



# FACULTAD DE INGENIERÍA

Carrera de Ingeniería de Sistemas Computacionales

**“Algoritmo basado en Deep Learning para predecir la aproximación del precio de los activos de renta variable en tiempo real por Yahoo Finance”**

**Tesis para optar al título profesional de:**

**Ingeniero de Sistemas Computacionales**

**Autores:**

**Israel Junior Rojas Agüero**

**Luigi Fabrizio Paz Salinas**

**Asesor:**

**Mg. Ing. Jorge Rosvin Narvaez Villacorta**

**Código ORCID**

**<https://orcid.org/0000-0003-4818-4139>**

**Lima - Perú**

**2025**

**JURADO EVALUADOR**

Jurado 1 Presidente(a)	Laura Bazan Diaz
	Nombre y Apellidos

Jurado 2	Betty Suarez Torres
	Nombre y Apellidos

Jurado 3	Jorge Narvaez Villacorta
	Nombre y Apellidos

## Informe de Similitud

(Copie y pegue como imagen la hoja del reporte global)



Página 2 de 60 - Descripción general de integridad

Identificador de la entrega trn:oid::1:3401981658




### 19% Similitud general

El total combinado de todas las coincidencias, incluidas las fuentes superpuestas, para ca...

#### Exclusiones

- N.º de fuente excluida
- N.º de coincidencias excluidas

#### Fuentes principales

- 18%  Fuentes de Internet
- 9%  Publicaciones
- 6%  Trabajos entregados (trabajos del estudiante)

#### Marcas de integridad

##### N.º de alertas de integridad para revisión

No se han detectado manipulaciones de texto sospechosas.

Los algoritmos de nuestro sistema analizan un documento en profundidad para buscar inconsistencias que permitirían distinguirlo de una entrega normal. Si advertimos algo extraño, lo marcamos como una alerta para que pueda revisarlo.

Una marca de alerta no es necesariamente un indicador de problemas. Sin embargo, recomendamos que preste atención y la revise.

## **Dedicatoria**

El presente trabajo de investigación lo dedicamos a Dios por darnos un día más de vida, por bendecirnos cada día sin pedir nada a cambio y por siempre darnos fuerzas para luchar cada obstáculo que se nos ha presentado; y a nuestros padres que nos apoyaron en todo nuestro camino universitario y son la principal fuente de inspiración y fuerzas.

## **Agradecimiento**

Agradezco a mi tutor por guiarme en la presente investigación y a nuestros docentes que nos supieron orientar y enseñar de manera correcta y dinámica todos los temas de investigación, por las experiencias brindadas en cada examen y ejercicios de clases y por los consejos al momento de tener algunos problemas en diferentes situaciones.

**Tabla de contenidos**

JURADO EVALUADOR.....	2
Informe de Similitud.....	3
Dedicatoria.....	4
Agradecimiento.....	5
Índice de tablas .....	7
Índice de Figuras.....	8
Resumen .....	10
Palabras Claves .....	10
CAPÍTULO I: INTRODUCCIÓN.....	11
CAPÍTULO II: METODOLOGÍA .....	17
CAPÍTULO III: RESULTADOS .....	26
CAPÍTULO IV: DISCUSIÓN Y CONCLUSIONES.....	31
<b>REFERENCIAS</b> .....	<b>35</b>
ANEXOS .....	37

## Índice de tablas

<i>Tabla 1. Lista de Cotejo.....</i>	288
<i>Tabla 2. Tipo de Datos y Descripción de las Variables antes del preprocesamiento .....</i>	40
<i>Tabla 3. Tipo de Datos y Descripción de las Variables después del tratamiento .....</i>	42
<i>Tabla 4. Tipo de Datos y Descripción de las Variables después de la Estandarización.....</i>	455
<i>Tabla 5. Matriz de Consistencia .....</i>	50
<i>Tabla 6. Matriz de Operacionalización .....</i>	51

## Índice de Figuras

Figura 1. Metodología CRISP-DM.....	199
<i>Figura 2. Gráfica de comparación entre el precio real y el precio predicho del activo del Dow Jones .....</i>	<i>266</i>
<i>Figura 3. Gráfica de diferencia de error entre precios del Dow Jones .....</i>	<i>277</i>
<i>Figura 4. Métricas obtenidas para el índice del Oro .....</i>	<i>299</i>
<i>Figura 5. Comparación entre valores reales y predichos del Russell 2000 ....</i>	<i>300</i>
<i>Figura 6. Comparación entre valores reales y predichos del Nasdaq.....</i>	<i>30</i>
<i>Figura 7. Librerías de Yahoo Finance .....</i>	<i>388</i>
<i>Figura 8. Elección del activo de renta variable .....</i>	<i>399</i>
<i>Figura 9. Datos históricos del activo del Dow Jones antes del preprocesamiento .....</i>	<i>399</i>
<i>Figura 10. Gráfico de precios del activo del Dow Jones .....</i>	<i>41</i>
<i>Figura 11. Eliminación de datos innecesarios .....</i>	<i>42</i>
<i>Figura 12. Estacionalización de los precios de cierre .....</i>	<i>433</i>
<i>Figura 13. División de los datos de entrenamiento y testeo.....</i>	<i>444</i>
<i>Figura 14. Estandarización de los valores de entrenamiento y testeo .....</i>	<i>444</i>
<i>Figura 15. Preparación de los datos de entrenamiento por bloques .....</i>	<i>455</i>
<i>Figura 16. Armado del Modelo LSTM .....</i>	<i>466</i>
<i>Figura 17. Entrenamiento del modelo .....</i>	<i>477</i>
<i>Figura 18. Gráfica de errores del entrenamiento .....</i>	<i>488</i>

<i>Figura 19. Preparación de los datos de entrenamiento por bloques .....</i>	<i>488</i>
<i>Figura 20. Predicción de los valores entrenados.....</i>	<i>499</i>
<i>Figura 21. Flujograma de Proceso del Funcionamiento del Algoritmo .....</i>	<i>522</i>

## Resumen

Este estudio presenta un algoritmo basado en Deep Learning, específicamente el modelo LSTM, para predecir la aproximación del precio de los activos de renta variable. Basado en la relevancia de los algoritmos de aprendizaje profundo en el campo financiero y la importancia del análisis de sentimiento en la predicción de precios, el objetivo principal es mejorar la precisión en la predicción de precios mediante este enfoque innovador. Se empleó la metodología CRISP-DM y se recopilaron datos históricos, utilizando 9191 datos de entrenamiento y 2298 datos de prueba. Los resultados obtenidos fueron altamente prometedores, con una precisión del 86.18% demostrando la eficacia del algoritmo, superando los enfoques tradicionales y capturando patrones y tendencias en los datos financieros. Además, se destacan las implicaciones prácticas de este estudio, ya que se ha demostrado la capacidad del algoritmo para analizar múltiples mercados, ofreciendo una herramienta prometedora que mejora la precisión en la predicción de la aproximación de precios. En resumen, este estudio proporciona una solución efectiva y avanzada basada en Deep Learning para la predicción de aproximación de precios de activos de renta variable, mejorando significativamente la precisión de la aproximación y ofreciendo nuevas oportunidades para comprender y aprovechar los patrones del mercado financiero.

## Palabras Claves

predicción, machine learning, deep learning, LSTM, activos

## CAPÍTULO I: INTRODUCCIÓN

### 1.1 Realidad problemática

El estudio utiliza como fuentes de información bibliográficas relacionadas a los algoritmos de aprendizaje profundo, siendo la mayoría sobre predicciones para el precio de los activos de renta variable. Aldhyani y Alzahrani et al. (2022) sostienen que en un mundo cada vez más impulsado por los datos, el uso de algoritmos de Deep Learning para predecir dichos precios ha despertado un gran interés en la comunidad financiera. En particular, el modelo LSTM se ha propuesto como una forma efectiva de poder predecir los cambios de precios en tiempo real y evaluar el sentimiento del mercado. Además, explican cómo los activos de renta variable, un mercado en el que los inversores pueden comprar y vender diversos activos en bolsas públicas, privadas o mixtas, son cotizados por las acciones de empresas. En otras palabras, los valores que cotizan en la bolsa son negociados a un precio, el cual logra ser determinado por la oferta y demanda del mercado, explicando también la relación que tienen el precio del mercado con los activos de renta variable para que se pueda tener una mejor percepción sobre dichos conceptos.

Lin, Lai y Pai et al. (2022) sustentan que el uso de dichos algoritmos de aprendizaje profundo se basa en la teoría de que los precios de las acciones están influenciados por una variedad de factores, incluyendo el sentimiento del mercado. Según la teoría del mercado eficiente, el precio de una acción refleja toda la información disponible públicamente y, por lo tanto, es difícil predecir los cambios de precios futuros.

Pagolu, Reddy, Panda y Majhi et al. (2016) sustentan que el análisis de sentimiento es un enfoque basado en la teoría del procesamiento del lenguaje natural que tiene como objetivo identificar la polaridad del sentimiento en un texto. Este enfoque está basado en la hipótesis de que el lenguaje humano contiene información sobre el sentimiento y la opinión, y que esta información puede ser utilizada para predecir el

comportamiento del mercado de valores. En particular, el análisis de sentimiento puede ser útil para detectar patrones de sentimiento positivos o negativos en las noticias financieras, los tweets o las publicaciones en redes sociales que puedan afectar el precio de las acciones. Asimismo, el análisis de sentimiento se ha utilizado en una variedad de aplicaciones financieras, incluyendo la predicción del precio de las acciones.

La investigación analizó diferentes documentos académicos, revistas y artículos. Farida et al. (2017) muestra cómo el modelo matemático conocido como movimiento Browniano Geométrico es utilizado para predecir el precio futuro de valores en el mercado financiero. Antes de realizar dicha predicción, se lleva a cabo una fase en la que se determina el precio esperado de las acciones y se calcula a nivel de confianza del 95%. Para aplicar este modelo al pronóstico del precio de las acciones, se realiza un cálculo del valor de retorno, seguido de la estimación del valor de la volatilidad y se obtiene el pronóstico del precio de las acciones. Además, se calcula el pronóstico MAPE y el precio esperado de la acción, y se determina el nivel de confianza de 95%. Según dicha investigación, se demostró que el modelo de movimiento Browniano Geométrico es una técnica de predicción altamente precisa, con una tasa de error pronosticado de  $MAPE \leq 20\%$ .

