

A menudo, los pequeños inversores no pueden comprar valores en el mercado primario porque la empresa y sus banqueros de inversión quieren vender todos los valores disponibles en poco tiempo para alcanzar el volumen requerido. Deben centrarse en la comercialización de la venta a grandes inversores que puedan comprar más valores a la vez (Hayes, 2024).

La comercialización de la venta a los inversores puede incluir a menudo un ‘roadshow’ o un ‘dog and pony show’ en el que los banqueros de inversión y la dirección de la empresa viajan para reunirse con inversores potenciales y convencerles del valor del título que se va a emitir (Hayes, 2024).

El mercado secundario incluye los lugares supervisados por un organismo regulador como la SEC, donde estos valores previamente emitidos se negocian entre inversores. Las empresas emisoras no participan en el mercado secundario. La Bolsa de Nueva York (NYSE) y el Nasdaq son ejemplos de mercados secundarios (Hayes, 2024).

El mercado secundario tiene dos categorías: el ‘mercado de subastas’ y el ‘mercado de intermediarios’. El mercado de subastas es el hogar del sistema ‘open outcry’, en el que compradores y vendedores se reúnen en un mismo lugar y anuncian los precios a los que están dispuestos a comprar y vender sus valores. La Bolsa de Nueva York es un ejemplo de ello. En los mercados de intermediarios se negocia a través de ‘redes electrónicas’. La mayoría de los pequeños inversores negocian a través de los mercados de intermediarios (Clara Fabiola, 2020).

3.2.2 Teoría de los Mercados Eficientes.

La Hipótesis del Mercado Eficiente (HME) postula que los precios de los activos en los mercados financieros siguen movimientos aleatorios e independientes, lo que los hace impermeables a la predicción, incluso cuando se analizan los datos históricos mediante el análisis técnico. Esto se debe a la incertidumbre inherente a las noticias que es amplio e instantáneamente accesible (Fama, 1965).

Por consiguiente, se cree que el precio actual refleja con exactitud el valor intrínseco del activo, lo que hace innecesario el análisis fundamental del activo. La hipótesis afirma que la expectativa más razonable para el precio futuro es el precio actual, y cualquier rendimiento superior a la media del mercado se considera excepcional (Santos, 2023).

3.2.3 Hipótesis del Mercado Adaptativo.

La Hipótesis del Mercado Adaptativo (HMA) plantea que los mercados financieros no son completamente eficientes ni siguen patrones predecibles a lo largo del tiempo. En cambio, estos mercados fluctúan cíclicamente entre estados de eficiencia e ineficiencia, dependiendo de factores externos como eventos geopolíticos, cambios en políticas y avances

tecnológicos. La HMA se diferencia de la Hipótesis del Mercado Eficiente (EMH) al sugerir que, en vez de mantener una eficiencia constante, los mercados evolucionan y se adaptan a los cambios de su entorno, haciendo que las estrategias de inversión exitosas en un momento puedan resultar ineficaces en otro (Lo, 2004).

Esta teoría reconoce que los inversores no son siempre racionales, ya que exhiben racionalidad limitada y patrones de comportamiento predecibles, especialmente en condiciones de alta incertidumbre. Esto incluye tendencias como el pánico, la aversión a las pérdidas y el seguimiento de la multitud, que pueden llevar a burbujas o caídas de precios. Según la HMA, los inversores aprenden y adaptan sus estrategias en función de experiencias pasadas, lo que contribuye a ciclos recurrentes en los mercados. Además, al igual que en la evolución biológica, las estrategias de inversión menos eficaces tienden a desaparecer, mientras que las más adaptativas sobreviven en un entorno competitivo (Lo, 2004).

Para los inversores, la HMA implica la necesidad de una gestión dinámica y flexible, dado a que no hay una estrategia óptima en todas las condiciones. La diversificación y el monitoreo continuo del mercado se vuelven esenciales para adaptarse a las fluctuaciones entre eficiencia e ineficiencia. Esta teoría también destaca la importancia de la psicología del inversor, sugiriendo que los mercados están influenciados no solo por datos objetivos, sino también por factores emocionales y comportamentales. En resumen, la HMA proporciona una visión más realista del mercado financiero, invitando a una estrategia de inversión adaptable y un análisis que considere las complejidades humanas y la evolución del entorno económico (Lo, 2004).

3.3 Teoría de Portafolio.

Los fundamentos teóricos de la gestión de activos son amplios, con varias facetas clave que merecen reconocimiento. La Teoría Moderna de Portafolios (MPT) introducida por Markowitz en 1952, representa una perspectiva fundamental que ha recibido gran atención tanto en el discurso académico como entre los profesionales de la inversión.

En esencia, la MPT subraya la oportunidad de que los inversores pueden aumentar la rentabilidad mitigando el riesgo mediante la diversificación estratégica, basándose en un análisis del rendimiento y la volatilidad históricos de los activos (Markowitz, 1952).

Posteriormente, una formulación matemática sustentada en un modelo de optimización calcula la asignación de cada activo dentro de la cartera. Dicha optimización tiene por objeto maximizar los rendimientos esperados para un determinado nivel de riesgo (Markowitz, 1952).

Es importante señalar que un mayor número de activos en una cartera no equivale necesariamente a una diversificación prudente. Por ejemplo, los activos pueden concentrarse en un único sector, lo que podría amplificar los rendimientos y exponer al mismo tiempo la cartera a niveles de riesgo equivalentes.

Las evoluciones que envuelven a la teoría emanan de diversos frentes, principalmente atribuibles a los retos asociados al cumplimiento de los supuestos subyacentes, que, de no cumplirse, podrían viciar cualquier análisis o crítica de la MPT. En particular, los supuestos de la racionalidad de los participantes en el mercado y la eficiencia del mercado representan características que se cuestionan incesantemente en la literatura (Wilford, 2012).

Por lo tanto, según la MPT, un inversor debe ser compensado por un mayor nivel de riesgo a través de mayores rendimientos esperados empleando la idea central de la diversificación. Para ello es necesario considerar las siguientes expresiones:

3.3.1 Esperanza de Retorno de un Portafolio.

La esperanza de retorno es el rendimiento promedio esperado de un portafolio, basado en los rendimientos esperados de los activos individuales y sus proporciones en el portafolio (Müller, 2014).

La rentabilidad del portafolio no depende explícitamente del riesgo de cada uno de los activos. El Retorno se expresa de la siguiente forma:

$$E(r_p) = \sum_{i=1}^{N} E(r_i) \cdot w_i$$

Donde:

- $E(r_p)$ como la esperanza del retorno del portafolio.
- w_i es el peso del activo i del portafolio.
- $E(r_i)$ es la esperanza de retorno del activo i.
- N la cantidad de activos del portafolio.

3.3.2 Riesgo.

El riesgo, en este contexto, se refiere a la volatilidad o variabilidad de los retornos del portafolio, y se mide comúnmente a través de la desviación estándar del retorno del portafolio (Müller, 2014). Se denota con la siguiente expresión:

$$\sigma_p^2 = \sum_{i=1}^{N} \sum_{j=1}^{N} w_i \cdot w_j \cdot \sigma_i \cdot \sigma_j \cdot \rho_{i,j}$$

Donde:

- σ_p^2 *es la varianza del retorno de portafolio*
- $w_i \cdot w_j$ los pesos de activos i y j en el portafolio
- $\sigma_i \cdot \sigma_j$ *son las desviciones estándar de los retornos de los activos i y j*
- $\rho_{i,j}$ es la correlación entre los retornos de los activos i y j.

3.3.3 Índice de Sharpe.

El índice de Sharpe es una medida de relación en un portafolio entre el riesgo y retorno, estando este último por encima de la tasa libre de riesgo. Es utilizado para evaluar el rendimiento ajustado por el riesgo de una inversión (Müller, 2014). El índice de Sharpe se denota con la siguiente expresión:

$$S_p = \frac{E(r_p) - r_f}{\sigma_p}$$

Donde:

- $E(r_p)$ *siendo la esperanza de retorno del portafolio.*
- r_f *la taza libre de riesgo.*
- σ_p *la desviación estándar o riesgo de retorno del protafolio.*

3.4 Estrategia de Buy & Hold.

La estrategia de "comprar y mantener" (también conocida como Buy & Hold), representa un enfoque de inversión pasivo y conservador en el cual los inversores adquieren acciones y las conservan durante un período prolongado, sin importar las fluctuaciones en el mercado. Este enfoque se basa en la premisa de que al mantener una inversión a largo plazo se puede generar rendimientos más altos al reducir la exposición a la volatilidad diaria o mensual. En este contexto, esta estrategia de inversión puede ser aplicada dentro del marco de la teoría de compensación riesgo-rentabilidad, la cual plantea que los rendimientos esperados de una inversión deben ser proporcionales al nivel de riesgo asumido (Ling, 2014).

Dicha estrategia puede ser aplicada sin problema en los distintos mercados bursátiles para la reducción del riesgo asociado a las acciones, ya que lo que busca es disminuir la volatilidad del rendimiento. Sin embargo, puede haber casos en que el análisis no refleja una mejora notable en los rendimientos de las inversiones si es que no se realiza un análisis debido a las acciones que conforman un portafolio, por ello, llevar un análisis pertinente a la empresa que representa cada acción no puede ser inverosímil a la hora de tomar una decisión final por parte del inversor (Ling, 2014).

3.5 Redes Neuronales.

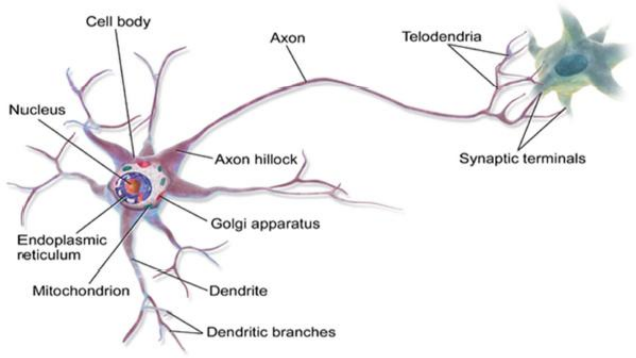
Las redes neuronales son un tipo de inteligencia artificial diseñada para imitar el funcionamiento del cerebro humano. A diferencia de los modelos digitales convencionales, que realizan cálculos manipulando exclusivamente ceros y unos, una red neuronal artificial establece conexiones entre unidades de procesamiento, que son el equivalente informático de las neuronas biológicas. La organización de estas conexiones y los pesos que se asignan a cada una determinan las respuestas y predicciones de la red neuronal, lo que permite obtener salidas adaptativas y precisas a partir de los datos de entrada que recibe (Islam, 2019).

Las redes neuronales son especialmente eficaces para tareas de predicción y clasificación cuando disponen de una base de datos amplia de ejemplos previos. Aunque en un sentido estricto una red neuronal implicaría un tipo de computación que no es digital, la realidad es que estas redes pueden ser simuladas en computadoras digitales. A través de esta simulación, las redes neuronales logran imitar las capacidades de reconocimiento de patrones como un cerebro biológico, identificando relaciones y estructuras complejas en los datos (Islam, 2019).

Desde una perspectiva técnica, una red neuronal es un conjunto de algoritmos inspirados en el funcionamiento del cerebro humano. Su propósito principal es reconocer patrones en los datos, interpretándolos mediante una especie de "percepción artificial" que permite etiquetar o agrupar la información en bruto. Los patrones que las redes neuronales son capaces de identificar están representados en forma numérica, generalmente mediante vectores. Esta estructura es versátil, ya que puede aplicarse para analizar y procesar datos provenientes de diferentes fuentes, como imágenes, sonidos, textos o series temporales (Islam, 2019).

Figura 1

Estructura de una Neurona.



Nota. Tomado de American Journal of Neural Networks and Applications, 2019.

En su forma más simple, el cerebro biológico es una vasta red de neuronas interconectadas. Cada neurona biológica recibe señales químicas y eléctricas a través de sus dendritas y, una vez procesada la señal, transmite la respuesta a otras neuronas a través de su axón. Estas conexiones ocurren en las sinapsis, que son puntos de contacto especializados que permiten la transmisión de impulsos eléctricos de una neurona a otra, repitiendo el proceso millones de veces en todo el sistema nervioso (Islam, 2019).

Inspiradas en esta estructura, las redes neuronales artificiales consisten en un conjunto de unidades conectadas, también conocidas como neuronas o nodos. Las conexiones entre estas neuronas artificiales transportan señales entre ellas, y cada conexión tiene un valor numérico asociado, llamado peso, que representa la importancia o "fuerza" de la señal. Estos pesos son ajustados durante el proceso de aprendizaje de la red, lo cual permite a la red neuronal adaptarse y optimizar su capacidad para reconocer patrones y realizar predicciones cada vez más precisas (Islam, 2019).

3.5.1 Tipos de Redes Neuronales.

Existen varios tipos de redes neuronales artificiales. Estos tipos de redes se implementan basándose en las operaciones matemáticas y en conjuntos de parámetros que se requieren para determinar la salida (Islam, 2019).

- **Redes Neuronales Feed Forward**

Las redes neuronales feed forward son una de las formas más simples de redes neuronales, donde la información fluye en una sola dirección, desde las capas de entrada hasta las capas de salida. Estas redes pueden tener o no capas ocultas y no utilizan retropropagación. La salida se calcula mediante la suma ponderada de las entradas, y se aplica una función de activación para determinar el resultado. Son comúnmente utilizadas en tareas de clasificación y regresión. Su simplicidad las hace adecuadas para problemas donde la relación entre las entradas y salidas es directa (Islam, 2019).

- **Redes Neuronales de Base Radial (RBF)**

Las redes RBF son un tipo de red neuronal que utiliza funciones de base radial como función de activación. Estas redes son especialmente efectivas para problemas de clasificación y regresión, ya que pueden modelar relaciones no lineales. La arquitectura típica incluye una capa de entrada, una capa oculta con neuronas RBF y una capa de salida. La activación de las neuronas en la capa oculta depende de la distancia entre la entrada y un centro específico, lo que permite una respuesta local a las entradas. Son valoradas por su capacidad de generalización y su rapidez en el entrenamiento (Islam, 2019).

- **Redes Neuronales de Kohonen**

Las redes de Kohonen, también conocidas como mapas autoorganizados, son un tipo de red neuronal no supervisada que se utiliza para la reducción de dimensionalidad y la visualización de datos. Estas redes organizan los datos de entrada en un espacio de menor dimensión, preservando la topología de los datos originales. Cada neurona en la red se activa en función de la similitud con la entrada, lo que permite agrupar datos similares. Son útiles en aplicaciones como la segmentación de imágenes y el análisis de patrones. Su capacidad para aprender sin supervisión las hace ideales para explorar grandes conjuntos de datos (Islam, 2019).

- **Redes Neuronales Recurrentes (RNN)**

Las redes neuronales recurrentes son un tipo de red que permite conexiones entre neuronas en capas diferentes, lo que les da la capacidad de mantener información en el tiempo. Esto las hace especialmente adecuadas para tareas que involucran secuencias, como el procesamiento de lenguaje natural y la predicción de series temporales. Las RNN pueden recordar información de entradas anteriores, lo que les permite capturar dependencias temporales. Sin embargo, pueden enfrentar problemas de desvanecimiento del gradiente durante el entrenamiento. Variantes como LSTM y GRU han sido desarrolladas para abordar estas limitaciones (Islam, 2019).

- **Redes Neuronales Convolucionales (CNN)**

Las redes neuronales convolucionales son especialmente diseñadas para el procesamiento de datos con una estructura de cuadrícula, como imágenes. Utilizan capas convolucionales que aplican filtros para extraer características locales de las imágenes, lo

que permite identificar patrones y objetos. Las CNN son altamente efectivas en tareas de visión por computadora, como reconocimiento de imágenes y detección de objetos. Su arquitectura jerárquica permite aprender representaciones complejas a partir de datos de entrada. Además, son robustas frente a variaciones en la posición y escala de los objetos en las imágenes (Islam, 2019).

- **Redes Neuronales Modulares**

Las redes neuronales modulares dividen un problema complejo en sub-tareas más pequeñas, utilizando múltiples redes que operan de manera independiente. Cada módulo se especializa en una parte del problema, lo que reduce la complejidad general y mejora la eficiencia del procesamiento. Esta estructura modular permite una mayor flexibilidad y escalabilidad en el diseño de redes neuronales. Sin embargo, la interacción entre módulos es limitada, lo que puede ser una desventaja en ciertos contextos. Son útiles en aplicaciones donde se requiere un enfoque distribuido para el aprendizaje y la toma de decisiones (Islam, 2019).

3.5.2 Estructura de las Redes Neuronales.

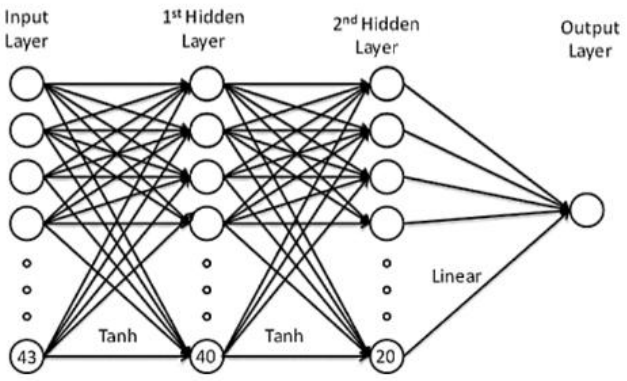
Las Estructura de las NN consiste en la conexión de distintos nodos distribuidas en capas de entrada, intermedias y de salida, donde se les asigna un peso $w^{(l)}$ y un sesgo $b^{(l)}$ dado una capa l a partir de un optimizador a elección (Kady Sako, 2022).

$$\widehat{Y^{(l)}} = \sigma\left(w^{(l)}x + b^{(l)}\right)$$

El principal objetivo es que la NN encuentre los parámetros $w^{(l)}$ y $b^{(l)}$ óptimos para la predicción del valor de salida $\widehat{Y^{(l)}}$, siendo σ su función de activación el cual validará o no los parámetros óptimos si la condición de pronóstico es satisfecha, para lo anterior es necesario desarrollar una 'Propagación hacia adelante' en donde mediante la capa de entrada (input) logre una salida en la capa de salida (output) mediante la estructura típica de la NN (Kady Sako, 2022).

Figura 2

Estructura de una Red Neuronal.



Nota. Tomado de IJORIS, 2020.

Por otro lado, el objetivo principal de cualquier red neuronal es transformar los datos de entrada no separables linealmente en características abstractas más separables linealmente mediante una jerarquía de capas. Estas capas son combinaciones de funciones lineales y no lineales en donde distintos hiperparámetros van configurando sus pesos y sesgos a medida que la NN es entrenada (Kady Sako, 2022).

3.5.2.1 Hiperparámetros de las Redes Neuronales.

Es importante considerar que por cada estructura o tipo de Red Neuronal por decisión de quién la usa es posible definir ciertos hiperparámetros los cuales pueden mejorar o no la performance del pronóstico de una red neuronal los cuales son completamente manipulables. Dentro de los hiperparámetros principales podemos mencionar:

- **Tasa de Aprendizaje (Learning Rate).**

La Tasa de Aprendizaje o Learning Rate sirve primordialmente para ver la frecuencia con que se buscan valores óptimos influenciando en la performance de la red neuronal dado a su estricta relación con el optimizador elegido. Incrementar este parámetro puede acelerar la convergencia de dos pasos consecutivos a la misma dirección del gradiente pudiendo ser fija, cíclica y adaptativa, dependiendo cual sea la elección del usuario o inversor (Raiaan, 2024).

- **Épocas (Epochs).**

Las Épocas en una red neuronal se definen para determinar el número de pasadas completas hacia delante (Forward Propagation) y hacia atrás (Backward Propagation) en una red neuronal durante el entrenamiento. Pasar todos los datos de entrenamiento una vez es inadecuado para recuperar las propiedades de la red neuronal. Puede que sea necesario

introducir el conjunto de datos varias veces para mejorar la generalización, posiblemente muchas veces para evitar el sobreajuste. Se desconoce cuál es el tamaño óptimo de las épocas para todos los conjuntos de datos (Raiaan, 2024).

- **Optimizadores.**

Los optimizadores son cruciales para entrenar redes neuronales ajustando sus pesos y ritmos de aprendizaje para minimizar la función de pérdida. En el contexto del contexto de las Redes Neuronales, existen varios optimizadores, cada uno con ventajas e idoneidad para distintos conjuntos de datos. Para el desarrollo de los pronósticos de acciones se utilizó el optimizador Adam, pero también existen otros tipos los cuales sirvieron de inspiración para la construcción del optimizador Adam. Entre los principales puntos rescatables podemos mencionar RMSProp y AdaGrad (Raiaan, 2024).

- **Funciones de Activación.**

Las funciones de activación juegan un papel muy crucial en las redes neuronales aprendiendo las características abstractas a través de transformaciones no lineales, en donde su principal función es generar una convergencia de entrenamiento dentro de la red sin aumentar la complejidad del modelo y sin obstaculizar el flujo de gradiente durante el entrenamiento (Andrea Apicella, 2021). Dentro de las principales Funciones de activación podemos encontrar:

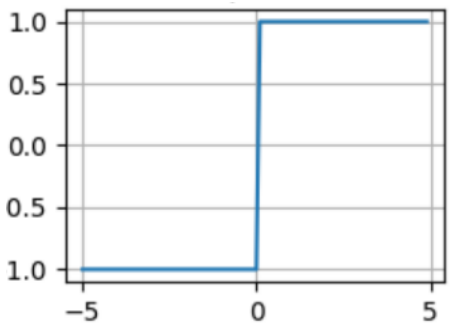
Función de Escalón Unitario:

Los valores que arroja esta función de activación son de 1 o 0, es decir, la función es de carácter binario {0,1} (Andrea Apicella, 2021).

$$f(x) = \begin{cases} 0 & para \quad x < 0 \\ 1 & para \quad x \geq 0 \end{cases}$$

Figura 3

Función de Escalón Unitario.



Nota. Tomado desde el ISTC, 2021.

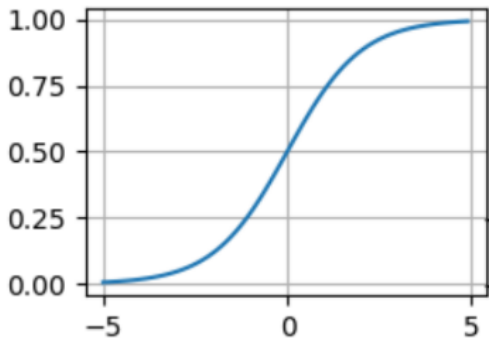
Función Sigmoide:

Los valores que arroja esta función de activación van desde 1 a 0, es decir, no son binarias como los valores de la Función de Escalón Unitario ya que pertenecen al intervalo [0,1] (Andrea Apicella, 2021).

$$f(x) = \frac{1}{1 + e^{-1}}$$

Figura 4

Función Sigmoide.



Nota. Tomado desde ISTC, 2021.

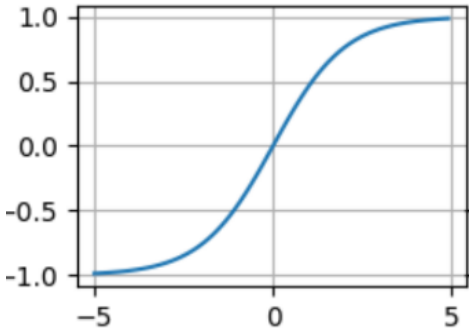
Función Tangente Hiperbólica:

Los valores de esta función son desde el -1 al 1, pertenecientes al intervalo [-1,1], siendo su ecuación no lineal (Andrea Apicella, 2021).

$$f(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}}$$

Figura 5

Función Tangente Hiperbólica.



Nota. Tomado desde el ISTC, 2021.

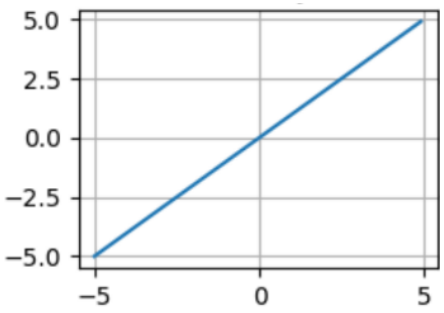
Función Lineal:

Los valores que arroja esta función de activación en comparación al resto van desde el -∞ al +∞, perteneciendo al intervalo [-∞; +∞] (Andrea Apicella, 2021).

$$f(x) = x$$

Figura 6

Función Lineal.



Nota. Tomado desde el ISTC, 2021.

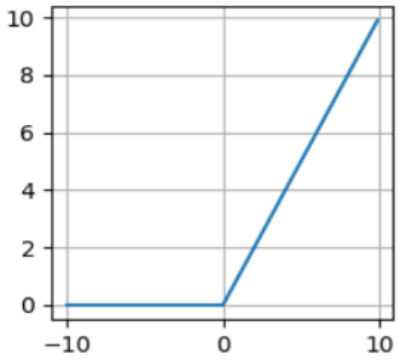
Función ReLU:

La función ReLU o Rectified Linear Unit genera una salida igual a cero cuando la entrada (*x*) sea negativa, y una salida igual a la entrada cuando dicha esta última sea positivo. Hoy en día es la más utilizada en el mundo dado a su implicación computacional (Andrea Apicella, 2021).

$$\boldsymbol{f(x) = max(0, x) = \begin{cases} 0 & para \quad x < 0 \\ x & para \quad x \geq 0 \end{cases}}$$

Figura 7

Función ReLU.



Nota. Tomado desde el ISTC, 2021.

3.5.3 Optimizador Adam.

Para la obtención de los parámetros $w^{(l)}$ y $b^{(l)}$ es necesario utilizar una optimización estocástica que sea eficiente, bajo esta condicionante, el optimizador Adam sólo requiere gradientes de primer orden con pocos requisitos de memoria. El método calcula tasas de aprendizaje adaptativas individuales para diferentes parámetros a partir de estimaciones del primer y segundo momento de los gradientes (Diederik P. Kingma, 2015).

Algunas de las ventajas que este modelo presenta en comparación a los modelos RMSProp y AdaGrad, son que las magnitudes de las actualizaciones de los parámetros son invariables ante variaciones del gradiente por los que sus tamaños de iteración están aproximadamente limitados por el parámetro de tamaño de paso (Diederik P. Kingma, 2015).

La forma en que este optimizador funciona es mediante la definición de una función objetivo $f_t(\Theta)$ con respecto a su parámetro Θ_t, el cual busca minimizar el valor esperado de dicha función $\mathrm{E}[f_t(\Theta)]$ con respecto a su parámetro Θ_t. Luego se denota la función estocástica con pasos consecutivos de 1 a *T*. La estocasticidad puede provenir de la evaluación en submuestras aleatorias (minibatches) de puntos de datos o surgir del ruido inherente a la función. Denotamos la función $g_t = \nabla_\Theta f_t(\Theta)$ como la gradiente siendo el vector de derivadas parciales de f_t, con respecto a Θ evaluado en el paso de tiempo t (Diederik P. Kingma, 2015).

El algoritmo se va actualizando con las medias móviles de la gradiente ($\mathrm{m_t}$) y de la gradiente al cuadrado (v_t), en donde los hiperparámetros β_1 y β_2 pertenecientes al intervalo [0,1) controlan las tasas de decaimiento exponencial de estas medias móviles (Diederik P.

