



**UNIVERSIDAD
DE ANTIOQUIA**

Estimación de Tendencia de Precio de Cierre del Bitcoin

Usando Redes Neuronales

Iver Johan Hincapie Betancur

Eliana Carolina Herrán Logreira

Monografía presentada para optar al título de Especialista en Analítica y Ciencia de Datos

Asesor

Dr. Sergio Sanes Negrete

Universidad de Antioquia

Facultad de Ingeniería

Especialización en Analítica y Ciencia de Datos

Medellín, Antioquía, Colombia

2022

Cita	(Herran Logreira & Hincapie Betancur, 2022)
Referencia	Herran Logreira, E., & Hincapie Betancur, I. J. (2022). <i>Estimación de tendencias de Precio de cierre del Bitcoin aplicando redes neuronales</i> Trabajo de grado especialización. Universidad de Antioquia, Medellín, Colombia.
Estilo APA 7 (2020)	



Centro de Documentación de Ingeniería (CENDOI)

Repositorio Institucional: <http://bibliotecadigital.udea.edu.co>

Universidad de Antioquia - www.udea.edu.co

Rector: John Jairo Arboleda Céspedes

Decano/director: Jesús Francisco Vargas Bonilla

Jefe departamento: Diego José Luis Botia Valderrama

El contenido de esta obra corresponde al derecho de expresión de los autores y no compromete el pensamiento institucional de la Universidad de Antioquia ni desata su responsabilidad frente a terceros. Los autores asumen la responsabilidad por los derechos de autor y conexos.

TABLA DE CONTENIDO

1. RESUMEN EJECUTIVO	1
2. DESCRIPCIÓN DEL PROBLEMA	2
2.1 PROBLEMA DE NEGOCIO	2
2.2 APROXIMACIÓN DESDE LA ANALÍTICA DE DATOS	3
2.3 ORIGEN DE LOS DATOS	5
2.4 MÉTRICAS DE DESEMPEÑO	6
3. DATOS	7
3.1 DATOS ORIGINALES	7
3.2 DATASETS	8
4. PROCESO DE ANALÍTICA	9
4.1 PIPELINE PRINCIPAL	9
4.2 PREPROCESAMIENTO	11
4.3 MODELOS	16
5. METODOLOGÍA	24
5.1 BASELINE	24
5.2 VALIDACIÓN	30
5.3 EXPERIMENTOS	31
5.4 HERRAMIENTAS	37
6. RESULTADOS	38
6.1 MÉTRICAS	38
6.2 EVALUACIÓN CUALITATIVA	39
6.3 CONSIDERACIONES DE PRODUCCIÓN	41
7. CONCLUSIONES	43
8. REFERENCIAS BIBLIOGRÁFICAS	45
9. ANEXOS	47

ÍNDICE DE FIGURAS

Figura 1: Diagrama de alto nivel de los servicios implementados en la construcción de la solución.	10
Figura 2: Diagrama de alto nivel del procesamiento de datos para el modelo analítico.	11
Figura 3: Gráfica de los componentes de la serie de tiempo para el bitcoin.	12
Figura 4: Gráfica de los componentes de la serie de tiempo para el bitcoin en escala logarítmica. 14	
Figura 5: Gráfica de los componentes de la serie de tiempo para el bitcoin derivada de la escala logarítmica.	15
Figura 6: Gráfica de Autocorrelación para el bitcoin	16
Figura 7: Gráfica de predicción del entrenamiento modelo ARIMA (1,0,1)	26
Figura 8: Gráfica de predicción del test modelo ARIMA (1,0,1)	26
Figura 9: Gráfica de tendencia de las criptomonedas	28
Figura 10: Gráfica de predicción del entrenamiento modelo ARIMAX (1,0,2)	29
Figura 11: Gráfica de predicción del testing modelo ARIMAX(1,0,2)	29
Figura 12: Comportamiento predicción y error red LSTM	32
Figura 13: Comportamiento predicción y error red RNN	34
Figura 14: Comportamiento predicción y error GRU	35
Figura 15: Comportamiento predicción y error Red CONV	37
Figura 16: comportamiento de Pérdidas Train y Validación en 40 épocas para la red	
Figura 17: Comportamiento predicción y Test LSTM	40
Figura 18: Estimación de los 15 valores siguientes.	41
Figura 19: Arquitectura despliegue en producción	42

1. RESUMEN EJECUTIVO

La presente monografía tiene como objetivo el estudio de series de tiempo en un problema de economía, que consiste en la estimación de los valores de cierre futuros del bitcoin.

El proyecto se divide en dos etapas, la primera es la construcción de diferentes modelos base como ARIMA y ARIMAX haciendo uso de variables exógenas. La segunda se basa en el uso de modelos más modernos, donde se explora el uso de técnicas de Deep Learning y las aplicaciones de diferentes arquitecturas de redes neuronales.

Así mismo, dado que el desarrollo contempla diversos tipos de modelos, la manera de elegir el más adecuado es por medio del uso de métricas de desempeño como el error cuadrático medio, el error relativo medio absoluto y el error absoluto medio, con lo cual se escoge el modelo con menor valor en dichas métricas. De este modo, se decide realizar una comparación entre los modelos clásicos y los modernos, tomando como base el mejor modelo clásico: ARMAX que es el modelo que obtiene el mejor desempeño. Por otro lado para el caso de los modelos de redes neuronales el que mejor desempeño presenta es la red LSTM, ya que presenta un balance entre el error y overfitting con relación a los demás modelos.

Finalmente se concluye que las redes neuronales son mejores que los modelos clásicos, ya que a pesar que estos últimos usan variables exógenas al aumentar la cantidad de parámetros de entrada al modelo, las redes neuronales dan mejor resultados al solo utilizar el valor de cierre del bitcoin como parámetro de entrada obteniendo un valor de error menor. Sin embargo se evidencia que la estimación de valores futuros no es una tarea sencilla, ya que las serie de tiempo están permeadas por factores externos que insertan un alto grado de incertidumbre, debido a que por cada n valor adicional que se quiere predecir, en cada iteración se agregan errores que imposibilitan una estimación a largo plazo, es por esto que se considera como mejora estar entrenando el modelo cada cierto tiempo.

2. DESCRIPCIÓN DEL PROBLEMA

Antes del año 2009 el comercio electrónico dependía de la intermediación de las entidades financieras para crear una relación de confianza entre el vendedor y el comprador para completar la transacción realizada desde internet, generando gastos adicionales a la venta por este canal. Es por esto que a partir de ese año, en el mercado financiero entra la primera moneda electrónica creada para eliminar el intermediario y garantizar la seguridad de la transacción por medio de autenticación con pruebas criptográficas, a esta moneda se le conoce como criptomoneda. Este trabajo se enfocará en la criptomoneda denominada como bitcoin [1]. Este tipo de dinero transformaría por completo el paradigma de transacciones que se tenía hasta ese entonces de forma electrónica, al posicionarse como una nueva manera de abrir mercado sin la necesidad de tener la validación de la banca mundial, ni organizaciones gubernamentales [2]. Con esta divisa se avecinaba una nueva etapa de descentralización de la moneda. Y aunque años anteriores en los Estados Unidos ya funcionaba esta idea de criptografía, con la denominada eCash y posterior digiCash [3], en la década de los 90's, no fue hasta la crisis económica que vivió ese país en el año 2008 [4] que esta idea cobró una mayor fuerza, generando un rápido crecimiento del bitcoin que impulsó la creación de más criptomonedas en años posteriores y con esto los analistas financieros empezaron a incluir en sus informes el comportamiento de este activo y su interacción con el mercado existente.

Los analistas basan sus informes en la observación de los datos generados por las criptomonedas, cuyo patrón tiene la característica de eventos que suceden en determinado momento de tiempo de manera secuencial y por lo tanto son considerados una serie de tiempo [5].

Finalmente, dado el interés alrededor de las variaciones que se presentan en los mercados de criptomonedas por su relación con el mundo moderno y por sus altos indicadores de ganancias y fluctuaciones, se tiene la motivación de desarrollar y seleccionar entre diferentes modelos el que tenga el menor error y de esta forma diseñar una herramienta útil en la toma de decisiones de inversión.

2.1 PROBLEMA DE NEGOCIO

Uno de los puntos que ocupa la economía financiera es el de la predicción del comportamiento a futuro con bajo margen de error de las diferentes monedas en el mercado bursátil, y por lo tanto se busca tener herramientas para el análisis de series de tiempo, que reproduzcan las tendencias de las variables económicas y de esta manera tener indicios de hacia dónde ir en términos de inversión. Pero, es de conocimiento general que el comportamiento de estas variables económicas tiene un grado de incertidumbre elevado debido a que en su valor final confluyen un sin número de elementos exógenos de carácter social, político y/o natural[6]. Dado lo anterior, lo que se pretende en este proyecto de monografía es brindar por medio de Deep Learning herramientas y técnicas de análisis, que reduzca la incertidumbre en la predicción de la tendencia, para toda aquella persona interesada en las inversiones de bitcoin a corto y largo plazo, con un riesgo bajo. Es así cómo se desarrollan, modelos de series temporales utilizados en econometría: ARIMA y ARIMAX [7] y modelos basados en redes autorregresivas (RNN) [8], Redes neuronales convolucionales (CNN) [9], Long Short-Term Memory (LSTM) [10] y Gated Recurrent Units (GRU) [11], para estimar el comportamiento del valor del bitcoin.

2.2 APROXIMACIÓN DESDE LA ANALÍTICA DE DATOS

En este trabajo se evalúan diferentes arquitecturas de redes neuronales para elegir la que presente mejores métricas, usando como referencia el análisis de la serie de tiempo y las métricas de referencia de un modelo clásico de econometría.

Debido a que se estudian fenómenos económicos que por su naturaleza tienen tendencias crecientes, decrecientes o exponenciales [12] se hace un análisis exploratorio de los datos para determinar los componentes principales que permiten modelar las tendencias en la variación de los precios de cierre de la criptomoneda, que describe el comportamiento de los datos según tres componentes principales: Estacionariedad, Tendencia y Aleatoriedad [13].

Estos componentes se definen de la siguiente forma:

- **Estacionariedad:** Es la variación de la serie de tiempo entre periodos, un ejemplo de estacionariedad en ventas puede ser la temporada navideña, en esta temporada la cantidad de artículos vendidos aumenta.

- **Tendencia:** Es el cambio a largo plazo de los valores medios de los datos
- **Aleatoriedad:** Es el resultado de hechos fortuitos que afectan de forma aislada los valores de la serie sin tener un patrón de comportamiento.

Para determinar si una serie de tiempo es estacionaria o no, se aplica la prueba de Dickey-Fuller [14], donde el contraste de hipótesis se hace calculando un estadístico de pruebas y comparándolo con los valores críticos de la tabla de Dickey Fuller [15], si el estadístico de prueba es menor que el valor crítico, entonces rechazamos la hipótesis Nula y decimos que los datos provienen de un fenómeno estacionario. Por lo tanto, la hipótesis nula es la serie estacionaria y la hipótesis aceptada es que la serie es no estacionaria.

Otro concepto utilizado para el análisis es el de autocorrelación. Este determina la similitud que tiene la serie consigo misma, con el objetivo de hallar algún patrón de repetición entre periodos anteriores y el valor actual de esta, de tal manera que se puede determinar la frecuencia en la que se presenta dicho patrón, que puede ser sea dado en días, meses o años. La secuencia de varios periodos atrás se conoce como *Lag* y es el que permite tener como base si la serie presenta alguna estacionalidad y permite estimar la cantidad de datos necesarios que se deben considerar para poder hallar los valores futuros de la serie. [16]

Una vez determinadas las características de la serie, se aplican algunos métodos para identificar la estacionalidad de la serie y luego se selecciona el modelo econométrico base. En esta monografía usamos el modelo ARIMA como uno de los modelos clásicos, el cuál consiste en la unión entre dos modelos diferentes, el modelo autorregresivo (AR) descrito en por la ecuación:

Modelo AR

$$x_t = c + \varphi x_{t-1} + \epsilon_t$$

Ecuación 1: *Ecuación modelo Autorregresivo*

donde x_t es el valor de interés, del período actual, de nuestra serie de tiempo, c es la constante, φ es el coeficiente que debemos estimar, ϵ_t es el residuo en el período actual, y x_{t-1} es el valor de la serie en el período anterior. La ecuación nos permite estimar el valor actual a partir de los

valores pasados más un error. Por su parte, el modelo de medias móviles MA [17], calcula el valor actual a partir de los residuos pasados como se indica en la ecuación

Modelo MA

$$x_t = c + \theta\epsilon_{t-1} + \epsilon_t$$

Ecuación 2: *Ecuación modelo de medias móviles*

donde x_t es el valor de interés, del período actual, de nuestra serie de tiempo, c es la constante, θ es el coeficiente que debemos estimar, ϵ_t es el residuo en el período actual y ϵ_{t-1} es el valor del residuo en el período anterior[18].