Srivastava, Zuopeng y Eachempati et al. (2021) exponen cómo la utilización del aprendizaje profundo para predecir el precio de los activos de renta variable ha ganado importancia, demostrando que las redes neuronales pueden ser efectivas para predecir el precio de las acciones. Además, utilizan algoritmos de aprendizaje automático como soporte de vectores, bosque aleatorio, gradiente de impulso y redes neuronales para predecir la tendencia de la bolsa de la nación de la India. Utilizaron un conjunto de entrenamiento de datos históricos de precios de cierre de la NSE, así como indicadores técnicos como la media móvil y la convergencia-divergencia móvil. También, las redes

neuronales profundas tienen un mayor poder predictivo que el resto de las técnicas de aprendizaje automático, sugiriendo su potencial para la predicción de series temporales multiparamétricas en el futuro.

Ampomah, Nyame, Qin, Clement, Opanin y Gyan et al. (2021) exhiben cómo el mercado de valores en la economía es tan importante y cómo la predicción del comportamiento del mercado es crucial para reducir el riesgo de inversión. También, para lograr predecir el mercado, utilizaron un modelo de aprendizaje automático, en específico el algoritmo gaussiano Naive Bayes machine learning. Dicho algoritmo fue evaluado para poder predecir el movimiento del precio de las acciones y concluyendo que la combinación de diferentes técnicas de escalado y extracción de características mejora su rendimiento. Por último, sugieren que el algoritmo GNB puede ser una herramienta útil para la predicción del movimiento del precio de las acciones.

Albahli, Irtaza, Nazir, Mehmood, Alkhalifah y Albattah et al. (2022) manifiestan que la realización de cualquier actividad útil para la humanidad requiere finanzas y los mercados financieros, como los de valores, divisas y bolsas mercantiles, ofrecen oportunidades para invertir y generar ingresos. Para obtener el máximo beneficio de estos mercados, es crucial tomar decisiones eficaces al identificar las direcciones de las operaciones y analizar factores influyentes como la evolución de los precios, las políticas económicas y la oferta y demanda. Además, dan a conocer que los análisis de noticias financieras y publicaciones en Twitter pueden predecir el comportamiento futuro de los mercados financieros, mostrando que su objetivo es analizar los datos de Twitter y Google Finance para predecir el comportamiento futuro de los mercados bursátiles a través de la red de palabras StockSentiWordNet (SSWN). Este modelo amplía el léxico de opinión estándar de SentiWordNet (SWN) y utiliza máquinas de aprendizaje extremo y redes neuronales recurrentes para predecir el precio de las acciones, demostrando una precisión

del 86.06%, superando a otros métodos en general, y planeando mejorar la capacidad de dicho método agregando otras redes sociales populares como Facebook, Google News, entre otros.

Mehtab, Sen y Dutta et al. (2020) enseñan cómo los análisis técnicos de las acciones sirven para poder identificar los patrones utilizando técnicas avanzadas de minería de datos. Además, proponen un enfoque híbrido para predecir el precio de las acciones utilizando diferentes modelos de machine learning y Deep learning. Utilizando datos del índice NIFTY 50 de la bolsa nacional de valores de la India, se construyeron ocho modelos de regresión utilizando los datos de entrenamiento y se predijo el valor de apertura del NIFTY 50 para un periodo posterior. Se aumentó el poder predictivo mediante la construcción de cuatro modelos de regresión basados en LSTM y se optimizaron los hiperparámetros de los modelos mediante la técnica de búsqueda en cuadrícula. Se concluyó que el modelo univariante basado en LSTM que utiliza datos previos de una semana como entrada es el más preciso.

Park y Lee et al. (2019) presentan un artículo sobre cómo el movimiento del índice Standard & Poor's 500 es predicho por técnicas de minería de datos y reconocimiento de patrones. Por lo tanto, para lograr esto se utilizaron clasificadores de formas funcionales variables para predecir los datos de series temporales financieras y evaluar el rendimiento de varios clasificadores. Además, de utilizar el modelo ARIMA de series temporales para pronosticar el índice Standard & Poor's 500, se comparan las tasas de precisión de clasificación de los modelos bagging y boosting con varios clasificadores, como las máquinas de vectores de soporte, el k-nearest neighbor, la red neuronal probabilística y el árbol de clasificación y regresión, utilizando el algoritmo AdaBoost y sus extensiones. Por último, concluyeron que el clasificador boosting con AdaBoost es el que mejor

pronostica los movimientos del índice Standard & Poor's 500, logrando que el resultado sea útil para las empresas que deseen predecir los precios de las acciones.

La presente investigación de este estudio radica en la importancia que tienen dichos algoritmos de aprendizaje profundo para poder lograr predecir el precio de los activos de renta variable en el mercado financiero; sin embargo, dicha investigación se realiza en tratar de aproximarse a los precios lo máximo posible. La comunidad financiera ha mostrado gran interés en el uso de estas técnicas, especialmente en el modelo LSTM, debido a su capacidad para capturar patrones temporales complejos en los datos históricos, lo cual resulta crucial para pronosticar el comportamiento de los activos en el futuro. Además, la investigación analiza diferentes modelos matemáticos y algoritmos de aprendizaje automático utilizados para predecir el comportamiento del mercado de valores, con el fin de reducir el riesgo de inversión, a pesar de que el mercado bursátil se vea obstaculizado por la alta correlación entre los precios de las acciones y los enfoques de procesamiento por lotes. Gracias a todo ello, este estudio innova en el medio de la dinamicidad de la elección de cualquier divisa en la base de datos de Yahoo Finance y poder predecir la aproximación de manera más exacta que otros estudios utilizando el modelo LSTM, teniendo un resultado de precisión de resultados en un 86.18% en promedio.

## **1.2 Formulación del problema**

Dado los párrafos anteriores, surge la pregunta de investigación: ¿Cómo puede un algoritmo basado en Deep Learning predecir con mayor precisión el precio de los activos de renta variable?

### **1.3 Objetivos**

Objetivo General: Diseñar un algoritmo basado en Deep Learning que optimice la predicción del precio de los activos de renta variable a partir de datos históricos financieros.

Objetivo Específico 1: Validar la precisión del algoritmo basado en Deep Learning mediante la comparación entre los precios predichos y los precios reales de los activos de renta variable.

Objetivo Específico 2: Evaluar la capacidad del algoritmo para realizar predicciones dinámicas de los precios futuros de los activos de renta variable bajo diferentes condiciones del mercado.

### **1.4 Hipótesis**

Hipótesis General: La predicción de precios de los activos de renta variable experimenta una mejora significativa mediante el uso de un algoritmo basado en Deep Learning.

Hipótesis Específica 1: El algoritmo basado en Deep Learning presenta una diferencia mínima entre los precios predichos y los precios reales de los activos de renta variable.

Hipótesis Específica 2: El algoritmo basado en Deep Learning mantiene una precisión elevada en la predicción de precios futuros de los activos de renta variable, incluso en escenarios de comportamiento dinámico del mercado.

## CAPÍTULO II: METODOLOGÍA

### 2.1 Tipo de investigación

El presente estudio adopta un enfoque cuantitativo, ya que se centra en la recolección y el análisis de datos numéricos mediante técnicas estadísticas y computacionales. Asimismo, según Ng et al. (2018) indicó que la investigación cuantitativa “es fundamental para el éxito del aprendizaje automático, ya que los científicos de datos deben diseñar experimentos rigurosos y analizar los resultados de manera efectiva para mejorar continuamente sus modelos”, además, también implica el uso de herramientas tecnológicas y matemáticas para la obtención de resultados. A diferencia del enfoque cualitativo, que se basa en la comprensión de fenómenos subjetivos y contextuales, el enfoque cuantitativo busca identificar patrones medibles, relaciones y predicciones a partir de datos estructurados.

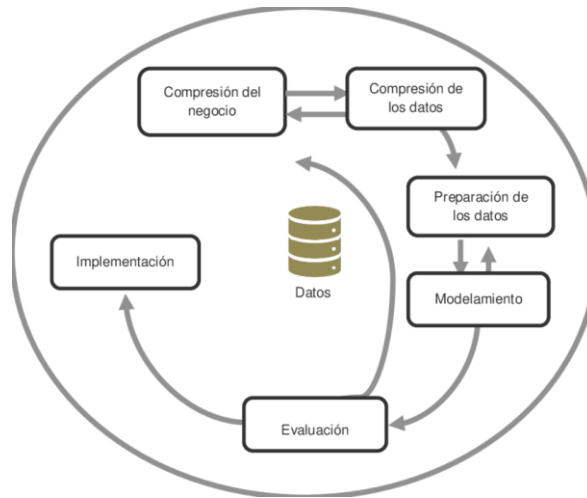
El diseño metodológico es de tipo no experimental, ya que no se manipulan las variables independientes, sino que se observan tal como ocurren en su entorno natural. Igualmente, el diseño es longitudinal porque analiza datos históricos a lo largo del tiempo, y aplicado, dado que busca utilizar el conocimiento generado para resolver un problema práctico: predecir precios de activos financieros con técnicas de aprendizaje automático.

El estudio se orienta al nivel predictivo, puesto que tiene como objetivo anticipar el valor futuro de una variable (precio de los activos financieros) a partir del análisis y modelado de patrones históricos y actuales.

De igual manera, la investigación presentó una metodología CRISP-DM (Cross-Industry Standard Process for Data Mining), que es ampliamente utilizada en la industria financiera para abordar proyectos de análisis de datos y minería de datos. La metodología CRISP-DM proporciona un marco estructurado que guía a los profesionales financieros a través de sus 6 fases:

- **Entendimiento del negocio:** Se definen los objetivos del estudio, enfocados en la predicción intradiaria de los activos de renta variable.
- **Entendimiento de los datos:** Se recolectaron, examinaron y comprendieron los datos históricos disponibles desde plataformas como Yahoo Finance.
- **Preparación de los datos:** Se procesaron los datos mediante limpieza, transformación y selección de características relevantes para el modelo.
- **Modelamiento:** Se aplicaron modelos de Deep Learning para el entrenamiento con los datos históricos.
- **Evaluación:** Se evaluaron los resultados del modelo mediante métricas como MSE, RMSE y MAE.
- **Implementación:** Se planteó la aplicación práctica de los modelos en contextos reales de predicción intradiaria.