Kingma, 2015).

$$m_t \leftarrow \beta_1^t \cdot m_{t-1} + (1 - \beta_1^t) \cdot g_t$$

$$v_t \leftarrow \beta_2^t \cdot v_{t-1} + (1 - \beta_2^t) \cdot g_t^2$$

Las propias medias móviles son estimaciones del 1er momento (media) y del 2do momento (varianza no centrada) del gradiente. Sin embargo, estas medias móviles se inicializan como vectores de ceros, lo que da lugar a estimaciones de momentos que están sesgadas hacia cero, especialmente durante los pasos de tiempo iniciales y sobre todo cuando las tasas de decaimiento son pequeñas con $\beta's$ cercanas a 1. Aquí los sesgos de inicialización se pueden contrarrestar fácilmente dando lugar a estimaciones corregidas de sesgo $\widehat{m_t}$ y $\widehat{v_t}$ (Diederik P. Kingma, 2015).

$$\widehat{m_t} \leftarrow \frac{m_t}{(1 - \beta_1^t)}$$

$$\widehat{v_t} \leftarrow \frac{v_t}{(1 - \beta_2^t)}$$

Considerando los valores iniciales β_1, β_2 y α, son iguales a 0.9 y 0.999 y 0.001 respectivamente, además de considerar siempre el valor del parámetro ϵ con una magnitud de 10^{-8}. Para hacer que el algoritmo sea eficiente y que pueda mejorarse, el orden de los cálculos en el bucle debe ser:

$$\alpha = \frac{\alpha \cdot \sqrt{1 - \beta_2^t}}{(1 - \beta_1^t)}$$

$$\theta_t = \theta_{t-1} - \alpha \cdot \frac{\widehat{m_t}}{\left(\sqrt{\widehat{v_t}} + \epsilon\right)}$$

Al final de los ciclos del optimizador Adam se obtendrán como resultado distintos valores de Θ_t óptimos los cuales ayudarán a obtener los pesos y sesgos de la red neuronal, siendo el optimizador Adam una forma más sofisticada y directa que el método de descenso por gradiente estocástico (Diederik P. Kingma, 2015).

3.5.4 Backpropagation.

La propagación hacia atrás o Backpropagation, es una función vital en las redes neuronales ya que permite encontrar los gradientes necesarios para que los pesos y sesgos sean actualizados mediante el optimizador Adam de manera eficiente. Para comprenderlo, esto se explica con las siguientes expresiones.

- Cálculo del error en la capa de salida:

$$\delta_j^L = \frac{\partial C}{\partial a_j^L}\sigma'\left(z_j^L\right) = \nabla_a C \cdot \sigma'\left(z_j^L\right)$$

Esta es una expresión natural. El primer termino $\frac{\partial C}{\partial a_j^L}$ mide que tanto la función de costo o error cambia como una función de la activación en la salida j, no dependiendo sólo de la salida, sino también de que tanto la función de activación σ' cambia respecto a z_j^L. Desde un punto de vista matricial, se utiliza la expresión $\nabla_a C$ como un vector el cual contiene las derivadas parciales en cuanto a la variación de a_j^L (Nielsen, 2015).

- Cálculo del error δ^l en términos del error en la capa siguiente δ^{l+1}:

$$\delta^l = ((w^{l+1})^T \delta^{l+1}) \cdot \sigma'(z^l)$$

Siendo w^{l+1} la transpuesta de los pesos de la matriz para la capa $(l+1)$, en donde a partir del error δ^{l+1} que ya se sabe, al aplicar los pesos w^{l+1} se puede pensar intuitivamente que el error se está moviendo hacia atrás a través de la red entregando una medición proveniente de la capa l, por ello, cuando aplicamos el producto por Hadamard con $\sigma'(z^l)$, dicha operación mediante la función de activación nos da el error δ^l perteneciente a la capa anterior. Esta expresión nos permite modelar el cómo la derivada

parcial de Costo o Error respecto a los pesos y sesgos funciona (Nielsen, 2015).

- Tasa de cambio del coste con respecto a cualquier sesgo:

Usando la ecuación error δ^l en términos del error en la capa siguiente δ^{l+1}, al derivar el costo por un sesgo, queda con la misma expresión, dado a que la derivada del sesgo es igual a 1 (Nielsen, 2015).

$$\frac{\partial C}{\partial b_j^l} = \delta_j^l$$

- Tasa de cambio del coste con respecto a cualquier peso:

Usando la ecuación error δ^l en términos del error en la capa siguiente δ^{l+1}, al derivar el costo por el peso, esta no queda con la misma expresión dado a que el peso en la capa l multiplica al output de la capa $(l-1)$ correspondiente a la expresión a_k^{l-1}, siendo esta última el input de la capa $(l-1)$ en el nodo k (Nielsen, 2015).

$$\frac{\partial C}{\partial w_{jk}^l} = a_k^{l-1}\delta_j^l$$

3.5.5 Métricas de Desempeño para Pronósticos.

Una vez se obtengan los pronósticos de nuestra Red Neuronal, es necesario calcular el error entre los valores reales y pronosticados con la finalidad de cuantificar el nivel de eficacia de nuestra Red Neuronal. Entre las distintas métricas a considerar tenemos:

- **MSE (Mean Squared Error)**:

El MSE es el promedio de los errores cuadrados entre los valores reales y los predichos. Penaliza los errores más grandes de manera más severa que el MAE, ya que los errores se elevan al cuadrado. Al ser una métrica que penaliza más fuertemente los grandes errores, es útil cuando se desea evitar errores grandes en las predicciones (M. Z. Naser, 2021).

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

- **MAE (Mean Absolute Error)**:

El MAE es la media de las diferencias absolutas entre los valores reales y los valores predichos por un modelo. Mide el promedio de los errores en términos absolutos, sin tener en cuenta la dirección de estos (si son positivos o negativos). Cuanto menor sea el MAE, mejor será la precisión del modelo, ya que indica que los errores promedio son más pequeños (M. Z. Naser, 2021).

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$$

- **MAPE (Mean Absolute Percentage Error)**:

El MAPE es el promedio de los errores absolutos en porcentaje respecto al valor real. Mide el error de predicción como un porcentaje, lo que permite una fácil interpretación de la magnitud del error en relación con los valores reales. Es útil para comparar errores entre diferentes modelos y escalas de datos, ya que está en forma de porcentaje. Un MAPE más bajo indica una mayor precisión (M. Z. Naser, 2021).

$$MAPE = \frac{100}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right|$$

- **RMSE (Root Mean Squared Error)**:

El RMSE es la raíz cuadrada del MSE. Mide la desviación estándar de los errores, lo que da una indicación de cuánto varían las predicciones del modelo en relación con los valores reales. Como está en las mismas unidades que las predicciones, el RMSE es fácil de interpretar. Un valor de RMSE más bajo indica una mejor precisión del modelo (M. Z. Naser, 2021).

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

3.6 Reinforcement Learning en Finanzas.

En los últimos años Reinforcement Learning (RL) ha acaparado un interés significativo en problemas relacionado con decisiones en secuencias complejas dado a su gran capacidad para resolver o complementar problemas en el trading demostrando gran potencial con las distintas técnicas que provee.

Reinforcement Learning (RL) ha sido de gran ayuda para las estrategias de inversión relacionadas con diversos modelos matemáticos y estadísticos para identificar oportunidades de inversión, como es el caso del Capital Asset Pricing Model (CAPM), el modelo Almgren-Chriss, Teoría de Portafolio de Markowitz, modelo del factor Fama & French, entre otros (Santos, 2023).

A lo anterior se le suma el avance de la inteligencia Artificial lo que refuerza aún más la necesidad de innovar o buscar otras alternativas que impacten positivamente a las estrategias de inversión, ante esto, también se destacan los modelos de Machine Learning los cuales han sido bastante atractivos. Desde este punto Reinforcement Learning se cataloga como un subcampo de Machine Learning dado a que provee de una formulación matemática basada en la extracción de información para su prueba y error, en donde dependiendo del modelo se genera un Actor cuyo comportamiento se busca que sea el óptimo (Santos, 2023). Reinforcement Learning aplicado a las finanzas ha destacado en rendimiento en cuanto a su gestión de portafolio y orden de ejecución, pudiendo entregar las siguientes ventajas en comparación a los modelos teóricos de inversión convencionales como, por ejemplo:

- Los modelos de RL permiten entrenar a un agente de extremo a extremo, que toma la información disponible sobre el mercado como estado de entrada y emitiendo un plan de acción.

- Los métodos basados en RL evitan la tarea extremadamente difícil de predecir el precio futuro y optimizar el beneficio global directamente.

- Las restricciones específicas de la tarea (por ejemplo, el coste de la transacción y el deslizamiento) pueden importarse fácilmente en los objetivos de RL.

- Otorga una exploración más controlada bajo entornos inciertos ajustando su política de manera estable, siendo ideal para lidiar con la volatilidad.

3.6.1 Algoritmo de Actor-Crítico con Ventaja (A2C)

El Algoritmo de Actor-Crítico con Ventaja o Advantage Actor-Critic (A2C), siendo uno de los modelos más utilizados de Reinforcement Learning en el contexto de robótica y juegos, es una versión sincrónica y determinista del algoritmo de Actor-Crítico con Ventaja Asincrónico (A3C). En A3C, cada agente actualiza los parámetros globales de forma independiente haciendo que agentes específicos de cada hilo estén jugando con políticas de diferentes versiones logrando a veces que la actualización agregada sea óptima. Para resolver el problema de la inconsistencia, A2C proporciona un Coordinador que se utiliza para actualizar los parámetros globales. El coordinador espera a que todos los actores paralelos terminen su trabajo antes de actualizar los parámetros globales y luego, en la siguiente iteración, los actores paralelos parten de la misma política. La actualización sincronizada del gradiente mantiene el entrenamiento más cohesionado y puede hacer que la convergencia sea más rápida (Sewak, 2019).

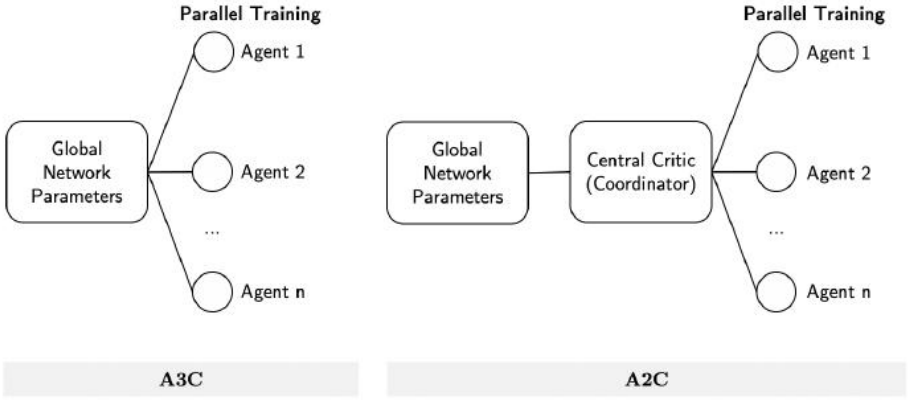
Además, se ha observado que A2C utiliza las GPU de forma más eficiente y funciona mejor con lotes de gran tamaño, consiguiendo rendimientos superiores al modelo de A3C, en donde el algoritmo A2C tiene un coordinador que controla la actualización de cada uno de los actores (Santos, 2023).

El algoritmo de Actor-Crítico se nombra de tal manera dado a que:

- El ‘Crítico’ estima la función de valor. Puede ser el valor-acción Q o el valor-estado V(s).
- El ‘Actor’ actualiza la distribución de la política en la dirección sugerida por el Crítico (por ejemplo, con gradientes de política).

Figura 8

Diferencia estructural de los modelos A3C y A2C.

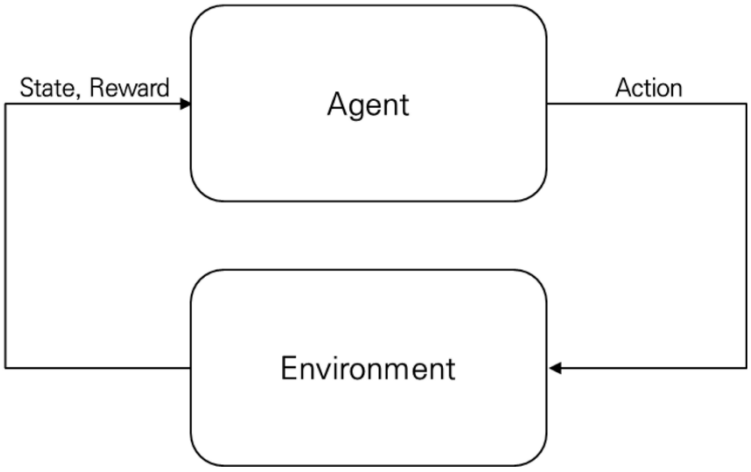


Nota. Tomado desde ResearchGate, 2020.

El algoritmo A2C comparte las variables comunes en su estructura con los demás modelos de RL, los cuales podemos mencionar el Actor (Agent), Estado (State), Acción (Action), Ambiente (Environment) y Ganancia (Reward), en donde la red de actores propone la mejor política para el agente basándose en el estado actual y la red crítica evalúa la calidad de la política y sugiere posibles mejoras (Sewak, 2019).

Figura 9

Variables comunes de los modelos de Reinforcement Learning.



Nota. Tomado desde ElSevier, 2023.

3.6.2 Markov Decision Process.

El algoritmo A2C al igual que la mayoría de los modelos de Reinforcement Learning están formulados a partir de los elementos del Markov Decision Process (MDP) el cual se describe como la tupla de $M = (s, a, P, r, \rho_0, \gamma)$ (Yang, 2023).

Un MDP se define por:

- **Conjunto de estados (s)**: Representa todas las posibles situaciones en las que se puede encontrar el sistema.
- **Estado inicial del estado (s_0):** Denotado con $\rho_0(\cdot)$: $S \rightarrow [0, 1]$ siendo el estado inicial de la distribución.
- **Conjunto de acciones (a)**: Conjunto de todas las acciones posibles que el agente puede tomar en cualquier estado.
- **Función de transición de estado (P)**: Describe la probabilidad de que el sistema pase de un estado a otro dado que se ha tomado una acción específica.
- **Función de recompensa (r)**: Proporciona una recompensa inmediata obtenida al pasar de un estado a otro como resultado de una acción.
- **Factor de descuento (γ)**: Un valor entre 0 y 1 que refleja la importancia de las recompensas futuras en comparación con las recompensas inmediatas. Un valor de γ cercano a 1 indica que las recompensas futuras son casi tan importantes como las inmediatas, mientras que un valor cercano a 0 indica que solo las recompensas inmediatas importan.

El objetivo en un MDP es encontrar una ‘política óptima’ en función a la mejor acción a tomar en cada estado, para así maximizar la recompensa acumulada a lo largo del ambiente (Yang, 2023).

Una política de Markov estacionaria π es una distribución de probabilidad definida en $S \times A$, en donde $\pi(a_t|s_t)$ denota la probabilidad de tomar la decisión *a* en el estado *s*, por lo que π denota el conjunto que recoge todas las políticas estacionarias de Markov (Yang, 2023).

$$\tau \rightarrow \{s_t, a_t, r_{t+1}\}_{t\geq 0} \sim \pi$$

Por otra parte, para definir la trayectoria τ tomada por la política π es necesario seguir una serie de pasos:

$$s_o \sim \rho_o(\cdot)\,,\ \ a_t \sim \pi(\cdot\,|s_t)\,,\ \ s_{t+1} \sim P(\cdot\,|s_t, a_t)\,,\ \ r_{t+1} = \mathrm{r}(s_{t+1}|s_t, a_t)$$

3.6.3 Matriz de probabilidades de transición de cada paso.

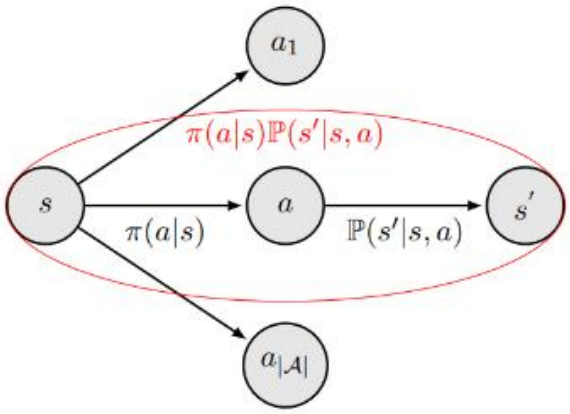
Para definir la matriz de probabilidad por transición, definimos $P_\pi \in$ R |S|×|S| como una matriz de probabilidad respecto a la transición entre estados, pudiendo ser expresado como:

$$P_\pi[s, s'] = \sum_{a \epsilon A} \pi(a|s) P\,(s'|s, a) =: \mathrm{P}_\pi(s|s')$$

Dicha expresión, denota la probabilidad de transformación de estado desde un paso s al s′ (siendo s' el estado consecutivo a s) mediante la ejecución de π (Yang, 2023). La transición de estado de un paso bajo una política π la ilustramos en la siguiente Figura.

Figura 10

Visualización de la probabilidad en la transición de un solo paso.



Nota. Tomado desde School of Artificial Intelligence, 2023.

3.6.4 Objetivo de la Política de Reinforcement Learning.

A raíz del MDP ahora podemos explicar de mejor forma cómo el Algoritmo de Actor-Crítico con ventaja (A2C) logra buenas decisiones de Inversión. De acuerdo con la política óptima seguida en base a un ambiente, el propósito de la Política y/o su gradiente es 'Maximizar los retornos esperados', los cuales son obtenidos desde la función objetivo de la gradiente de la política $J(\Theta)$ (Yang, 2023):

$$J(\Theta) = E\left[\sum_{t=0}^{T-1} r_{t+1}\right]$$

$$J(\Theta) = E\left[\sum_{t=0}^{T-1} r_{t+1}\,|\pi_{\Theta}\right]$$

$$J(\Theta) = \sum_{t=i}^{T-1} P(s_t, a_t|\tau)\, r_{t+1}$$

Consideramos i como un punto de comienzo arbitrario y $P(s_t\,, a_t|\tau)$ como la probabilidad de la ocurrencia de $s_t\, y\, a_t$ dado una trayectoria τ (Kouridi, 2020)**.** Luego diferenciando ambos lados de la expresión con respecto a la política del parámetro Θ tenemos:

$$\nabla_{\Theta} J(\Theta) = \sum_{t=i}^{T-1} \nabla_{\Theta} P(s_t, a_t\,|\tau)\, r_{t+1}$$

$$\nabla_{\Theta} J(\Theta) = \sum_{t=i}^{T-1} P(s_t, a_t \tau) \frac{\nabla_{\Theta} P(s_t, a_t|\tau)}{P(s_t, a_t|\tau)}\, r_{t+1}$$

$$\nabla_{\Theta} J(\Theta) = \sum_{t=i}^{T-1} P(s_t, a_t \tau) \nabla_{\Theta} log\, P(s_t, a_t \tau)\, r_{t+1}$$

$$\nabla_{\Theta} J(\Theta) = E\left[\sum_{t=i}^{T-1} \nabla_{\Theta} log\, P(s_t, a_t \tau)\, r_{t+1}\right]$$

Sin embargo, en la aplicación de la política no se aplica de manera directa y explícitamente, ya que se obtiene mediante muestras de episodios aleatorias, por lo que nos permite aproximar y expresar el valor esperado de $\nabla_{\Theta} J(\Theta)$ como:

$$\nabla_{\Theta} J(\Theta) = \sum_{t=i}^{T-1} \nabla_{\Theta} \log \pi_{\Theta}\,(s_t, a_t \tau) \left(\sum_{t'=t+1}^{T} r_{t'+1}\right)$$

$$\nabla_{\Theta} J(\Theta) = \sum_{t=i}^{T-1} \nabla_{\Theta} \log \pi_{\Theta}\,(s_t, a_t \tau)\, G_t$$

En consecuencia, la alta variabilidad de las probabilidades logarítmicas y de los valores de recompensa acumulativos creará gradientes ruidosos y provocará un aprendizaje inestable y/o una desviación de la distribución de políticas en una dirección no óptima (Yang, 2023).

Además de la alta varianza de los gradientes, otro problema con los gradientes de política ocurre cuando las trayectorias tienen una recompensa acumulativa de 0. La esencia del gradiente de política es aumentar las probabilidades de las acciones ‘buenas’ (incremento de la recompensa final) y disminuir las de las acciones ‘malas’ (disminución de la recompensa final) en la distribución de políticas. Tanto las acciones buenas como las malas no se aprenderán si la recompensa acumulativa es muy cercano o igual a cero (Silva, 2022).

Lo anterior contribuye a la inestabilidad y lenta convergencia de los métodos de gradiente de política, en el cual una forma de reducir la varianza y aumentar la estabilidad es restar la recompensa acumulada por una 'línea de base' (Kouridi, 2020).

$$\nabla_{\Theta} J(\Theta) = E\left[\sum_{t=0}^{T-1} \nabla_{\Theta} \log \pi_{\Theta}\left(a_t \mid s_t\right)\left(G_t - b(s_t)\right)\right]$$

Intuitivamente, hacer la recompensa acumulativa más pequeña restándola con una 'línea de base' hará gradientes más pequeños, y por lo tanto actualizaciones más pequeñas y estables (Yousefi, 2022). La línea de base puede tomar varios valores, en donde utilizando la función V como la función de línea de base, restamos el término de valor Q con el valor V. Intuitivamente, esto significa cuánto mejor es tomar una acción específica en comparación con el promedio, la acción general en el estado dado será mucho mejor (Yousefi, 2022), llamándose este valor el valor de ventaja:

$$A(s_t, a_t) = Q_w(s_t, a_t) - V_v(s_t)$$

Por otro lado, podemos utilizar la relación entre Q y V de la ecuación de optimalidad de Bellman, siendo:

$$Q_w(s_t, a_t) = E[r_{t+1} + \gamma V(s_{t+1})]$$

Al reescribir la expresión anterior, la ventaja queda igual a:

$$A(s_t, a_t) = r_{t+1} + \gamma V_v(s_{t+1}) - V_v(s_t)$$

Así que podemos reescribir la ecuación de actualización como la siguiente formulación que sería la gradiente de política contenida dentro del algoritmo A2C (Yousefi, 2022):

$$\nabla_{\Theta} J(\Theta) \sim \sum_{t=0}^{T-1} \nabla_{\Theta} \log \pi_{\Theta} (a_t s_t)\big(r_{t+1} + \gamma V_v(s_{t+1}) - V_v(s_t)\big)$$

$$\nabla_{\Theta} J(\Theta) \sim \sum_{t=0}^{T-1} \nabla_{\Theta} \log \pi_{\Theta} (a_t s_t) A(s_t, a_t)$$

Un aspecto atractivo de este enfoque es la capacidad de aprender una política continua con probabilidades de acción que cambian suavemente durante el aprendizaje en lugar de tomar máximos discontinuos. Esto puede mejorar la estabilidad y, por tanto, la convergencia en comparación con los métodos basados en valores (Kouridi, 2020).

Además, la parametrización continua de la política permite el entrenamiento con optimización basada en gradiente dejando la siguiente expresión:

$$\theta_t = \theta_{t-1} + \alpha \nabla_{\theta} J(\theta)$$

Donde θ_t es el parámetro de la política, $\nabla_{\theta} J(\theta)$ es la gradiente de la política y α es la tasa de aprendizaje. Los gradientes de la muestra sólo tienen que ser proporcionales al gradiente verdadero porque cualquier constante de proporcionalidad puede absorberse en el tamaño del paso que es arbitrario. Esto confiere al modelo por refuerzo o Reinforce buenas propiedades de convergencia teóricas garantizando una mejora en el rendimiento esperado para tamaños de paso suficientemente pequeños para su convergencia a un óptimo local en condiciones de aproximación estocástica estándar (Kouridi, 2020), además un punto valioso para considerar es que la línea de base no influye en el valor esperado del gradiente estimado, sino que reduce su varianza (Kouridi, 2020).

4. Metodología

A continuación, se detallarán los procedimientos necesarios para la combinación y aplicación del pronóstico generado por la Red Neuronal (NN) y el Algoritmo de Actor-Crítico con Ventaja (A2C), para los cuales es necesario la utilización de métricas y definiciones de parámetros para calcular la Rentabilidad Acumulada Final que generen las decisiones de compra y venta sobre los precios de acción pronosticados y sobre los precios reales. Además, con la rentabilidad acumulada final de esta estrategia de inversión, se comparará con los resultados que arrojará la estrategia de Buy & Hold al final del horizonte de evaluación, para así evaluar si la propuesta de inversión utilizando NN y el algoritmo A2C supera a este método tradicional inversión. A partir de lo anterior, se establecen los siguientes puntos.

4.1 Definición del Problema.

De acuerdo con los objetivos, la idea principal es poder generar portafolios de inversión eficientes con una buena capacidad de obtener retornos de manera casi autónoma en la definición de la decisión final de compra y venta de acciones implementando algoritmos que estén a la vanguardia en este contexto, en donde para este caso y como se ha estado mencionando, se utilizarán los modelos de Redes Neuronales (basado en Deep Leraning) y del Algoritmo de Actor-Crítico con Ventaja (basado en Reinforcement Learning).

4.2 Datos Utilizados.

Para la ejecución y utilización de los modelos de Red Neuronal y A2C se utilizaron precios de acciones importadas en formato .csv desde ‘Yahoo Finance’ comenzando desde el 1 de enero del 2015 hasta el 31 de julio del 2024.

Con respecto a los datos de entrenamiento se consideró los precios contenidos desde el 1 de enero del 2015 hasta el 29 de agosto del 2022, y para los datos de testeo desde el 30 de agosto del 2022 hasta el 31 de Julio del 2024.

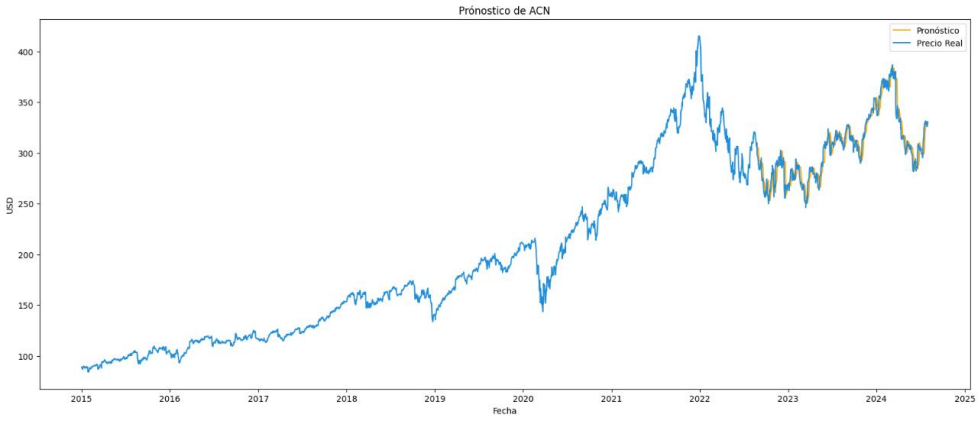
En total se analizaron los precios de acción de 17 empresas, las cuales son:

ACN (Accenture plc):

Accenture es una multinacional de servicios profesionales que se enfoca en consultoría de gestión, tecnología y outsourcing. Fundada en 1989, su sede está en Dublín, Irlanda. Ofrece soluciones digitales, de nube e inteligencia artificial a nivel global. Sus clientes provienen de diversas industrias, incluidos los sectores financieros, salud y productos de consumo. Tiene un enfoque fuerte en la transformación digital y optimización de procesos empresariales. Accenture es conocida por su capacidad para implementar estrategias tecnológicas innovadoras.

Figura 11

Pronóstico acción ACN.



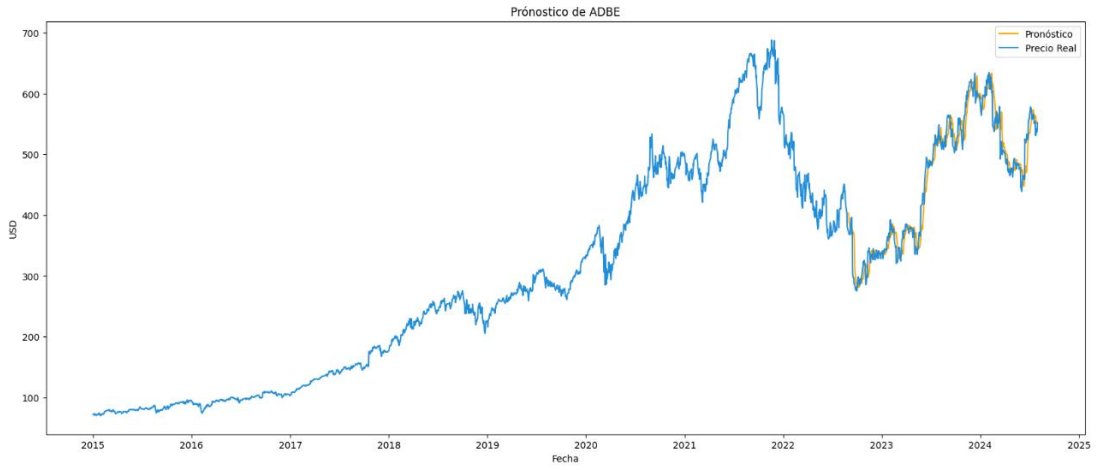
Nota. Elaboración Propia.