Por otro lado, al modelo ARIMA se le agrega un componente de integración el cual es aplicado a la serie original para posteriormente ajustar el modelo ARMA al resultado de la integración. De tal forma que ARIMA(p,d,q) donde p describe el grado de la función de autorregresión, d el grado de la integración y el grado de la ecuación de media móvil, denotado por q .

La familia de modelos ARIMA, tiene una última versión y es el modelo ARIMAX que consta de la agregación de variables exógenas al proceso de entrenamiento del modelo, dichas variables son variables externas a la variable de interés, pero que pueden guardar una relación con ella.

Por último, con el resultado del análisis previo y la validación del modelo anterior se desarrollan algunas soluciones de Redes neuronales con los que se busca estimar el valor de cierre de la criptomoneda del día siguiente y evaluar el modelo de Deep Learning más adecuado que permita saber si es beneficioso o no realizar una compra o venta de la divisa.

2.3 ORIGEN DE LOS DATOS

El dataset es extraído de la página web CoinMarketCap¹ que contiene los datos históricos y actuales de las tendencias de las criptomonedas del mercado. Con ayuda de la extensión proporcionada por Google Chrome llamada Web Scraper se realiza una conexión a este sitio, para

¹ <https://coinmarketcap.com/>

acceder a la información y obtener los datos históricos hasta el 06 de junio del año 2021. Estos se guardarán y están disponibles en un repositorio creado para el proyecto².

2.4 MÉTRICAS DE DESEMPEÑO

Las métricas de desempeño se utilizan para establecer la confianza de la predicción y las que se eligieron para analizar el resultado de la tendencia del bitcoin son aquellas que miden la distancia entre el valor predicho y el valor real de la divisa. En este orden de ideas, las métricas usadas en esta monografía son el error medio cuadrático(MSE), el error absoluto medio (MAE) y error relativo medio absoluto (MRAE), y los componentes básicos para su cálculo son:

n: Número de observaciones del fenómeno

i: Indica una observación puntual

y_i : Valor predicho

f_i : Valor real

A continuación, se describe el cálculo y la interpretación de las métricas

- **Error cuadrático medio (MSE)** [19]

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

Ecuación 3: *Ecuación error cuadrático medio*

Este error mide la aproximación de los puntos de la curva de predicción y la curva de prueba y arregla el problema de cancelación entre errores negativos y positivos.

- **Error absoluto medio (MAE)** [20]

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - f_i|$$

Ecuación 4: *Ecuación error absoluto medio*

² <https://github.com/Eherran/monografia.git>

Este error es el promedio del valor absoluto de la diferencia entre los datos observados y los datos predichos, al ser un valor absoluto todas las diferencias individuales se ponderan por igual en el promedio

- **Error relativo medio absoluto (MREA) [21]**

$$MREA = \frac{1}{n} \sum_{i=1}^n \left| \frac{y_i - f_i}{f_i} \right|$$

Ecuación 5: *Ecuación error absoluto medio*

ϵ : es cualquier valor entero positivo muy pequeño, para evitar la división por cero.

es una métrica comúnmente utilizada para problemas de regresión, que busca ser sensible a los errores relativos, teniendo en cuenta las pequeñas desviaciones, contrario a MAE que toma los valores mayores

3. DATOS

3.1 DATOS ORIGINALES

En el sitio CoinMarketCap, las 5 criptomonedas más transadas para el mes de julio de 2021 fueron bitcoin, ethereum, binance Coin, tether y cardano. Posicionando al bitcoin como la moneda principal con relación a las demás y por tal razón se toma con un mayor interés como objeto de estudio. Las monedas diferentes al bitcoin se tomarán como información exógena al valor de cierre del bitcoin, con el fin de tener más datos para alimentar el modelo ARIMAX. Para esto, se usa el registro histórico de estas monedas desde sus inicios hasta junio de 2021.

La TABLA I describe los campos de los datasets y la TABLA II ilustra las propiedades de estos, tales como tamaño, datos históricos y la cantidad de registros con los que se dispone para hacer el análisis de tendencia.

TABLA I
DATASET CRIPTOMONEDAS

Atributos	Descripción
Date	Día de observación
Open	Precio de apertura en el día de la observación en dólares
High	El precio más alto obtenido en el día de la observación en dólares
Low	El precio más bajo obtenido en el día de la observación en dólares
Close	Precio de cierre en el día de la observación en dólares
Volume	Volumen de transacciones realizadas en el día de la observación
Market Cap	Valorización en dólares en el mercado, de la moneda el día la observación.

TABLA II
CARACTERÍSTICAS DE ARCHIVOS HISTÓRICOS PARA ENTRENAMIENTO DEL MODELO

Moneda	Fecha Inicial	Cantidad Registros	Tamaño (KB)
Bitcoin	2013-04-29	3254	382
Etherium	2015-08-08	2423	280
BinanceCoin	2017-07-26	1705	186
Tether	2015-02-26	2586	265
Cardano	2017-10-02	1637	175

3.2 DATASETS

En el desarrollo se implementaron varios tipos de modelos, entre ellos ARIMA, Redes neuronales recurrentes y convolucionales para la predicción del precio de cierre del bitcoin, para los cuales se usa el mismo dataset.

- Arimax: Este modelo requiere que se comparen los precios de cierre del bitcoin con las variables exógenas, que en este caso corresponden a los precios de cierre de Ethereum, BinanceCoin y Cardano, entonces es necesario realizar un ajuste en cada dataset para tener la misma fecha de inicio de transacciones en las criptomonedas usadas y se establece 2017-10-02 por ser la más reciente entre todos los dataset. En este caso se partieron los datos en entrenamiento ni testing en una proporción de 80/20.
- Arima: En este caso se tomaron los datos en entrenamiento y testing en una proporción de 80/20.
- Red Neuronal: Para los diferentes modelos de redes neuronales como: LSTM, Conv1D, RNN y GRU se aplicó la misma relación de conjunto de datos, usando la siguiente distribución:
 1. Un dataset con un 70% de los datos como entrenamiento,
 2. 10% de datos de validación
 3. 20% de los datos como test.

Cabe destacar que al trabajar con series de tiempo, la separación se hace de manera secuencial y no aleatoria y en ambos casos se usan los datos transformados en escala logarítmica base 10, paso siguiente se debe revertir la operación para obtener el valor en las unidades iniciales (USD).

4. PROCESO DE ANALÍTICA

En esta sección se describe de manera detallada los diferentes procesos que se han realizado en el desarrollo del problema, desde el punto de vista de diseño y análisis de la solución, partiendo desde adquisición de los datos, el análisis, el procesamiento de la información y la construcción de la solución al problema de series de tiempo aplicado al bitcoin por medio de modelos de redes neuronales.

4.1 PIPELINE PRINCIPAL

Como se ha indicado anteriormente, los datos históricos se encuentran alojados en el sitio web CoinMarketCap, donde son descargados utilizando el plugin de google chrome llamado *web scraper*. Los datos son almacenados en github de forma manual junto con el código del proyecto. La Figura 1, indica el flujo de adquisición de los datos y el código del proyecto desde el repositorio github por medio de un notebook de Google Colaboratory³ que es externo al proyecto y que se conecta por medio de una autenticación entre github y el google drive de la cuenta personal.

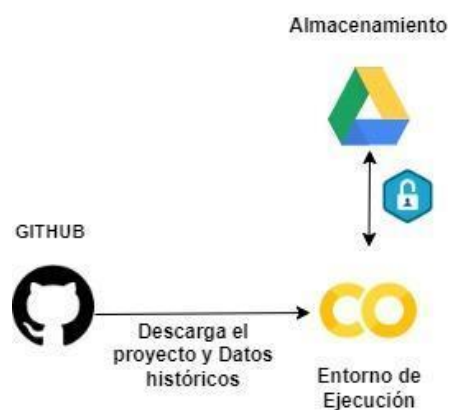


Figura 1: Diagrama de alto nivel de los servicios implementados en la construcción de la solución.

Descripción del flujo paso a paso:

Una vez se han descargado los notebooks y los datasets, se inicia la etapa de ejecución de los archivos de trabajo correspondientes al análisis y creación de los modelos, cuyo flujo está descrito en la Figura 2. Cada paso se ejecuta al interior de un notebook creado en la herramienta de Google Colaboratory.

³ <https://colab.research.google.com/notebooks/welcome.ipynb?hl=es-419>

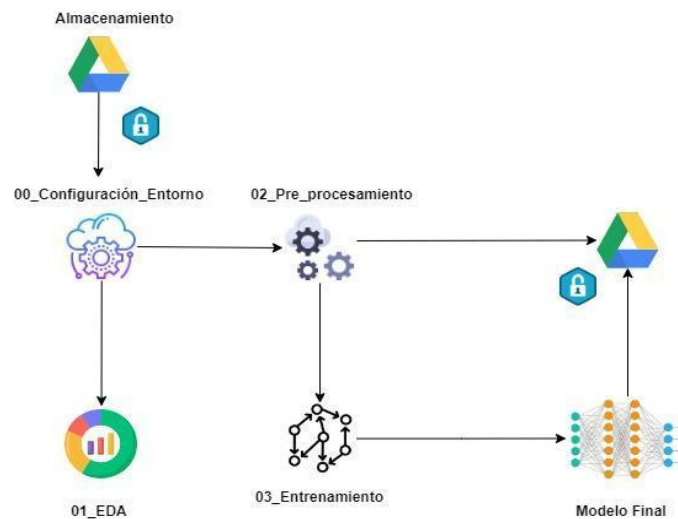


Figura 2: Diagrama de alto nivel del procesamiento de datos para el modelo analítico.

Configuración de entorno, este notebook se conecta a google drive, importa las librerías necesarias y carga los datasets de las criptomonedas en dataframes de pandas con el fin de disponibilizar el entorno de trabajo para ser usado por los demás notebooks.

1. **Análisis exploratorio de los datos (EDA)**, la primera acción que se realiza en este paso es la ejecución del notebook **Configuración de entorno** descrito anteriormente. Paso seguido, se utilizan las librerías y data frames definidos en el notebook anterior y es sobre estos últimos que se analiza su comportamiento y se aplica la teoría de series de tiempo.
2. **Preprocesamiento** de los datos, tomando nuevamente los data frames que se cargaron en el notebook de Configuración de entorno, se les aplica una serie de transformaciones generando un nuevo dataset que se guarda en el google drive de la cuenta respectiva.
3. **Entrenamiento del modelo** Una vez los datos son pre-procesados y almacenados en el google drive, el archivo que ha sido generado en el paso 2 se carga en un nuevo data frame para hacer la validación de los datos y separarlos en los diferentes subconjuntos de entrenamiento, pruebas y validación, luego se disponen como entrada a los diferentes modelos de redes neuronales con el fin de entrenar el modelo y evaluarlo. Finalmente se guarda la mejor red neuronal en google drive.

4. **Validación del Modelo**, desde el google drive se carga el modelo seleccionado, que previamente es entrenado, de esta manera se realizan algunas predicciones de prueba, generando nuevos valores a la serie de tiempo.

4.2 PREPROCESAMIENTO

Para el preprocesamiento de los datos se siguieron las siguientes estrategias:

1. Se homogeneizan los datasets para que estén en el mismo rango de fechas para hacer las comparaciones respectivas entre las diferentes monedas.
2. Se grafican los componentes principales de la serie del bitcoin, como se muestra en la Figura 4, donde se puede ver que los datos tienen una tendencia al alza de 2020 a 2021, sin embargo, se ve que esa tendencia empieza a bajar en el 2021. Adicionalmente, los datos no muestran estacionalidad, como se puede ver el componente seasonal(estacionalidad) de la Figura 3 cuya gráfica no define variaciones.