Figura 1. Metodología CRISP-DM



Nota: Se muestran las fases de la metodología y cómo se interrelacionan en un proceso iterativo de análisis de datos.

Cabe mencionar que, en el presente estudio, se desea demostrar y probar que el algoritmo basado en Deep Learning prediga el futuro de los activos de renta variable en un proceso intradiario.

Finalmente, la investigación presentó un nivel predictivo con un enfoque cuantitativo que se relaciona con el aprendizaje automático, el cual es un proceso iterativo que requiere que los científicos de datos prueben diferentes enfoques y experimenten con diferentes algoritmos y parámetros, lo que lo hace más similar a la investigación cuantitativa que a la ingeniería. Además, los científicos de datos deben tener una sólida comprensión de las matemáticas y la estadística, así como la capacidad de trabajar con grandes conjuntos de datos y aplicar técnicas de visualización de datos para encontrar patrones y relaciones interesantes.

## A. Población

En la población se consideran los datos históricos de los activos de renta variable que se encuentran en la página de Yahoo Finance. Se consideraron únicamente los índices bursátiles Dow Jones (30 activos), S&P 500 (500 activos), Nasdaq (más de 4000 activos), Russell 2000 (2000 activos) para este estudio. Sin embargo, para este estudio se toman los valores del índice agregado de cada uno como representativo del comportamiento general del mercado, y no los datos individuales de cada acción que los conforma.

## **B. Muestra**

Se utilizaron datos históricos obtenidos de la página de Yahoo Finance. La selección del periodo se realizó en función de la disponibilidad histórica de datos para cada índice bursátil, considerando como fecha de corte el 15 de julio de 2025 para un mejor estudio y entrenamiento del algoritmo basado en Deep Learning.

- Para Russell 2000, los datos disponibles desde el 10 de setiembre de 1987 hasta la fecha de corte fueron de aproximadamente 9600.
- Para Dow Jones, los datos disponibles desde el 02 de enero de 1992 hasta la fecha de corte fueron de aproximadamente 8300.

Por consiguiente, el criterio de selección de estos datos se basó en su disponibilidad completa y continua en Yahoo Finance, buscando obtener la mayor cantidad de información histórica posible para un mejor estudio y entrenamiento del algoritmo.

## **2.2 Técnicas e instrumentos de recolección y análisis de datos**

### **A. Técnicas**

#### **a. Técnicas de recolección de datos**

La técnica de recolección de datos utilizada en la presente investigación fue el análisis documental, a través de la recopilación de información secundaria proveniente de la base de datos de Yahoo Finance. Se obtuvieron registros históricos de índices bursátiles como el Dow Jones, S&P 500, Nasdaq y Russell 2000, así como un conjunto de acciones representativas (Apple, Microsoft, Amazon, Tesla y Google).

Para garantizar la confiabilidad de los datos, se seleccionaron únicamente fuentes reconocidas, completas y consistentes dentro del Yahoo Finance, asegurando que la información histórica utilizada fuese oficial y verificable.

#### **b. Técnicas de procesamiento**

Asimismo, se aplicaron técnicas de preprocesamiento de datos que incluyeron:

- Limpieza de datos, eliminando valores nulos o inconsistentes.
- Normalización y escalado de variables numéricas para mejorar el desempeño del modelo.
- Transformación de los datos en secuencias temporales para ser procesados por la red neuronal LSTM.

De esta manera, se garantizó que los datos recolectados fueron aptos y confiables para su posterior análisis. Además, se empleó LSTM (Long Short-Term Memory) por su capacidad para capturar dependencias temporales de largo plazo, fundamentales en series de datos financieros.

### **B. Instrumentos**

Para la recolección y análisis de los datos se emplearon los siguientes instrumentos:

- a. Instrumentos de recolección de datos

- Fuente de datos: Yahoo Finance, como repositorio digital de información histórica.
- Entorno de programación: Jupyter Notebook (Anaconda), utilizado para manipular, limpiar y procesar los datos.
- Librerías de Python:
  - Pandas y NumPy: para la manipulación y el análisis de datos tabulares.
  - Scikit-Learn: para la división de los conjuntos de entrenamiento y prueba.
  - TensorFlow/Keras: para la implementación de los modelos de redes neuronales LSTM.
- b. Instrumentos de análisis y validación del modelo
  - Métricas matemáticas de evaluación: se emplearon fórmulas como Mean Squared Error (MSE), Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), Mean Absolute Percentage Error (MAPE) y el coeficiente de determinación ( $R^2$ ), que permitieron cuantificar la precisión y el margen de error de las predicciones del modelo.

En conjunto, estos instrumentos facilitaron la recolección, procesamiento y análisis de los datos, asegurando la validez y confiabilidad de los resultados obtenidos en la investigación.

### **2.3 Procedimiento**

El estudio se realizó con un enfoque cuantitativo, basado en el análisis de datos numéricos mediante técnicas estadísticas y computacionales. Se sitúa en un nivel predictivo, ya que busca anticipar el comportamiento futuro de los activos financieros. Tiene un alcance aplicado y correlacional, pues intenta resolver un problema práctico en

el ámbito financiero mientras examina relaciones entre variables. Además, utiliza un diseño no experimental y longitudinal, observando datos históricos en su contexto natural a lo largo del tiempo, sin manipular las variables independientes. La población de la investigación incluyó todos los datos históricos de activos bursátiles disponibles en Yahoo Finance, mientras que la muestra estuvo compuesta por registros de índices como Dow Jones, S&P 500, Nasdaq y Russell 2000, además de acciones representativas como Apple, Microsoft, Amazon, Tesla y Google, recopilados hasta el 15 de julio de 2025. La selección se basó en la disponibilidad continua y completa de la información para garantizar la calidad y coherencia de los datos utilizados en el modelado predictivo. La técnica de recolección fue el análisis documental de registros secundarios en Yahoo Finance. Para el procesamiento, se aplicaron procedimientos de limpieza, estandarización, normalización y transformación en secuencias temporales, además de dividir los datos en conjuntos de entrenamiento y prueba. Los instrumentos utilizados incluyeron Jupyter Notebook como entorno de programación y librerías de Python (Pandas, NumPy, Scikit-Learn y TensorFlow/Keras), complementadas con métricas de validación como MSE, RMSE, MAE, MAPE y  $R^2$ , garantizando la validez y fiabilidad de los resultados.

Realizamos un flujograma de los procesos que se requieren para diseñar el algoritmo y tener su estructura (Anexo N° 4). Teniendo los procesos mapeados, procedemos a utilizar la metodología CRISP-DM. En la sección I, podemos encontrar la comprensión del negocio; en el Anexo N° 1.1, la comprensión de los datos; en el Anexo N° 1.2, la preparación de los datos; en el Anexo N° 1.3, el modelamiento; y en la sección III, la evaluación y la implementación.

Primero seleccionaron algunos activos de renta variable de la página Yahoo Finance para poder analizar sus datos históricos (Anexo N° 1.1.2).

Luego de haber extraído toda la base de datos de los datos históricos de los activos seleccionados en el algoritmo, tratamos los datos faltantes (Anexo N° 1.2.1) y los estacionalizamos para poder entrenar el algoritmo de manera óptima (Anexo N° 1.2.2).

Todos esos datos ya tratados, los separamos en dos conjuntos: el 80% para el entrenamiento, que incluye la validación, y el 20% para su testeo (Anexo N° 1.2.3).

Después, estandarizamos los datos de entrenamiento y testeo para un correcto entrenamiento del algoritmo (Anexo N° 1.2.4). El 80% de los datos históricos se utiliza para entrenar la red neuronal y enseñarle a predecir el precio de los activos de renta variable (Anexo N° 1.3.1).

Por último, realizamos el testeo con el 20 % de los datos históricos restantes y se espera la salida de una predicción del precio del activo seleccionado (Anexo N° 1.3.2). Una vez obtenida, procedemos a realizar la comparación, análisis e interpretación.

Finalmente, se contrastaron los resultados con la hipótesis planteada.

Tras completar el procedimiento técnico según la metodología CRISP-DM, fue necesario realizar un análisis estadístico adicional para evaluar la precisión y validez de los resultados. Este análisis incluyó métricas de error usadas en la validación del modelo, además de un nivel de confianza estándar en investigaciones cuantitativas, lo que fortaleció la solidez y el respaldo académico de las conclusiones.

#### **2.4 Análisis Estadístico**

El análisis estadístico se llevó a cabo mediante el uso de métricas de error comúnmente aceptadas en estudios de predicción de series temporales, entre ellas, Mean Squared Error (MSE), Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE) y el coeficiente de determinación ( $R^2$ ). Estas métricas permitieron cuantificar con precisión el desempeño del modelo en términos de exactitud y confiabilidad de las predicciones realizadas. Además, y en coherencia con el

carácter cuantitativo de la investigación, se consideró un nivel de confianza del 95% ( $\alpha = 0.05$ ) como referencia estadística para la interpretación de los resultados. Dicho estándar otorga validez a las conclusiones y asegura que las diferencias observadas en el desempeño del modelo no se deben al azar, sino a patrones identificados en los datos históricos. Es importante mencionar que no se realizaron pruebas de hipótesis tradicionales como t-test o ANOVA, ya que el objetivo de la investigación no era comparar medias entre grupos, sino evaluar la capacidad predictiva de un algoritmo. Por lo tanto, la validación estadística del estudio se basa en la precisión de las métricas de desempeño, en la reproducibilidad del método y en la coherencia de los resultados en diferentes conjuntos de datos.