ADBE (Adobe Inc.):

Adobe es una empresa multinacional estadounidense de software, conocida por sus productos como Photoshop, Acrobat y Creative Cloud. Fundada en 1982, tiene su sede en San José, California. Adobe es líder en soluciones para diseño gráfico, edición de videos, web y marketing digital. Su modelo de negocio está centrado en suscripciones a su software basado en la nube. Además, ofrece herramientas de análisis y personalización para la creación de experiencias digitales. Adobe también es un referente en la gestión de documentos y firma electrónica.

Figura 12

Pronóstico acción ADBE.



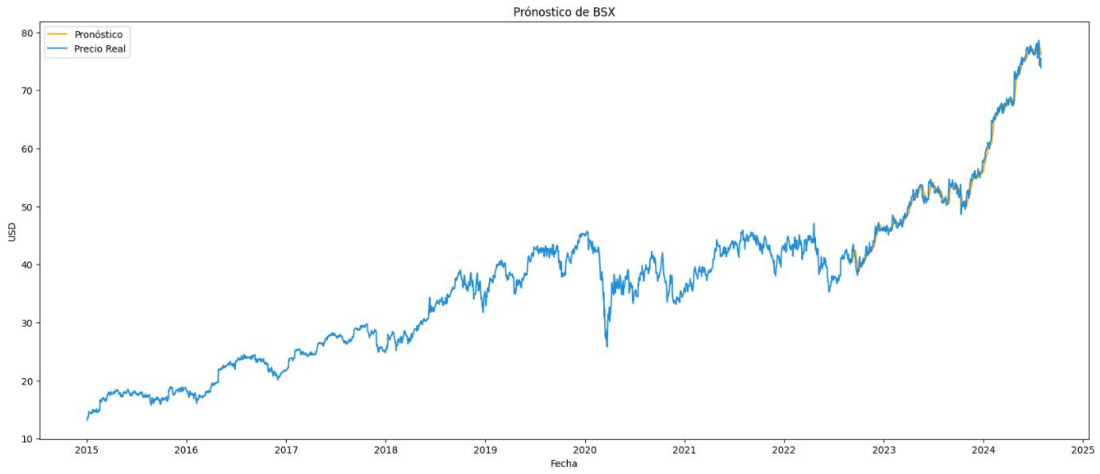
Nota. Elaboración Propia.

BSX (Boston Scientific Corporation):

Boston Scientific es una empresa global de tecnología médica que desarrolla dispositivos innovadores para una amplia gama de especialidades médicas. Fundada en 1979, tiene su sede en Marlborough, Massachusetts. Su portafolio incluye soluciones para cardiología, urología, oncología y neurocirugía, entre otras áreas. La compañía se enfoca en procedimientos mínimamente invasivos, buscando mejorar la calidad de vida de los pacientes. Boston Scientific es un actor clave en la innovación médica, con un fuerte enfoque en investigación y desarrollo.

Figura 13

Pronóstico acción BSX.



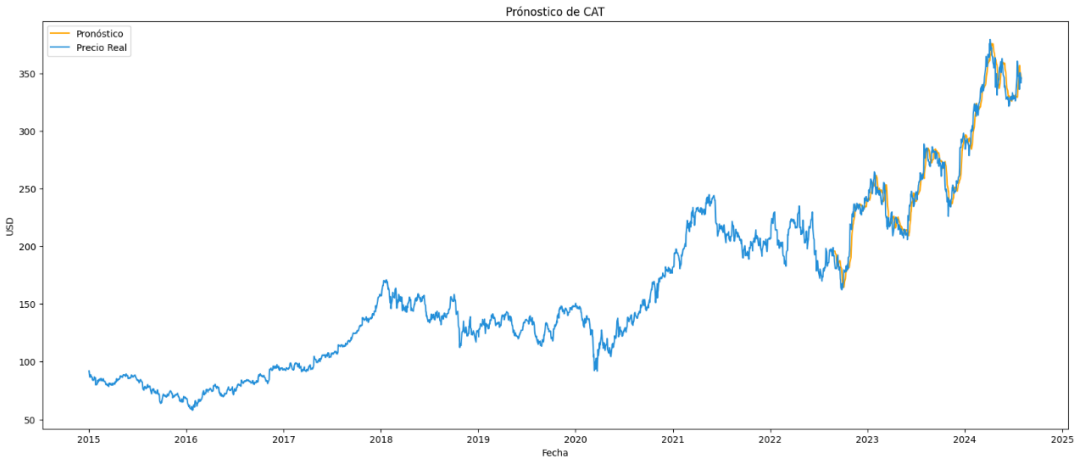
Nota. Elaboración Propia.

CAT (Caterpillar Inc.):

Caterpillar es el fabricante líder mundial de maquinaria pesada para construcción y minería, así como de motores y turbinas. Fundada en 1925, tiene su sede en Deerfield, Illinois. La compañía produce tractores, excavadoras, camiones y otros equipos para sectores industriales, energéticos y de transporte. Además, ofrece soluciones de energía y generación eléctrica. Caterpillar se distingue por su red global de distribución y servicio. Su tecnología también incluye soluciones avanzadas en automatización y eficiencia energética para mejorar la productividad.

Figura 14

Pronóstico acción CAT.



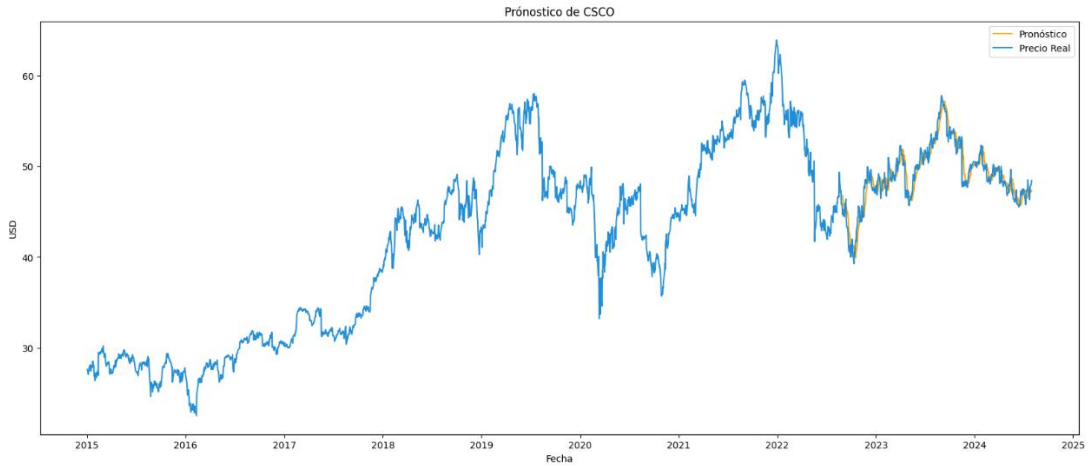
Nota. Elaboración Propia.

CSCO (Cisco Systems, Inc.):

Cisco es una empresa multinacional estadounidense especializada en equipos de redes y telecomunicaciones. Fundada en 1984 y con sede en San José, California, Cisco proporciona infraestructura para internet, routers, switches, y soluciones de seguridad. Es líder en servicios de ciberseguridad, comunicación en la nube y redes empresariales. La compañía también impulsa el desarrollo de tecnologías para la automatización de redes y la Internet de las Cosas (IoT). Sus productos y servicios permiten a empresas y gobiernos conectarse de manera segura y eficiente.

Figura 15

Pronóstico acción CSCO.



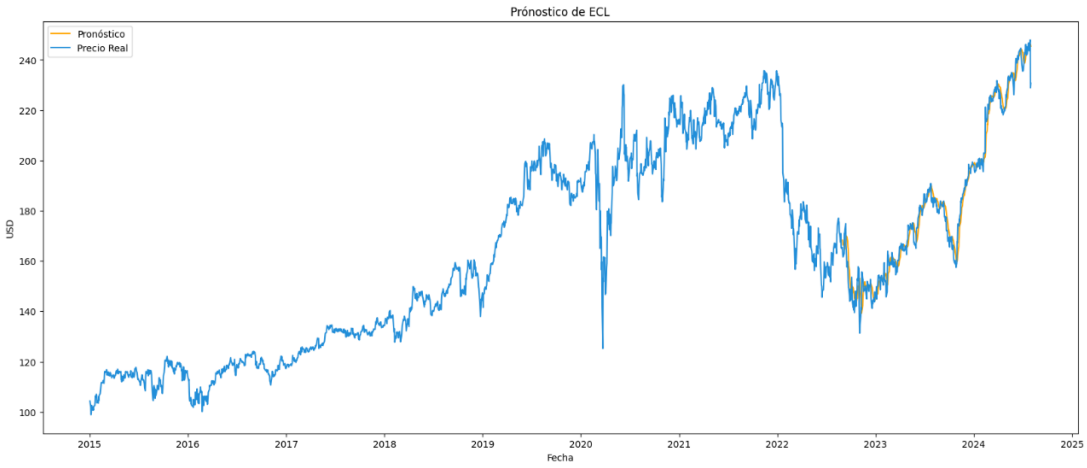
Nota. Elaboración Propia.

ECL (Ecolab Inc.):

Ecolab es una empresa global de servicios y tecnología para la higiene, la energía y el tratamiento del agua. Fundada en 1923, tiene su sede en St. Paul, Minnesota. Sus productos y soluciones son utilizados en la industria alimentaria, hospitales, hoteles y plantas de manufactura. Ecolab se centra en la sostenibilidad, ayudando a sus clientes a reducir el consumo de agua, energía y mejorar la eficiencia operativa. Ofrecen productos químicos y equipos especializados para asegurar la limpieza y desinfección en operaciones críticas.

Figura 16

Pronóstico acción ECL.



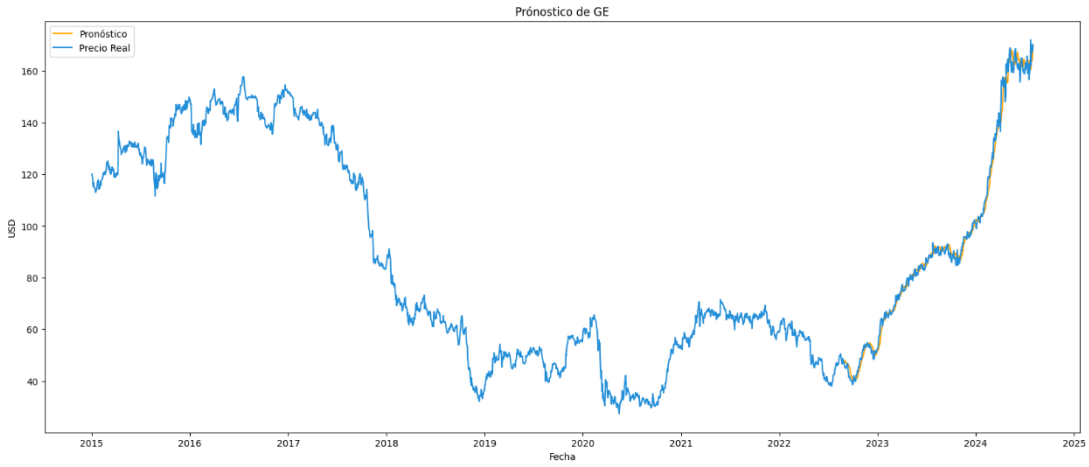
Nota. Elaboración Propia.

GE (General Electric Company):

General Electric es una empresa multinacional estadounidense que opera en varios sectores industriales, incluida la aviación, energía y salud. Fundada en 1892 y con sede en Boston, Massachusetts, GE ha sido pionera en la fabricación de productos de energía eléctrica y tecnología industrial. Hoy, es líder en turbinas para generación eléctrica, motores de avión, y equipos médicos. También ha impulsado la transformación digital de la industria a través de software y soluciones de IoT. GE se enfoca en innovación y sostenibilidad.

Figura 17

Pronóstico acción GE.



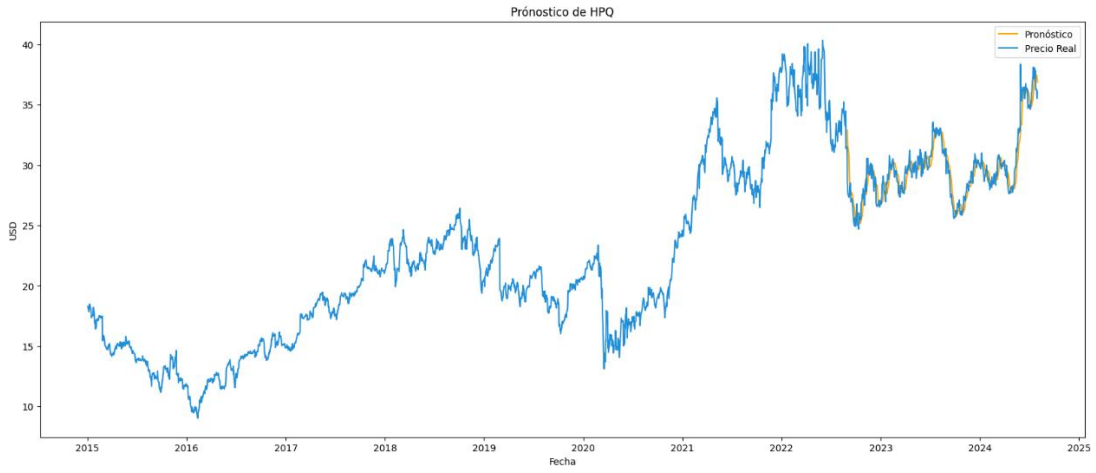
Nota. Elaboración Propia.

HPQ (HP Inc.):

HP Inc. es una empresa estadounidense de tecnología que fabrica computadoras personales, impresoras y dispositivos relacionados. Fundada en 1939 y con sede en Palo Alto, California, HP es conocida por sus soluciones de hardware y software para el consumidor y empresas. Ofrece productos que van desde laptops y PCs hasta impresoras 3D y sistemas de impresión de gran escala. HP también tiene un enfoque creciente en la sostenibilidad y la economía circular, buscando reducir el impacto ambiental de sus productos.

Figura 18

Pronóstico acción HPQ.



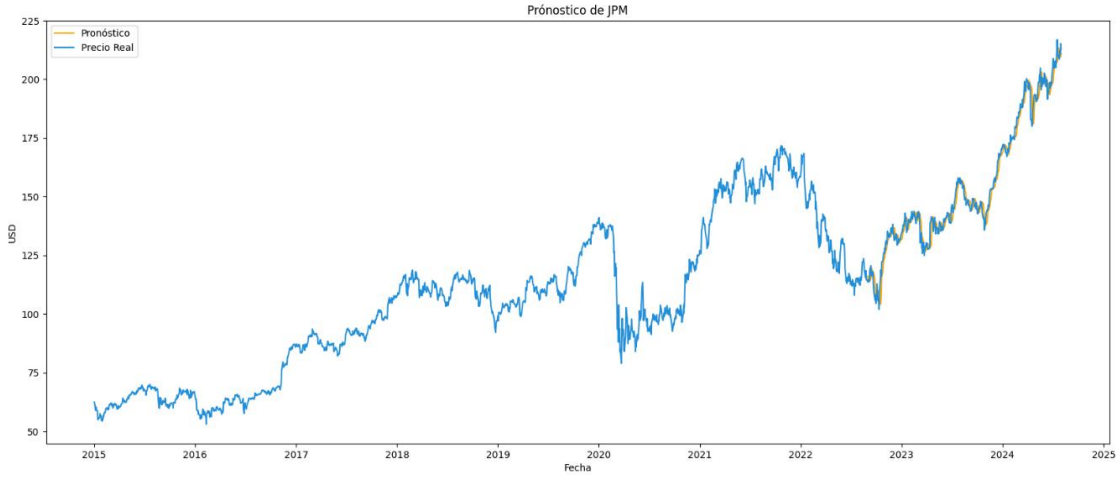
Nota. Elaboración Propia.

JPM (JPMorgan Chase & Co.):

JPMorgan Chase es el mayor banco de Estados Unidos por activos, con operaciones en banca de inversión, gestión de activos, y servicios financieros. Fundado en 2000 tras la fusión de J.P. Morgan & Co. y Chase Manhattan Bank, tiene su sede en Nueva York. JPMorgan ofrece servicios a individuos, empresas e instituciones, incluyendo financiamiento, asesoría y gestión de riesgo. Es líder global en mercados de capitales, fusión y adquisiciones, y soluciones de banca digital. JPMorgan también juega un papel clave en la economía global con una presencia fuerte en mercados internacionales.

Figura 19

Pronóstico acción JPM.



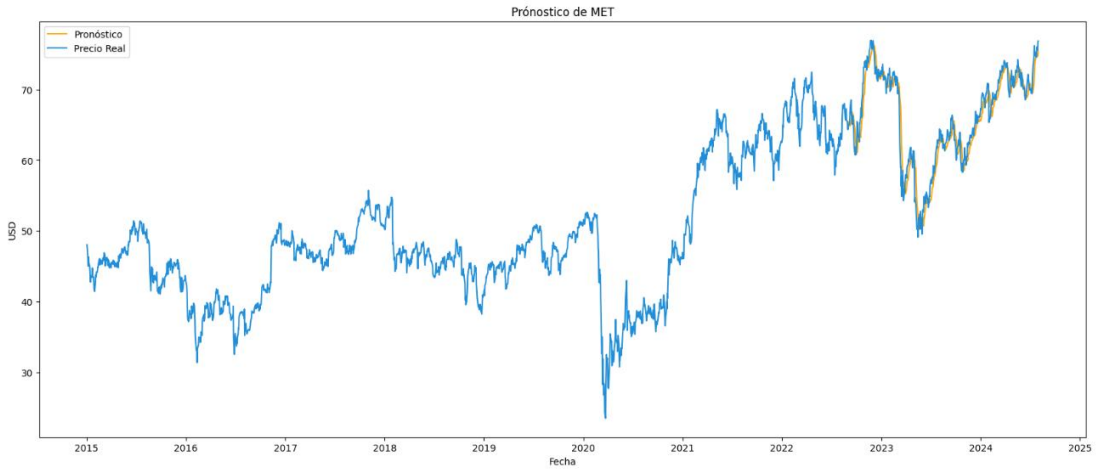
Nota. Elaboración Propia.

MET (MetLife, Inc.):

MetLife es una de las compañías de seguros más grandes del mundo, especializada en seguros de vida, de salud y rentas vitalicias. Fundada en 1868 y con sede en Nueva York, MetLife opera en más de 40 países, ofreciendo soluciones financieras tanto a individuos como a empresas. La compañía también ofrece productos de jubilación y gestión de activos. Su enfoque está en la protección financiera a largo plazo y la previsión de riesgos. MetLife es un actor importante en la gestión de pensiones y planes de beneficios para empleados.

Figura 20

Pronóstico acción MET.



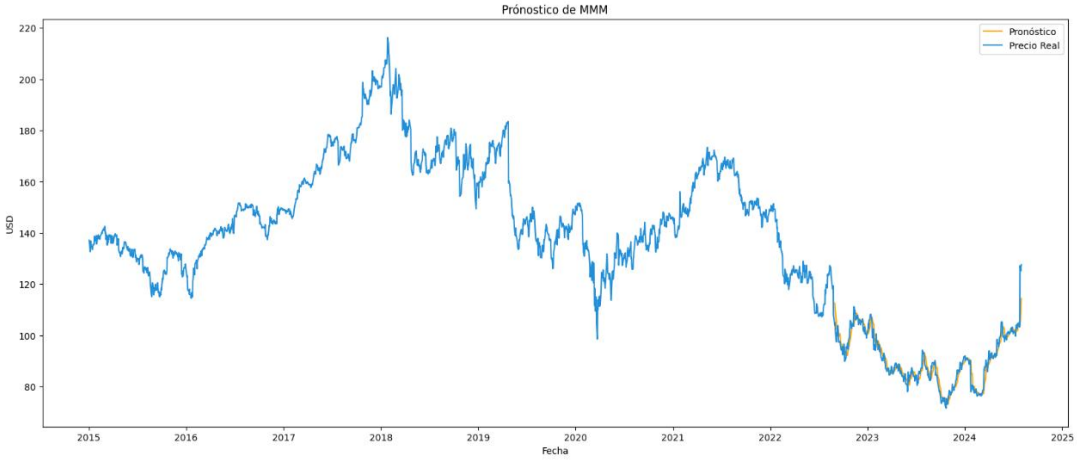
Nota. Elaboración Propia.

MMM (3M Company):

3M es una empresa multinacional estadounidense diversificada que opera en sectores como salud, industria, seguridad, y consumo. Fundada en 1902, tiene su sede en St. Paul, Minnesota. Conocida por sus productos innovadores como los adhesivos, cintas y equipos de protección personal, 3M cuenta con una extensa cartera de más de 60,000 productos. La compañía se distingue por su capacidad de investigación y desarrollo, aplicando tecnología avanzada a una amplia gama de industrias. También tiene un fuerte compromiso con la sostenibilidad y la responsabilidad social.

Figura 21

Pronóstico acción MMM.



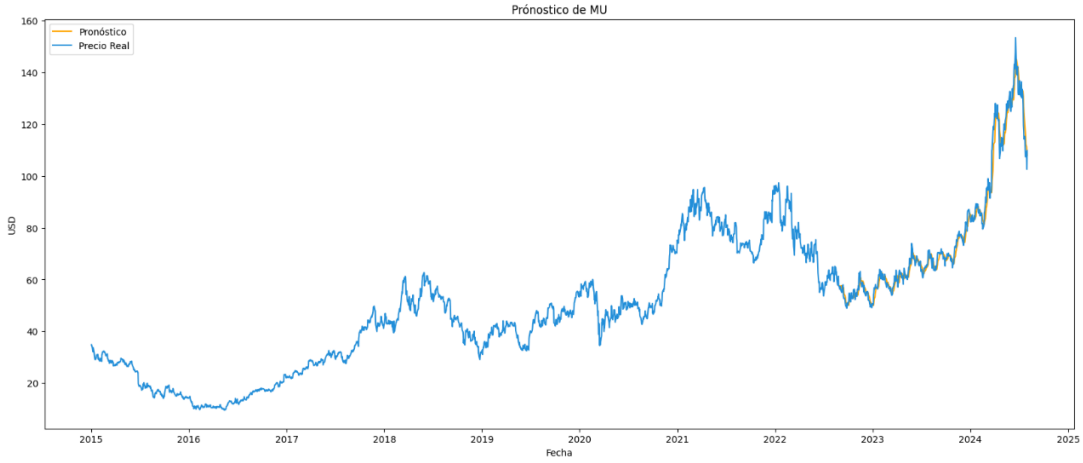
Nota. Elaboración Propia.

MU (Micron Technology, Inc.):

Micron Technology es una empresa estadounidense especializada en la fabricación de semiconductores y soluciones de memoria. Fundada en 1978 y con sede en Boise, Idaho, Micron produce DRAM, NAND y otros tipos de memoria flash utilizadas en dispositivos electrónicos. Sus productos son esenciales para smartphones, computadoras, servidores y vehículos autónomos. Micron juega un papel clave en la industria de la tecnología, proporcionando componentes para inteligencia artificial, cloud computing y la Internet de las Cosas (IoT). La compañía está altamente comprometida con la innovación en chips de memoria.

Figura 22

Pronóstico acción MU.



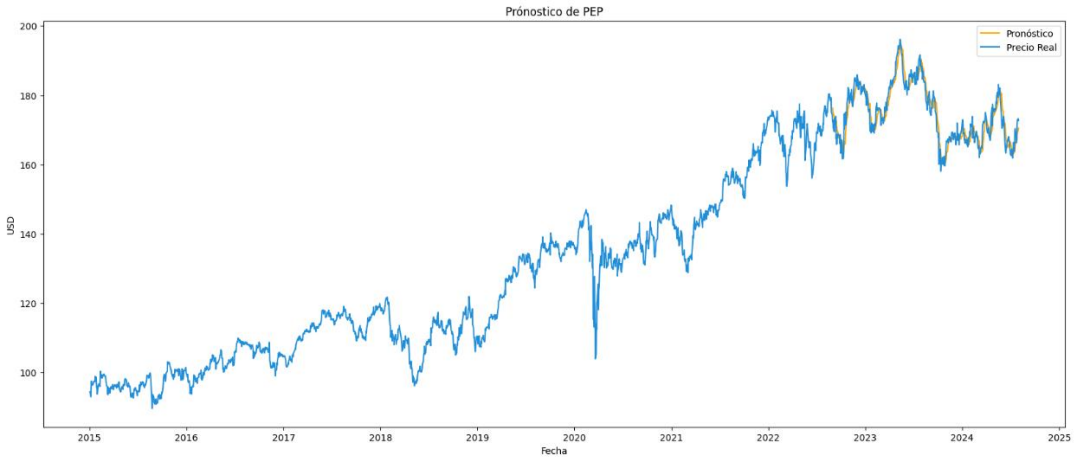
Nota. Elaboración Propia.

PEP (PepsiCo, Inc.):

PepsiCo es una multinacional estadounidense de alimentos y bebidas, conocida por marcas como Pepsi, Lay's, y Gatorade. Fundada en 1965 tras la fusión de Pepsi-Cola y Frito-Lay, su sede está en Purchase, Nueva York. PepsiCo es uno de los mayores fabricantes de alimentos y bebidas a nivel mundial, operando en más de 200 países. La empresa se enfoca en ofrecer productos saludables y sostenibles, con un compromiso en la reducción de azúcares y el uso de envases reciclables. Su portafolio incluye productos de bebidas, snacks y cereales.

Figura 23

Pronóstico acción PEP.



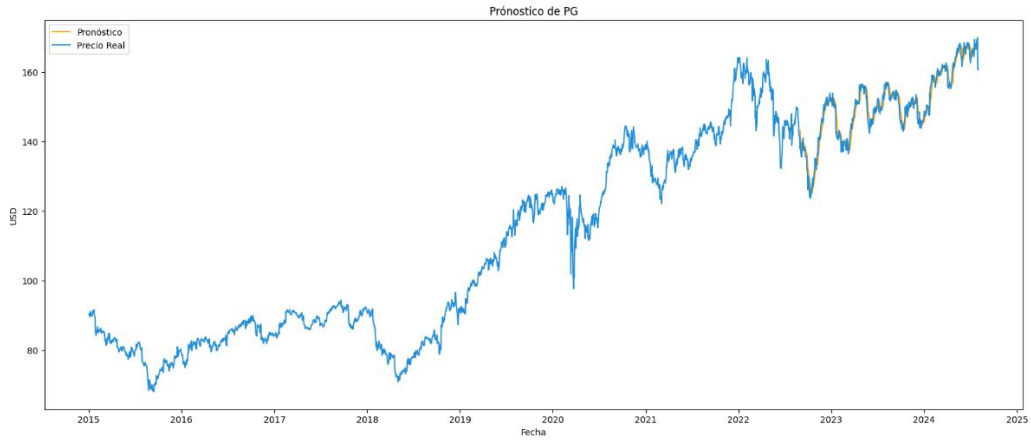
Nota. Elaboración Propia.

PG (The Procter & Gamble Company):

Procter & Gamble es una empresa multinacional estadounidense que fabrica productos de consumo masivo, especialmente en higiene y cuidado personal. Fundada en 1837, con sede en Cincinnati, Ohio, P&G es conocida por marcas como Pampers, Gillette, y Head & Shoulders. La compañía se enfoca en innovación de productos, con un fuerte enfoque en sostenibilidad y responsabilidad social. Además de su gran portafolio de marcas, P&G ha liderado iniciativas de reducción de residuos y mejor uso de recursos en la fabricación de productos.

Figura 24

Pronóstico acción PG.