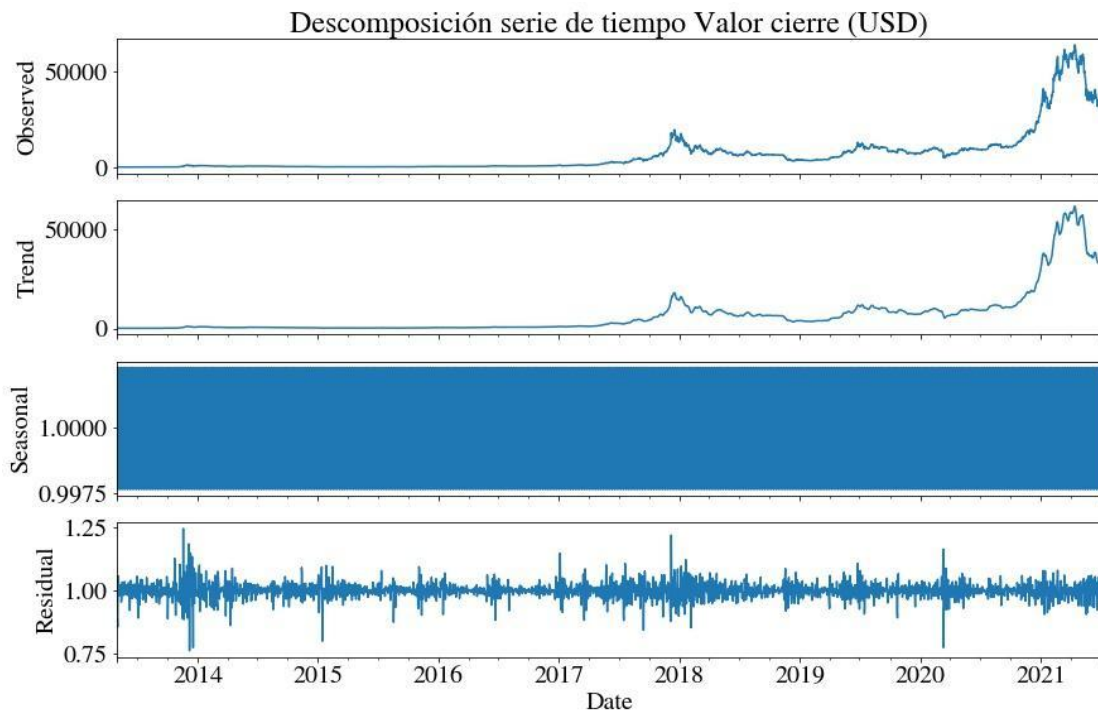


Figura 3: Gráfica de los componentes de la serie de tiempo para el bitcoin.

3. Se calcula la estacionalidad del valor de cierre con la prueba Dickey Fuller. El resultado obtenido para el bitcoin se muestra en la TABLA III:

TABLA III
RESULTADO DE LA PRUEBA ADF PARA UN NIVEL DE SIGNIFICANCIA DEL 5%

Valor Crítico	P-Value	Lags	No de observaciones	T Valor
-0.895	0.790	29	2961	'5%': -2.862

El resultado de la TABLA III lleva a concluir que se acepta la hipótesis nula, debido a que el valor estadístico, -0.895, toma un valor superior al referenciado en el nivel de significancia, que corresponde a -2.862; adicionalmente, el p-value es un valor superior al nivel de significancia. Esta prueba confirma lo observado en la figura 4, los datos no tienen un comportamiento estacionario.

4. Se calculan los logaritmos en base 10 para cada una de tendencias de las monedas, con el fin de conocer las pendientes de crecimiento y decrecimiento para cada una de ellas. El resultado obtenido para el bitcoin puede ser observado en la Figura 4. Los datos siguen sin marcar la estacionalidad.

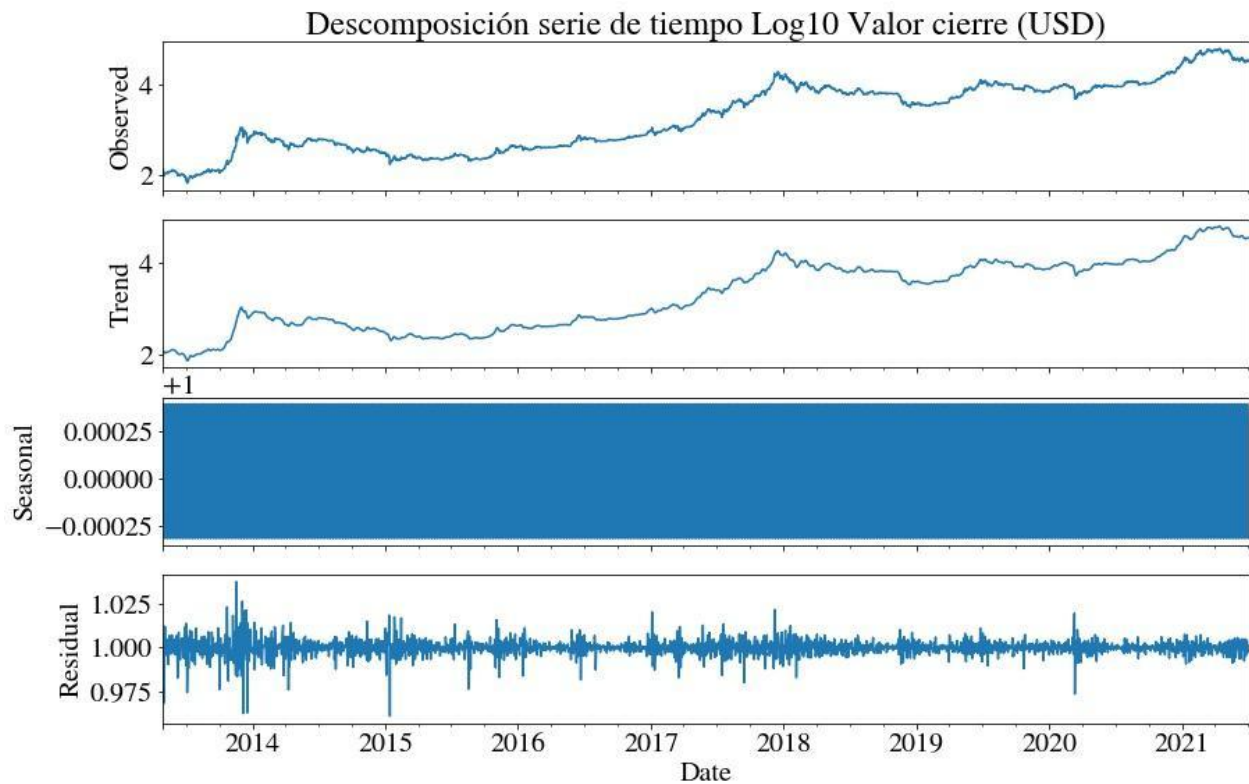


Figura 4: *Gráfica de los componentes de la serie de tiempo para el bitcoin en escala logarítmica.*

A partir de estas transformaciones evidenciamos una mejora en la tendencia y un suavizado en la serie de tiempo, donde se puede ver una posible estacionalidad, por lo tanto, se realiza la prueba de Dickey-Fuller para comprobarlo. El resultado obtenido para el bitcoin se muestra en la TABLA IV.

TABLA IV
RESULTADO DE LA PRUEBA ADF PARA UN NIVEL DE SIGNIFICANCIA DEL 5% DE LOS DATOS EN ESCALA LOGARÍTMICA.

Valor Crítico	P-Value	Lags	No de observaciones	T Valor
-0.777	0.826	23	2602	5%: -2.863

Al aplicar nuevamente la prueba, se nota que el valor estadístico aún es mayor que el esperado y por tal motivo se toma nuevamente la hipótesis nula: Que la serie es no estacionaria.

5. Luego se calcula el diferencial del logaritmo para identificar puntos de cambios en la función, con el fin de un comportamiento periódico. El resultado para el bitcoin se muestra en la TABLA V:

TABLA V
RESULTADO DE LA PRUEBA ADF PARA UN NIVEL DE SIGNIFICANCIA DEL 5% DE LOS DATOS DE LA DERIVADA A ESCALA LOGARÍTMICA.

Valor Crítico	P-Value	Lags	No de observaciones	T Valor
-1.928	0.319	22	2602	5%: -2.863

De esta manera al realizar la derivada se nota que los resultados para el valor estadístico y el valor p-value son los esperados para tomar la hipótesis alternativa: La serie es estacionaria como se muestra en la figura 5 y lo corroboran los resultados de la prueba ADF.

- Valor estadístico $-1.928 < -2.863$
- Valor-p $0.3191 < 0.05$

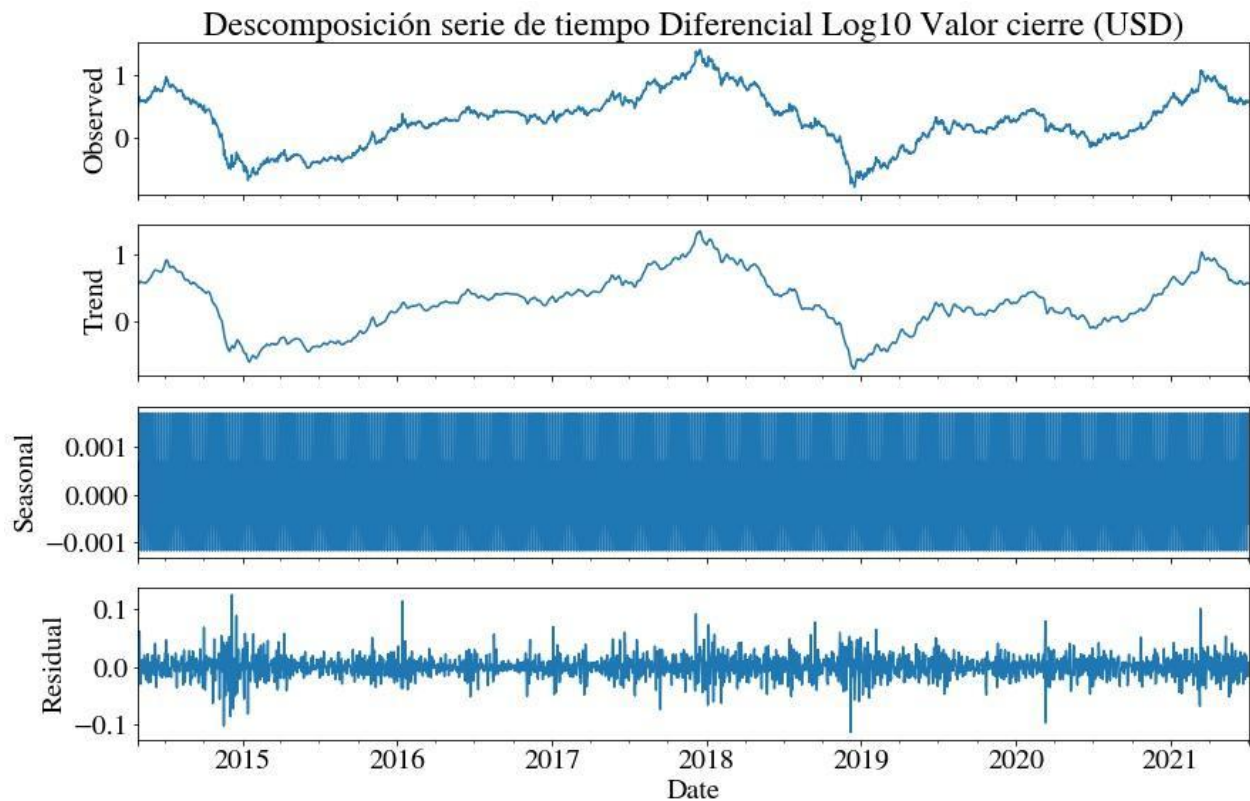


Figura 5: Gráfica de los componentes de la serie de tiempo para el bitcoin derivada de la escala logarítmica.

6. Por último se realiza la autocorrelación de un mes para ver el comportamiento de los datos e identificar el valor de días necesarios en los que se puede presentar un proceso cíclico de la serie, que permita hallar una estacionalidad, este valor se conoce como *look back* y ayuda a conocer el valor aproximado de datos previos que se requieren para hacer la predicción, que están dados por la frecuencia de la serie, para este caso la frecuencia está dada en días. Luego de hallar la autocorrelación se determina un valor de 7 con el cual se empieza a identificar un ciclo de bajada nuevamente. La Figura 6, muestra la autocorrelación de los datos en un periodo de retraso de 30 días.