## **2.5 Aspectos Éticos**

Finalmente, en cuanto a los aspectos éticos, se respeta la línea de investigación establecida por la escuela, que es Tecnologías Emergentes. Por otro lado, se respeta la sublínea de investigación, que se delimita por “Data mining, machine/deep learning, Internet of Things (IoT), realidad aumentada (AR) y warehousing inteligente”. También se considera la confidencialidad de los datos que serán utilizados antes, durante y después de la ejecución del algoritmo basado en Deep Learning, ya sea en las pruebas o en la predicción del precio de los activos de renta variable. Asimismo, se contempla la aprobación y/o el consentimiento por parte de la Universidad Privada del Norte para la realización del trabajo de investigación. Por último, respetando las líneas de referencia y citación de la documentación, ésta se realiza en formato APA 7<sup>a</sup> edición, con el fin de salvaguardar el valor y esfuerzo de la población de investigación científica en sus definiciones, mostradas para ser insumos teóricos en nuevos conceptos.

## CAPÍTULO III: RESULTADOS

A continuación, se muestran los resultados del algoritmo de Deep Learning utilizado para predecir precios de activos de renta variable. El modelo se implementó en Anaconda con Jupyter (versión 6.4.12) y se ejecutó en un equipo con Windows 11, procesador Intel Core i7, 12 GB de RAM y una tarjeta gráfica NVIDIA GeForce MX130.

### 3.1 Resultados del Objetivo General

El objetivo general fue desarrollar un algoritmo de Deep Learning para mejorar la predicción de precios de activos de renta variable. Se implementó con éxito un modelo LSTM que cuenta con una capa de entrada de 365 neuronas (correspondientes a los días del año), una capa oculta y una capa de salida. Se utilizó el optimizador Adam y la función de pérdida elegida fue el Error Cuadrático Medio (MSE).

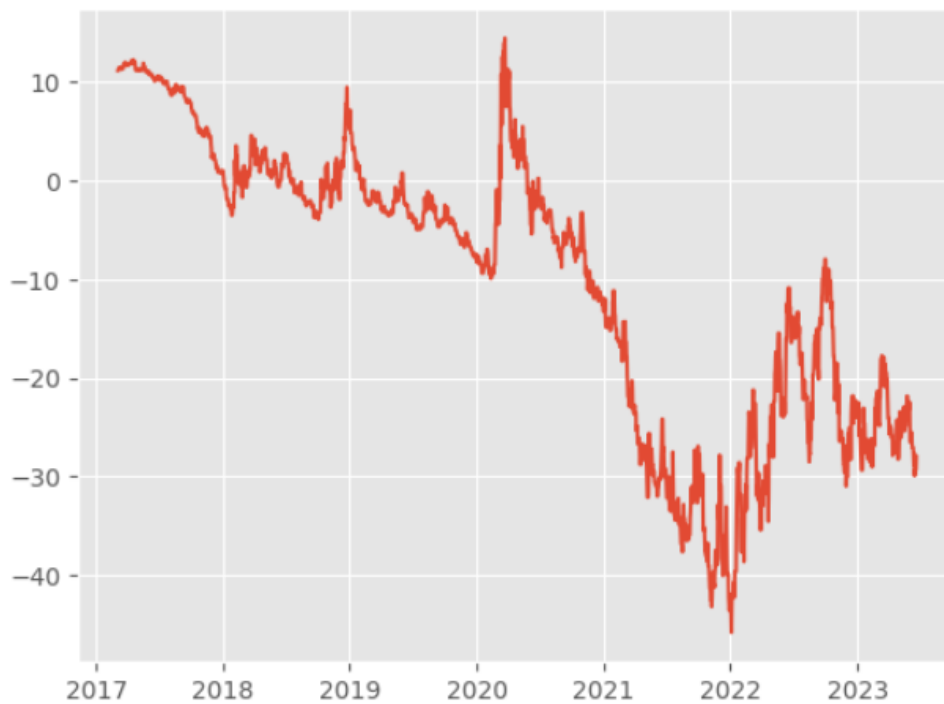
*Figura 2. Gráfica de comparación entre el precio real y el precio predicho del activo del Dow Jones*



*Nota:* En esta gráfica, hacemos una gráfica con la recta de los precios reales y de los precios predichos. Como dice en la leyenda, la recta azul viene a ser el valor de los precios en el activo real, y la recta roja viene a ser la predicción del precio con el modelo LSTM. Con este gráfico también apreciamos los días y los valores de cada segmento.

Figura 3. Gráfica de diferencia de error entre precios del Dow Jones

```
Date
2017-03-05    11.167969
2017-03-06    11.267578
2017-03-07    11.324219
2017-03-08    11.453125
2017-03-09    11.453125
...
2023-06-16   -29.304688
2023-06-17   -29.304688
2023-06-18   -29.304688
2023-06-19   -29.304688
2023-06-20   -27.984375
Freq: D, Length: 2299, dtype: float64
```



Las métricas de validación del Dow Jones fueron:  $MSE = 308.52$ ,  $MAE = 13.07$ ,  $RMSE = 17.56$ ,  $MAPE = 0.04$  y  $R^2 = 0.9999$ . Estas cifras reflejan un ajuste muy preciso. En particular, un MAPE de 0.04 implica un error promedio relativo de solo 4 %, que es muy bajo en el ámbito financiero. El coeficiente de determinación ( $R^2 = 0.9999$ ) indica que el modelo captura casi toda la variabilidad de los datos, reforzando su eficacia. No obstante, un  $R^2$  tan cercano a 1 también podría señalar un posible sobreajuste, por lo que se recomienda realizar pruebas adicionales en contextos de alta volatilidad.

Tabla 1. Lista de Cotejo

Activo de Renta Variable	MSE	MAE	RMSE	MAPE	R2
<b>Dow Jones</b>	308.5247519 1458973	13.07520168415 6156	17.56487267003634 4	0.04195100516266 3035	0.999985059381 9053
<b>S&amp;P 500</b>	207.4574431 9262363	8.986068808728 177	14.40338304679229 3	0.30494258118945 29	0.999804782446 8405
<b>NASDAQ</b>	3268.217288 646923	39.92209176573 527	57.16832417210533 6	0.42442126651799 73	0.999738974043 3804
<b>Russell 2000</b>	58.50922211 086878	7.132165163094 1565	7.649132114878705 5	0.41417055610875 025	0.999419449250 0649

Se observa que el índice Nasdaq muestra errores principales (MSE = 3268.21, MAE = 39.92, MAPE = 0.42). Esto se debe a su carácter más volátil, lo que provoca mayores desviaciones y representa un desafío extra para predecir activos muy inestables.

### 3.2 Resultados del Objetivo Específico 1

El primer objetivo específico era validar la precisión del algoritmo mediante la comparación entre los precios predichos y los precios reales de los activos de renta variable. Los experimentos realizados indican que el modelo puede representar con alta precisión la dinámica de los activos.

Figura 4. Métricas obtenidas para el índice del Oro

```
#Colocamos una fórmula para mostrar la validación del algoritmo en base al MSE, MAE, RMSE, MAPE y R2
def evaluacion_metrica(y_true, y_pred):
    def mean_absolute_percentage_error(y_true, y_pred):
        y_true, y_pred = np.array(y_true), np.array(y_pred)
        return np.mean(np.abs((y_true - y_pred) / y_true)) * 100
    print("-Resultados de las métricas de evaluación:-")
    print(f"El MSE es : {metrics.mean_squared_error(y_true, y_pred)}")
    print(f"El MAE es : {metrics.mean_absolute_error(y_true, y_pred)}")
    print(f"El RMSE es : {np.sqrt(metrics.mean_squared_error(y_true, y_pred))}")
    print(f"El MAPE es : {mean_absolute_percentage_error(y_true, y_pred)}")
    print(f"El R2 es : {metrics.r2_score(y_true, y_pred)}", end = "\n\n")

[24] #Mostramos los resultados de la validación de las fórmulas matemáticas
evaluacion_metrica(test_data["Precio Real"], test_data["Predicciones_LSTM"])

-Resultados de las métricas de evaluación:-
El MSE es : 3268.217288646923
El MAE es : 39.92209176573527
El RMSE es : 57.168324172105336
El MAPE es : 0.4244212665179973
El R2 es : 0.9997389740433804
```

La interpretación de las métricas respalda este descubrimiento: un MAE bajo indica que los errores absolutos promedio son pequeños en tamaño, mientras que el RMSE penaliza más severamente los errores grandes, que también permanecen controlados. Estos indicadores confirman que el modelo se generaliza adecuadamente en los activos analizados, validando así su precisión en la estimación de precios reales.

### 3.3 Resultados del Objetivo Específico 2

El segundo objetivo específico era evaluar la capacidad del algoritmo para realizar predicciones dinámicas de los precios futuros de distintos activos de renta variable. Este comportamiento se comprobó al entrenar el modelo de manera independiente para varios índices, lo que permitió observar su capacidad de adaptación a diferentes series temporales y condiciones del mercado.

Figura 5. Comparación entre valores reales y predichos del Russell 2000

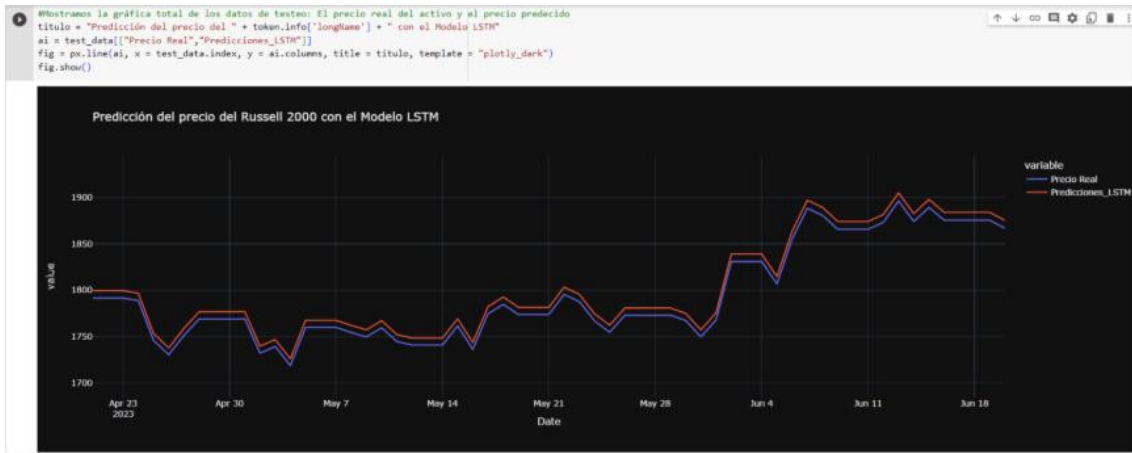


Figura 6. Comparación entre valores reales y predichos del Nasdaq



Desde una perspectiva científica, el carácter dinámico del modelo LSTM mejora su utilidad práctica, ya que posibilita realizar predicciones en diversos activos en tiempo real. Esto aumenta su valor para la gestión de portafolios y la formulación de estrategias de inversión basadas inteligencia artificial.