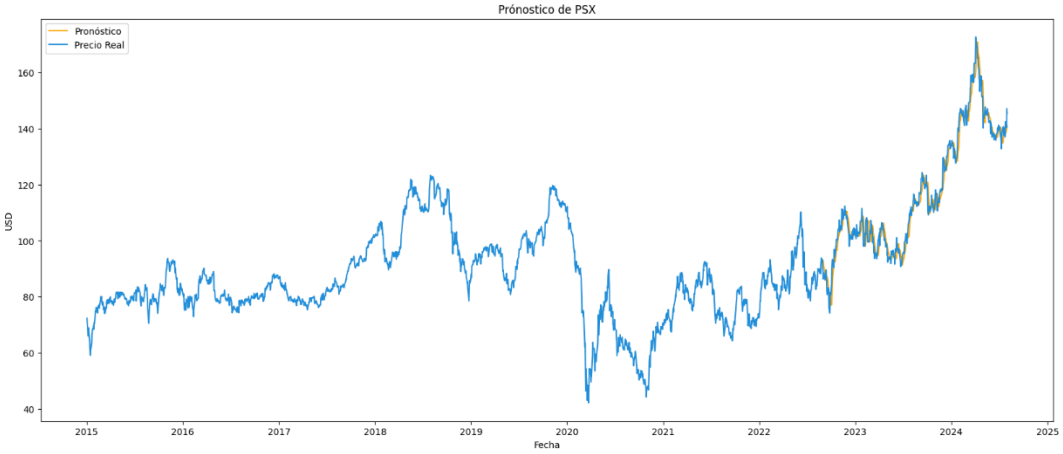
Nota. Elaboración Propia.

PSX (Phillips 66):

Phillips 66 es una empresa multinacional de energía especializada en refinación, marketing y distribución de productos petroleros. Fundada en 2012 tras la escisión de ConocoPhillips, tiene su sede en Houston, Texas. Sus operaciones incluyen refinación de crudo, producción de productos petroquímicos y la gestión de infraestructura energética. Phillips 66 también se centra en el desarrollo de combustibles limpios y energías renovables. La empresa juega un papel clave en la infraestructura energética de EE.UU. y tiene una fuerte red de estaciones de servicio.

Figura 25

Pronóstico acción PSX.



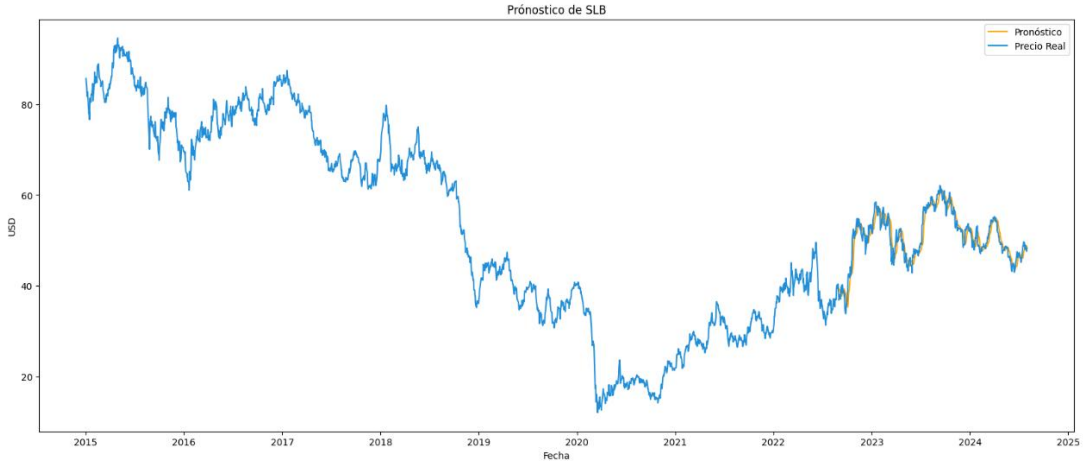
Nota. Elaboración Propia.

SLB (Schlumberger Limited):

Schlumberger es la mayor empresa de servicios petroleros del mundo, proporcionando tecnología para la perforación y exploración de petróleo y gas. Fundada en 1926, con sede en Houston, Texas, y París, Francia, la empresa ofrece soluciones en ingeniería de pozos, análisis de datos geológicos y sistemas submarinos. Schlumberger también impulsa la transformación digital en la industria energética mediante el uso de inteligencia artificial y análisis de datos. La compañía está invirtiendo en tecnologías limpias y energías renovables para diversificar su portafolio en respuesta a la transición energética.

Figura *26*

Pronóstico acción SLB.



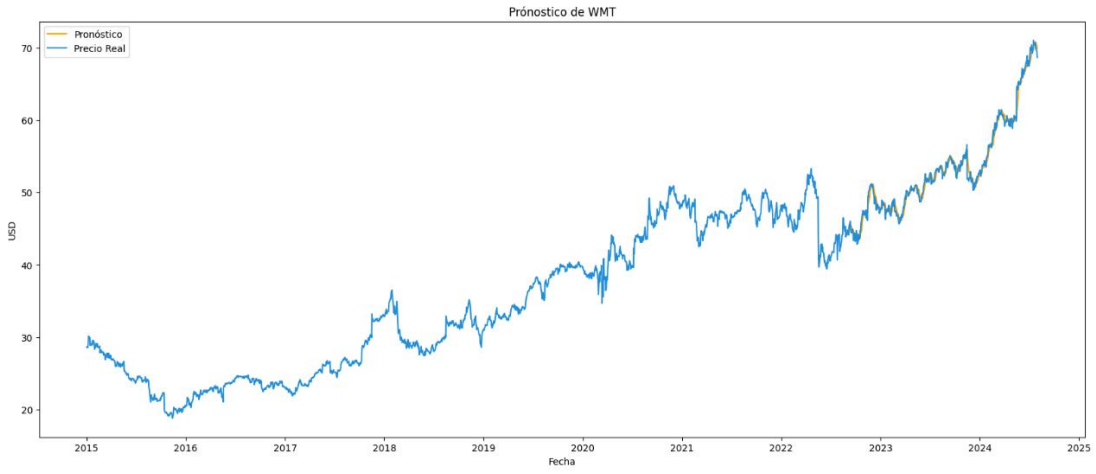
Nota. Elaboración Propia.

WMT (Walmart Inc.):

Walmart es la mayor cadena minorista del mundo, fundada en 1962 y con sede en Bentonville, Arkansas. Con más de 11,000 tiendas en 27 países, Walmart se especializa en la venta al por menor de productos a bajo costo. La empresa también tiene una fuerte presencia en el comercio electrónico a través de Walmart.com. Walmart ofrece una amplia gama de productos, desde alimentos y ropa hasta electrónica y productos del hogar. Con un enfoque en la sostenibilidad y la tecnología, Walmart trabaja para mejorar su cadena de suministro y reducir su huella de carbono.

Figura 27

Pronóstico acción WMT.



Nota. Elaboración Propia.

4.3 Librerías Utilizadas.

En el desarrollo de proyectos de análisis de datos, aprendizaje automático y simulación de entornos, para el uso de modelos basados en Deep Learning y Reinforcement Learning, el uso de librerías en Python es fundamental para maximizar la eficiencia y la precisión de los resultados. Las librerías proporcionan herramientas y funcionalidades preconstruidas que simplifican tareas complejas, permitiendo enfocarse en la creación he importación de modelos de manera sencilla. Dentro de las librerías utilizadas podemos mencionar:

- **TensorFlow:** Es una biblioteca de código abierto desarrollada por Google para el cálculo numérico y el aprendizaje automático. Proporciona herramientas y recursos para construir y entrenar modelos de aprendizaje profundo y otros algoritmos de machine learning. TensorFlow facilita el trabajo con redes neuronales, desde su definición hasta el entrenamiento y la implementación en producción a partir de la definición de sus principales Hiperparámetros.
- **TensorFlow.keras:** Consiste en una API de alto nivel dentro de TensorFlow que proporciona una interfaz sencilla para construir y entrenar modelos de redes neuronales. Es una implementación de Keras en TensorFlow que simplifica la creación de modelos mediante una sintaxis más intuitiva y accesible, ofreciendo una amplia gama de capas, optimizadores y funciones de pérdida para desarrollar modelos de aprendizaje profundo.
- **Gym:** Biblioteca de código abierto creada por OpenAI que proporciona una variedad de entornos para el desarrollo y evaluación de algoritmos de aprendizaje por refuerzo. Los entornos en Gym incluyen simulaciones y tareas estándar que permiten a los

agentes aprender a través de la interacción con el ambiente, facilitando la experimentación y comparación de diferentes enfoques de aprendizaje por refuerzo.

- **Gym-anytrading:** Es una extensión de Gym que proporciona entornos específicos para la simulación de tareas en trading y finanzas. Esta biblioteca incluye entornos que modelan datos de series temporales financieras, permitiendo el desarrollo y prueba de algoritmos de aprendizaje por refuerzo en el contexto de trading y predicción de mercados financieros.
- **stable-baselines3:** Es una biblioteca de código abierto que implementa algoritmos de aprendizaje por refuerzo en Python, basada en el framework Gym. Proporciona una implementación sencilla y confiable de varios algoritmos de aprendizaje por refuerzo, como PPO, A2C, y DQN, con el objetivo de facilitar la investigación y el desarrollo siendo fácil de usar y extender, ofreciendo herramientas para la experimentación y la implementación en producción.

Cada una de estas librerías fueron utilizadas para el desarrollo del código utilizado en la construcción de la Red Neuronal y del algoritmo de Actor-Crítico con Ventaja (A2C) por medio de Google Colab.

4.4 Parámetros elegidos para cada modelo.

Para la aplicación de las Redes Neuronales (NN) y el modelo de Reinforcement Learning de Actor-Crítico con Ventaja (A2C) es necesario la elección de sus parámetros a elegir o establecidos por defecto, para así determinar si poseen algún grado de influencia en los resultados finales. Por ello se designan los siguientes parámetros.

4.4.1 Parámetros relevantes utilizados para la Red Neuronal.

Para la construcción de la red neuronal es ecesario la utilización de TensorFlow y Keras para generar los valores pronosticados a partir de las series temporales con los datos de entrenamiento, para lo cual fue necesario una previa transformación de los datos en 'ventanas' y 'horizontes'. Respecto a los datos de ventanas se tienen asignados a las variables "train_windows" y "test_windows" un conjunto precios de 7 días consecutivos, pero además se utilizaron como horizontes "train_labels" y "test_labels" los cuales contenian los precios del día 8 (que venía después del último día de ventana). Esta transformación en los datos se hizo con la finalidad de que la Red Neuronal pueda acaparar una mayor cantidad de datos dentro de sus Batches.

Para la importación de los datos en la red neuronal, se establece una semilla aleatoria con tf.random.set_seed(42) para garantizar que los resultados sean reproducibles. Esto es importante en experimentos con redes neuronales ya que si no se inicia mediante elementos aleatorios (como la inicialización de pesos), el modelo puede que produzca los mismos resultados si se ejecuta bajo las mismas condiciones.

Luego, se define el modelo utilizando tf.keras.Sequential, que es una estructura que permite apilar capas de forma lineal, donde cada capa se conecta a la siguiente. La primera

capa es una capa densa (totalmente conectada) con 128 neuronas y activación ReLU. La función de activación ReLU (Rectified Linear Unit) es una no linealidad común que ayuda a la red a aprender representaciones complejas sin problemas de saturación, como ocurre con funciones de activación más antiguas (como sigmoide o tanh). Esta capa se encarga de capturar patrones no lineales en los datos. La segunda capa es otra capa densa, pero con un número de neuronas igual a HORIZON, estando relacionado con el horizonte de predicción en un problema de series temporales. Aquí se utiliza una activación lineal, que es adecuada para tareas de regresión donde se espera que las salidas sean valores continuos y no transformados.

El modelo se compila con la función de pérdida de error cuadrático medio (MSE), que mide la diferencia entre las predicciones del modelo y los valores reales, penalizando más los errores grandes. Se utiliza el optimizador Adam, que es un optimizador basado en el gradiente descendente capaz de ajustarse dinámicamente al ritmo de aprendizaje, pero también porque combina las ventajas de dos optimizadores más simples, AdaGrad y RMSProp, lo que lo hace adecuado para problemas con datos ruidosos o dinámicos (Diederik P. Kingma, 2015). Además, se especifican dos métricas para monitorear el rendimiento: MSE y MAE (error absoluto medio), lo que proporciona una mejor idea de cómo el modelo está aprendiendo y qué tan precisas son sus predicciones.

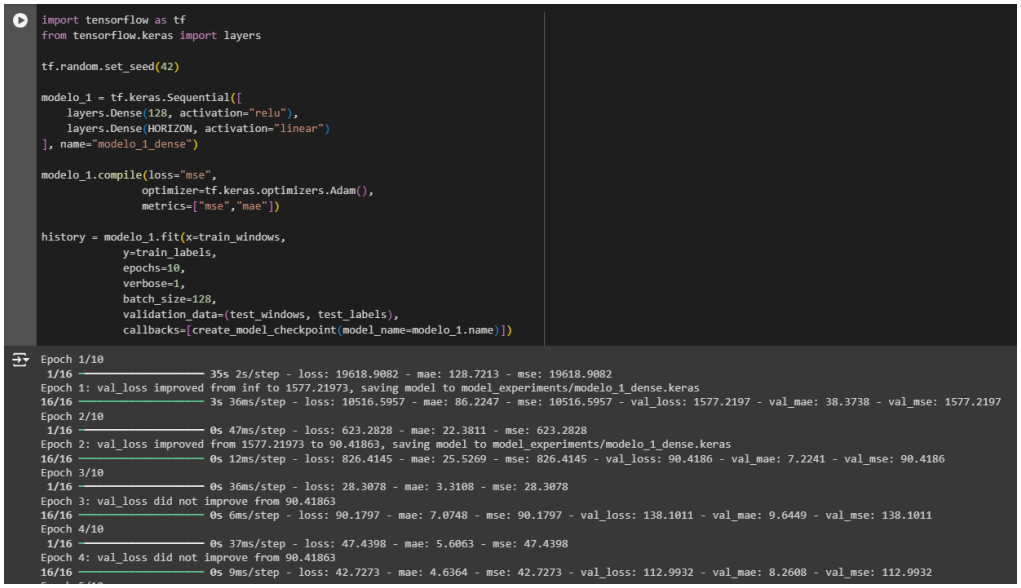
Finalmente, el modelo se entrena usando el método fit(), pasando como entrada las ventanas y horizonte de datos de entrenamiento definidas por (train_windows) y (train_labels). El modelo se entrena durante 10 épocas, lo que significa que verá todo el conjunto de datos de entrenamiento 10 veces, con un tamaño de lote de 128. Esto significa que, en cada iteración, se ajustarán los pesos del modelo basándose en 128 muestras antes de

realizar una actualización del gradiente. También se proporciona un conjunto de validación (test_windows y test_labels) para evaluar el rendimiento del modelo después de cada época, ayudando a monitorear si está sobreajustando los datos de entrenamiento.

Además de lo anterior, también se utiliza un callback mediante la funicón create_model_checkpoint, el cual se usa para guardar automáticamente el mejor modelo durante el entrenamiento, asegurando que el modelo final no solo esté bien ajustado a los datos de entrenamiento, sino que también generalice bien a los datos no vistos (validación).

Figura 28

Código de la Red Neuronal Utilizado.



```
import tensorflow as tf
from tensorflow.keras import layers

tf.random.set_seed(42)

modelo_1 = tf.keras.Sequential([
    layers.Dense(128, activation="relu"),
    layers.Dense(HORIZON, activation="linear")
], name="modelo_1_dense")

modelo_1.compile(loss="mse",
                 optimizer=tf.keras.optimizers.Adam(),
                 metrics=["mse","mae"])

history = modelo_1.fit(x=train_windows,
            y=train_labels,
            epochs=10,
            verbose=1,
            batch_size=128,
            validation_data=(test_windows, test_labels),
            callbacks=[create_model_checkpoint(model_name=modelo_1.name)])
```

```
Epoch 1/10
 1/16 ━━━━━━━━━━━━━━━━━━━━ 35s 2s/step - loss: 19618.9082 - mae: 128.7213 - mse: 19618.9082
Epoch 1: val_loss improved from inf to 1577.21973, saving model to model_experiments/modelo_1_dense.keras
16/16 ━━━━━━━━━━━━━━━━━━━━ 3s 36ms/step - loss: 10516.5957 - mae: 86.2247 - mse: 10516.5957 - val_loss: 1577.2197 - val_mae: 38.3738 - val_mse: 1577.2197
Epoch 2/10
 1/16 ━━━━━━━━━━━━━━━━━━━━ 0s 47ms/step - loss: 623.2828 - mae: 22.3811 - mse: 623.2828
Epoch 2: val_loss improved from 1577.21973 to 90.41863, saving model to model_experiments/modelo_1_dense.keras
16/16 ━━━━━━━━━━━━━━━━━━━━ 0s 12ms/step - loss: 826.4145 - mae: 25.5269 - mse: 826.4145 - val_loss: 90.4186 - val_mae: 7.2241 - val_mse: 90.4186
Epoch 3/10
 1/16 ━━━━━━━━━━━━━━━━━━━━ 0s 36ms/step - loss: 28.3078 - mae: 3.3108 - mse: 28.3078
Epoch 3: val_loss did not improve from 90.41863
16/16 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - loss: 90.1797 - mae: 7.0748 - mse: 90.1797 - val_loss: 138.1011 - val_mae: 9.6449 - val_mse: 138.1011
Epoch 4/10
 1/16 ━━━━━━━━━━━━━━━━━━━━ 0s 37ms/step - loss: 47.4398 - mae: 5.6063 - mse: 47.4398
Epoch 4: val_loss did not improve from 90.41863
16/16 ━━━━━━━━━━━━━━━━━━━━ 0s 9ms/step - loss: 42.7273 - mae: 4.6364 - mse: 42.7273 - val_loss: 112.9932 - val_mae: 8.2608 - val_mse: 112.9932
```

Nota. Elaboración Propia.

4.4.2 Parámetros relevantes del Algoritmo de Actor-Crítico con ventaja (A2C).

Para el caso del Algoritmo de Actor-Crítico con Ventaja (A2C) y su construcción del código en Python, se entrenó un modelo de aprendizaje por refuerzo utilizando la librería stable_baselines3 y gym_anytrading. Los parámetros y componentes clave utilizados del código fueron los siguientes:

- env_maker: Esta función crea un entorno para el modelo utilizando datos de acciones ('stocks-v0') utilizando el DataFrame en la variable "df_RL" siendo la cantidad de precios de cierre de los distintos días, a su vez, configurando el entorno con frame_bound el cual determinaba la ventana de tiempo sobre la cual se hacen las predicciones.

- A2C: Es el algoritmo Actor-Critic Advantage (A2C) utilizado que ayuda al modelo a aprender políticas de acciones óptimas basadas en los datos de entrada del entorno.

- ActorCriticPolicy: Es la política que el modelo A2C utilizará para hacer predicciones. Esta política integra tanto el actor que propone acciones como el crítico que evalúa esas acciones.

- total_timesteps: Es el número total de pasos de tiempo que el modelo aprenderá antes de detenerse. Estando configurado para 20000 pasos.

- verbose=2: Este parámetro configura el modelo para que muestre mensajes detallados sobre su aprendizaje, lo cual es útil para depurar y entender el rendimiento del modelo y los cálculos que en ella realiza de acuerdo con su funcionamiento.

A lo que respecta a los resultados del entrenamiento, son arrojados en tablas que contienen varios indicadores del proceso de aprendizaje del modelo:

- time/fps: Frames por segundos procesados, que indica la velocidad de procesamiento.
- iterations: Número de iteraciones completadas.
- time_elapsed: Tiempo total transcurrido durante el entrenamiento en segundos.
- entropy_loss: Pérdida de entropía, que mide cuán predecible o incierta es la política del actor. Una entropía menor generalmente indica una política más determinística (Shah, 2023).

$$H(\pi) = -\sum_{a} \pi(a|s) \log \pi\,(a|s)$$

Con $\pi(a|s)$ siendo la probabilidad de tomar la acción a dado un estado s. La pérdida de entropía es negativa porque el objetivo es minimizar la pérdida total del modelo. Durante el entrenamiento, esta métrica puede disminuir conforme el agente desarrolla una política más determinista al aumentar la probabilidad de acción en un estado.

- explained_variance: Muestra qué tan bien la función de valor predice la recompensa real obtenida.

$$\text{Explained Variance} = 1 - \frac{\text{Var}\big(R - V(s)\big)}{\text{Var}(R)}$$

Siendo R las recompensas reales obtenidas, $V(s)$ el valor estimado por la red en el

estado s, y Var(R) la varianza de las recompensas.

Un valor de 1 indica una perfecta predicción de las recompensas, mientras que valores cercanos a 0 indican que la función de valor no está capturando bien las recompensas.

- learning_rate: Tasa de aprendizaje utilizada en el entrenamiento.
- n_updates: Número de veces que la red neuronal se actualizó.
- policy_loss: Indicador de qué tan bien el modelo está optimizando su rendimiento, midiendo el rendimiento de la política del modelo y se calcula generalmente mediante una función de pérdida basada en el gradiente de política, siendo $A(s, a)$ la ventaja estimada, que indica cuánto mejor fue la acción a en el estado s en comparación con la media esperada (Shah, 2023).

$$L_{\text{policy}} = -E[\log \pi(a|s) A(s, a)]$$

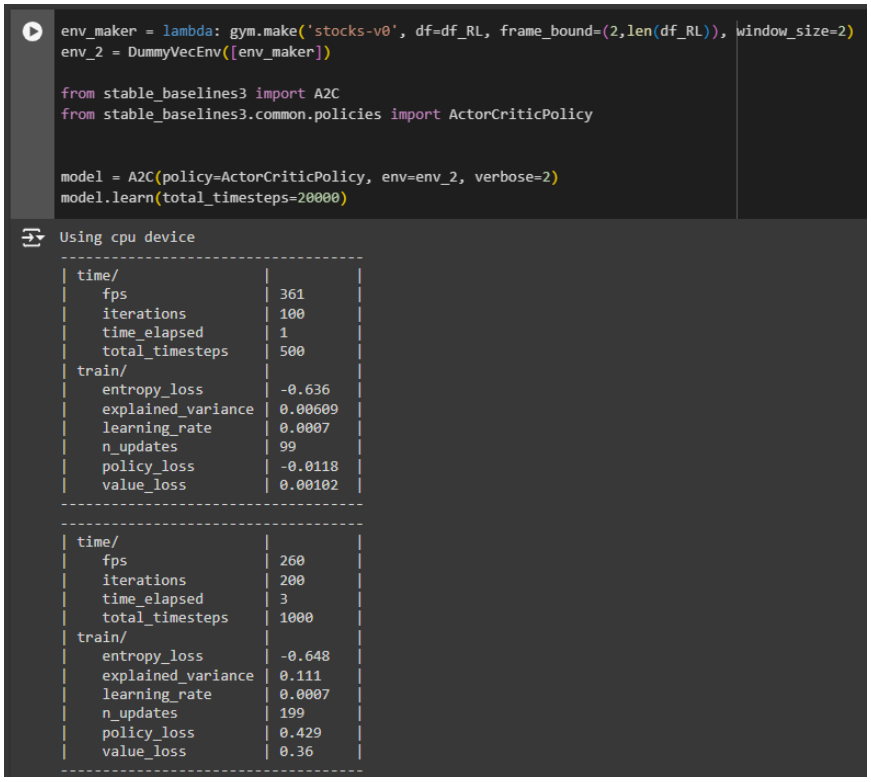
- value_loss: La pérdida de valor mide el error en la estimación del valor del estado. Esto se calcula mediante una función de pérdida de error cuadrático medio (MSE) entre el valor estimado y el valor de la recompensa real, siendo R las recompensas reales obtenidas y $V(s)$ el valor estimado por la red en el estado s (Femminella, 2024).

$$L_{\text{value}} = \frac{1}{2} E\left[\left(R - V(s)\right)^2\right]$$

Cada una de estas métricas ayuda a entender el rendimiento y la eficiencia del modelo durante el entrenamiento, permitiéndote ajustar parámetros para mejorar los resultados.

Figura 29

Código del algoritmo A2C entrenando.



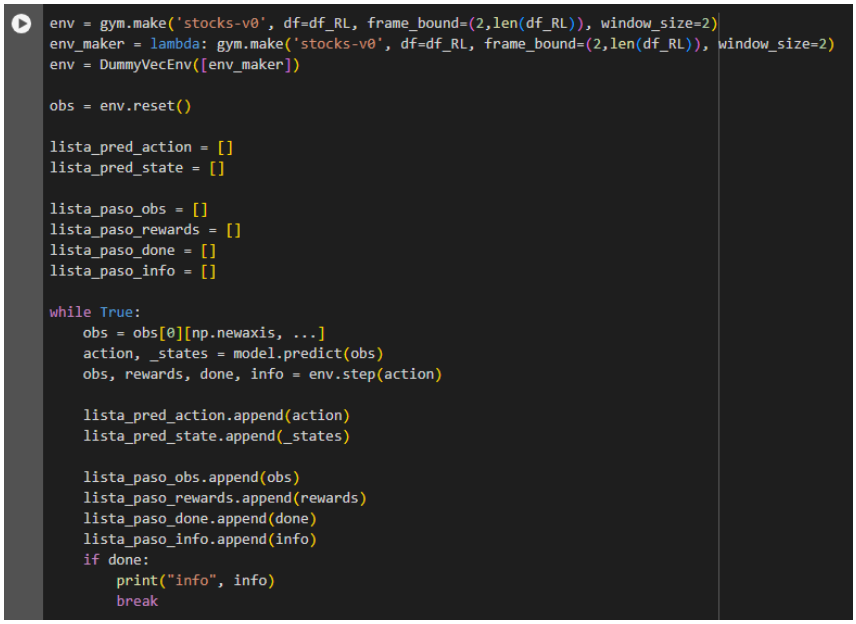
```
env_maker = lambda: gym.make('stocks-v0', df=df_RL, frame_bound=(2,len(df_RL)), window_size=2)
env_2 = DummyVecEnv([env_maker])

from stable_baselines3 import A2C
from stable_baselines3.common.policies import ActorCriticPolicy


model = A2C(policy=ActorCriticPolicy, env=env_2, verbose=2)
model.learn(total_timesteps=20000)
```

```
Using cpu device
-----------------------------------
| time/                 |         |
|    fps                | 361     |
|    iterations         | 100     |
|    time_elapsed       | 1       |
|    total_timesteps    | 500     |
| train/                |         |
|    entropy_loss       | -0.636  |
|    explained_variance | 0.00609 |
|    learning_rate      | 0.0007  |
|    n_updates          | 99      |
|    policy_loss        | -0.0118 |
|    value_loss         | 0.00102 |
-----------------------------------
-----------------------------------
| time/                 |         |
|    fps                | 260     |
|    iterations         | 200     |
|    time_elapsed       | 3       |
|    total_timesteps    | 1000    |
| train/                |         |
|    entropy_loss       | -0.648  |
|    explained_variance | 0.111   |
|    learning_rate      | 0.0007  |
|    n_updates          | 199     |
|    policy_loss        | 0.429   |
|    value_loss         | 0.36    |
-----------------------------------
```

Nota. Elaboración Propia.

Figura 30

Resultados del algoritmo A2C utilizado.



```
env = gym.make('stocks-v0', df=df_RL, frame_bound=(2,len(df_RL)), window_size=2)
env_maker = lambda: gym.make('stocks-v0', df=df_RL, frame_bound=(2,len(df_RL)), window_size=2)
env = DummyVecEnv([env_maker])

obs = env.reset()

lista_pred_action = []
lista_pred_state = []

lista_paso_obs = []
lista_paso_rewards = []
lista_paso_done = []
lista_paso_info = []

while True:
    obs = obs[0][np.newaxis, ...]
    action, _states = model.predict(obs)
    obs, rewards, done, info = env.step(action)

    lista_pred_action.append(action)
    lista_pred_state.append(_states)

    lista_paso_obs.append(obs)
    lista_paso_rewards.append(rewards)
    lista_paso_done.append(done)
    lista_paso_info.append(info)
    if done:
        print("info", info)
        break
```

Nota. Elaboración Propia.