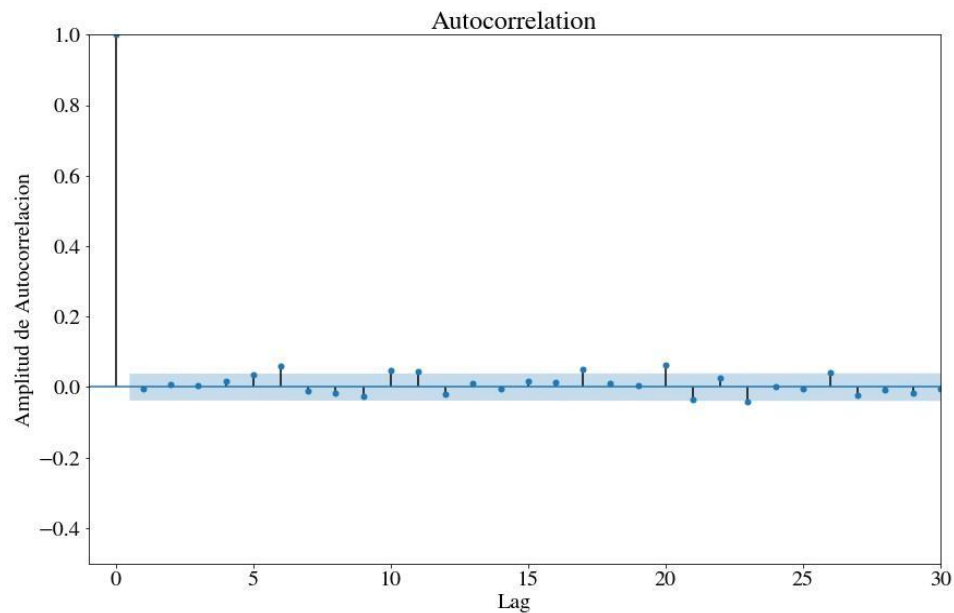


Figura 6: *Gráfica de Autocorrelación para el bitcoin*

4.3 MODELOS

Las criptomonedas son series de valores que obedecen a procesos estocásticos [22], dado a que el valor varía en función del tiempo de manera aleatoria. Para generalizar suponga que se quiere conocer el valor de la criptomoneda en un periodo de tiempo determinado, para este ejemplo, se establece el periodo de tiempo de una hora, la variable aleatoria se define como “valor de la criptomoneda en una hora”, si ese tiempo se amplía a dos horas, la distribución de probabilidad de esa nueva variable va a ser distinta a la anterior, esto significa que para cada momento en el tiempo existirá una variable diferente, por lo tanto el comportamiento del grupo de variables depende del tiempo.

Para el ejemplo anterior se define el proceso estocástico $X(t)$, como el valor que puede tener la criptomoneda en un intervalo de tiempo $(0, t)$ por lo tanto se puede definir como $\{X(t) : t \in T\}$, donde T representa al tiempo, X tiene un componente de aleatoriedad que hace difícil predecir el valor que tomará el activo en cada momento del tiempo.

Adicional a lo anterior los procesos estocásticos se dividen en dos:

- Procesos estocásticos estacionarios: tienen características que lo hacen eventualmente predecibles
- Procesos estocásticos no estacionarios: Es impredecible su comportamiento.

El componente aleatorio de la variable X , puede ser cualquier evento que se presente en el instante de tiempo observado, para el caso de esta monografía se toma como variables exógenas los precios de cierre de las criptomonedas más transadas en el momento de la elaboración de este documento.

Desde la econometría se conocen algunos modelos estadísticos que se han usado durante muchos años para hallar el comportamiento de tendencias de series de tiempo de variables económicas estocásticas, como lo son las criptomonedas. Para el desarrollo del proyecto se eligen dos para comparar resultados entre sí y seleccionar el modelo de referencia que mejor se comporte con la serie de datos usados.

Primeramente se construye un modelo ARMA, sin variables exógenas, esto con el fin de analizar las diferencias entre ambos modelos y definir cual permite definir una tendencia más acertada durante el proceso de análisis del valor de cierre del bitcoin, la configuración del modelo ARMA se indica en la TABLA VI.

TABLA VII
CONFIGURACIÓN MODELO ARMA

ARMA Model Results							
Dep. Variable:	bitcoinClose	No. Observations:	1099				
Model:	ARMA(1, 1)	Log Likelihood	2836.570				
Method:	css-mle	S.D. of innovations	0.018				
Date:	Sun, 05 Jun 2022	AIC	-5665.140				
Time:	04:45:36	BIC	-5645.131				
Sample:	10-02-2017	HQIC	-5657.570				
	- 10-04-2020						
	coef	std err	z	P> z 	[0.025	0.975]	
const	3.8720	0.076	50.991	0.000	3.723	4.021	
ar.L1.bitcoinClose	0.9937	0.003	302.427	0.000	0.987	1.000	
ma.L1.bitcoinClose	-0.0222	0.029	-0.773	0.440	-0.079	0.034	
	Roots						
	Real	Imaginary	Modulus	Frequency			
AR.1	1.0063	+0.0000j	1.0063	0.0000			
MA.1	45.0562	+0.0000j	45.0562	0.0000			

Por otro lado, se construye un modelo ARMAX, con la intención de determinar el comportamiento de la serie de tiempo con influencia de los precios de cierre de Binance, Ethereum y Cardano. Esta aproximación de predicción se hace con una vista clásica de la situación y así poder observar qué tan bueno sería el pronóstico de la tendencia. La TABLA VII, a continuación, muestra la configuración del modelo ARMAX desarrollado en el primer experimento.

TABLA VII
CONFIGURACIÓN MODELO ARMAX

ARMA Model Results						
Dep. Variable:	bitcoinClose	No. Observations:	1099			
Model:	ARMA(1, 2)	Log Likelihood	3388.466			
Method:	css-mle	S.D. of innovations	0.011			
Date:	Sun, 05 Jun 2022	AIC	-6760.931			
Time:	04:45:39	BIC	-6720.914			
Sample:	10-02-2017	HQIC	-6745.792			
	- 10-04-2020					
	coef	std err	z	P> z 	[0.025	0.975]
const	2.6367	0.080	33.001	0.000	2.480	2.793
binanceClose	0.1751	0.017	10.604	0.000	0.143	0.207
cardanoClose	0.0469	0.014	3.405	0.001	0.020	0.074
ethereumClose	0.4526	0.022	20.140	0.000	0.409	0.497
ar.L1.bitcoinClose	0.9937	0.003	322.385	0.000	0.988	1.000
ma.L1.bitcoinClose	0.1341	0.031	4.370	0.000	0.074	0.194
ma.L2.bitcoinClose	-0.0176	0.030	-0.592	0.554	-0.076	0.041
Roots						
	Real	Imaginary	Modulus	Frequency		
AR.1	1.0063	+0.0000j	1.0063	0.0000		
MA.1	-4.6320	+0.0000j	4.6320	0.5000		
MA.2	12.2319	+0.0000j	12.2319	0.0000		

De esta manera, se observa que para cada caso descrito anteriormente, se consigue la mejor representación en la tendencia con un modelo ARMAX, ya que la parte integral d toma el valor de 0.

Por otro lado las redes neuronales, poseen algunas características que las hacen atractivas a la hora de proponer soluciones para algunos de los problemas más conocidos en el pronóstico (o forecasting) del comportamiento de las series de tiempo, es por esta razón que, surge la motivación para explorar los siguientes modelos de redes neuronales.

Redes Neuronales convolucionales (CNN)

Aunque las redes neuronales convolucionales han sido normalmente utilizadas para la clasificación de imágenes por medio de la aplicación de filtros que permiten la extracción automática de características a partir de sus entradas, las CNN también se aplican a modelos de series de tiempo debido a la invariancia de distorsión, es decir que logra extraer las características de manera independiente de cómo llegan los datos. De esta manera una serie de tiempo se puede

tratar como una *imagen de una dimensión* para extraer las características del comportamiento de esta secuencia de entrada[23].

Este modelo se compone de 3 capas convolucionales, entre ellas capas dropout para controlar el overfitting, la capa flatten para aplanar la entrada de los datos a la capa Densa que cierra la arquitectura. La TABLA VIII, muestra la configuración de la red.

TABLA VIII
RESUMEN DE CONFIGURACIÓN DE LA RED CNN

Layer (type)	Output Shape	Param #
conv1d_3 (Conv1D)	(None, 5, 64)	256
dropout_8 (Dropout)	(None, 5, 64)	0
conv1d_4 (Conv1D)	(None, 3, 64)	12352
dropout_9 (Dropout)	(None, 3, 64)	0
conv1d_5 (Conv1D)	(None, 1, 64)	12352
dropout_10 (Dropout)	(None, 1, 64)	0
flatten_1 (Flatten)	(None, 64)	0
dense_7 (Dense)	(None, 1)	65
=====		
Total params: 25,025		
Trainable params: 25,025		
Non-trainable params: 0		

El modelo se compila usando una función de activación Sigmoides, la función de pérdida es el MSE y se entrena con 40 épocas.

Redes Neuronales Recurrentes

Adicionalmente, se construye un modelo basado en redes neuronales recurrentes, estas son redes neuronales que tienen dentro de sus bondades la capacidad de *recordar* características de las entradas de las neuronas y de las capas anteriores, permitiendo que sean muy útiles en el momento de trabajar con datasets que sean secuenciales, como es el caso de las series de tiempo [24].

Arquitectura RNN: Esta red se compone de dos capas, una SimpleRNN y una Densa y la función de costo, está dada por MSE y MAE, tanto para los datos de entrenamiento como para validación. La TABLA IX muestra la configuración de la red.

TABLA IX
RESUMEN DE LA ARQUITECTURA RNN CONSTRUIDA.

```

Model: "sequential_1"

```

Layer (type)	Output Shape	Param #
simple_rnn (SimpleRNN)	(None, None, 20)	440
simple_rnn_1 (SimpleRNN)	(None, 20)	820
dense_1 (Dense)	(None, 10)	210

```

=====
Total params: 1,470
Trainable params: 1,470
Non-trainable params: 0

```

Red Neuronal Long Short-Term Memory (LSTM)

La siguiente arquitectura que se usa para buscar mejorar el error es una Long Short-Term Memory (LSTM), por ser una configuración diseñada para trabajar con valores que cambian en periodos de tiempo corto, como es el caso de las criptomonedas que tiene una ventana de media móvil entre 5 y 7 días, correspondiente al valor de look back.

Dicho valor, como se explica anteriormente, corresponde al análisis exploratorio donde se identifica que es la cantidad de días en los cuales se evidencia un comportamiento auto correlacionado.

Por otro lado, se usan 100 neuronas con un kernel Glorot uniform[25], con una función de activación tangente hiperbólica (tanh)[26], una función de pérdida de error de *MAE* y un

inicializador tipo ADAMAX [27]. El entrenamiento se hace con 40 épocas, la TABLA X ilustra la configuración de la red LSTM.

TABLA X
RESUMEN DE LA ARQUITECTURA LSTM CONSTRUIDA

Layer (type)	Output Shape	Param #
lstm_17 (LSTM)	(None, 1, 50)	11600
dropout_16 (Dropout)	(None, 1, 50)	0
lstm_18 (LSTM)	(None, 1, 50)	20200
dropout_17 (Dropout)	(None, 1, 50)	0
dense_9 (Dense)	(None, 1, 1)	51
=====		
Total params: 31,851		
Trainable params: 31,851		
Non-trainable params: 0		

Redes Gated Recurrent Unit (GRU)

Las redes Gated Recurrent Unit (GRU) son un tipo de redes recurrentes creadas en 2014, la capa GRU contiene dos puertas que controlan el flujo de la información para entrar y salir de la capa. Estas redes tienen menos parámetros que las LSTM por lo tanto son más simples para configurar, generalmente pueden tener un mejor rendimiento y trabajan bien con conjuntos pequeños de datos [28].

Esta arquitectura consta de 4 capas GRU, capas dropout entre ellas para disminuir el overfitting y una función de activación tangente hiperbólica (\tanh), este modelo se compila con un optimizador tipo Sigmoid (SGD) y la pérdida se mide con el MSE.

La TABLA XI muestra la configuración de la red.

TABLA XI
RESUMEN DE LA ARQUITECTURA GRU CONSTRUIDA

Layer (type)	Output Shape	Param #
gru (GRU)	(None, 7, 64)	12864
dropout_7 (Dropout)	(None, 7, 64)	0
gru_1 (GRU)	(None, 7, 100)	49800
dropout_8 (Dropout)	(None, 7, 100)	0
gru_2 (GRU)	(None, 7, 128)	88320
dropout_9 (Dropout)	(None, 7, 128)	0
gru_3 (GRU)	(None, 256)	296448
dropout_10 (Dropout)	(None, 256)	0
dense_3 (Dense)	(None, 1)	257
=====		
Total params: 447,689		
Trainable params: 447,689		
Non-trainable params: 0		

5. METODOLOGÍA

Con base a la metodología de Box-Jenkins, se plantea que el modelo de referencia sobre el que se contrastan los demás modelos debe ser uno de la familia ARIMA, de tal manera que en el análisis exploratorio de los datos se busque reducir la serie de tiempo a una serie estacionaria con diferentes técnicas matemáticas, como el logaritmo natural, la derivada de la función y/o la autocorrelación.