## CAPÍTULO IV: DISCUSIÓN Y CONCLUSIONES

### 4.1 Discusión

De acuerdo con nuestros resultados de la investigación, el modelo LSTM aplicado a la predicción de precios de activos de renta variable no solo alcanza una precisión elevada, sino que también representa un avance significativo frente a enfoques tradicionales. En el caso del Dow Jones, se logró un  $R^2$  de 0.99998 y un MAPE de apenas 0.0419, indicadores que reflejan una capacidad predictiva muy sólida. Estos resultados afirman que el algoritmo es capaz de reconocer patrones temporales complejos en datos históricos, minimizando el margen de error y reforzando su beneficio práctico para el análisis financiero.

Nuestros resultados se comparan con investigaciones como Farida et al. (2017), quien utilizó un modelo de movimiento Browniano Geométrico, obtuvo un  $MAPE \leq 20\%$ , una cifra considerablemente superior al error promedio del 4 % que se registra en los principales índices en este estudio. Este contraste pone de manifiesto la superioridad de los enfoques basados en Deep Learning. De manera similar, Srivastava et al. (2021) demostraron que las redes neuronales profundas superan a otros métodos de aprendizaje automático en predicción bursátil en la India. Nuestros resultados refuerzan esta afirmación, evidenciando que un modelo LSTM bien configurado puede ofrecer un ajuste casi perfecto en mercados internacionales, más allá de las fronteras y particularidades regionales.

Por otro lado, Mehtab et al. (2020) determinaron que un modelo univariante LSTM optimizado era el más adecuado para el índice NIFTY 50. En cambio, nuestra investigación afirma que un LSTM correctamente parametrizado puede mantener altos niveles de precisión incluso al aplicarse a múltiples índices globales, como Dow Jones, Nasdaq o Russell 2000, lo que amplía su potencial de uso.

Asimismo, Park y Lee et al. (2019) habían destacado la eficacia de técnicas de boosting como AdaBoost en la predicción del S&P 500. No obstante, el enfoque LSTM presentado aquí va un paso más allá al ofrecer predicciones dinámicas y adaptables a distintos activos, sin restringirse a un único índice y aumentando su valor para los analistas financieros.

### **Limitaciones**

El modelo presenta un valor de  $R^2$  tan elevado que podría indicar cierto sobreajuste, por lo que se recomienda validar el algoritmo en contextos de alta volatilidad o crisis financieras para garantizar su robustez. Tener en cuenta que el trabajo se basa únicamente en datos históricos de precios, dejando fuera variables externas como tasas de interés, inflación o indicadores macroeconómicos que también influyen en el mercado. El acceso a datos de calidad en tiempo real sigue siendo un obstáculo, pues a menudo está condicionado por plataformas de pago, limitando la reproducibilidad y generalización del modelo en entornos reales.

### **Implicancias**

En términos teóricos, este estudio refuerza la idea de que el Deep Learning es una herramienta confiable para el análisis financiero, mostrando que la arquitectura LSTM puede identificar de forma efectiva dependencias temporales de largo plazo en series financieras.

Desde una perspectiva metodológica, la investigación proporciona evidencia sustancial sobre la eficacia del enfoque cuantitativo y el uso de la metodología CRISP-DM en proyectos de predicción financiera basados en datos. Además, exhibe que la utilización de métricas como MAE, RMSE y MAPE conforma un marco de validación adecuado para la evaluación de modelos de aprendizaje profundo aplicados a activos de renta variable.

En el ámbito académico y práctico, estos resultados abren el camino a futuros desarrollos de modelos híbridos, combinando LSTM con análisis de sentimiento proveniente de redes sociales o noticias financieras (como sugieren Mehtab y Park), para capturar tanto patrones cuantitativos como cualitativos y mejorar la toma de decisiones en entornos financieros reales.

#### 4.2 Conclusión

- La investigación logró diseñar y desarrollar un **algoritmo de Deep Learning (LSTM)** que mejora significativamente la predicción de precios de activos de renta variable. Este logró, respaldado por  $R^2$  cercano a 1 y con un MAPE promedio del 4%, no solo respalda la hipótesis inicial, sino que demuestra que la arquitectura LSTM puede convertirse en una herramienta sólida y confiable para el análisis financiero. Esto nos afirma que los inversionistas y analistas pueden contar con un sistema capaz de detectar patrones ocultos en los datos históricos y transformar esa información en predicciones más precisas.
- Con respecto al objetivo 1 e hipótesis 1, se tiene que el algoritmo valida su precisión al estimar precios reales de los activos a partir de datos históricos con altos niveles de precisión. Las métricas de validación (MAE, RMSE y MAPE) confirmaron que los errores absolutos permanecen en niveles bajos, reforzando la fiabilidad del modelo para aplicarse a diferentes activos y contextos de mercado.
- Finalmente, con respecto al objetivo 2 e hipótesis 2, se evaluó la capacidad del algoritmo para realizar predicciones dinámicas bajo diferentes condiciones del mercado. El modelo evidenció flexibilidad y adaptación al ser entrenado y aplicado sobre diversos activos de renta variable con distintos niveles de volatilidad. Aunque índices más inestables, como el **Nasdaq**, presentaron

mayores errores, la tendencia general fue correctamente captada, lo que confirma la hipótesis específica 2 sobre la **naturaleza dinámica y adaptable** del algoritmo. Este resultado sugiere que el LSTM puede ser una base sólida para sistemas predictivos que operen en tiempo real y en múltiples mercados.

Esta investigación demuestra que las redes neuronales LSTM constituyen un recurso significativo para analistas e inversionistas, permitiéndoles complementar su juicio humano con predicciones objetivas y precisas. También, contribuye al campo del *Deep Learning* aplicado a las finanzas, sentando las bases para soluciones tecnológicas que acerquen la inteligencia artificial a la toma de decisiones estratégicas en un entorno tan incierto y cambiante como el mercado de valores.

#### **4.3 Recomendaciones**

- Se recomienda integrar variables macroeconómicas, análisis de sentimiento y datos provenientes de redes sociales. Además, evaluar su rendimiento en mercados emergentes y escenarios de alta volatilidad. Del mismo modo, sería conveniente aplicar técnicas de regularización que ayuden a reducir posibles problemas de sobreajuste y aporten mayor solidez a las predicciones.
- Se recomienda aprovechar el potencial de las redes neuronales LSTM como apoyo para los analistas e inversionistas, ya que estas permiten respaldar la intuición y la experiencia humana con proyecciones más objetivas y confiables. En este sentido, el presente estudio no solo refuerza el aporte académico en la aplicación del Deep Learning a las finanzas, sino que también abre la posibilidad de desarrollar soluciones tecnológicas que acerquen la inteligencia artificial a la toma de decisiones estratégicas dentro de un entorno financiero dinámico e incierto.

## REFERENCIAS

- Albahli, S., Irtaza, A., Nazir, T., Mehmood, A., Alkhalifah, A., & Albattah, W. (2022). A machine learning method for prediction of the stock market using real-time Twitter data. *Electronics* 2022, Vol. 11, Page 3414, 11(20), 3414. <https://doi.org/10.3390/ELECTRONICS11203414>
- Aldhyani, T. H. H., & Alzahrani, A. (2022). Framework for Predicting and Modeling Stock Market Prices Based on Deep Learning Algorithms. *Electronics* 2022, Vol. 11, Page 3149, 11(19), 3149. <https://doi.org/10.3390/ELECTRONICS11193149>
- Ampomah, E. K., Nyame, G., Qin, Z., Addo, P. C., Gyamfi, E. O., & Gyan, M. (2021). Stock market prediction with Gaussian naïve Bayes machine learning algorithm. *Informatica (Slovenia)*, 45(2), 243–256. <https://doi.org/10.31449/inf.v45i2.3407>
- Farida Agustini, W., Affianti, I. R., & Putri, E. R. (2018). Stock price prediction using geometric Brownian motion. *Journal of Physics: Conference Series*, 974, 012047. <https://doi.org/10.1088/1742-6596/974/1/012047>
- Lin, Y. L., Lai, C. J., & Pai, P. F. (2022). Using Deep Learning Techniques in Forecasting Stock Markets by Hybrid Data with Multilingual Sentiment Analysis. *Electronics* 2022, Vol. 11, Page 3513, 11(21), 3513. <https://doi.org/10.3390/ELECTRONICS11213513>
- Mehtab, S., Sen, J., & Dutta, A. (2021). Stock Price Prediction Using Machine Learning and LSTM-Based Deep Learning Models. *Communications in Computer and Information Science*, 1366, 88–106. [https://doi.org/10.1007/978-981-16-0419-5\\_8/COVER](https://doi.org/10.1007/978-981-16-0419-5_8/COVER)
- Ng, A. (2018). *Machine Learning Yearning: Technical Strategy for AI Engineers, In the Era of Deep Learning*.

- Pagolu, V. S., Reddy, K. N., Panda, G., & Majhi, B. (2016). Sentiment analysis of Twitter data for predicting stock market movements. *2016 International Conference on Signal Processing, Communication, Power and Embedded System (SCOPES)*, 1345–1350. <https://doi.org/10.1109/SCOPES.2016.7955659>
- Park, M., Lee, M. L., & Lee, J. (2019). Predicting stock market indices using classification tools. *Asian Economic and Financial Review* , 9(2), 243–256. <https://doi.org/10.18488/JOURNAL.AEFR.2019.92.243.256>
- Srivastava, P. R., Zuopeng (Justin) Zhang, & Eachempati, P. (2021). Deep Neural Network and Time Series Approach for Finance Systems. *Journal of Organizational and End User Computing*, 33(5), 204–226. <https://doi.org/10.4018/JOEUC.20210901.0a10>

## ANEXOS

### ANEXO N°1 Desarrollo del algoritmo

En este capítulo, se gestiona el proceso de desarrollo del algoritmo basado en Deep Learning para predecir el precio de los activos de renta variable siguiendo la metodología CRISP-DM.