4.5 Estrategia de Buy & Hold.

Con el fin de concluir respecto a la fiabilidad en el uso de las NN y el algoritmo A2C, se calculó la rentabilidad con respecto a la compra y posterior venta de cada acción utilizando la estrategia de Buy & Hold. Para ello se compró en el primer día del horizonte evaluado (al 31/08/2022) y se vendió cada acción al final del periodo evaluado (al 31/07/2024).

5. Resultados

5.1 Resultados de la Red Neuronal.

Para cada una de las acciones, se utilizó la misma Red Neuronal para obtener un pronóstico por cada acción a partir de los datos de entrenamiento, pero además se utilizó como métrica principal el cálculo del error mediante MSE para así medir el desempeño de dicho entrenamiento por cada época.

Además, para comparar los resultados obtenidos en cada época para cada una de las 17 acciones se consideraron las métricas de MSE, MAE, RMSE y MAPE, a partir de una función desarrollada importando cada fórmula de error directamente desde Tensorflow.

Todo lo anterior se hizo con los parámetros establecidos y explicados desde un inicio sin aumentar la cantidad de épocas ni nodos, todo ello con el fin de evitar un sobreajuste, pero también dado a los buenos resultados numéricos obtenidos.

Tabla 1

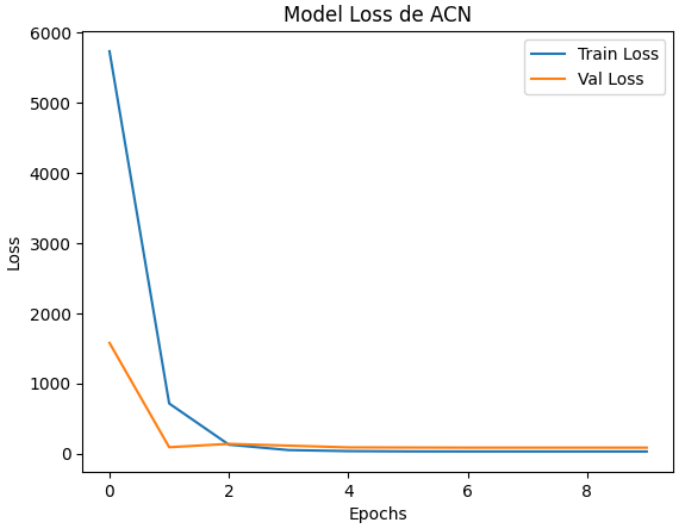
Errores obtenidos por cada acción.

ACCIÓN	MSE	MAE	RMSE	MAPE
ACN	84.29774	6.9261665	9.18138	2.320288
ADBE	436.72327	15.534253	20.897924	3.5368059
BSX	1.1314108	0.78778577	1.063678	1.4546757
CAT	104.01095	7.787251	10.198576	3.002686
CSCO	1.2073387	0.8429552	1.0987897	1.743368
ECL	19.500502	3.252942	4.4159374	1.8633366
GE	7.582764	2.0540779	2.753682	2.2671661
HPQ	0.92711	0.72724086	0.96286553	2.4571488
JPM	13.240559	2.7632473	3.638758	1.8478067
MET	3.4028628	1.4241692	1.844685	2.2122781
MMM	9.0584545	2.0782201	3.0097268	2.2536368
MU	10.511861	2.2449634	3.2422	2.7671204
PEP	7.160198	2.1027336	2.6758547	1.2050426
PG	4.5265555	1.6417483	2.1275704	1.1059706
PSX	12.831881	2.7962332	3.5821614	2.4454277
SLB	3.7084835	1.4661585	1.9257423	2.938165
WMT	0.5298363	0.5437894	0.72789854	1.0288585

Nota. Elaboración Propia.

Figura 31

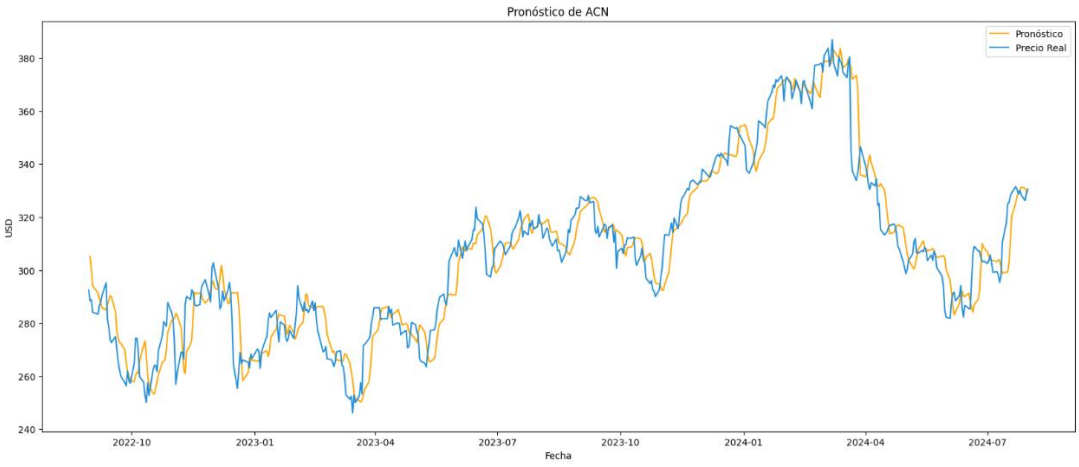
Pérdida de la acción ACN.



Nota. Elaboración Propia.

Figura 32

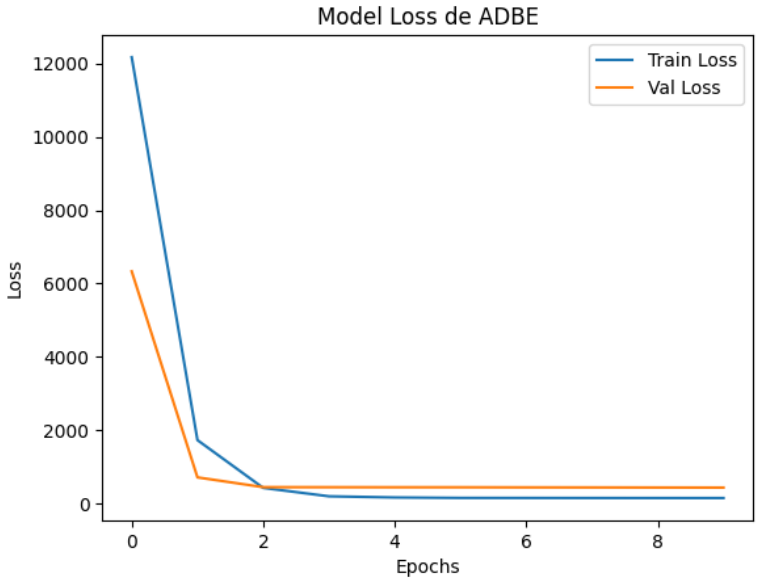
Pronóstico acción ACN.



Nota. Elaboración Propia.

Figura 33

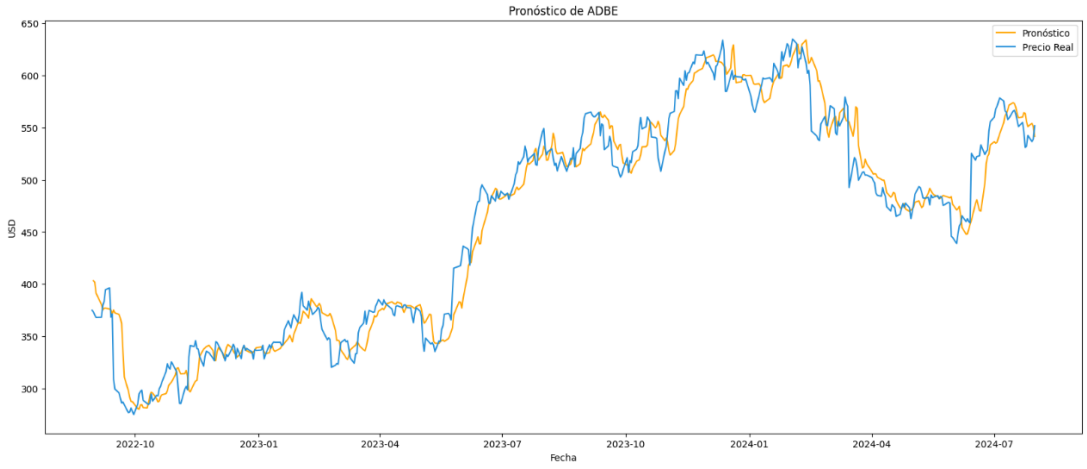
Pérdida de la acción ADBE.



Nota. Elaboración Propia.

Figura 34

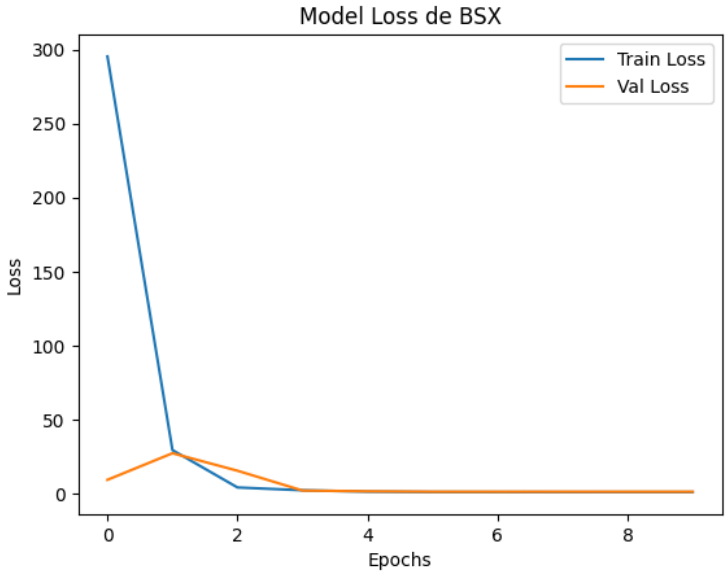
Pronóstico acción ADBE.



Nota. Elaboración Propia.

Figura 35

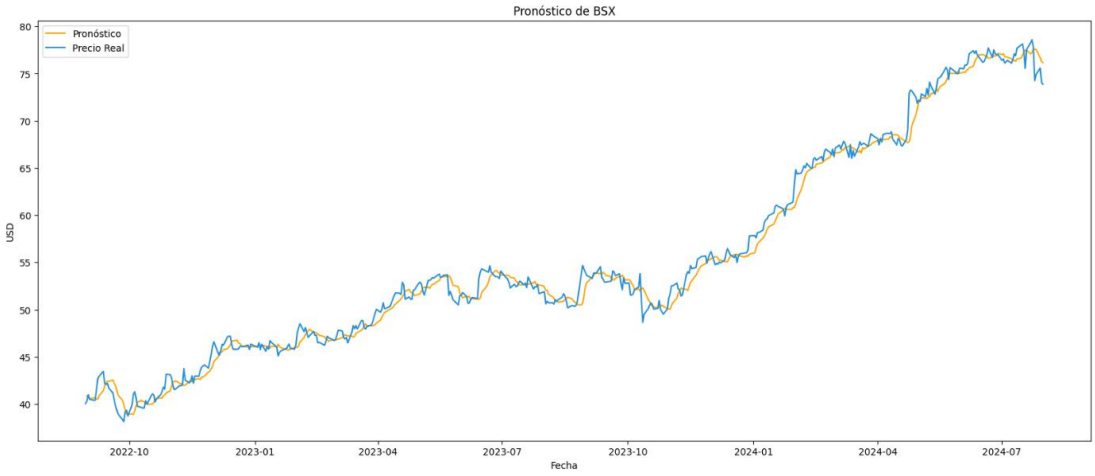
Pérdida de la acción BSX.



Nota. Elaboración Propia.

Figura 36

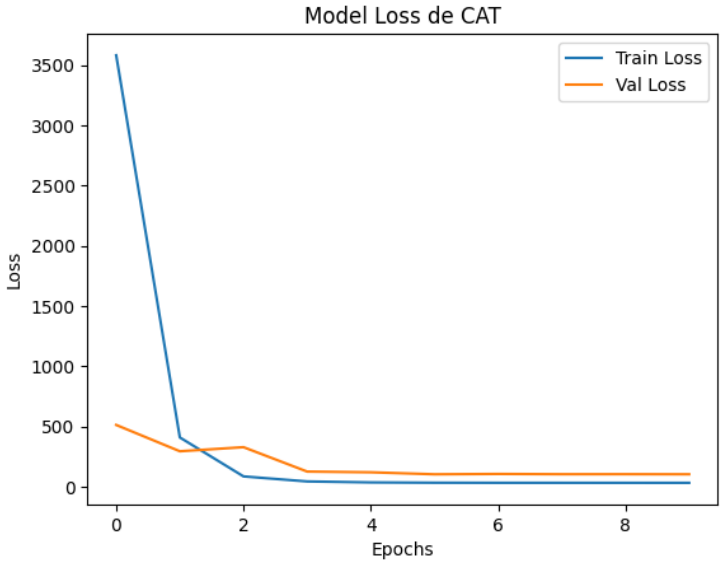
Pronóstico acción BSX.



Nota. Elaboración Propia.

Figura 37

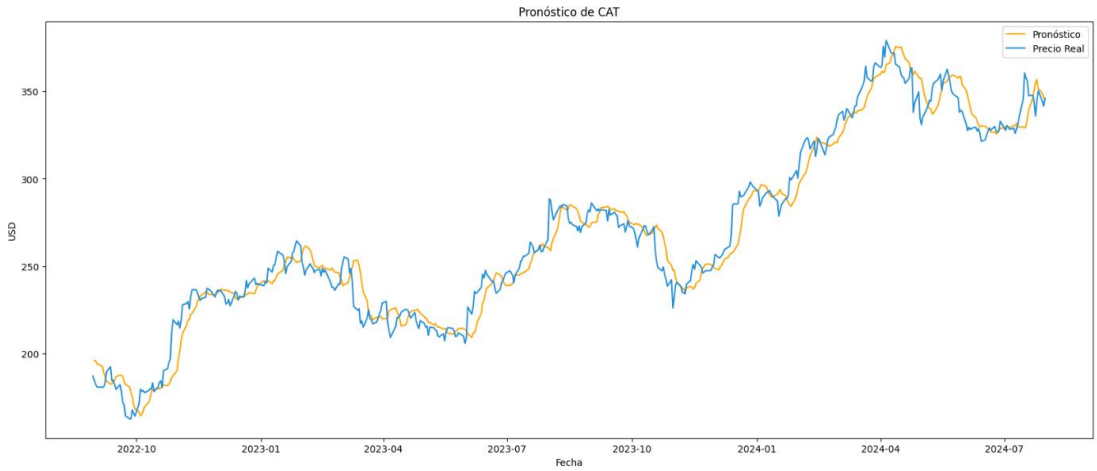
Pérdida de la acción CAT.



Nota. Elaboración Propia.

Figura 38

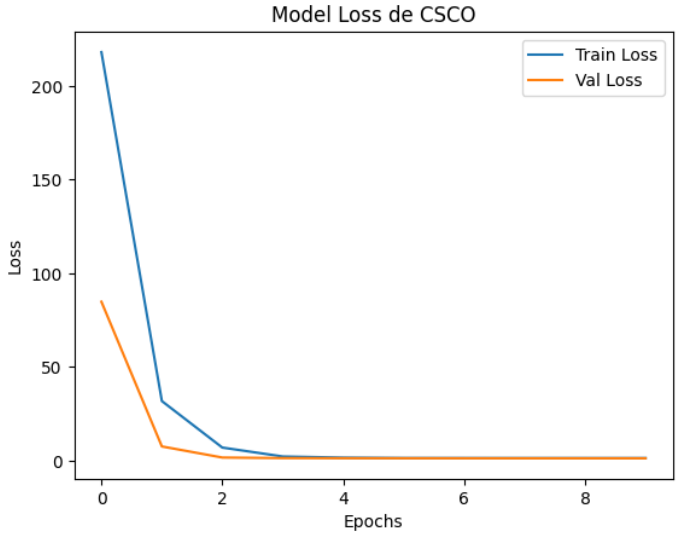
Pronóstico acción CAT.



Nota. Elaboración Propia.

Figura 39

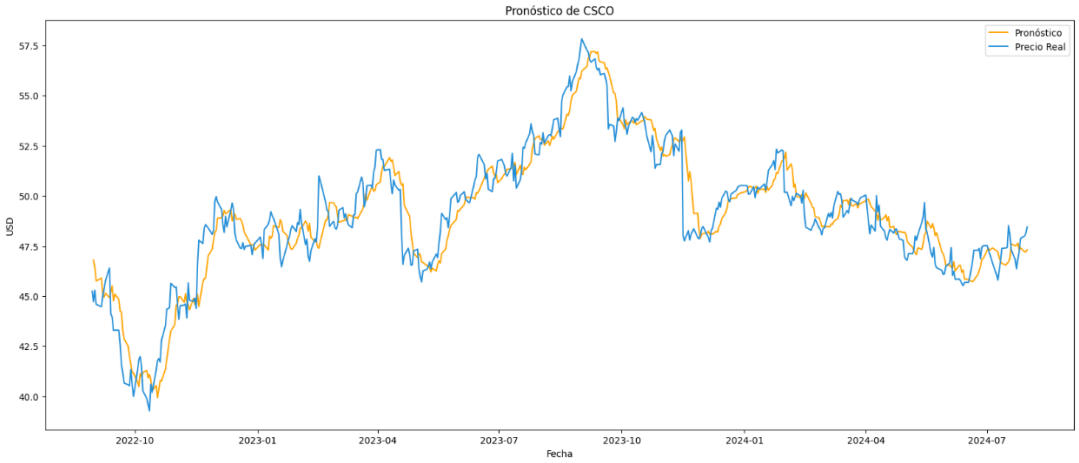
Pérdida de la acción CSCO.



Nota. Elaboración Propia.

Figura 40

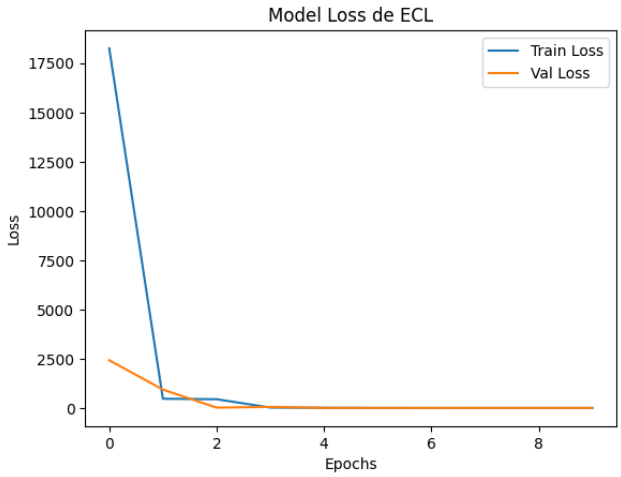
Pronóstico acción CSCO.



Nota. Elaboración Propia.

Figura 41

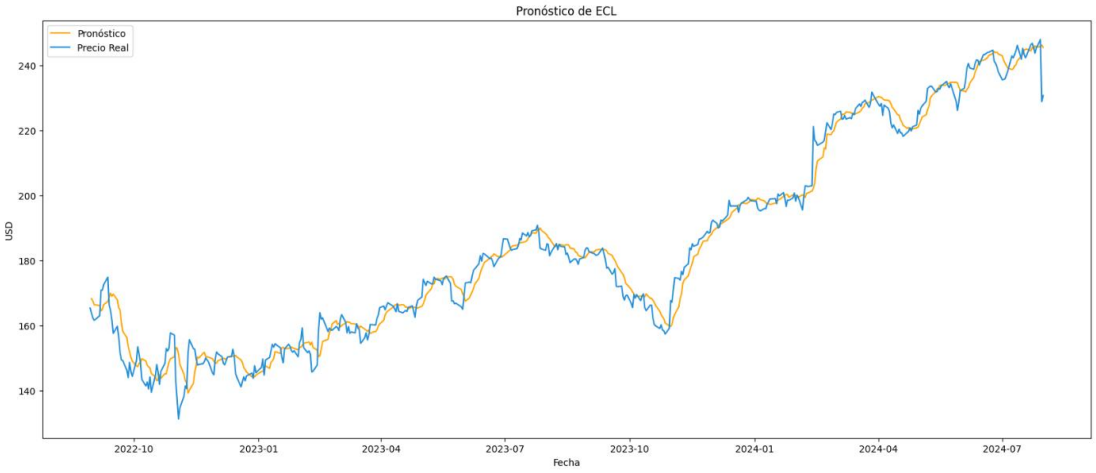
Pérdida de la acción ECL.



Nota. Elaboración Propia.

Figura 42

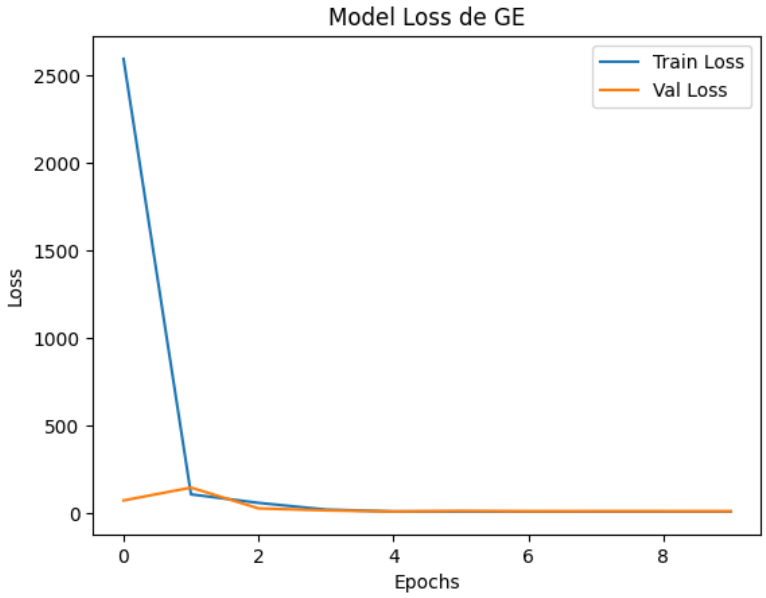
Pronóstico acción ECL.



Nota. Elaboración Propia.

Figura 43

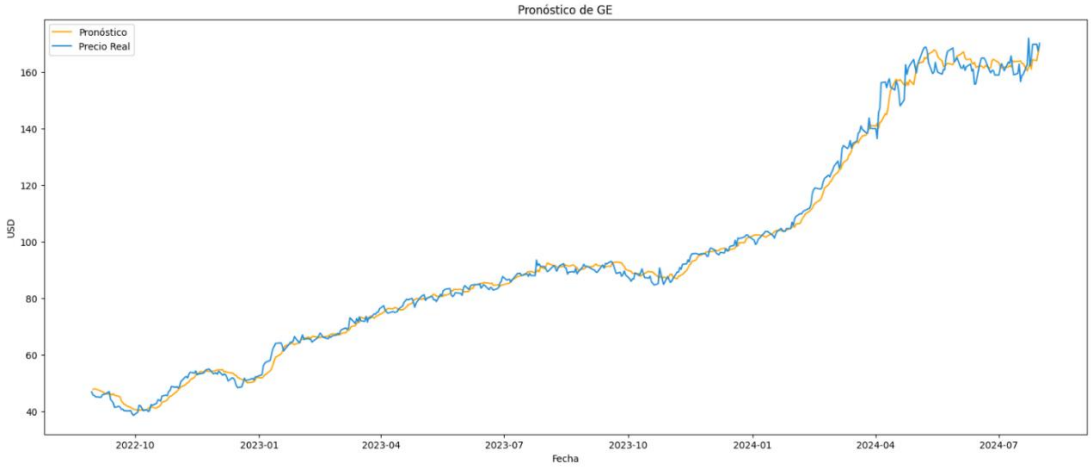
Pérdida de la acción GE.



Nota. Elaboración Propia.

Figura 44

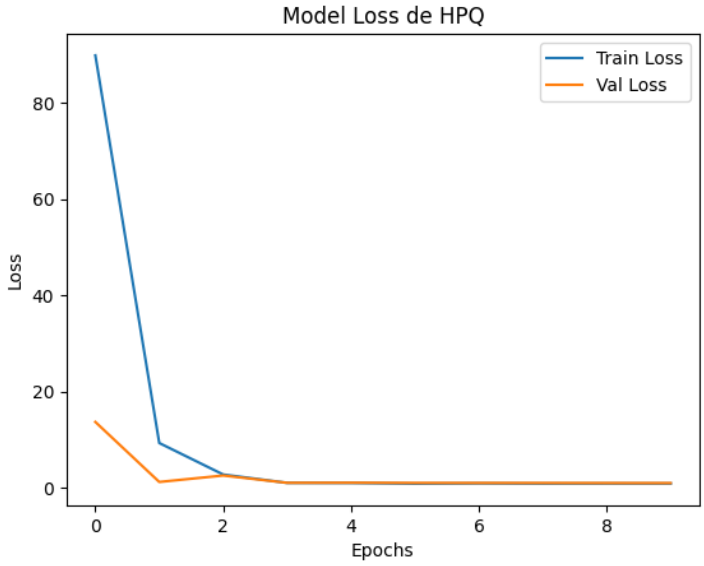
Pronóstico acción GE.



Nota. Elaboración Propia.

Figura 45

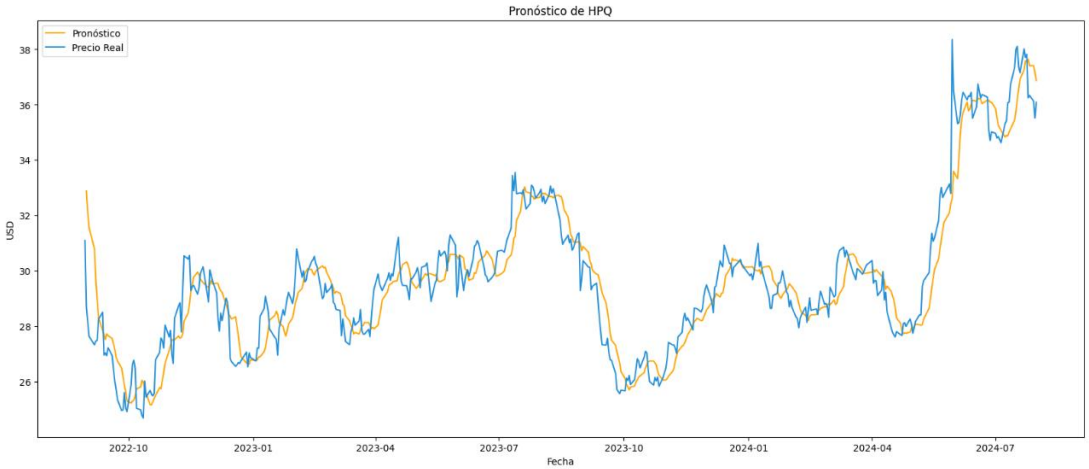
Pérdida de la acción HPQ.



Nota. Elaboración Propia.

Figura 46

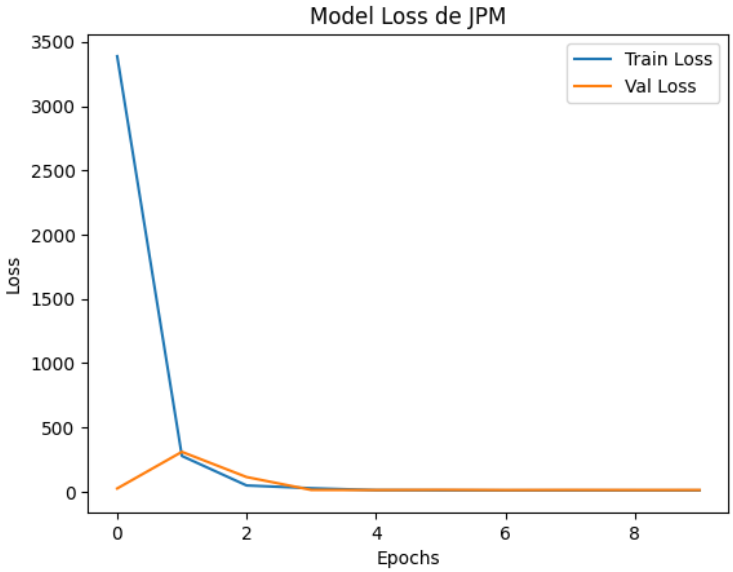
Pronóstico acción HPQ.