5.1 BASELINE

Para llegar a los resultados siguientes, es necesario escalar los datos a logaritmo base 10 para trabajar con las tendencias de las monedas debido a la diferencia de valores en el precio de las variables.

En un primer acercamiento para la selección del modelo, se usa un ARIMA, que es muy utilizado en estadística y econometría para la predicción de series temporales. Este modelo consiste en encontrar patrones de tendencias basándose en el comportamiento de la variable en periodos anteriores sumado a la consideración de los residuos medios de la serie; este modelo ARIMA es

la unión de otros dos modelos AR y MA, descritos en la Sección 2.2, además incluye la integral de la serie original para buscar su estacionalidad.

Luego de varios experimentos se determina que el mejor modelo para este caso es un modelo ARIMA(1,0,1) que, como se indica, su componente $d = 0$, quedando de esta manera un modelo ARMA de orden 1.

Las métricas de referencia se calculan con los datos en escala logarítmica base 10, los resultados obtenidos con este modelo están en la tabla XII que se muestra a continuación.

TABLA XII
MÉTRICAS DE REFERENCIA MODELO ARIMA

Error cuadrático medio (MSE)	0.38
Error absoluto medio (MAE)	0.56
Error medio relativo absoluto (MREA)	0.14

La Figura 7 muestra el resultado de la predicción en el entrenamiento para el modelo ARMA contra el valor real de los datos, donde se evidencia un ajuste casi idéntico entre ambas curvas y bajo esta se muestra la curva de error MREA, para identificar los valores de error y sus magnitudes que conservan un comportamiento similar en cada punto, cuando se presentan picos en la serie de tiempo.

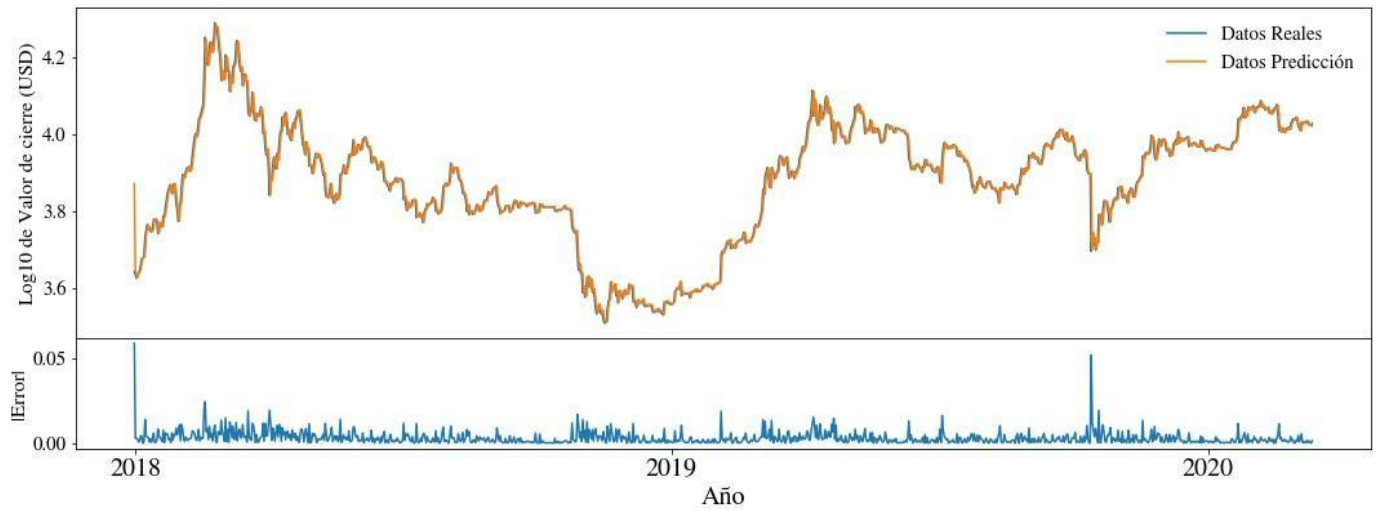


Figura 7: Gráfica de predicción del entrenamiento modelo ARIMA (1,0,1)

La Figura 8, por su lado muestra el resultado de la predicción en los datos de prueba para el modelo ARMA contra el valor real de los datos, donde se evidencia un error muy alto en el valor estimado, en el cual es prácticamente inexistente la tendencia en la gráfica de predicción, tal como se comprueba en la curva de error, donde se evidencia que hay una tendencia creciente a medida que avanza la serie.

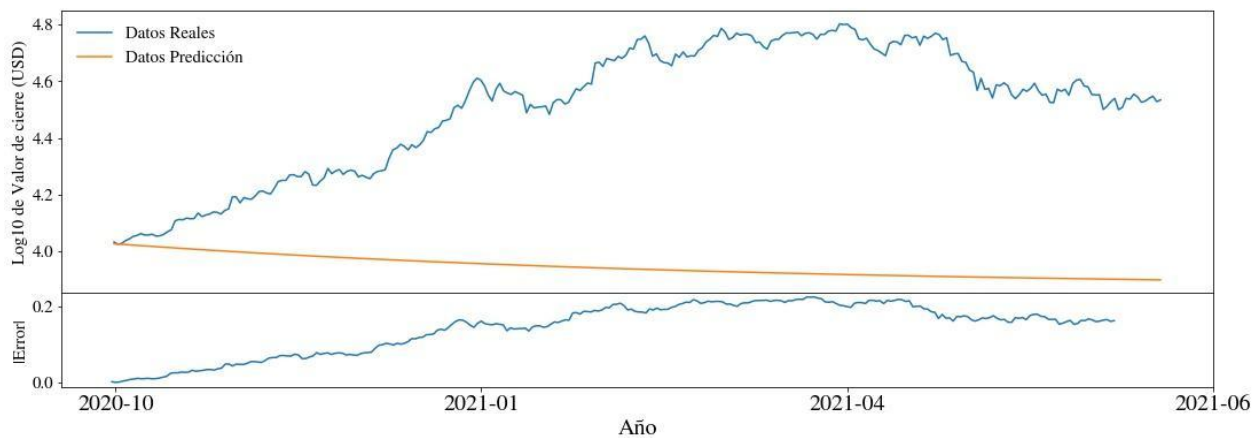


Figura 8: Gráfica de predicción del test modelo ARIMA (1,0,1)

En un Segundo acercamiento para la selección del modelo, se usa un ARIMAX. Este modelo es una “mejora” del modelo ARIMA, el cual incluye un nuevo parámetro, las variables exógenas, que consisten en variables externas a la variable de interés, pero lo que se busca es identificar si estas nuevas variables tienen una influencia en la variable objetivo y si presenta una relación entre ellas. Para el desarrollo del proyecto se utilizan los valores de cierre de otras criptomonedas, como se menciona en una sesión anterior en este documento.

Dicha decisión se basa en la gráfica de tendencia de las criptomonedas, detallada en la Figura 9, en la cual se puede identificar un comportamiento en su tendencia en el tiempo muy similar entre ellas, excepto Tether que es descartada por su comportamiento “ruidoso”.

Para este caso, cabe mencionar que dado que el modelo ARIMAX agrega variables exógenas al modelo, su parte integral tiene el mismo comportamiento que el modelo ARIMA, por lo tanto este también es de orden cero, pasando a ser modelo ARMA y ARMAX respectivamente. Con la diferencia que, para este caso, los resultados que se obtienen son mejores que en el anterior modelo.

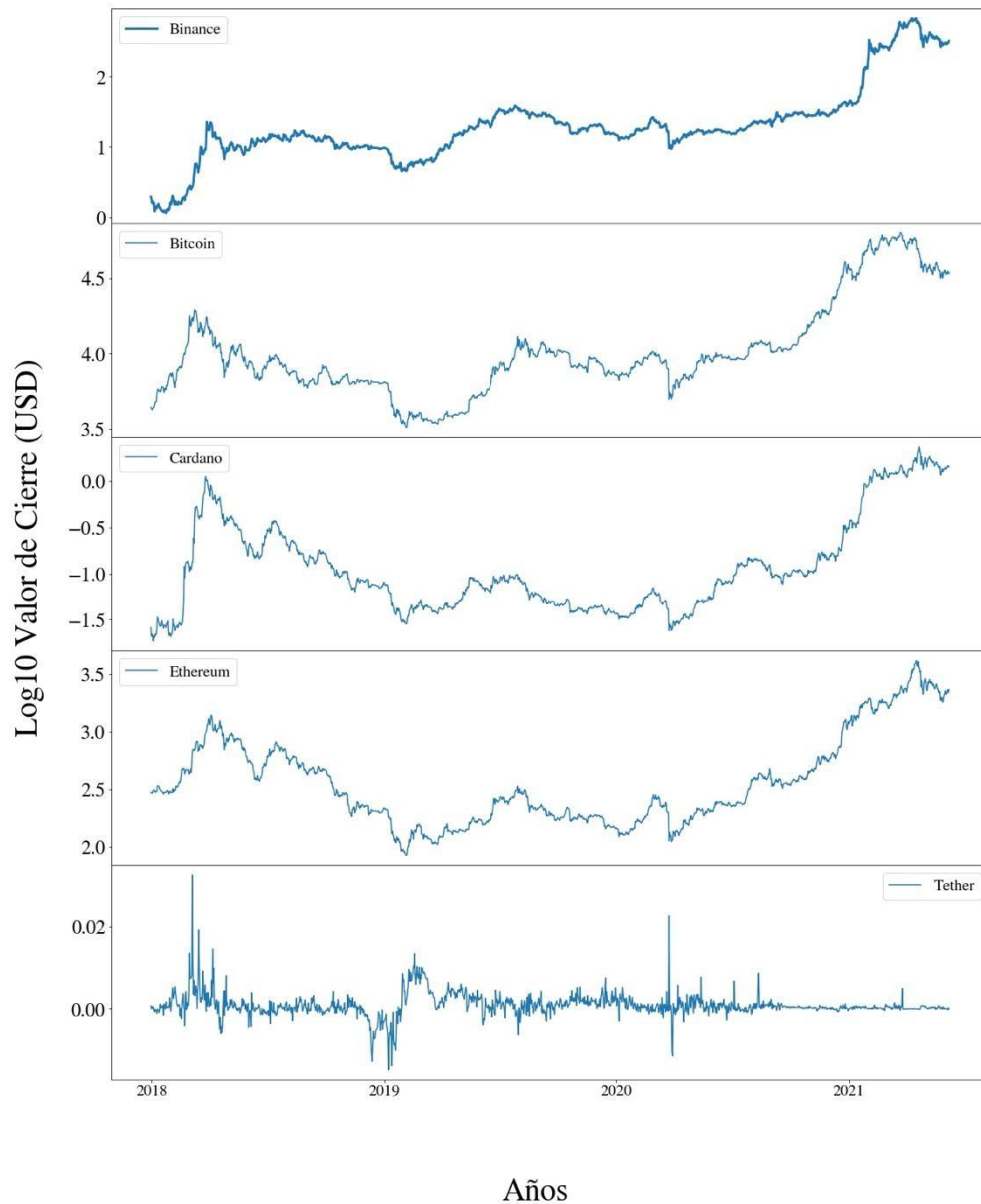


Figura 9: Gráfica de tendencia de las criptomonedas

La Figura 10 y 11 muestran el resultado de la predicción del modelo ARMAX, con relación al valor real de los datos de entrenamiento y de prueba respectivamente.