#### ANEXO N°1.1 Comprensión de los Datos

##### ANEXO N°1.1.1 Descargas de Librerías para la conexión con Yahoo Finance y el Funcionamiento del Algoritmo

Para este algoritmo, utilizamos los datos históricos de los precios de los activos de renta variable en Yahoo Finance. En la Figura 7, 8 y 9; colocamos las librerías para poder conectarnos a la base de datos de Yahoo Finance, colocamos el token de la divisa de la cual se quiere extraer los datos y el intervalo de tiempo de los últimos cierres en el mercado en cada día para que nos pueda mostrar los datos que están en la base de datos desde que se creó el activo de renta variable elegido.

Figura 7. Librerías de Yahoo Finance

```
Librerías Principales para el Funcionamiento del Algoritmo

In [ ]: pip install yfinance

In [ ]: pip install yahoo-finance

In [ ]: pip install yahoofinancials

In [ ]: # Manipulación y tratamiento de Datos
import numpy as np
import pandas as pd

# Visualización de datos
import plotly.express as px
import matplotlib.pyplot as plt
%matplotlib inline
plt.style.use('ggplot')

# Descargar datos de yahoo
import yfinance as yf
from yahoofinancials import YahooFinancials

# Métrica de Evaluación
from sklearn.metrics import mean_squared_error
from statsmodels.tools.eval_measures import rmse
from sklearn import metrics

# No presentar advertencias
import warnings
warnings.filterwarnings("ignore")
```

*Nota:* Las 3 librerías esenciales para la conexión a la base de datos de Yahoo Finance son las siguientes: yfinance, yahoo-finance y yahoofinancials. También extraemos para la manipulación y tratamiento de datos las librerías de numpy y pandas, las librerías de matplotlib y plotly para la visualización de los datos y la librería skLearn para las métricas de evaluación del algoritmo.

### ANEXO N°1.1.2 Extracción de los Datos Históricos del Activo de Renta Variable Seleccionado

Al momento de descargar y nombrar las librerías para conectarse a la base de datos de Yahoo Finance, realizamos los pasos para poder extraer los datos históricos del activo de renta variable elegido y analizar con detenimiento antes del preprocesamiento como se aprecian en la Figura 8 y 9.

Figura 8. Elección del activo de renta variable

▼ Extracción de los Datos Históricos del Activo de Renta Variable de su Preferencia

```
[5] #Se pide el token del activo de renta variable
while True:
    try:
        nombre_token = input("Ingrese el valor del token: ")
        token = yf.Ticker(nombre_token)
        #Extrae el nombre del token elegido y se muestra para validar que no se haya equivocado
        token.info["longName"]
        break
    except:
        #Si el valor del token no es válido, te vuelve a pedir un valor hasta que sea válido
        print("El token no es válido, ingrese de nuevo un token que se encuentre en Yahoo Finance...\n")
```

Ingrese el valor del token: ^DJI

*Nota:* Utilizamos una validación para los valores del token que se va a digitar, si se llega a digitar de manera correcta un activo de renta variable que se encuentre en la base de datos de Yahoo Finance. En este caso elegimos el token del Dow Jones y se digita de manera correcta, en cambio si no fuera válido, saldría el error que se muestra en el campo de “except” diciendo “El token no es válido, ingrese de nuevo un token que se encuentre en Yahoo Finance.”.

Figura 9. Datos históricos del activo del Dow Jones antes del preprocesamiento

```
[ ] #Dependiendo del token que se ha escogido, se extrae de la base de datos de Yahoo Finance los datos históricos de dicho token
df = yf.download(nombre_token,
                #El periodo viene a ser "max" para que se extraigan los precios desde el inicio de la apertura del activo de renta variable
                period = "max",
                progress = False)
df
```

Date	Open	High	Low	Close	Adj Close	Volume
1992-01-02	3152.100098	3172.629883	3139.310059	3172.399902	3172.399902	23550000
1992-01-03	3172.399902	3210.639893	3165.919922	3201.500000	3201.500000	23620000
1992-01-06	3201.500000	3213.330078	3191.860107	3200.100098	3200.100098	27280000
1992-01-07	3200.100098	3210.199951	3184.479980	3204.800049	3204.800049	25510000
1992-01-08	3204.800049	3229.199951	3185.820068	3203.899902	3203.899902	29040000
...	...	...	...	...	...	...
2023-06-12	33906.800781	34077.839844	33878.460938	34066.328125	34066.328125	308130000
2023-06-13	34111.078125	34310.281250	34107.980469	34212.121094	34212.121094	308790000
2023-06-14	34044.699219	34151.421875	33783.550781	33979.328125	33979.328125	354910000
2023-06-15	33945.980469	34488.980469	33945.980469	34408.058594	34408.058594	361400000
2023-06-16	34464.019531	34588.679688	34285.691406	34299.121094	34299.121094	595230000

7924 rows x 6 columns

*Nota:* En esta parte se muestra la descarga de los datos históricos dependiendo del token que se ha escogido. El periodo viene a ser "max" para que se extraigan los precios desde el inicio de la apertura del activo de renta variable y que se tenga un completo entrenamiento con mayor eficacia en el algoritmo.

En este caso nos muestra 7924 datos que tiene almacenados los datos con los siguientes atributos: Date, Open, High, Low, Close, Adj Close y Volume. Los tipos de datos de cada atributo se muestra en la Tabla 2.

*Tabla 2. Tipo de Datos y Descripción de las Variables antes del preprocesamiento*

<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
<b>Date</b>	Date (fecha)	Fecha de operación
<b>Open</b>	Float (decimal)	Precio de apertura del activo en ese día
<b>High</b>	Float (decimal)	Mayor precio del activo en ese día
<b>Low</b>	Float (decimal)	Menor precio del activo en ese día
<b>Close</b>	Float (decimal)	Precio de cierre del activo en ese día
<b>Adj Close</b>	Float (decimal)	Precio posible de cierre del activo en ese día
<b>Volume</b>	Float (decimal)	Volumen de precios del activo en ese día

*Nota:* Analizamos los datos para ver cuál nos ayudará en el entrenamiento y predicción del algoritmo.

Finalmente, planteamos elegir la variable de cierre del precio en el día ya que podemos ver cómo se comportará el activo de renta variable seleccionado en ese día y podremos ver cómo cerrará.

Figura 10. Gráfico de precios del activo del Dow Jones



*Nota:* En esta figura podemos apreciar la gráfica exacta del activo seleccionado, en este caso el activo del Dow Jones, en base al cierre del día. Esta gráfica toma los valores del cierre del precio desde el inicio del activo hasta la actualidad y se puede demostrar que se han extraído de manera exitosa y correcta los precios.

## ANEXO N°1.2 Preparación de los Datos

### ANEXO N°1.2.1 Tratamiento de los Datos Históricos del Activo de Renta Variable Seleccionado

En la etapa de tratamiento de los datos de entrenamiento, eliminamos los datos innecesarios para nuestro análisis y solo quedarnos con el cierre del precio del activo seleccionado, que en este ejemplo viene a ser el activo del Dow Jones, y colocar los valores con una temporalidad diaria.

Figura 11. Eliminación de datos innecesarios

```
[ ] #Eliminamos los datos que no vamos a usar en el análisis de nuestro algoritmo y renombramos la variable "Close"
df["Precio Real"] = df["Close"]
df = df.drop(["Open"], axis=1)
df = df.drop(["High"], axis=1)
df = df.drop(["Low"], axis=1)
df = df.drop(["Adj Close"], axis=1)
df = df.drop(["Volume"], axis=1)
df = df.drop(["Close"], axis=1)
df
```

Date	Precio Real
1992-01-02	3172.399902
1992-01-03	3201.500000
1992-01-06	3200.100098
1992-01-07	3204.800049
1992-01-08	3203.899902
...	...
2023-06-12	34066.328125
2023-06-13	34212.121094
2023-06-14	33979.328125
2023-06-15	34408.058594
2023-06-16	34299.121094

7924 rows x 1 columns

Nota: Eliminamos las variables “Open”, “High”, “Low”, “Adj Close” y “Volume”. También creamos otra variable llamada “Precio Real” y copiamos los datos de la variable “Close” para después hacer una comparación de precios. Terminando de crear la variable, borramos la variable “Close” que ya no será de utilidad.

Tabla 3. Tipo de Datos y Descripción de las Variables después del tratamiento

Atributo	Tipo	Descripción
Date	Date (fecha)	Fecha de operación
Close	Float (decimal)	Precio de cierre del activo en ese día

Nota: Podemos apreciar que se seleccionó los precios de cierre en el día del activo seleccionado, quedándonos con solo dos atributos y el mismo tipo de dato y valor.

### ANEXO N°1.2.2 Estacionalización del Precio de Cierre del Activo de Renta Variable Seleccionado

Después de haber elegido el atributo necesario para nuestro algoritmo; procedemos a estacionalizar los datos, de manera que nos ayude a llenar los vacíos en los días que no abre el mercado y no varía el precio de cierre del activo de renta variable seleccionado.

Figura 12. Estacionalización de los precios de cierre



*Nota:* En esta parte, colocamos los valores de manera intradiaria en la frecuencia con valor “D” para que haya una continuidad en los datos. Por último, se rellenan los valores faltantes que hay en el histórico del activo por temas de que esos días en activo no tuvo ninguna operación, colocamos que se tiene una frecuencia diaria y que los valores que son vacíos se rellenen con el día anterior de un valor vacío gracias al método “ffill”, dando a entender que el precio no se movió. El tamaño de los datos termina siendo 11489, a comparación de los 7924 que había antes del preprocesamiento.

### ANEXO N°1.2.3 División de los Datos de Entrenamiento y Testeo

En este punto dividimos el 80% de los datos para el entrenamiento y el 20% para el testeo.