Nota. Elaboración Propia.

Figura 47

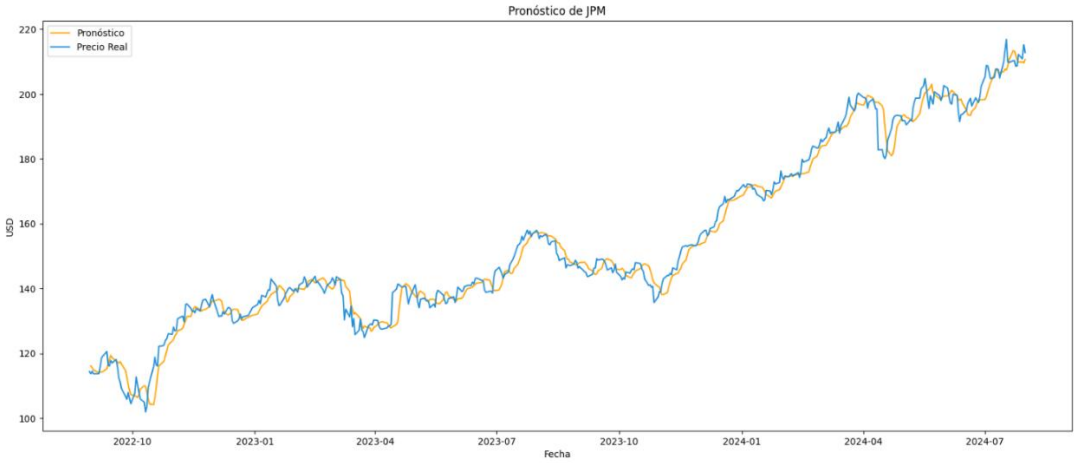
Pérdida de la acción JPM.



Nota. Elaboración Propia.

Figura 48

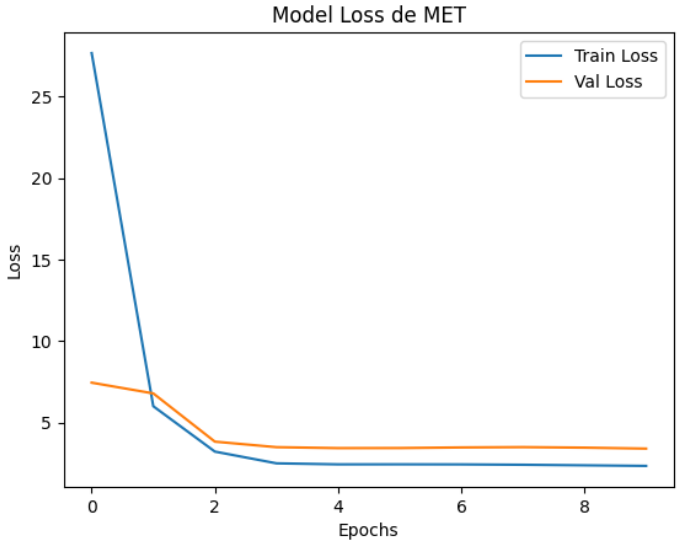
Pronóstico acción JPM.



Nota. Elaboración Propia.

Figura 49

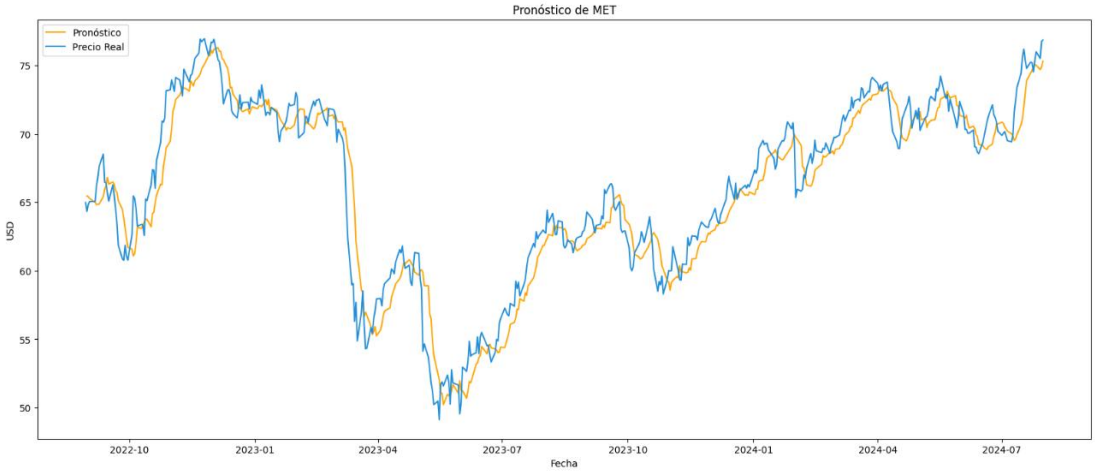
Pérdida de la acción MET.



Nota. Elaboración Propia.

Figura 50

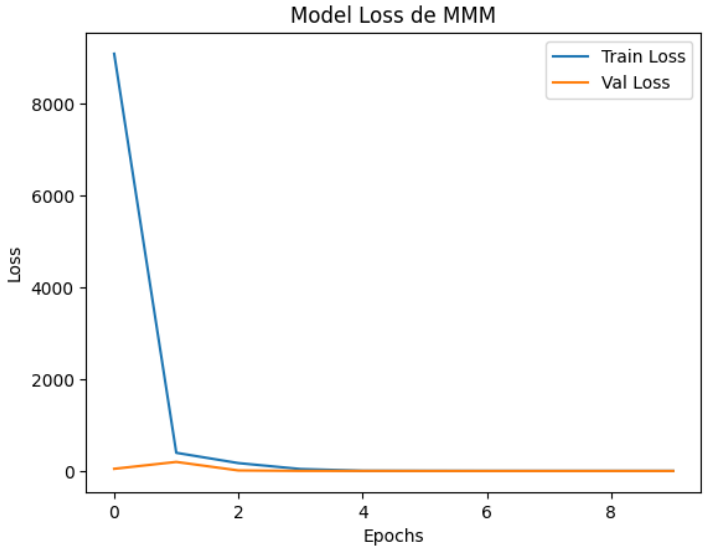
Pronóstico acción MET.



Nota. Elaboración Propia.

Figura 51

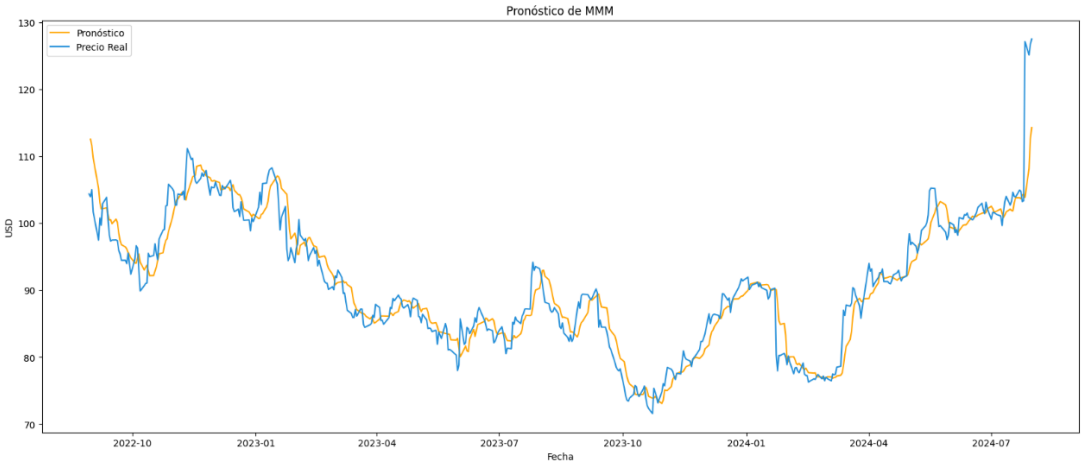
Pérdida de la acción MMM.



Nota. Elaboración Propia.

Figura 52

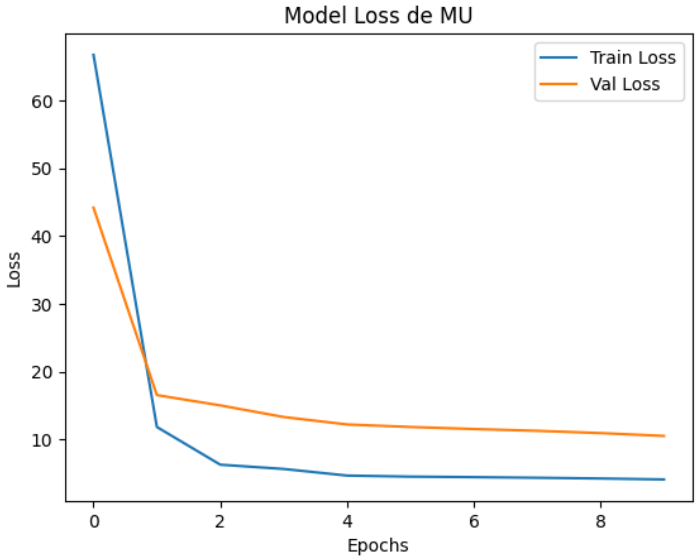
Pronóstico acción MMM.



Nota. Elaboración Propia.

Figura 53

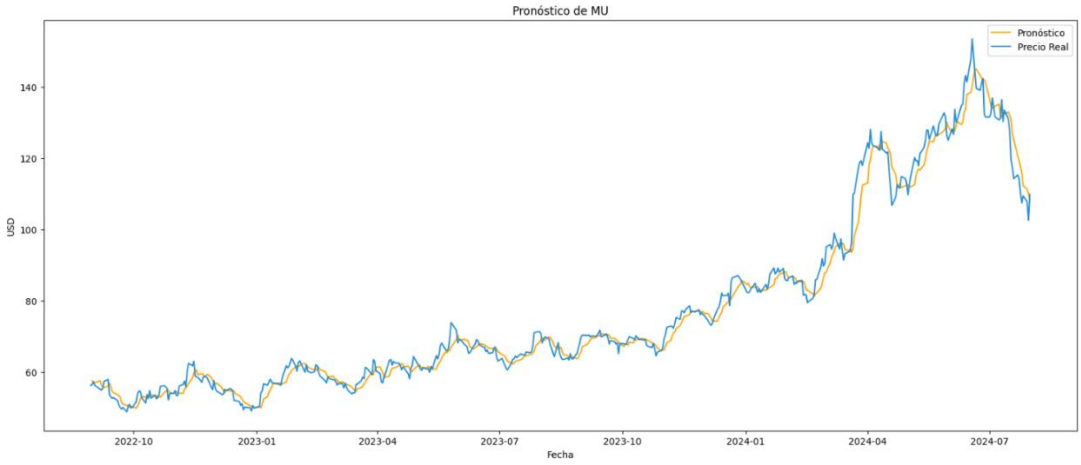
Pérdida de la acción MU.



Nota. Elaboración Propia.

Figura 54

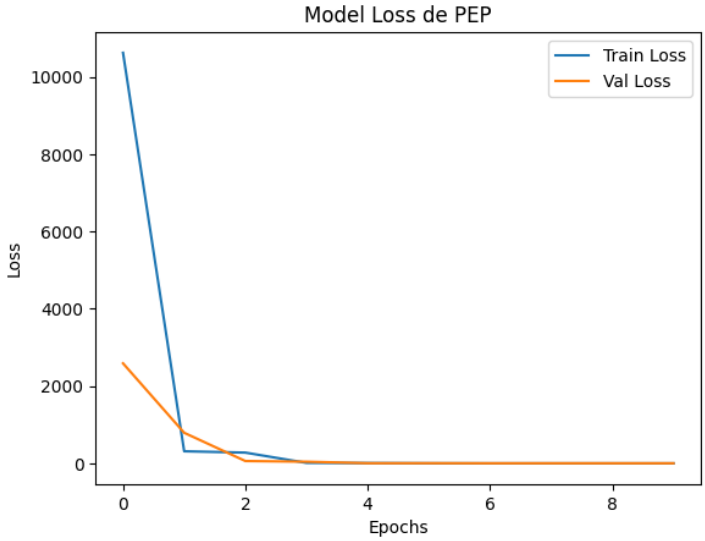
Pronóstico acción MU.



Nota. Elaboración Propia.

Figura 55

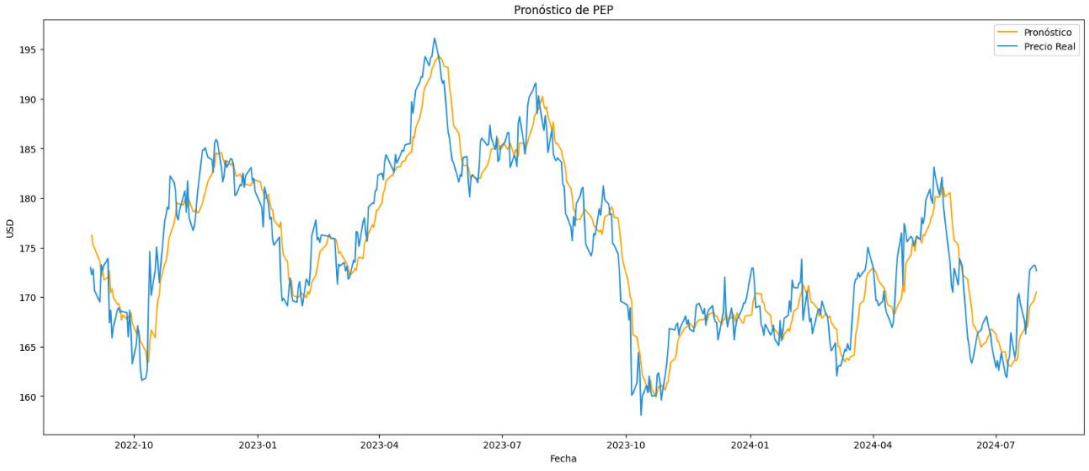
Pérdida de la acción PEP.



Nota. Elaboración Propia.

Figura 56

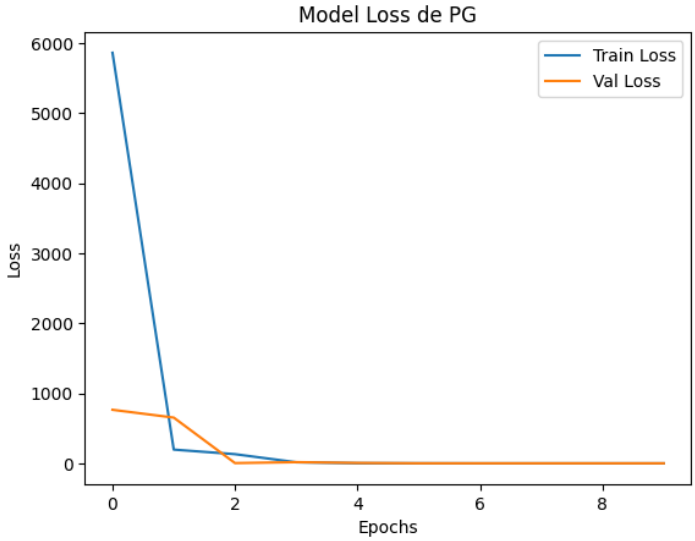
Pronóstico acción PEP.



Nota. Elaboración Propia.

Figura 57

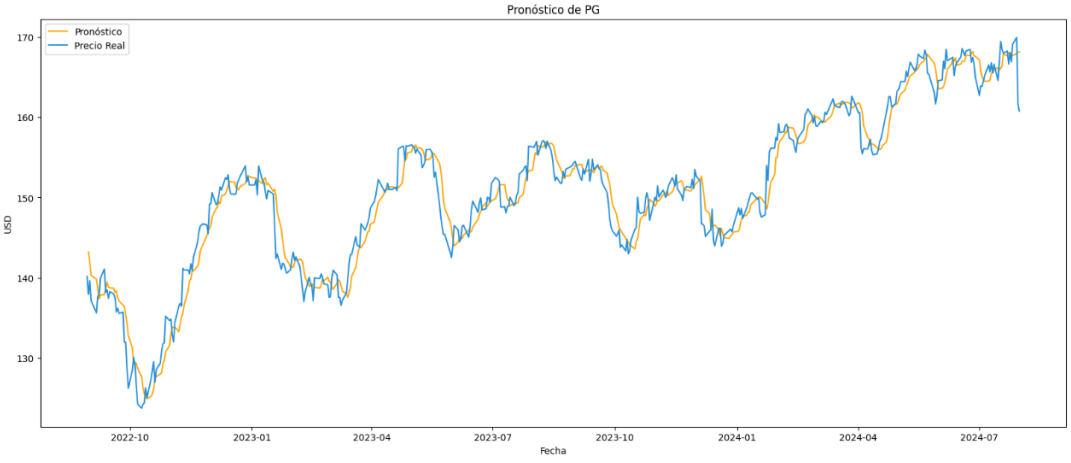
Pérdida de la acción PG.



Nota. Elaboración Propia.

Figura 58

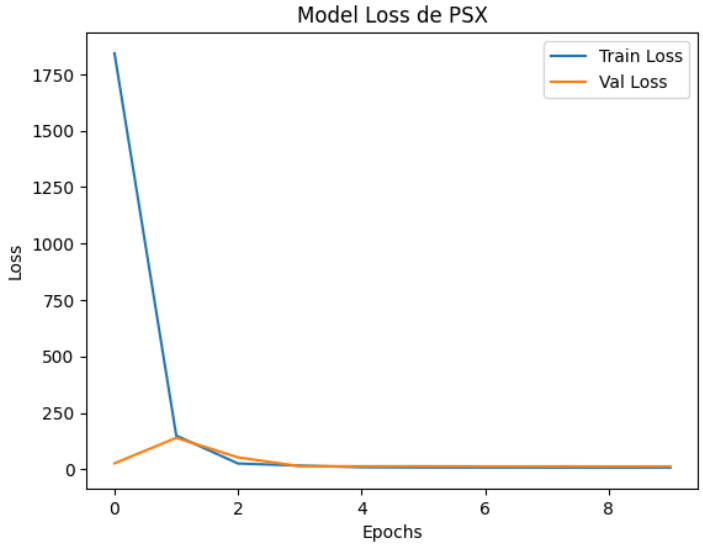
Pronóstico acción PG.



Nota. Elaboración Propia.

Figura 59

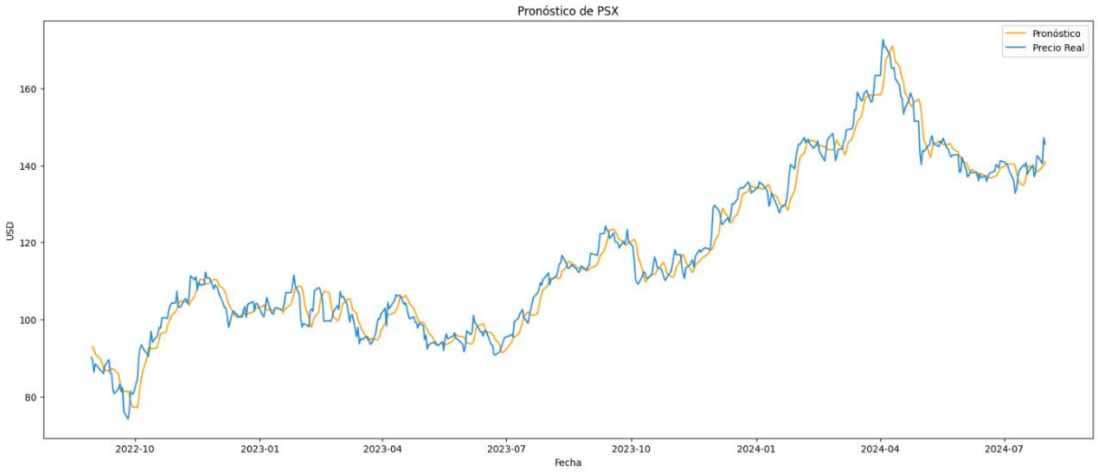
Pérdida de la acción PSX.



Nota. Elaboración Propia.

Figura 60

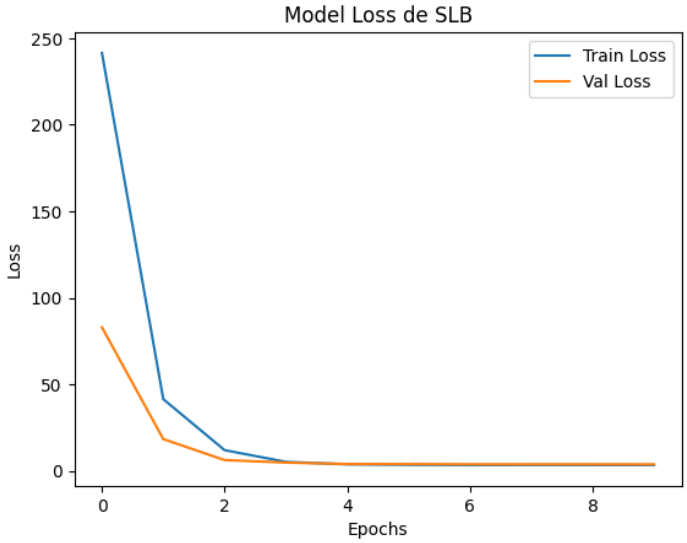
Pronóstico acción PSX.



Nota. Elaboración Propia.

Figura 61

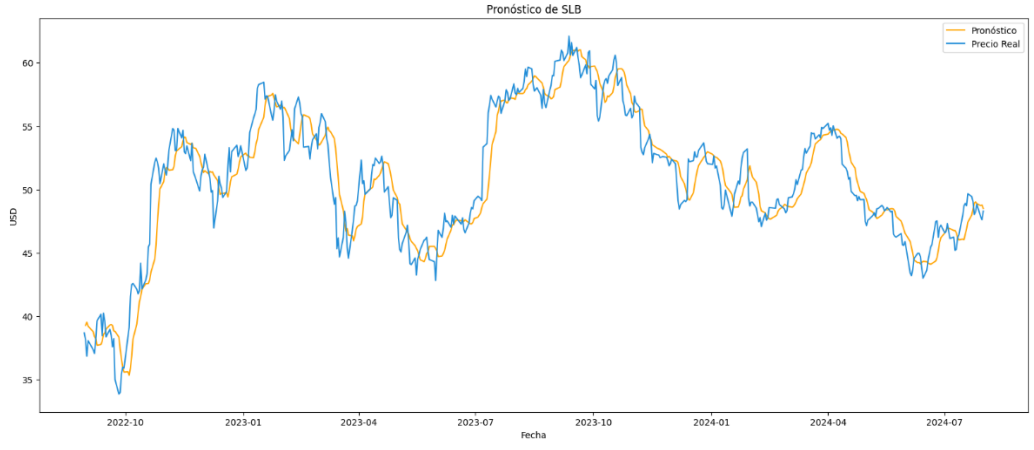
Pérdida de la acción SLB.



Nota. Elaboración Propia.

Figura 62

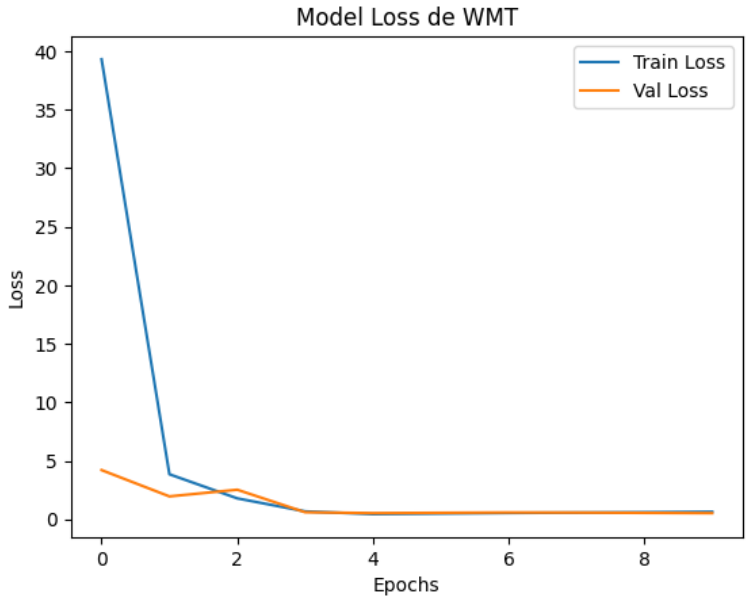
Pronóstico acción SLB.



Nota. Elaboración Propia.

Figura 63

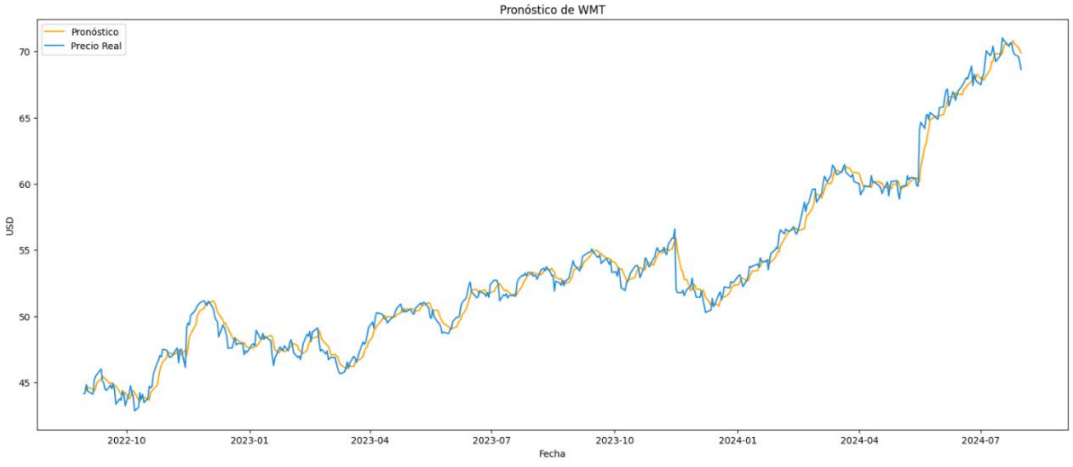
Pérdida de la acción WMT.



Nota. Elaboración Propia.

Figura 64

Pronóstico acción WMT.



Nota. Elaboración Propia.

5.2 Resultados del Algoritmo de Actor-Crítico con Ventaja (A2C).

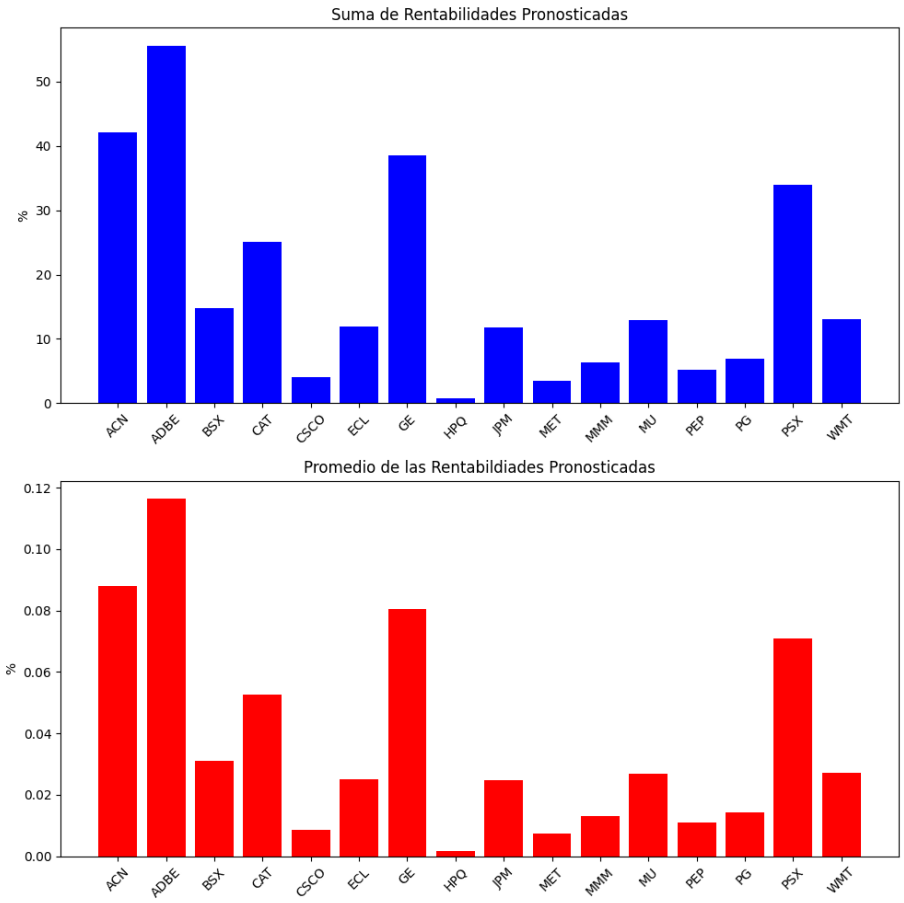
Los resultados demuestran que al entrenar el modelo A2C con el entorno de precios pronosticados por la Red Neuronal, el modelo A2C logra cumplir con el objetivo propuesto. Todas las acciones evaluadas generaron retornos positivos, lo cual sugiere que el modelo ha sido exitoso en identificar oportunidades de rentabilidad en este entorno de datos simulados. Esto refleja un desempeño positivo del modelo en las condiciones establecidas para maximizar la rentabilidad, siguiendo la política definida.