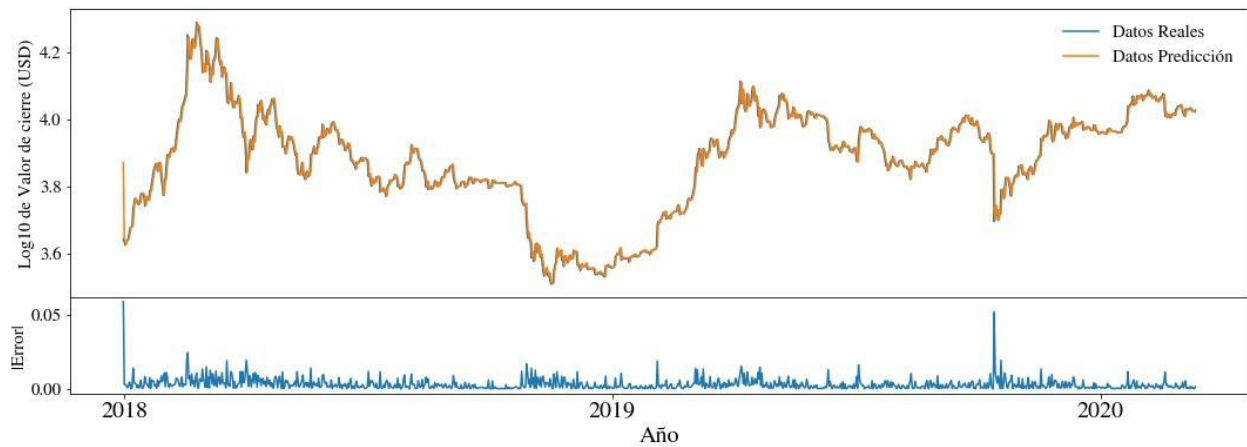


Figura 10: Gráfica de predicción del entrenamiento modelo ARIMAX (1,0,2)

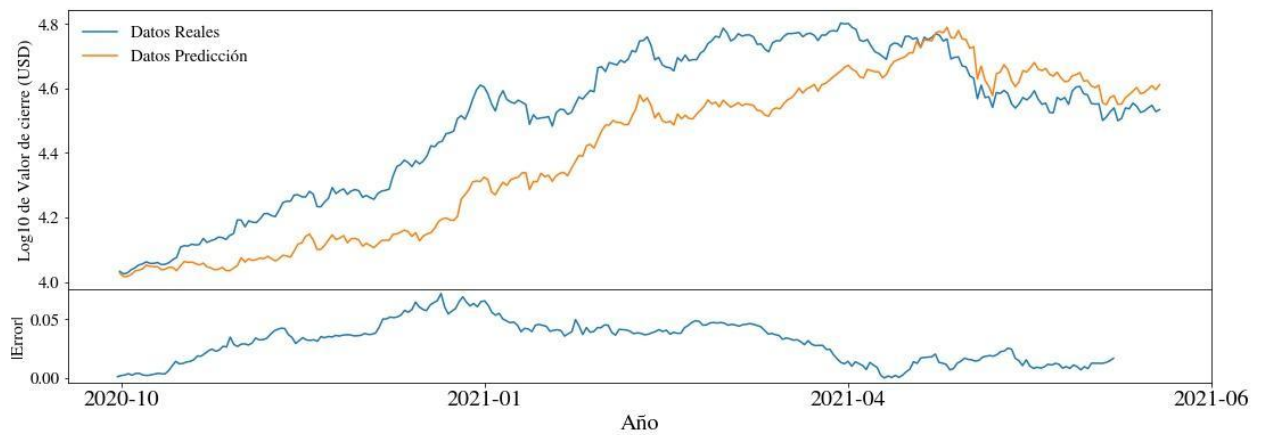


Figura 11: Gráfica de predicción del testing modelo ARIMAX(1,0,2)

Las métricas de referencias se calculan con los datos en escala logarítmica base 10 y los resultados de este modelo están en la tabla XIII que se muestra a continuación:

TABLA XIII
MÉTRICAS DE REFERENCIA MODELO ARIMAX

Error cuadrático medio (MSE)	0.02
Error absoluto medio (MAE)	0.13
Error medio relativo absoluto (MREA)	0.03

En resumen, a partir del trabajo realizado se toman dos resultados de gran relevancia. En primer lugar la anulación de la parte integral del modelo, que al aplicar esta integral lo que se logra es alejar aún más la serie de un estado estacionario; contrario al resultado obtenido en el análisis exploratorio, que al aplicar un diferencial a la serie esta tiende a ser estacionaria. Por otro lado, para los modelos ARMAX queda claro que la presencia de las variables exógenas tienen un gran peso en la estimación de los valores, porque permiten obtener un menor margen de error entre lo estimado y lo real, pero aún así no aportan mucha información al entrenamiento del modelo, que ayude a estimar mejores resultados, esto se da a que las series de tiempo que se están trabajando hacen parte de procesos estocásticos donde tienen un alto grado de incertidumbre y no se puede determinar una relación cercana entre las variables.

5.2 VALIDACIÓN

Debido a que el comportamiento de las series de tiempo es de carácter secuencial y los valores futuros dependen de los anteriores, no se pueden aplicar algunas otras técnicas de validación como cross-validation, que corresponden a saltos aleatorios en el muestreo de los datos. Por lo tanto los tres subconjuntos de datos para el valor de cierre del bitcoin se toman de la siguiente manera:

- Entrenamiento: 70% de la serie de tiempo para el entrenamiento de las redes neuronales.
- Validación: 10% de los datos.
- Prueba: 20% de los datos restantes para que sean tomados como prueba.

5.3 EXPERIMENTOS

Una vez realizado el preprocesamiento de los datos, como se describe en la Sección 4.2, de este documento, y establecida la métrica de referencia con base en los resultados del modelo ARMAX, se empieza a construir modelos basados en redes neuronales con el objetivo de encontrar la que minimice las funciones de costos MAE, MSE y MRAE, de cada red neuronal con el error de la métrica de referencia establecida.

Adicional con la construcción de redes neuronales se busca:

- Reducir los errores del modelo
- Reducir las oscilaciones de la regularización
- Buscar aproximar el modelo a una serie estacionaria.
- Reducir el overfitting

El diseño de los modelos se explica a continuación.

5.3.1 Modelo LSTM

La primera arquitectura que se usa para buscar mejorar el error es una LSTM, por ser una configuración diseñada para trabajar con valores que cambian en periodos de tiempo corto, como es el caso de las criptomonedas que tiene una ventana de media móvil entre 5 y 7 días.

5.3.1.1 Comparación de errores de Referencia LSTM

La TABLA XIV a continuación presenta la comparación entre las métricas de referencia y las métricas obtenidas en el modelo LSTM.

TABLA XIV
COMPARACIÓN DE MÉTRICAS LSTM VS ARMAX

MÉTRICAS DE REFERENCIA	MODELO ARMAX	MODELO LSTM
Error cuadrático medio (MSE)	0.02	0.00
Error absoluto medio (MAE)	0.13	0.01
Error relativo medio absoluto (MREA)	0.03	0.01

La figura 12, representa los datos de entrenamiento, validación y pruebas, con sus respectivas predicciones. Adicional se representa el error relativo entre los sets de datos reales y los de la predicción para cada set de datos.

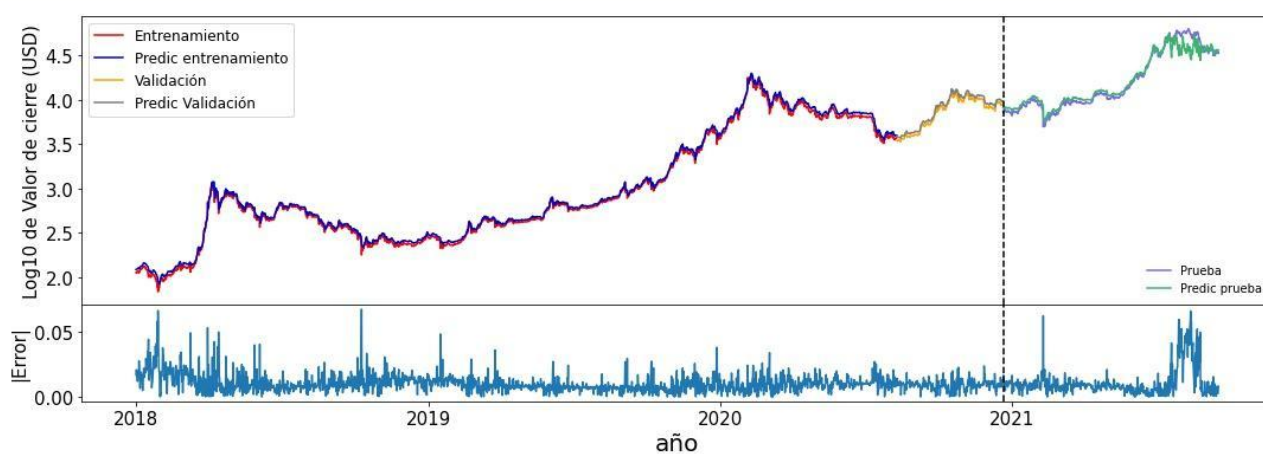


Figura 12: Comportamiento predicción y error red LSTM

De lo anterior se puede concluir que con relación al modelo ARMAX se tiene una mejora a resaltar que indica que no es tan relevante para las redes neuronales el uso de variables exógenas, por lo que en los siguientes experimentos no se ve la necesidad de usarlas. Adicionalmente, se nota en la parte de prueba, la presencia de unos picos de la parte final de la estimación, y con un posible overfitting.

5.3.2 Modelo Redes Recurrentes

La siguiente arquitectura es un modelo más simple empleando una red recurrente con una capa simple y una capa densa, esta configuración da las siguientes métricas:

5.3.2.1 Comparación de errores de Referencia RNN

La TABLA XV a continuación presenta la comparación entre las métricas de referencia y las métricas obtenidas en el modelo RNN

TABLA XV
COMPARACIÓN DE MÉTRICAS RNN VS ARMAX

MÉTRICAS DE REFERENCIA	MODELO ARMAX	MODELO RNN
Error cuadrático medio (MSE)	0.02	0.00
Error absoluto medio (MAE)	0.13	0.02
Error relativo medio absoluto (MREA)	0.03	0.01

La figura 13, representa los datos de entrenamiento, validación y pruebas, con sus respectivas predicciones. Adicional se representa el error relativo entre los sets de datos reales y los de la predicción para cada categoría, entrenamiento, validación y pruebas.

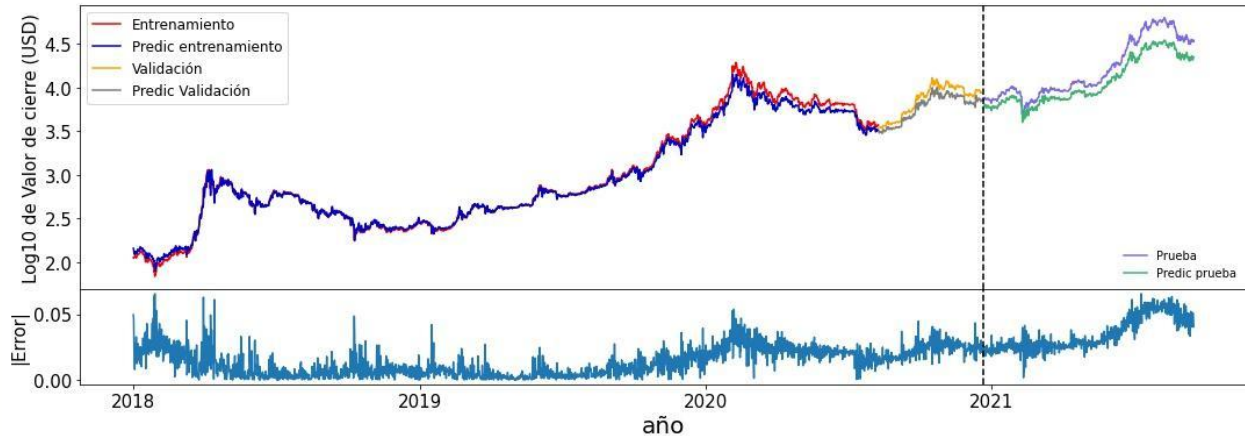


Figura 13: *Comportamiento predicción y error red RNN*

Para este caso, notamos que con relación al modelo base, se sigue presentando una mejora considerable en las métricas del modelo y del comportamiento de la serie, aún así para este caso en la parte de prueba la red neuronal simple no da un buen resultado con relación a la anterior, en un tema de escalamiento de los datos, pero se nota que la tendencia de la curva es muy semejante a la real, lo que podría dar un indicio que este modelo sería bueno en ese sentido, pero no en las estimaciones como tal. Adicionalmente, se evidencia, por la curva de error, que este va creciendo en el tiempo.

5.3.4 Modelo GRU (Gated Recurrent Unit)

La iteración final se realiza con una red GRU. Para esta red se elige una configuración de 4 capas GRU, seguidas de capas *dropout* que buscan controlar el overfitting y se termina con una capa densa, para la toma de decisiones.

5.3.4.1 Comparación de errores de Referencia GRU

La TABLA XVI presenta la comparación entre las métricas de referencia y las métricas obtenidas en el modelo GRU.