Figura 13. División de los datos de entrenamiento y testeo

```
[ ] #En esta parte, definimos los datos de entrenamiento y los datos de testeo (80% para el entrenamiento y 20% para el testeo)
train_data = df[:round(len(df) * 0.8)]
test_data = df[round(len(df) * 0.8):]
#En esta parte mostramos el total de datos que tiene cada variable y el total de datos que hay en total
train_data.shape, test_data.shape, df.shape

((9191, 1), (2298, 1), (11489, 1))
```

*Nota:* De los 11489 datos, 9191 datos son para el entrenamiento (que corresponde al 80% de los datos) y 2298 datos son para la prueba (que corresponde al 20% de los datos).

### ANEXO N°1.2.4 Estandarización del Precio de Cierre del Activo de Renta Variable Seleccionado

Para llegar a hacer la estandarización del precio de cierre del activo que se seleccionó, en este ejemplo el activo del Dow Jones, usamos la función del escalado `MinMaxScaler()` para que el modelo pueda procesar de manera eficiente los datos.

Figura 14. Estandarización de los valores de entrenamiento y testeo

```
[ ] #Utilizamos la librería del sklearn para poder escalar los valores de entrenamiento y testeo
from sklearn.preprocessing import MinMaxScaler
scaler = MinMaxScaler()

[ ] #Normalizamos los valores de entrenamiento y testeo para que el entrenamiento de éste sea óptimo
scaler.fit(train_data)
train_data_escalado = scaler.transform(train_data)
test_data_escalado = scaler.transform(test_data)
test_data_escalado

array([[0.99373823],
       [0.99389064],
       [0.99389064],
       ...,
       [1.715491 ],
       [1.73933724],
       [1.73327807]])
```

*Nota:* Como se muestra en la imagen, estandarizamos los valores de entrenamiento y de testeo. La importancia de la estandarizamos es para que no haya mucho ruido en la parte del entrenamiento del algoritmo, para la precisión de los precios predichos a comparación del precio real del activo y la reducción de los errores al momento de la validación.

Tabla 4. Tipo de Datos y Descripción de las Variables después de la Estandarización

Atributo	Tipo	Descripción
Date	Date (fecha)	Fecha de operación
Close	Float (decimal)	Precio estandarizado del

Nota: En esta parte nos quedamos con los dos atributos, con la diferencia que los valores varían en el atributo “Close” y vendrían a estar estandarizados.

## ANEXO N°1.3 Modelado

### ANEXO N°1.3.1 Entrenamiento de los Datos

Agrupamos los datos en función de los valores que se crea conveniente para tener un óptimo entrenamiento en el modelo LSTM que vamos a utilizar.

Figura 155. Preparación de los datos de entrenamiento por bloques

```
[ ] #Realizamos una agrupación de bloques para el lado del entrenamiento, quiere decir que se agrupará de time_step en time_step para que el siguiente dato sea la predicción
#Se tomará un dato del entrenamiento y su siguiente valor será una "Predicción"
time_step = 1
X_train = []
Y_train = []
#total es el total de datos de entrenamiento
total = len(train_data_escalado)

for i in range(time_step, total):
    # X: Bloques agrupados en time_step
    X_train.append(train_data_escalado[i - time_step:i, 0])
    # Y: el siguiente dato
    Y_train.append(train_data_escalado[i, 0])
X_train, Y_train = np.array(X_train), np.array(Y_train)
```

Nota: En esta función agrupamos los datos en función del hiperparámetro “time\_step”. Esta función lo que hace es juntar los datos de time\_step en time\_step. En este ejemplo, se agrupa de 1 en 1 cada dato y el siguiente valor, vendría a ser la “predicción” del precio en los datos de entrenamiento. Este hiperparámetro pude ir variando, dependiendo de la persona que configura o crea su modelo desde cero para hallar mejores valores de entrenamiento.

Para el entrenamiento de los datos, utilizamos el modelo LSTM univariado, que quiere decir que usa una variable.

Figura 16. Armado del Modelo LSTM

```
[ ] #En esta parte, gracias a la librería keras, armamos la cantidad de neuronas que tendrá nuestro modelo, qué activación tendrá y cuántos valores mostrará
from keras.models import Sequential
from keras.layers import Dense
from keras.layers import LSTM

#lo que estamos diciendo acá es que el modelo será secuencial, tendrá 365 neuronas en el hiperparámetro "units" en la capa de entrada,
#que tendrá una activación "relu" que es la más conocida y usada y que tendrá una neurona de capa de salida
#se le coloca 365 neuronas por la cantidad de días que tiene un año y que tenga mejor rendimiento
modelo_lstm = Sequential()
modelo_lstm.add(LSTM(units = 365, activation = "relu", input_shape = (x_train.shape[1], 1)))
modelo_lstm.add(Dense(1))
#El optimizador será el modelo de "adam" que es mejor que el "rmsprop" y las pérdidas serán en base al "error cuadrático medio (MSE)"
#que es el más usado en los datos supervisados clasificados
modelo_lstm.compile(optimizer = "adam", loss = "mse")

modelo_lstm.summary()
```

Layer (type)	Output Shape	Param #
lstm (LSTM)	(None, 365)	535820
dense (Dense)	(None, 1)	366

=====  
 Total params: 536,186  
 Trainable params: 536,186  
 Non-trainable params: 0  
 =====

*Nota:* En esta parte, gracias a la librería keras, armamos la cantidad de neuronas que tendrá nuestro modelo, qué activación tendrá y cuántos valores mostrará. Colocamos que: El modelo será secuencial, tendrá 365 neuronas en el hiperparámetro "units" en la capa de entrada, una activación "relu" que es la más conocida y usada, y que tendrá una neurona de capa de salida. Se le coloca 365 neuronas por la cantidad de días que tiene un año y que tenga mejor rendimiento. El optimizador será el modelo de "adam" que es mejor que el "rmsprop" y las pérdidas serán en base al "error cuadrático medio (MSE)" que es el más usado en los datos supervisados clasificados.

Al final, entrenamos el modelo LSTM para que vaya reconociendo patrones en los precios estandarizados del cierre del activo de renta variable seleccionado.

Figura 17. Entrenamiento del modelo

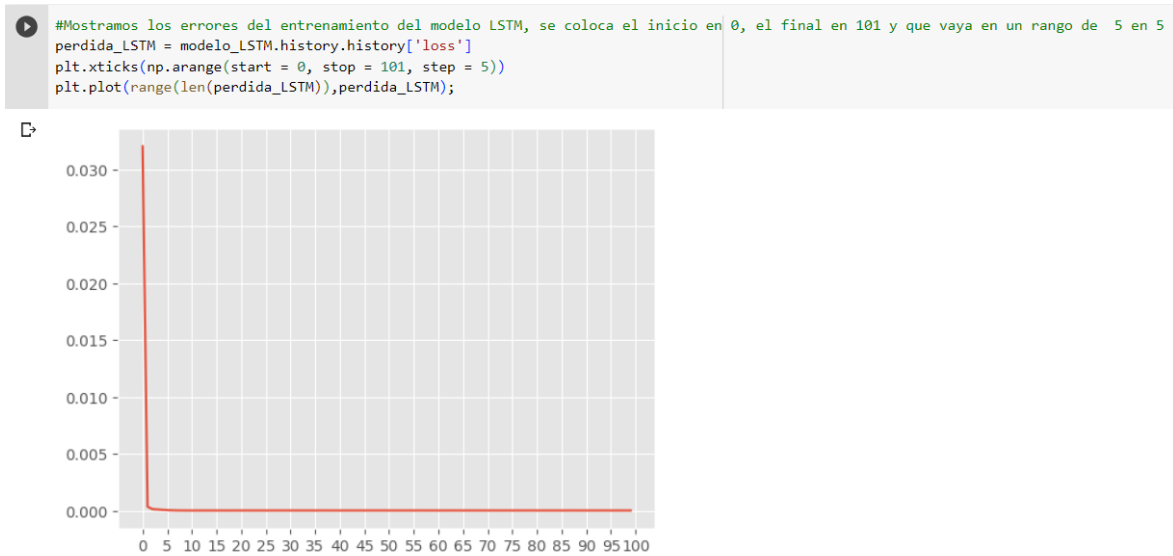
```
[ ] #En la parte de entrenamiento del algoritmo, colocamos los valores separados y ponemos el dato principal para el rendimiento que es el batch_size
#Le colocamos que hará 100 iteraciones de entrenamiento en el hiperparámetro epochs, y el batch_size es el número de muestras por actualización de degradado
#Una explicación sencilla de este hiperparámetro es que si no lo colocamos, predeterminadamente valdrá uno y analizará todos los datos totales del entrenamiento
#Esto produciría mucho ruido al momento del entrenamiento del modelo y que se demore mucho para entrenar
#En cambio, si colocamos el valor total de los datos del entrenamiento, no produciría ruido, pero a la vez se atascará el entrenamiento
#Entonces le ponemos un valor predeterminado o general para que el entrenamiento del modelo sea óptimo, que es 32
modelo_LSTM.fit(X_train, Y_train, epochs = 100, batch_size = 32)

288/288 [=====] - 6s 21ms/step - loss: 2.9862e-05
Epoch 73/100
288/288 [=====] - 6s 21ms/step - loss: 2.9412e-05
Epoch 74/100
288/288 [=====] - 6s 20ms/step - loss: 2.9235e-05
Epoch 75/100
288/288 [=====] - 5s 19ms/step - loss: 3.1263e-05
Epoch 76/100
288/288 [=====] - 7s 24ms/step - loss: 2.9032e-05
Epoch 77/100
288/288 [=====] - 7s 23ms/step - loss: 3.0197e-05
Epoch 78/100
288/288 [=====] - 7s 23ms/step - loss: 2.9989e-05
Epoch 79/100
288/288 [=====] - 5s 18ms/step - loss: 2.9571e-05
Epoch 80/100
288/288 [=====] - 6s 22ms/step - loss: 2.9264e-05
Epoch 81/100
288/288 [=====] - 6s 21ms/step - loss: 2.9516e-05
Epoch 82/100
288/288 [=====] - 7s 24ms/step - loss: 2.9634e-05
Epoch 83/100
288/288 [=====] - 5s 18ms/step - loss: 2.8736e-05
Epoch 84/100
288/288 [=====] - 6s 22ms/step - loss: 3.0604e-05
Epoch 85/100
288/288 [=====] - 6s 20ms/step - loss: 2.9567e-05
```

*Nota:* En la parte de entrenamiento del algoritmo, colocamos las iteraciones del entrenamiento en el hiperparámetro epochs, que es el número de iteraciones de entrenamiento por el lote dado, y el número de muestras por actualización de degradado en el hiperparámetro batch\_size. Una explicación sencilla del hiperparámetro batch\_size es que, si no lo colocamos, predeterminadamente valdrá uno y analizará todos los datos totales del entrenamiento. Esto produciría mucho ruido al momento del entrenamiento del modelo y que se demore mucho para entrenar. En cambio, si colocamos el valor total de los datos del entrenamiento, no produciría ruido, pero a la vez se atascará el entrenamiento. Entonces le ponemos un valor predeterminado o general para que el entrenamiento del modelo sea óptimo, que es 32.