Sin embargo, al aplicar estas mismas decisiones de compra y venta en un escenario diferente en donde esta vez se utilizan los precios reales de las acciones en el mismo horizonte temporal evaluado, los resultados fueron menos favorables. En este caso, se observaron pérdidas en algunas acciones específicas, lo que sugiere que la estrategia basada en pronósticos no siempre se traduce en ganancias cuando se enfrenta a datos reales. En particular, las acciones de Adobe Inc. (ADBE), HP Inc. (HPQ), MetLife Inc. (MET) y PepsiCo Inc. (PEP) experimentaron rentabilidades promedio negativas de -0.0389%, -0.0287%, -0.0159% y -0.0142%, respectivamente. Esto indica que, a pesar de haber optimizado para condiciones simuladas, el modelo no se desempeñó igual de bien cuando las decisiones se aplicaron a los precios observados en el mercado real.

En general, entre los retornos positivos en el entorno simulado y las pérdidas en los datos reales sugiere que las estrategias de compra y venta generadas por el modelo pueden ser sensibles a las diferencias entre los datos pronosticados y los reales. En términos prácticos, esto destaca la importancia de seguir evaluando y afinando el modelo, especialmente si se planea aplicar las decisiones en un entorno de trading real.

Figura 65

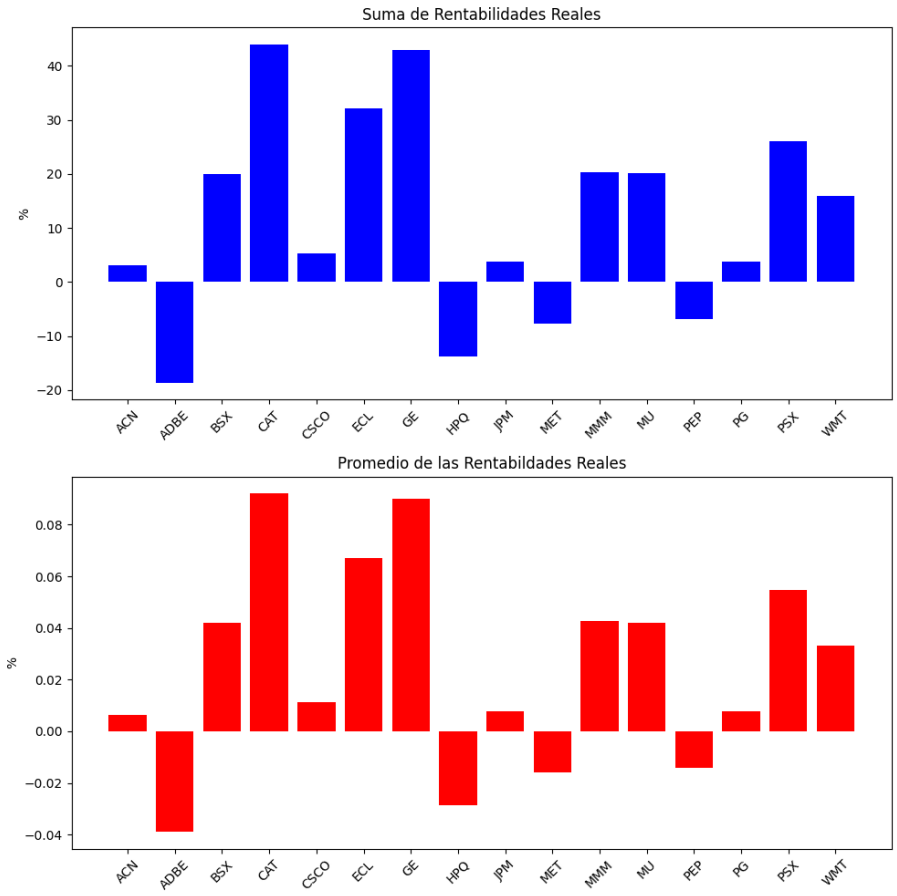
Sumas y Promedios de las rentabilidades diarias generadas por el Algoritmo A2C sobre el pronóstico generado con Redes Neuronales para cada acción.



Nota. Elaboración Propia.

Figura 66

Sumas y Promedios de las rentabilidades diarias generadas por las decisiones del Algoritmo A2C sobre los precios reales de cada acción.

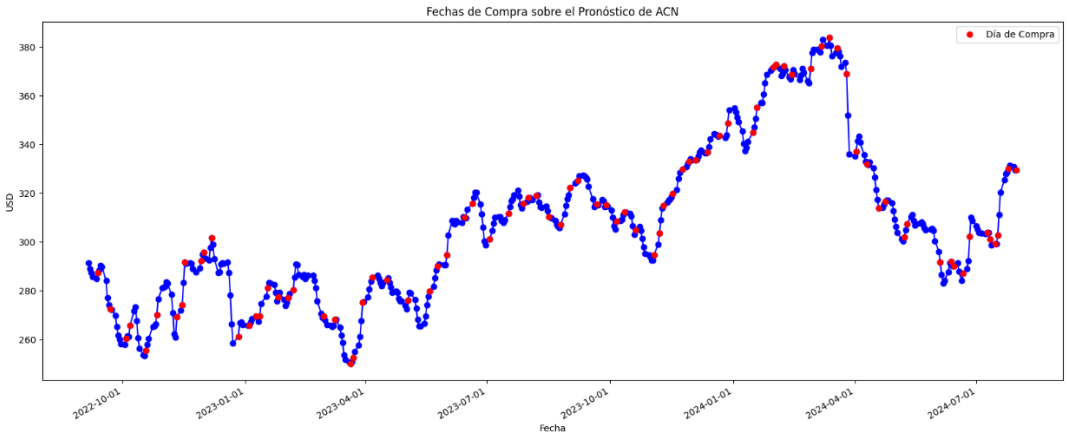


Nota. Elaboración Propia.

Con respecto al código construido, a partir de la importación directa del algoritmo de Actor-Crítico con Ventaja (A2C) con las librerías gym, gym_anytrading y stable_baselines3, las rentabilidades generadas por el modelo se obtuvieron gracias a las siguientes fechas de compra:

Figura 67

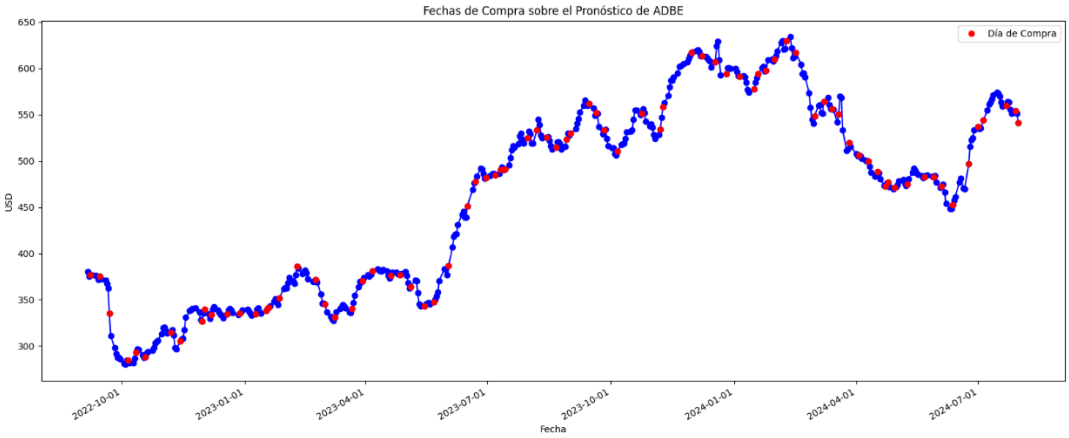
Decisiones de compra con la acción ACN.



Nota. Elaboración Propia.

Figura 68

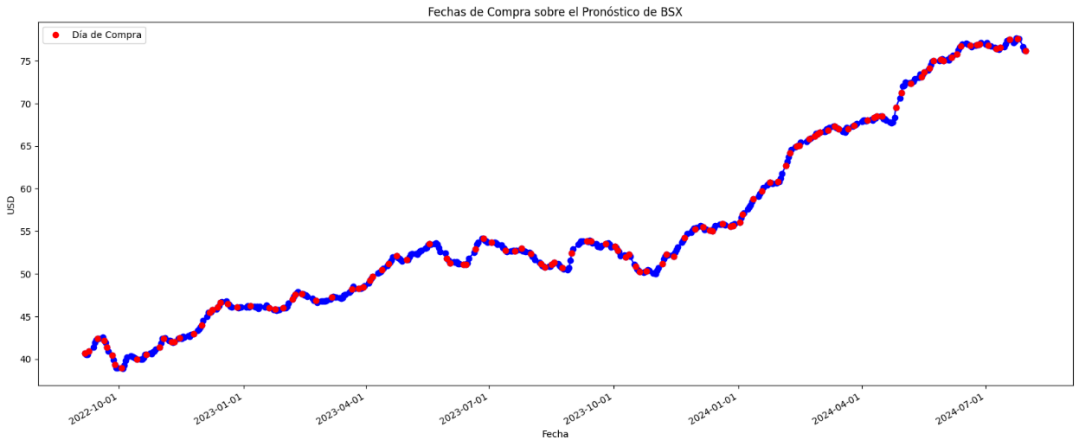
Decisiones de compra con la acción ADBE.



Nota. Elaboración Propia.

Figura 69

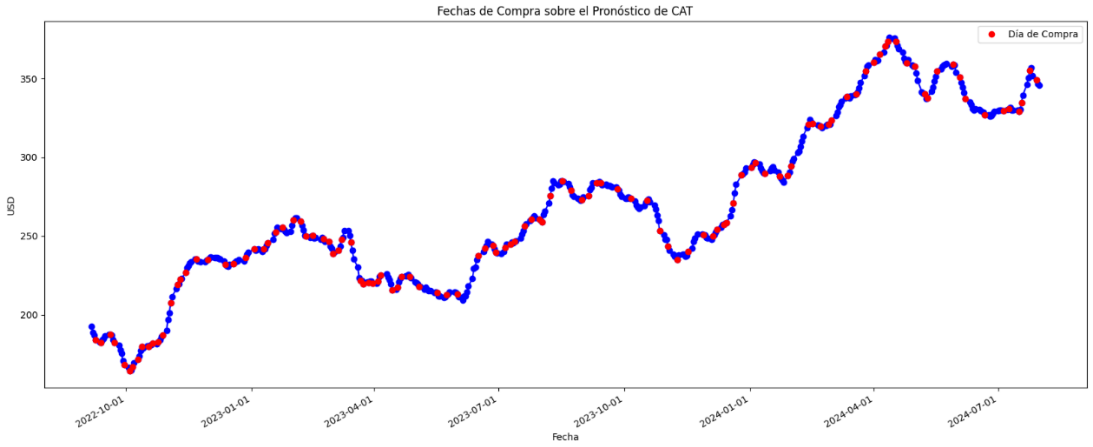
Decisiones de compra con la acción BSX.



Nota. Elaboración Propia.

Figura 70

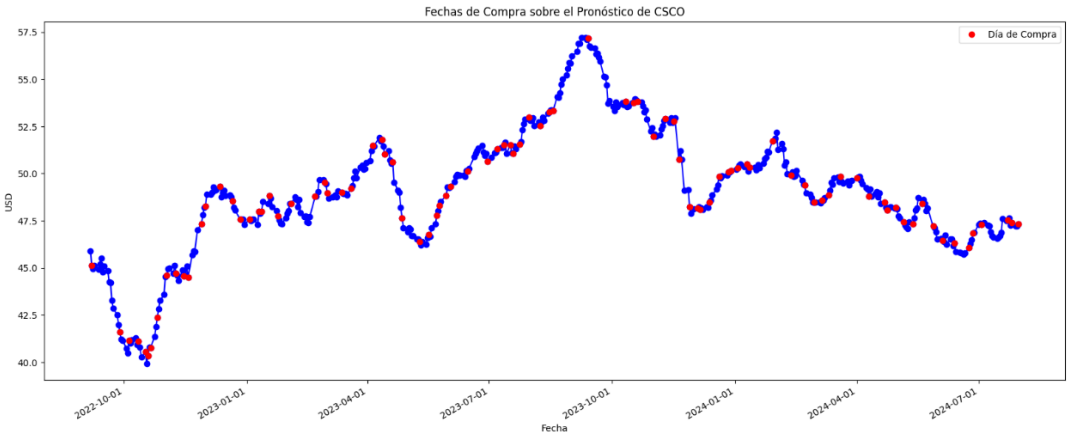
Decisiones de compra con la acción CAT.



Nota. Elaboración Propia.

Figura 71

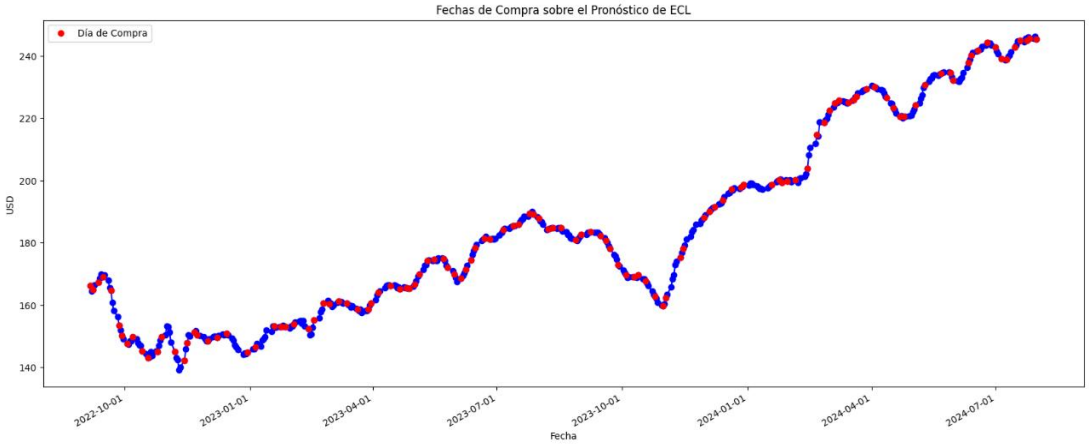
Decisiones de compra con la acción CSCO.



Nota. Elaboración Propia.

Figura 72

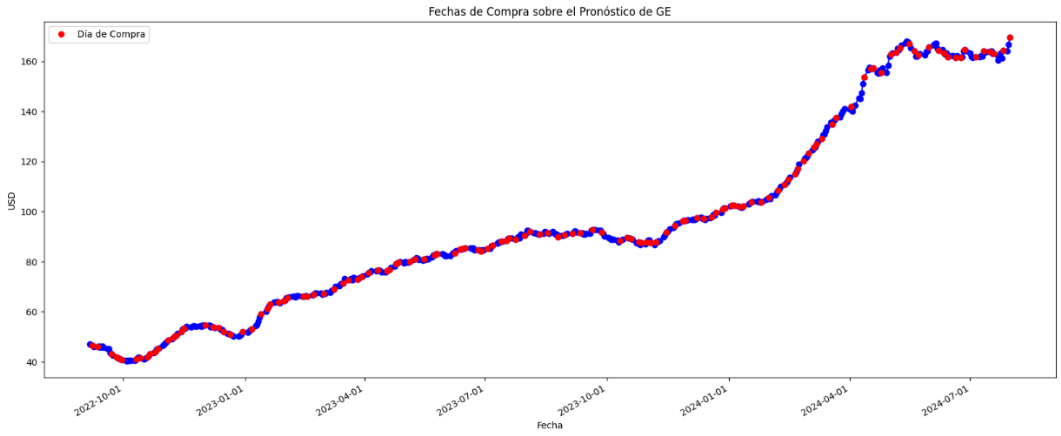
Decisiones de compra con la acción ECL.



Nota. Elaboración Propia.

Figura 73

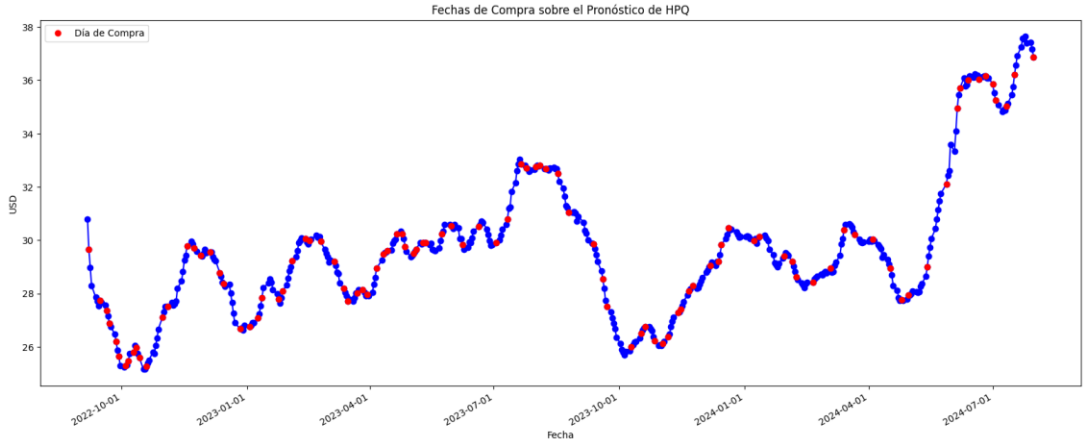
Decisiones de compra con la acción GE.



Nota. Elaboración Propia.

Figura 74

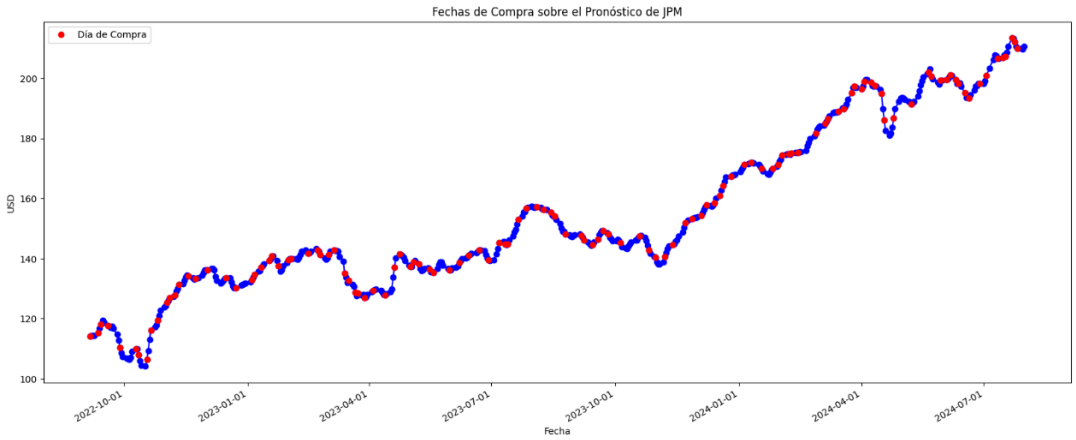
Decisiones de compra con la acción HPQ.



Nota. Elaboración Propia.

Figura 75

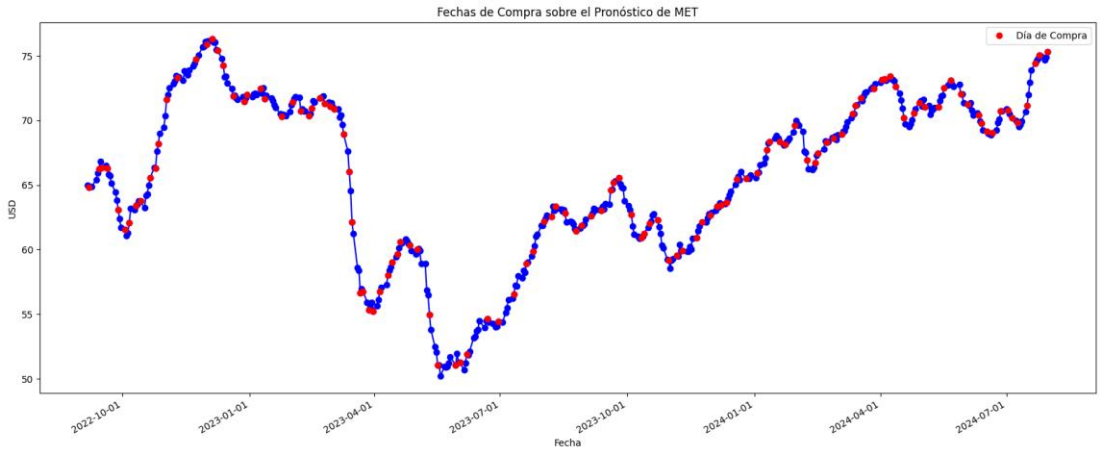
Decisiones de compra con la acción ACN.



Nota. Elaboración Propia.

Figura 76

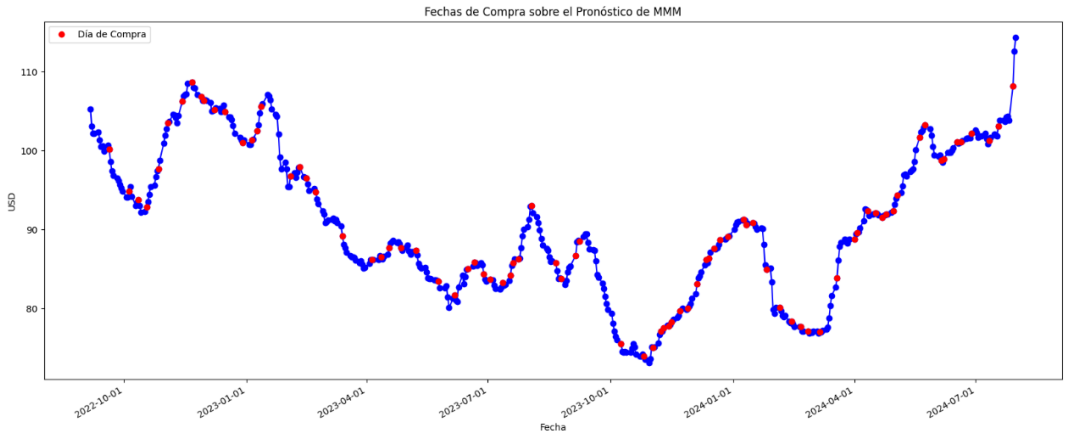
Decisiones de compra con la acción MET.



Nota. Elaboración Propia.

Figura 77

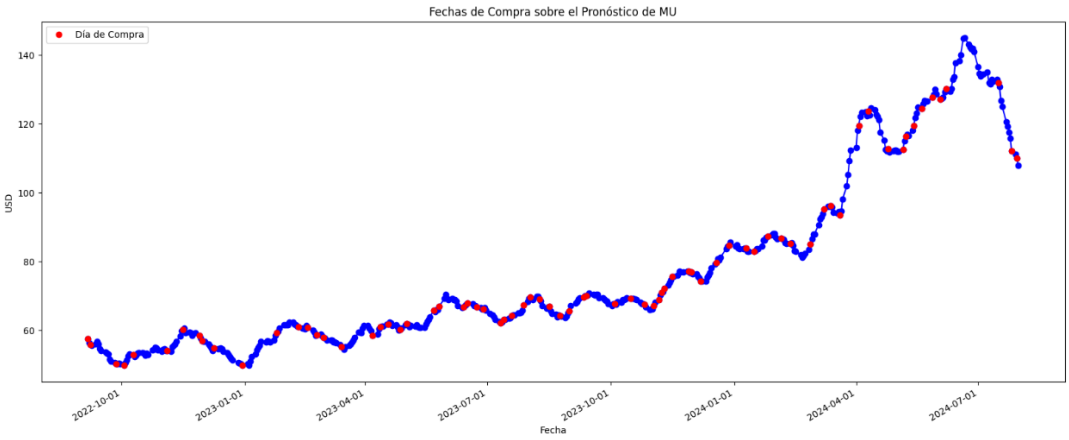
Decisiones de compra con la acción MMM.



Nota. Elaboración Propia.

Figura 78

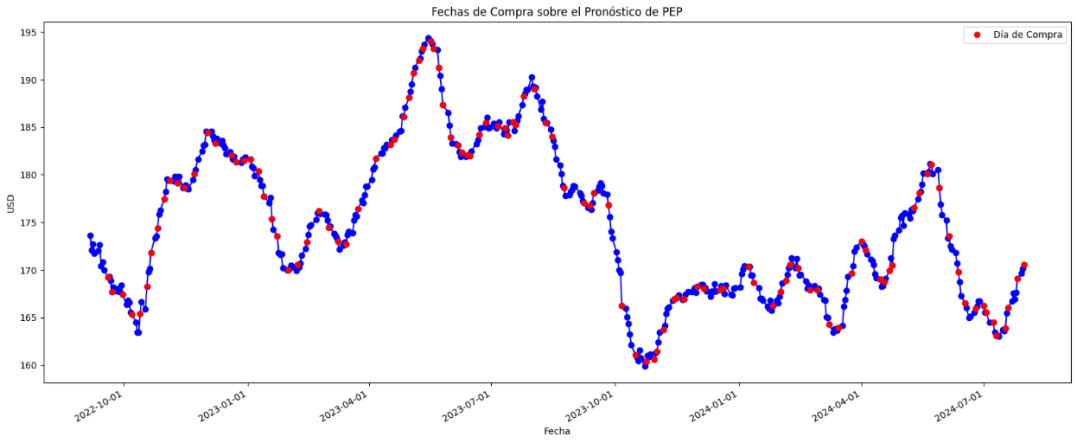
Decisiones de compra con la acción MU.



Nota. Elaboración Propia.

Figura 79

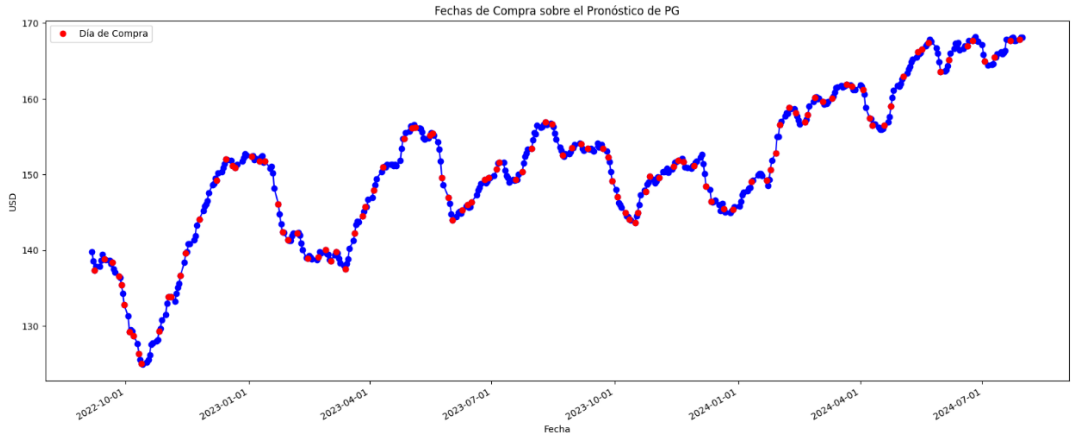
Decisiones de compra con la acción PEP.