TABLA XVI
COMPARACIÓN DE MÉTRICAS GRU VS ARMAX

MÉTRICAS DE REFERENCIA	MODELO ARMAX	MODELO GRU
Error cuadrático medio (MSE)	0.02	0.01
Error absoluto medio (MAE)	0.13	0.05
Error relativo medio absoluto (MREA)	0.03	0.02

La Figura 14, representa los datos de entrenamiento, validación y pruebas, con sus respectivas predicciones. Adicional se representa el error relativo entre los sets de datos reales y los de la predicción para cada categoría, entrenamiento, validación y pruebas.

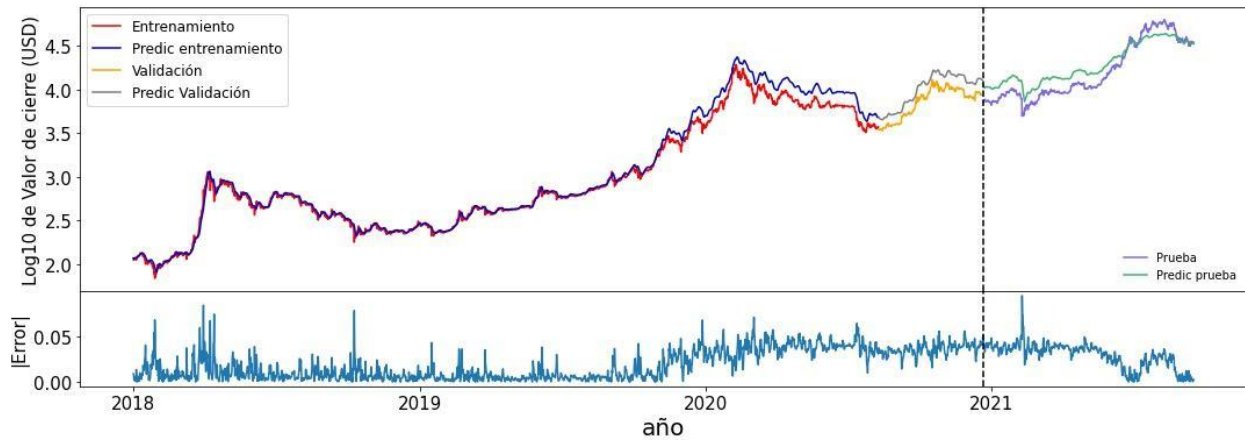


Figura 14: Comportamiento predicción y error GRU

La serie tiene inicialmente un comportamiento constante, y a medida que pasa el tiempo toma una tendencia creciente (véase figura 3). Esta tendencia se evidencia de forma más clara, al usar datos en escala logarítmica en base 10, de igual forma a medida que la serie crece abruptamente en tendencia, el modelo GRU es sensible a estos cambios, evidenciando que no se ajusta el valor real con la estimación, por lo tanto vemos que el error crece antes del año 2020, haciendo de este el peor modelo hasta ahora.

5.3.3 Modelo Convolutacional

También se implementa una red convolutacional con tres capas convolutacionales, capas dropout entre dichas capas para evitar el overfitting y finaliza con una capa densa. Los resultados de los errores se muestran a continuación.

5.3.3.1 Comparación de errores de Referencia CONV

La TABLA XVII presenta la comparación entre las métricas de referencia y las métricas obtenidas en el modelo.

TABLA XVII
COMPARACIÓN DE MÉTRICAS CONV VS ARMAX

MÉTRICAS DE REFERENCIA	MODELO ARMAX	MODELO CONV
Error cuadrático medio (MSE)	0.02	0.00
Error absoluto medio (MAE)	0.13	0.03
Error relativo medio absoluto (MREA)	0.03	0.01

La Figura 15, representa los datos de entrenamiento, validación y pruebas, con sus respectivas predicciones. Adicional se representa el error relativo entre los sets de datos reales y los de la predicción para cada categoría, entrenamiento, validación y pruebas.

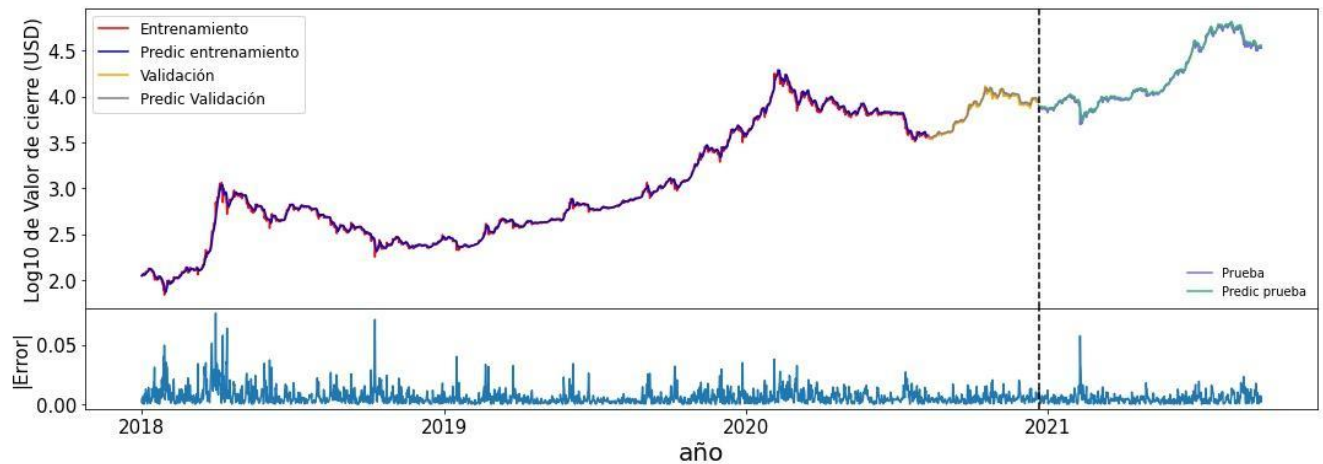


Figura 15: *Comportamiento predicción y error Red CONV*

Este modelo, tiene un comportamiento especial, y es el hecho que tanto el entrenamiento como las pruebas, existe una similitud muy grande entre ambas series, la real y la estimada, lo que indica que para este modelo el overfitting fue superior que en los anteriores, y por tal razón se descarta como modelo final.

5.4 HERRAMIENTAS

Retomando el diagrama de alto nivel de la arquitectura presentado en la Figura 1, se relacionan las herramientas que se usan en el desarrollo del proyecto.

- GitHub: Como Repositorio del código y del dataset.
- Google Drive: Como cuenta de almacenamiento para los datos y el modelo entrenado
- Google Colab: como Entorno de ejecución del código.

Por otro lado, se usan librerías de Python para el análisis, procesamiento y transformación de los datos y finalmente para el entrenamiento del modelo. (Ver Anexo 1 con listado de librerías).

6. RESULTADOS

Se describen los resultados en el desarrollo del proyecto, abordando las métricas que se obtienen en los diferentes experimentos y con las cuales se llega a conclusiones importantes, igualmente se evalúa el impacto de la solución con respecto al problema planteado con el fin de medir el alcance que se obtiene desde el punto de vista técnico y de negocio.

Para concluir, algunas consideraciones técnicas necesarias para industrializar el modelo y los puntos a tener en cuenta como servicios de nube, monitoreo, automatización y oportunidades de mejora.

6.1 MÉTRICAS

Para este análisis se toma como métricas de referencia las resultantes de un modelo ARMAX, y se confrontan con los modelos construidos en los diferentes experimentos planteados para elegir entre dichos experimentos el que mejor métrica presenta..

A pesar de que en términos generales el modelo con mejor métrica es el modelo convolucional esto se tiene un overfitting en su estimación, es por esta razón que se elige el modelo que tiene un menor error y a su vez presenta un mejor comportamiento con relación al overfitting, por lo tanto se selecciona el modelo LSTM como el de mejor resultado dentro de los experimentos realizados. Dado que los errores son tan pequeños, la elección del mejor modelo se realiza con el resultado de la métrica MAE. La TABLA XVIII muestra los resultados obtenidos del modelo de referencia, el modelo seleccionado LSTM y el CONV.

TABLA XVIII
COMPARACIÓN DE MÉTRICAS CONV VS ARMAX VS LSTM

MÉTRICAS DE REFERENCIA	MODELO ARMAX	MODELO CONV	MODELO LSTM
Error cuadrático medio (MSE)	0.02	0.00	0.00
Error absoluto medio (MAE)	0.13	0.03	0.01
Error relativo medio absoluto (MREA)	0.03	0.01	0.01

Adicionalmente, uno de los resultados que también llama la atención es el comportamiento de la curva del error de validación, el cuál es calculado con el error absoluto medio en el que se evidencia que a medida que se van incrementando la cantidad de épocas, el error no converge y presenta unas oscilaciones en la curva, lo que indica que para este tipo de serie de tiempo se presenta un problema de regularización. El cual en cada experimento se intenta alivianar con el uso de dropout y batch normalization, en la Figura 16 se puede observar el comportamiento descrito.

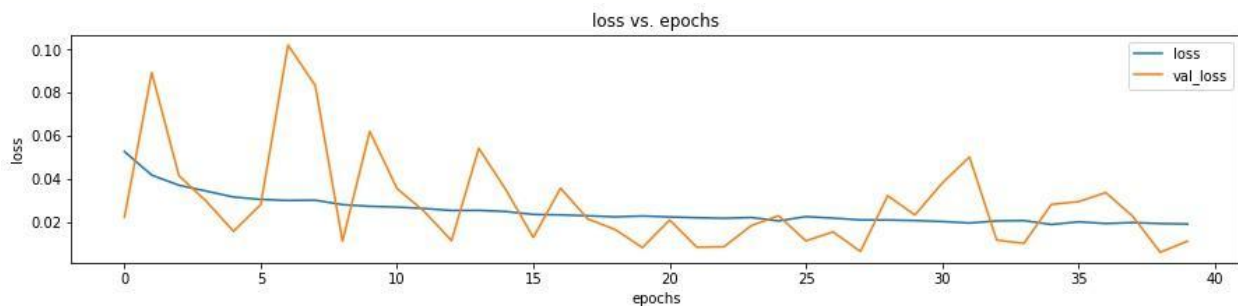


Figura 16: comportamiento de Pérdidas Train y Validación en 40 épocas para la red LSTM

6.2 EVALUACIÓN CUALITATIVA

Para la evaluación cualitativa, se identifica que para el caso de las redes neuronales se presenta overfitting en el entrenamiento del modelo, evento que se puede evidenciar con los valores arrojados por las métricas, donde da a entender que posiblemente para este escenario, la naturaleza de los datos es tan variable que la red puede entrar a aprender el comportamiento de los valores de entrenamiento muy bien. Sin embargo, este comportamiento con relación a los datos de prueba tiene un desempeño igual, en especial el de la red LSTM donde la tendencia de pruebas es la mejor, como se evidencia en la Figura 17.

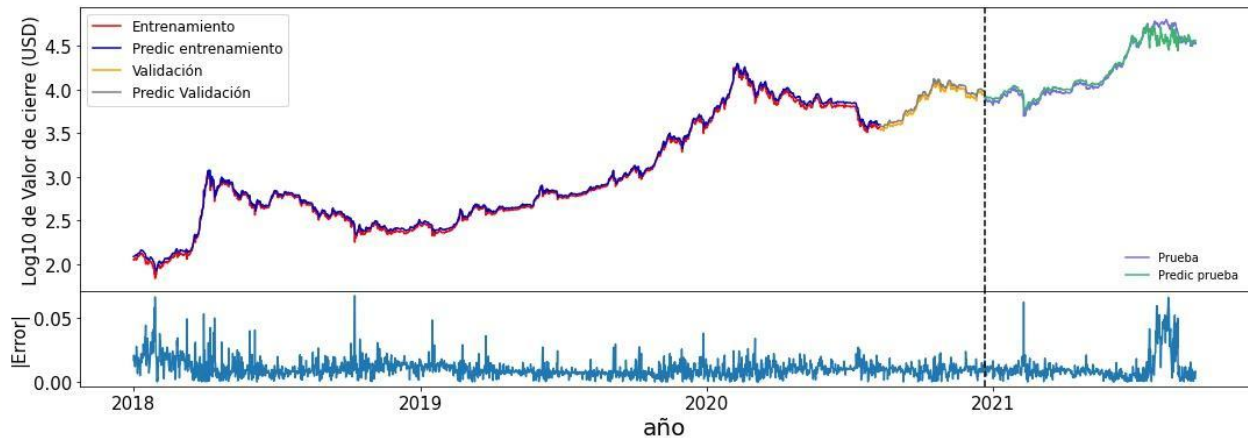


Figura 17: *Comportamiento predicción y Test LSTM*

Así mismo, con lo anterior se puede ver que las series de tiempo aplicada a este problema económico, tienen una alta incertidumbre y volatilidad, que implican que los resultados de predicción no sean muy buenos, a pesar de que las métricas indiquen lo contrario.