Al terminar de entrenar el modelo, mostramos una gráfica de errores que ha tenido al momento de entrenarse y evidenciar los resultados de un buen entrenamiento.

Figura 18. Gráfica de errores del entrenamiento



*Nota:* Se muestra los valores del porcentaje de errores que tiene el algoritmo al momento de entrenar. Llamamos la función history para poder graficarlo y con la función xticks podemos darle los parámetros de la gráfica.

### ANEXO N°1.3.2 Predicción de los Precios Futuros del Activos de Renta Variable Seleccionado

Por última etapa, agruparemos los datos de testeo para poder predecir los valores futuros del precio del activo seleccionado, que en este caso es el Dow Jones.

Figura 19. Preparación de los datos de entrenamiento por bloques

```
[ ] #Agrupamos los datos de testeo como se hizo en los datos de entrenamieto en base al valor del "time_step"
X_test = []
for i in range(time_step, len(test_data_escalado) + time_step):
    X_test.append(test_data_escalado[i - time_step:i, 0])
X_test = np.array(X_test)
X_test = np.reshape(X_test, (X_test.shape[0], X_test.shape[1], 1))
```

*Nota:* En esta función agrupamos los datos en función del hiperparámetro “time\_step”. Como se explicó en el agrupamiento de los datos de entrenamiento, haremos lo mismo con los datos de testeo y el hiperparámetro seguirá siendo el mismo.

Finalmente, invertimos los precios predichos estandarizados para poder apreciar que precio lanzará en las predicciones desde la creación del activo de renta variable hasta el día que nos encontramos.

*Figura 20. Predicción de los valores entrenados*

```
[ ] #Predecimos los valores agrupados del testeo para que nos muestre el "siguiente valor"  
prediccion = modelo_LSTM.predict(X_test)  
#E invertimos la normalización o escalado para poder ver el futuro precio  
prediccion = scaler.inverse_transform(prediccion)
```

```
72/72 [=====] - 1s 6ms/step
```

*Nota:* En esta parte predecimos los valores de los precios del activo seleccionado, que en este ejemplo es el activo del Dow Jones. Llamamos a la función predict para almacenar en el arreglo “prediccion” todos los valores predichos y después invertir los valores escalados de dicha función.

## ANEXO N° 2 Matriz de Consistencia

Tabla 5. Matriz de Consistencia

PROBLEMA	OBJETIVOS	HIPÓTESIS	VARIABLES - DIMENSION - INDICADORES		
PROBLEMA PRINCIPAL:	OBJETIVO PRINCIPAL	HIPÓTESIS PRINCIPAL	VARIABLE INDEPENDIENTE (Solución)		
¿Cómo el algoritmo basado en Deep Learning mejora la predicción del precio de los activos de renta variable?	Diseñar el algoritmo basado en Deep Learning para la mejora de la predicción del precio de los activos de renta variable	El algoritmo basado en Deep Learning mejora significativamente la predicción del precio de los activos de renta variable	Algoritmo basado en Deep Learning		
			VARIABLE DEPENDIENTE (Problema)		
			Predicción del precio		
PROBLEMAS ESPECÍFICOS	OBJETIVOS ESPECÍFICOS	HIPÓTESIS ESPECÍFICOS	DIMENSIONES	INDICADORES	UNIDAD DE MEDIDA
<b>PE1:</b> ¿Cómo el algoritmo basado en Deep Learning predice distintos activos de renta variable de manera dinámica?	<b>OE1:</b> Validar como el algoritmo basado en Deep Learning predice distintos precios futuros de los activos de renta variable de forma dinámica	<b>HE1:</b> El algoritmo basado en Deep Learning predice los precios de los distintos activos de renta variable de manera dinámica	Precio	Cierre del precio intradiario	Número
<b>PE2:</b> ¿Cómo el algoritmo basado en Deep Learning mejora la predicción del precio de los activos de renta variable utilizando los datos históricos?	<b>OE2:</b> Validar como el algoritmo basado en Deep Learning se aproxima al precio real de los activos de renta variable utilizando los datos históricos	<b>HE2:</b> El algoritmo basado en Deep Learning mejora la validación de la aproximación del precio real de los activos de renta variable utilizando los datos históricos			

*Nota:* En esta matriz podemos hallar y relacionar de forma detallada los problemas que vamos a tratar, los objetivos ante la situación, las posibles soluciones o hipótesis y, definir las variables y dimensiones del tema.

### ANEXO N° 3 Matriz de Operacionalización

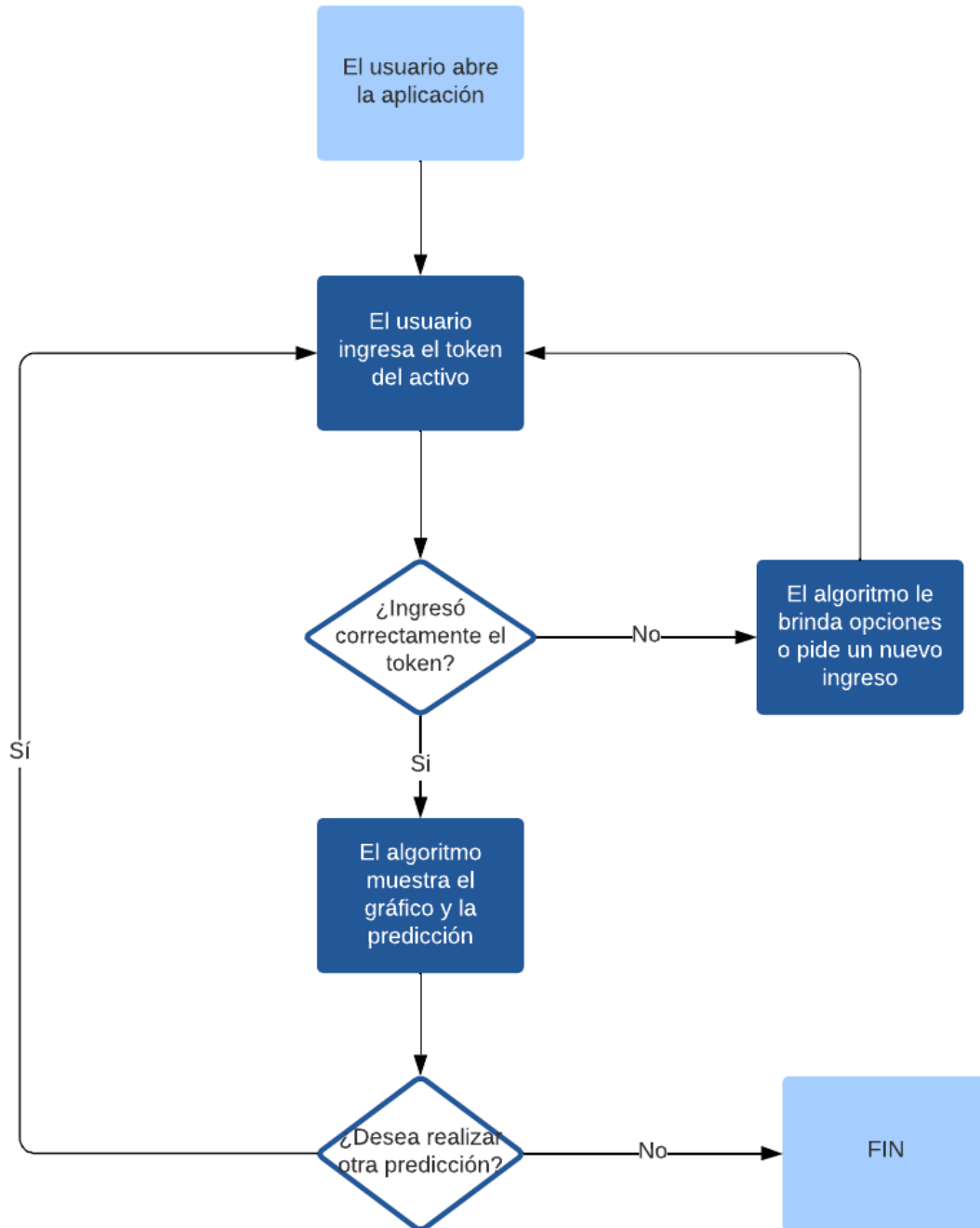
Tabla 6. *Matriz de Operacionalización*

Dimensiones	Indicador	Método	Fórmula
<b>Precio</b>	Porcentaje de eficacia de los precios predichos.	Cuantitativo	$EF = \frac{ T - A }{T} \times 100$ <p>Donde:            EF: Eficacia            A: Precio predicho del activo en el conjunto de Testeo            T: Precio real del activo en el conjunto de Testeo</p>

*Nota:* En esta matriz diseñamos la fórmula que se va a utilizar para indicar los hallazgos de cada dimensión.

**ANEXO N° 4 Flujograma de Procesos del Funcionamiento del Algoritmo**

Figura 21. Flujograma de Proceso del Funcionamiento del Algoritmo



*Nota:* El flujograma muestra cada proceso que sigue el algoritmo para hallar la predicción del precio en el activo de renta variable seleccionado.