Nota. Elaboración Propia.

Figura 80

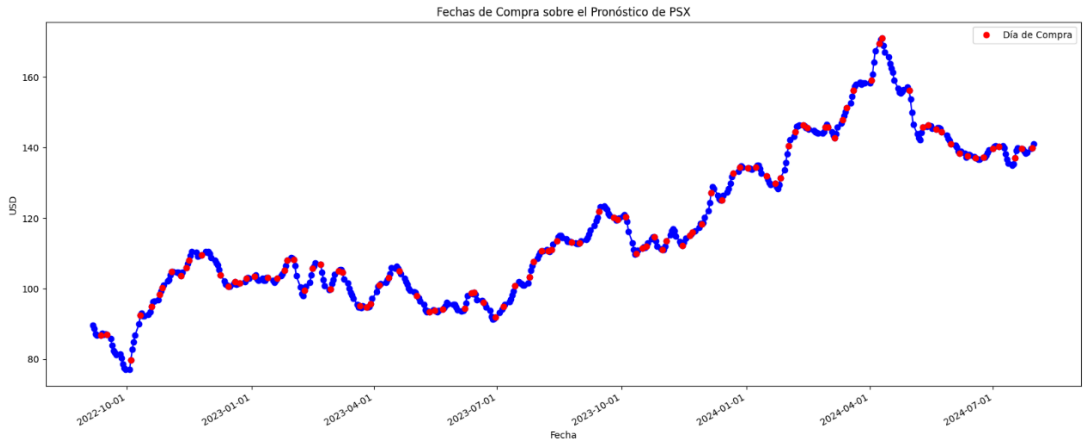
Decisiones de compra con la acción PG.



Nota. Elaboración Propia.

Figura 81

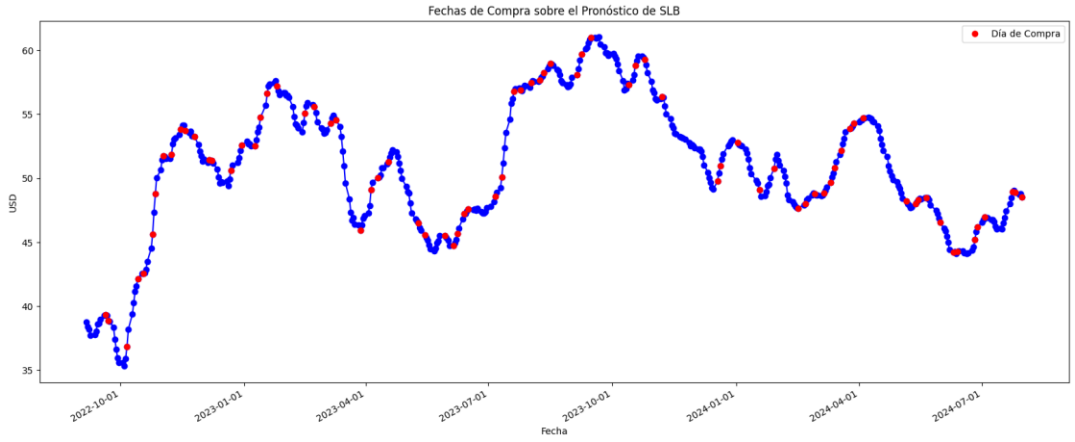
Decisiones de compra con la acción PSX.



Nota. Elaboración Propia.

Figura 82

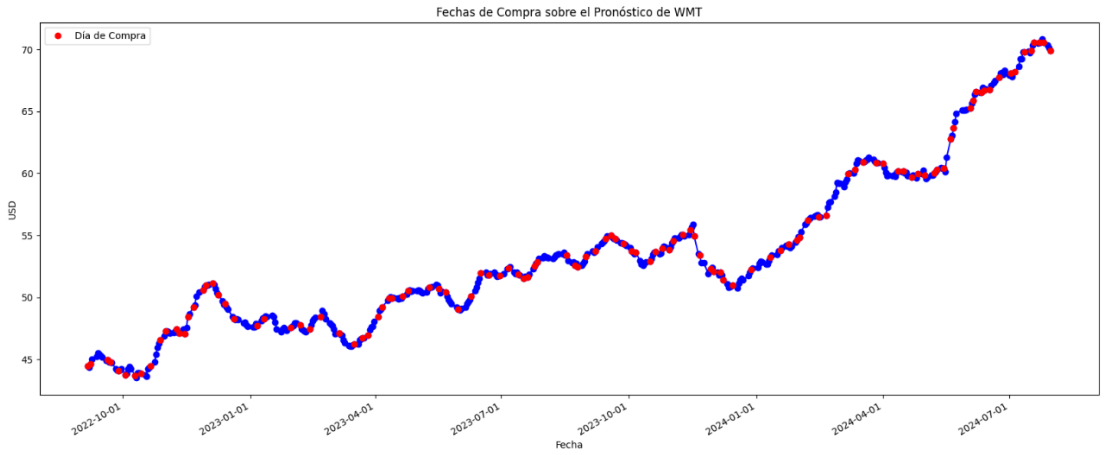
Decisiones de compra con la acción SLB.



Nota. Elaboración Propia.

Figura 83

Decisiones de compra con la acción WMT.



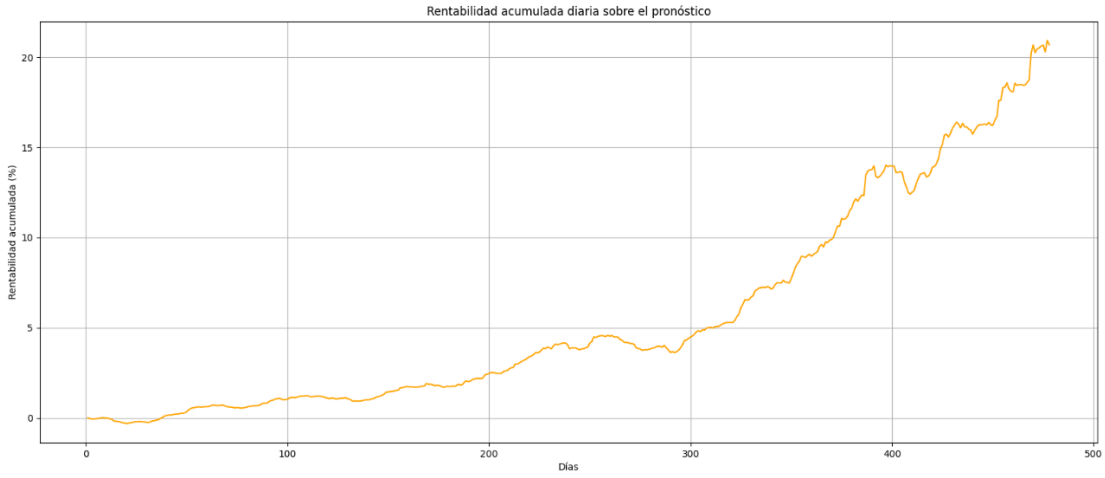
Nota. Elaboración Propia.

5.3 Rentabilidad Acumulada a lo largo del tiempo.

A raíz de los resultados obtenidos, se sumaron las rentabilidades diarias de todas las acciones generadas por el algoritmo A2C con los datos del pronóstico por cada día. Dichas decisiones de compra también se utilizaron sobre los precios reales de cada acción sumándolas de la misma manera, por lo que mediante la rentabilidad diaria total de estos dos casos se obtuvo la Rentabilidad Acumulada generado por el pronóstico y los precios reales de acción.

Figura 84

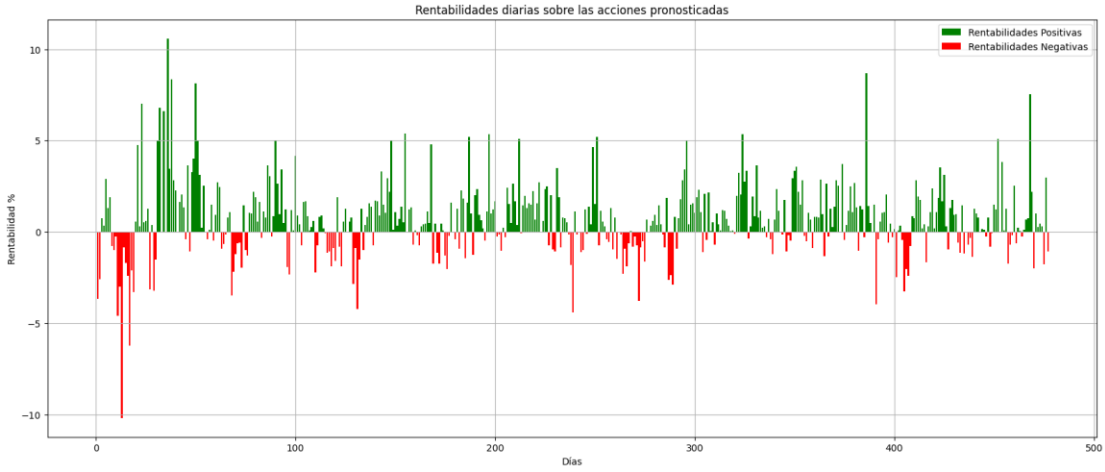
Rentabilidad acumulada diaria del pronóstico.



Nota. Elaboración Propia.

Figura 85

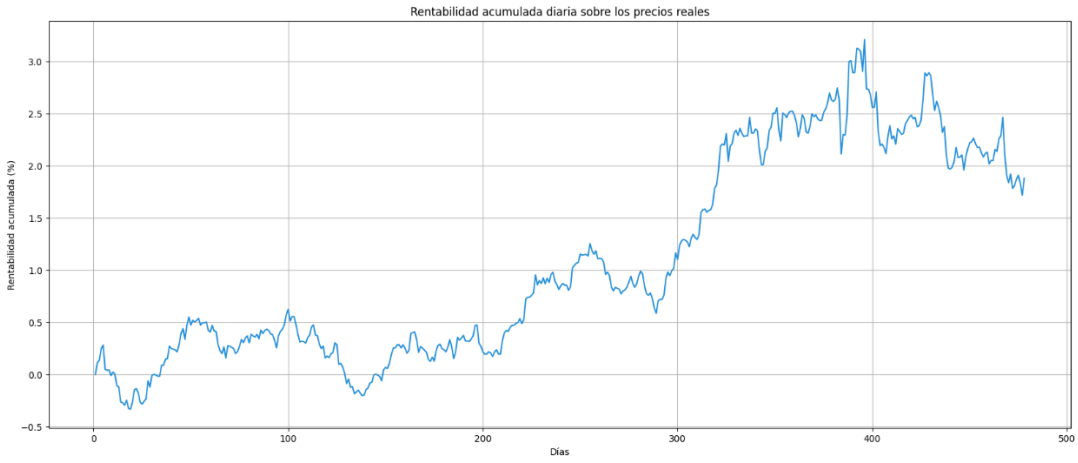
Rentabilidad diaria del pronóstico.



Nota. Elaboración Propia.

Figura 86

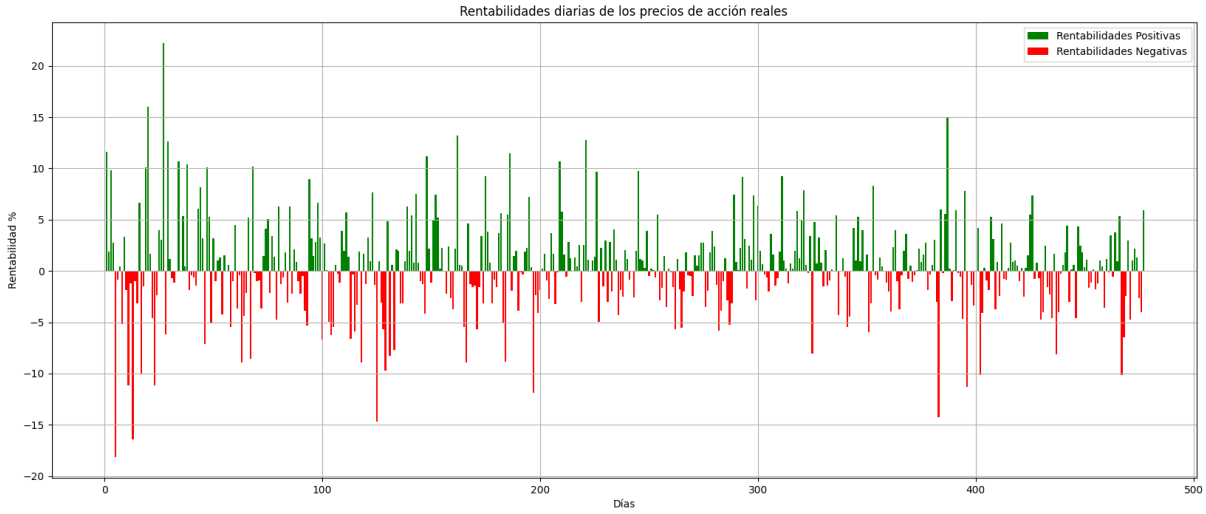
Rentabilidad acumulada diaria sobre los precios reales.



Nota. Elaboración Propia.

Figura 87

Rentabilidad diaria de los precios reales.

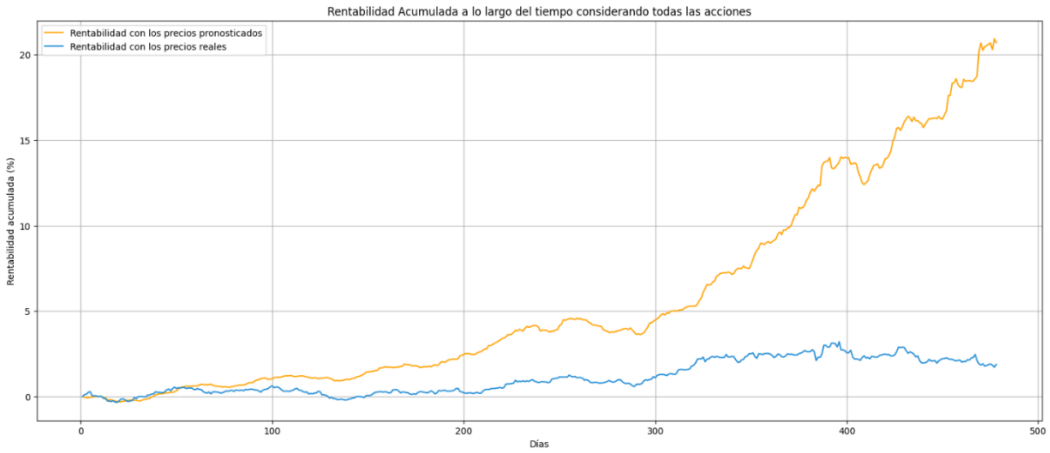


Nota. Elaboración Propia.

Una vez obtenidos los resultados finales sobre los pronósticos y sobre los precios reales, obtenemos el siguiente gráfico comparativo de las rentailidades acumuladas diarias a partir de las 17 acciones utilizadas.

Figura 88

Comparación final de la Rentabilidad Acumuladas de los precios pronosticados y reales.



Nota. Elaboración Propia.

En el gráfico podemos apreciar la gran diferencia entre las rentabilidades acumuladas, incrementandoce progrecibamente para las decisiones tomadas sobre el pronóstico, pero difiriendo en gran medida una vez que se aplican las mismas decisiones de compra sobre el precio real de cada acción.

Por otro lado, se comparó el Índice de Sharpe sobre ambas rentabilidades, considerando la Esperanza y Riesgo del portafolio como el promedio y desviación sobre los datos pronosticados y reales respectivamente, considerando una tasa libre de riesgo de un 0,01% diario a partir del 4,16% anual de los bonos del tesoro de EEUU a 2 años. Cada uno de los valores se consideró en porcentaje diario.

Tabla 2

Índices de Sharpe sobre la rentabilidad real y pronosticado.

	Sobre precios pronosticados	Sobre precios reales
Índice de Sharpe	0.3280	0.0713
Esperanza de Retorno	0.6661%	0.3249%
Riesgo	2.0301%	4.5584%

Nota. Elaboración Propia.

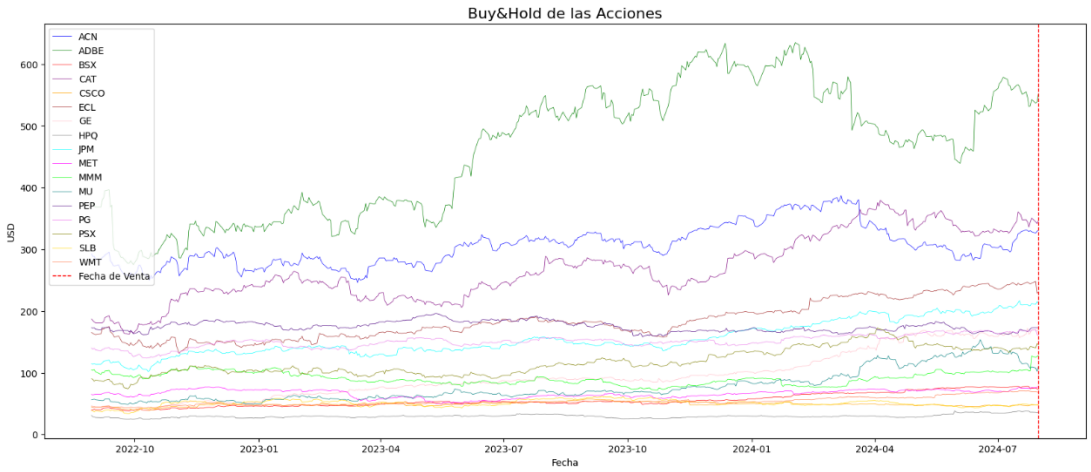
Con respecto Índice de Sharpe del pronóstico (0.3280) esta es más superior que al obtenido con los precios reales (0.0713), debido a que el pronóstico se utilizó como entrenamiento en la compra y venta al definirla como ambiente en el modelo A2C, por lo mismo, además se obtuvo la mejor rentabilidad, pero en cuanto al riesgo, la fluctuación de los datos fue menor para el pronóstico favoreciendo su Índice de Sharpe.

5.4 Cálculo de Rentabilidad del Buy & Hold al final del periodo evaluado.

Con la estrategia del Buy & Hold, se compró en el primer día y se vendió en el último dentro del horizonte de evaluación, dando como resultado un 56,75% en la rentabilidad promedio generada por la venta de todas las acciones al 31 de Julio del 2024.

Figura 89

Buy&Hold en las acciones elegidas.



Nota. Elaboración Propia.

5.5 Análisis de Resultados.

De acuerdo con los resultados arrojados respecto a los algoritmos utilizados podemos generar las siguientes opiniones:

- Con respecto a los pronósticos obtenidos mediante Redes Neuronales, se aprecia una gran similitud respecto a los precios reales gracias a su medición a partir de las métricas de error importadas directamente de la librería de TensorFlow, siendo bastante bajos para impulsar su credibilidad de ser utilizados, a excepción de las acciones de ACN, ADBE, CAT y ECL, cuyos valores de MSE fueron mayores que el resto con 84.30, 436.72, 104.01 y 19.50 respectivamente.

- Respecto a las pérdidas del error con respecto a MSE, en todas las acciones la perdida sobre los datos de testeo fue superior que las perdidas sobre los datos de entrenamiento. Esto es muy relevante, ya que es lo que normalmente se espera del entrenamiento de las redes neuronales a partir de sus distintas épocas ejecutadas, sugiriendo que logra una convergencia rápida y eficaz en un modelo de Red Neuronal demostrando gran estabilidad al largo de las épocas sin alcanzar un sobreajuste, el cual es lo que se busca evitar (Smets, 2022).

- Por otro lado, con respecto a la utilización del algoritmo A2C sobre los pronósticos obtenidos mediante las redes neuronales se logra conseguir una rentabilidad bastante buena gracias a la maximización de la rentabilidad esperada a raíz de las decisiones que toma y por la política óptima que logra para la obtención de rentabilidades positivas, pero a su vez, esta rentabilidad podría ser mucho mejor, dado a las pérdidas que muchas veces genera al comprar en días en donde el pronóstico tiende a la baja lo que genera un costo de oportunidad muy grande. Dicho comportamiento se

justifica a partir de la falta de memoria y secuencialidad impregnadas en el algoritmo de A2C, dado a que a medida que el modelo toma las decisiones de compra para generar rentabilidad, no es influenciada por lo que hizo antes.

- Por otro lado, si bien la combinación de Redes Neuronales más el Algoritmo de A2C logra resultados positivos y alentadores, estos no logran superar la rentabilidad lograda por la estrategia tradicional de Buy&Hold, siendo por motivos diversos, desde la elección de la data hasta el contexto macroeconómico de recuperación postpandemia.

- Finalmente, al aplicar las decisiones finales generadas por el modelo A2C sobre los precios reales de acción, representa un riesgo significativo para la decisión final del inversor, dado a que se corrompe el ambiente en donde el modelo A2C ya fue entrenado, por lo que es lógico esperar que la rentabilidad final fuese diferente. A pesar de eso, las decisiones finales de compra asignadas sobre los precios reales pudieron hacer frente de manera positiva en la rentabilidad acumulada, pero con una notoria disminución en los últimos días, lo que puede aun así representar una buena alternativa para aquello inversionistas amantes al riesgo.

6. Conclusión.

En conclusión, podemos decir que utilizar redes neuronales para el pronóstico de acciones es una alternativa bastante efectiva, dado a que al utilizarlas como métodos de pronóstico son capaces de captar patrones complejos y no lineales en los datos históricos de precios de acción, superando las limitaciones de los modelos tradicionales. Gracias a su capacidad para aprender representaciones abstractas y adaptarse a cambios en los datos, las redes neuronales pueden ofrecer predicciones más precisas en mercados financieros volátiles.

Por otro lado, el algoritmo de actor-crítico con ventaja (A2C), es eficaz para la toma de decisiones en la compra y venta de acciones, dado a que combina la estimación de valor y la política de acción en un solo marco. Gracias a la maximización de la rentabilidad a través de la búsqueda del parámetro óptimo en su política, es capaz de aprender estrategias efectivas al maximizar una función de recompensa diaria, adaptándose a la dinámica del mercado y ajustando las decisiones de compra y venta para maximizar las ganancias a lo largo del tiempo a través de distintas simulaciones.

En cuanto a la rentabilidad acumulada positiva generada por la combinación del modelo A2C con redes neuronales, se destaca su buena capacidad para obtener rentabilidades positivas. El A2C, como algoritmo de aprendizaje por refuerzo, toma decisiones secuenciales que optimizan las recompensas a corto plazo siendo útil para los inversores que decidan optar por esta alternativa de compra y venta en el corto plazo. Al incorporar las predicciones de redes neuronales, el modelo puede afinar sus decisiones de compra y venta evitando que se quede atrapado en soluciones subóptimas, gracias a la utilización del optimizador Adam y la optimización basada en el gradiente de política.

También, existen algunos desafíos y desventajas para tener en cuenta en los supuestos tomados para la combinación de ambos modelos. Por una parte, dado al contexto global u otros factores externos, el modelo tradicional de Buy & Hold demuestra tener mejores resultados que la sofisticación en la propuesta usando NN y el algoritmo de A2C, aunque esto a veces no puede llegar a ser así. Por otra parte, las redes neuronales al ser muy eficaces también pueden sobre ajustarse a los datos históricos, lo que resulta en predicciones menos precisas cuando las condiciones de mercado cambian con respecto al peso y sesgo definidos, sumándole además al coto de oportunidad que se aprecian a simple vista en cuanto a la decisión final del algoritmo A2C en comprar y vender secuencialmente cuando el pronóstico se va en pérdida, lo que puede ser arreglado descartando dichas rentabilidades negativas si el inversor así lo decide. Además, al aplicar las decisiones de compra cuando se cambia el ambiente del modelo A2C a otro en el que no fue entrenado, pueden implicar a posibles pérdidas temporales que representan un riesgo grande para aquellos inversionistas adversos al riesgo que buscan liquidez en el corto plazo, por lo cual es de vital importancia monitorear constantemente dichos modelos, en especial a las decisiones de compra final entregadas por el modelo A2C.

A pesar de estos desafíos, la combinación de A2C con redes neuronales es altamente eficaz pero no eficiente para estrategias de inversión debido a la adaptabilidad que ofrece. Mientras las redes neuronales proporcionan una visión predictiva sobre las tendencias del mercado, el A2C ajusta las decisiones de manera dinámica, lo que permite al sistema adaptarse a condiciones volátiles y cambiantes. Esta sinergia entre predicción y optimización secuencial es lo que hace que esta combinación sea de todas formas valiosa para maximizar la rentabilidad bajo la automatización.

7. Referencias

Andrea Apicella, F. D. (2021). A survey on modern trainable activation functions. *Cornell University*, 30.

Clara Fabiola, E. P. (2020). Primary Market vs. Secondary Market. *Social Science Research Network*, 7.

Diederik P. Kingma, J. B. (2015). Adam: A Method for Stochastic Optimization. *Cornell University*, 15.

Fama, E. F. (1965). The Behavior of Stock-Market Prices. *Journal of Business* , 73.

Femminella, M. (2024). Comparison of Reinforcement Learning Algorithms for Edge Computing Applications Deployed by Serverless Technologies. *MDPI*, 26.

Hayes, A. (13 de June de 2024). *Capital Markets: What They Are and How They Work*. Obtenido de Investopedia: https://www.investopedia.com/terms/c/capitalmarkets.asp

Islam, M. (2019). An Overview of Neural Network. *American Journal of Neural Networks and Applications*, 5.

Jian Huang, J. H. (2020). Deep learning in finance and banking: A literature review and classification. *Frontiers of Business Research in China*, 24.

Kady Sako, B. N. (2022). Neural Networks for Financial Time Series Forecasting. *Entropy*, 17.

Kouridi, C. (2020). Syntactic language understanding for compositional generalisation in reinforcement learning. *ResearchGate*, 114.

Ling, F. C. (2014). An Empirical Re-Investigation on the 'Buy-and-hold Strategy' in Four Asian Markets: A 20 Years' Study. *ResearchGate*, 12.

Lo, A. W. (2004). The Adaptive Markets Hypothesis: Market Eficiency from an Evolutionary Perspective. *Social Science Research Network*, 33.

M. Z. Naser, A. H. (2021). Error Metrics and Performance Fitness Indicators for Artificial Intelligence and Machine Learning in Engineering and Sciences. *ResearchGate*, 23.

Markowitz, H. (1952). Portfolio Selection. *The Journal of Finance*, 16.

Müller, H. H. (2014). Modern Portfolio Theory: Some Main Results. *Cambridge University Press*, 19
.

Nielsen, M. (2015). *Neural Networks and Deep Learning*. Determination Press.

Raiaan, M. A. (2024). A systematic review of hyperparameter optimization techniques in Convolutional Neural Networks. *ElSevier*, 32.

Santos, G. C. (2023). Management of investment portfolios employing reinforcement learning. *PeerJ Computer Science*, 27.

Sewak, M. (2019). Actor-Critic Models & the A3C, The Asynchronous Advantage Actor-Critic Model. *Springer*, 17.

Shah, D. (26 de January de 2023). *V7*. Obtenido de https://www.v7labs.com/blog/cross-entropy-loss-guide

Shuo Sun, R. W. (2023). Reinforcement Learning for Quantitative Trading. *Association for Computing Machinery*, 29.

Silva, J. A. (2022). *Portfolio Allocation using Deep Reinforcement Learning.* Porto : Instituto Superior de Ingeniería de Porto.

Smets, B. M. (2022). Mathematics of Neural Networks. *Cornell University*, 80.

The Economist. (7 de June de 2024). What happened to the artificial-intelligence investment boom? *Perhaps AI is a busted flush. Perhaps the revolution will just take time*, pág. 1
.

U.S. Securities and Exchange Commision . (2005). *Performance and Accountability Report.* Washington DC: SEC.

Yang, L. (2023). A General Perspective on Objectives of Reinforcement Learning. *Cornell University*, 20.

Yousefi, N. (2022). Deep Reinforcement Learning for Tehran Stock Trading. *Indonesian Journal of Data and Science (IJODAS)*, 10.