6.2.1 Validación del modelo

Como etapa final del proyecto, se valida el modelo LSTM. Para esto se decide arbitrariamente estimar los 15 valores siguientes a la última fecha que se tiene en el set de datos, con un look back de 7; de esta manera lo que se hace es calcular el valor inmediatamente siguiente, a partir del valor previamente predicho. Como resultado se observa una fuerte subida en la tendencia, comportamiento que el modelo aprendió de los datos, según podemos detallar en la Figura 18.

De igual manera se observa que los valores siguientes a partir de la fecha 2021-06-07, tienen una tendencia creciente en consecuencia del error que se agrega en cada iteración de la estimación, esto conlleva un aumento del error que provoca una tendencias más abrupta.



Figura 18: Estimación de los 15 valores siguientes.

6.3 CONSIDERACIONES DE PRODUCCIÓN

En esta sección se plasma un diagrama de alto nivel sobre cómo debería ser el flujo de ejecución del modelo en un ambiente productivo. Por lo tanto, es necesario para poder industrializar el modelo se cuente con algunos servicios de nube adicionales que ayuden a sostener la infraestructura del despliegue y del consumo del modelo.

La infraestructura en términos generales debe contener dos microservicios, uno a modo de consumo de información de la página web de las criptomonedas, para capturar los valores pasados del bitcoin y otro con el objetivo de disponibilizar el resultado del modelo en una herramienta de visualización.

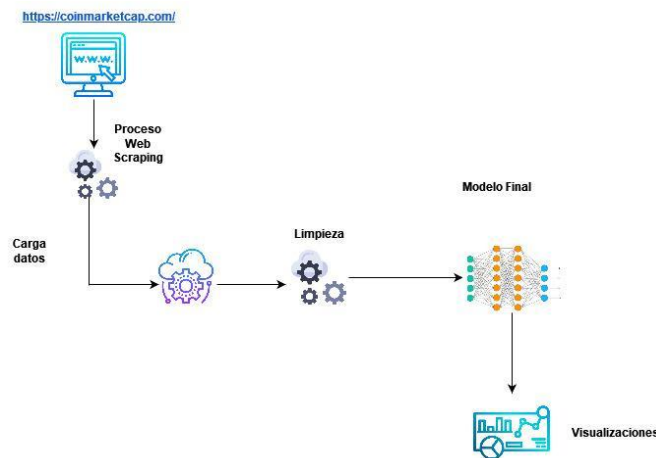


Figura 19: *Arquitectura despliegue en producción*

Ahora bien, como se muestra en la Figura 19, lo que se busca en el diseño es realizar un reentrenamiento continuo del modelo, es decir, inicialmente se tiene el modelo previamente entrenado y sobre cual se pueden realizar las estimaciones que posteriormente serán visualizadas por el usuario; Pero, debido a la gran incertidumbre de la serie de tiempo, se considera que el modelo debe ser reajustado cada semana, a modo de qué vaya incluyendo nuevas características en sus pesos.

Esto se debe diseñar de la siguiente manera:

1. Por medio de técnicas de web scraping se extraen los datos semanales del sitio web que contiene la información de la criptomoneda.
2. Se hace nuevamente el pre-procesamiento de los datos.
3. Se ajusta el modelo nuevamente y se guarda.
4. Se realizan las predicciones y se visualiza el resultado de la tendencia.

De esta forma se busca que las nuevas predicciones no aumenten el ruido de las estimaciones posteriores, y que el modelo se mantenga afinado.

7. CONCLUSIONES

La estimación de la tendencia de series de tiempos al tratarse de procesos estocásticos es un tema complejo de abordar, debido a que debe tenerse en cuenta la aleatoriedad como una característica que hace impredecible el valor que toma la variable en un momento del tiempo. Por lo tanto, para hacer del fenómeno algo que se pueda predecir es necesario identificar un comportamiento estacional que defina un patrón que dé indicios para una estimación de predicción. Es por ello que existen diferentes modelos bien estudiados desde el punto de vista de econometría que ayudan a modelar los diferentes escenarios que se puedan presentar en los datos usados para predecir tendencias futuras en series de tiempo.

En esta monografía se analizó el comportamiento del bitcoin como una serie de tiempo y se usaron dos modelos clásicos como son ARMA y ARMAX, para ambos casos es necesario un preprocesamiento de los datos para buscar la tendencia de la serie y de esta manera hacer un estimado de la estacionalidad.

Luego de analizar los resultados de los modelos antes mencionados, se toma la decisión de usar como métricas de referencia las arrojadas por el modelo ARMAX, debido a que este trabaja con variables exógenas que ayudan a estimar valores más acertados en las tendencias futuras.

Continuando con los experimentos, se concluye que uno de los problemas más frecuentes que se evidencia es la aparición de overfitting en el conjunto de entrenamiento y prueba, para lo cual se aplican técnicas de regularización con el objetivo de reducir el impacto de este, como la técnica de *dropout layer* y el *batch normalization*. Mostrando la primera un mejor resultado que la segunda, al permitir que se presentara una mayor flexibilidad en la etapa de aprendizaje, para lo cual funcionó muy bien la red LSTM. Aun así, se resalta que para el caso de pruebas, los resultados finales dieron muy semejantes a los reales, lo que da indicios que el modelo presenta un overfitting que podría ser atacado con algunas otras técnicas de regularización como L1-L2 [29], debido a que la generación de valores siguientes no presenta muy buenos resultados.

Por otro lado, los modelos de redes neuronales hoy en día son herramientas muy poderosas para el análisis de series de tiempo, ya que de manera experimental se puede detallar por ejemplo, que para cada experimento estás son superiores que los modelos clásicos de econometría, además no es necesario el uso de variables exógenas para obtener un resultado mejor que el del modelo de referencia, ARMAX. Es así como el modelo LSTM es la red con mejor comportamiento,

tomando en consideración las métricas y el overfitting, al presentar un balance entre estas dos características.

Por último, se propone que para enriquecer los experimentos realizados en esta monografía, se diseñen futuros trabajos utilizando como variables exógenas otros commodities, tales como el precio del petróleo, el oro u otro relevante en el mercado y ver cómo se correlacionan con la variación del precio de la criptomoneda influyendo de alguna forma en el valor que tome el activo en un momento dado en el tiempo.

8. REFERENCIAS BIBLIOGRÁFICAS

- [1] Nakamoto, S. (2008). Bitcoin: Un sistema de efectivo electrónico usuario-a-usuario. Bitcoin.org.
- [2] Tapscott, D., & Tapscott, A. (2017). La revolución blockchain. Descubre cómo esta nueva tecnología transformará la economía global. ediciones deusco. séptima edición. recuperado en [webdelprofesor.ula.ve/economia/oscard/materias/E_E_Mundial/Economia_Internacional_Krugman_Obstfeld.pdf](#).
- [3] O'mahony, D., Peirce, M., & Tewari, H. (1997). *Electronic payment systems* (pp. I-XII). Norwood: Artech House.
- [4] Pineda, L. (2011). La crisis financiera de los Estados Unidos y la respuesta regulatoria internacional. *Revista Aequitas: Estudios sobre historia, derecho e instituciones*, (1), 129-214.
- [5] Villavicencio, J. (2010). Introducción a series de tiempo. Puerto Rico.
- [6] Edwards, R. D., Magee, J., & Bassetti, W. H. C. (2021). Análisis técnico de las tendencias de los valores. Profit Editorial.
- [7] Chávez Quisbert, Nicolás. (1997). MODELOS ARIMA. *Revista Ciencia y Cultura*, (1), 23-30. Recuperado de http://www.scielo.org.bo/scielo.php?script=sci_arttext&pid=S2077-33231997000100005&lng=es&tlng=es.
- [8] Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1985). *Learning internal representations by error propagation*. California Univ San Diego La Jolla Inst for Cognitive Science.
- [9] Fukushima, K., & Miyake, S. (1982). Neocognitron: A self-organizing neural network model for a mechanism of visual pattern recognition. In *Competition and cooperation in neural nets* (pp. 267-285). Springer, Berlin, Heidelberg.
- [10] Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8), 1735-1780.
- [11] Chung, J., Gulcehre, C., Cho, K., & Bengio, Y. (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*.
- [12] González Casimiro, M. P. (2009). Análisis de series temporales: Modelos ARIMA.
- [13] Nava, F. A. (2015). Procesamiento de series de tiempo. Fondo de Cultura Económica.
- [14] Larios Meoño, J. F., & Alvarez, V. J. (2014). Análisis econométrico de series de tiempo. Universidad San Ignacio de Loyola.
- [15] Cheung, Y. W., & Lai, K. S. (1995). Lag order and critical values of the augmented Dickey–Fuller test. *Journal of Business & Economic Statistics*, 13(3), 277-280.
- [16] Robert H. Shumway & David S. Stoffer (2017). *Time Series Analysis and Its Applications With R Examples*, Fourth Edition: New York, NY: Springer, pp 19-21
- [17] De Arce, R., & Mahía, R. (2003). Modelos Arima. Programa CITUS: Técnicas de Variables Financieras. pp 5-6
- [18] De Arce, R., & Mahía, R. (2003). Modelos Arima. Programa CITUS: Técnicas de Variables Financieras. pp 6-7

- [19] Sklearn.Metrics.Mean_squared_error. (s/f). Scikit-Learn. Recuperado el 28 de mayo de 2022, de https://scikit-learn.org/stable/modules/generated/sklearn.metrics.mean_squared_error.html
- [20] Sklearn.Metrics. Mean_absolute_error. (s/f). Scikit-Learn. Recuperado el 28 de mayo de 2022, de https://scikit-learn.org/stable/modules/generated/sklearn.metrics.mean_absolute_error.html
- [21] R.J. Hyndman, A.B. Koehler, "Another look at measures of forecast accuracy", *International Journal of Forecasting*, 22 (2006), pp. 679-688
- [22] Margalef Roig, J., Miret-Artés, S., & Outerelo Domínguez, E. (2014). *Probabilidad y Economía 3. Procesos estocásticos*. Sanz y Torres.
- [23] Brownlee, J. (2018). Deep learning for time series forecasting: predict the future with MLPs, CNNs and LSTMs in Python. *Machine Learning Mastery*, pp. 87-92
- [24] Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. MIT Press, pp 388-390.
- [25] Glorot, X., & Bengio, Y. (2010, March). Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics* (pp. 249-256). JMLR Workshop and Conference Proceedings.
- [26] Sharma, S., Sharma, S., & Athaiya, A. (2017). Activation functions in neural networks. *towards data science*, 6(12), 310-316.
- [27] Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.
- [28] Naranjo, M. A. G. (s/f). *Redes neuronales recurrentes con puertas Gated Recurrent Neural Networks*. Cs.us.es. Recuperado el 29 de mayo de 2022, de http://www.cs.us.es/~fsancho/ficheros/IAML/Intro_LSTM.pdf
- [29] Chollet, F. (2021). *Deep learning with Python*. Simon and Schuster. pp 107-108.

9. ANEXOS

ANEXO 1: Librerías utilizadas para el desarrollo del proyecto

1. Creación de punto de montaje

```
from google.colab import drive
```

2. Análisis, procesamiento y visualización de los datos:

```
import pandas as pd
```

```
import numpy as np
```

```
import matplotlib
```

```
import seaborn as sns
```

```
import matplotlib.pyplot as plt
```

```
from datetime import datetime
```

```
from sklearn.preprocessing import MinMaxScaler
```

3. Validación de modelos estadísticos

```
import statsmodels.tsa.stattools as sts
```

```
import statsmodels.api as sm
```

```
from statsmodels.tsa.seasonal import seasonal_decompose
```

```
from statsmodels.tsa.arima_model import ARIMA
```

4. Redes Neuronales

```
from keras.layers import LSTM, GRU, Dense, Conv1D, RNN, Dropout, Flatten
```

```
from keras.models import Sequential
```

5. Cálculo de Métricas de validación

```
from sklearn.metrics import mean_squared_error, mean_absolute_percentage_error
```

```
from sklearn.metrics import median_absolute_error
```