



Maestría en Explotación de Datos y Gestión del Conocimiento

Cohorte 2021-22

Trabajo final

**Predicción de Ventas de Juguetes por Segmento de Negocio para MyC en el Mercado de Argentina en los próximos 12 meses con Modelos Basados en Series Temporales.**

Tesista:

Cristian Gonzalo Di Bono

Directora:

Mg. Fernanda Méndez

Codirector:

Mg. Braian Drago

9 de Marzo de 2025

## **Agradecimientos**

Quiero expresar mi más profundo agradecimiento a todas las personas que han contribuido al desarrollo de esta tesis.

En primer lugar, quiero agradecer a mi directora, Fernanda Méndez, por su invaluable guía, paciencia y apoyo durante todo el proceso de investigación. Sus conocimientos y consejos han sido fundamentales para la realización de este trabajo.

A mis compañeros de clase y amigos, especialmente a Antonio Rial y Vladimiro Bellini, gracias por su amistad y por los momentos de apoyo y motivación compartidos durante estos años. Su compañía ha hecho este viaje mucho más llevadero y enriquecedor.

A mi familia por su ejemplo y sus sacrificios, han sido una fuente constante de inspiración y fuerza.

Profesores y personal administrativo de la Universidad Austral, gracias por su dedicación y profesionalismo.

Finalmente a mi querida esposa Melanie por acompañarme día a día en cada paso de mi carrera tanto laboral como profesional.

Este logro no habría sido posible sin el apoyo y la contribución de todos ustedes.

9 mar 2025

<b>Resumen</b>	<b>4</b>
<b>1. Introducción</b>	<b>5</b>
1.1. Motivación e importancia del tema bajo estudio	5
1.2. Objetivo, alcance y límites	6
<b>2. Contexto organizacional</b>	<b>8</b>
<b>3. Series de tiempo, conceptos preliminares</b>	<b>9</b>
3.1. Patrones de series de tiempo	9
3.2. Estacionariedad	9
3.3. Función de Autocorrelación	10
3.4. Transformaciones	10
3.5. Pruebas de Raíces Unitarias	11
3.6. Modelos Autorregresivos	11
3.7. Modelos de Media Móvil	11
3.8. Modelos ARIMA	12
3.9. Árboles de Decisión	12
3.10. Redes Neuronales LSTM	13
3.11. XGBoost	14
3.12. Prophet	15
3.13. Algoritmos automáticos (TPOTRegressor)	16
3.14. Separación de los datos	17
3.15. Validación cruzada	18
3.16. Validación cruzada de datos de series temporales	18
3.17. Métricas de performance	18
3.17.1. Error Medio Absoluto (MAE)	18
3.17.2. Error Cuadrático Medio (MSE)	19
3.17.3. Raíz del Error Cuadrático Medio (RMSE)	19
3.17.4. Error Porcentual Absoluto Medio (MAPE)	19
<b>4. Origen, exploración y descripción de los datos</b>	<b>20</b>
4.1. Origen y consideraciones de los datos	20
4.2. Exploración y descripción de los datos	21
4.3. Análisis de Componentes Principales (PCA)	27
4.4. Descomposición de la serie	28
<b>5. Implementación de algoritmos</b>	<b>32</b>
5.1. Test de raíces unitarias	32
5.2. Función de Autocorrelación (ACF):	33
5.3. Gráfico de la función de autocorrelación parcial (PACF):	36
5.4. Proceso Autorregresivo (AR) y Proceso de Media Móvil (MA)	38
5.5. Modelos Univariados	39
5.5.1. Modelos ARIMA	39
5.5.2. Modelo de Redes Neuronales Long Short-Term Memory (LSTM)	43
5.5.3. Modelos Random Forest Regressor	50
5.5.4. Modelos Xgboost	54
5.5.5. Modelos Prophet (Facebook)	56
5.5.6. Modelos Automáticos (TPOTRegressor)	61
5.6. Modelos Multivariados (para variable "Total MyC")	62
5.6.1. Random Forest Multivariado	62
5.6.2. Gradient Boosting Regressor Multivariado	64
5.6.3. LSTM Multivariado	66
5.6.4. Híbrido Random Forest con residuos de LSTM	69
5.6.5. Híbrido Autoarima con residuos de LSTM	72
5.6.6. Random Feature Addition / Random Noise Injection ("pajaritos")	73
5.6.7. Híbrido Random Forest con residuos de LSTM (usando las 10 variables seleccionadas con RFE)	73

5.6.8. Híbrido Autoarima con residuos de LSTM (usando las 10 variables seleccionadas con RFE)	74
<b>6. Comparaciones de mejores modelos y conclusiones generales</b>	<b>78</b>
6.1. Conclusiones para variable “Total MyC”	78
6.2. Conclusiones para variable “Total M”	81
6.3. Conclusiones para variable “Total C”	83
6.4. Conclusiones generales	85
<b>7. Trabajos Futuros</b>	<b>87</b>
<b>Bibliografía</b>	<b>89</b>
<b>Anexo I</b>	<b>90</b>

## ***Resumen***

Los juguetes son esenciales para el entretenimiento y desarrollo infantil, además de impulsar la industria del ocio y la calidad de vida. Su consumo per cápita refleja la actividad económica, y su demanda está sujeta a tendencias y estacionalidades. Por ello, es crucial anticipar con precisión, especialmente en el mercado argentino. Dado que la empresa utiliza una ventana de presupuestación de 12 meses, este estudio propone modelos predictivos de series temporales para estimar la demanda de juguetes en Argentina en dicho horizonte, segmentando por unidad de negocio y a nivel total. También se identifican los principales indicadores de esta demanda. Los datos provendrán de bases internas de la empresa, complementados con información macroeconómica de consultoras. La estimación precisa permitirá mejorar la planificación interna y optimizar decisiones de inversión en capital de trabajo e infraestructura.

## **Palabras claves**

Juguetes, demanda, series temporales, modelos predictivos, ARIMA, Random Forest Regressor, Redes Neuronales LSTM, Prophet, TPOTRegressor, Xgboost, Gradient Boosting Regressor.

## 1. Introducción

### 1.1. Motivación e importancia del tema bajo estudio

Existen diversas razones que justifican el estudio de modelos de series temporales para la predicción de ventas de juguetes en Argentina. Las principales son:

- **Ayudar a la industria del juguete a tomar decisiones estratégicas:** las predicciones de ventas futuras pueden ser útiles para la industria del juguete al momento de planificar la producción, definir estrategias de comercialización e inversión, y optimizar la gestión de inventarios.
- **Mejorar la eficiencia y rentabilidad del sector:** contar con estimaciones precisas de la demanda permite ajustar la producción y optimizar los recursos de manera más eficiente, lo que contribuye a una mayor rentabilidad y sostenibilidad del negocio.
- **Proporcionar información valiosa para inversores y analistas financieros:** las proyecciones de ventas de juguetes pueden ser de interés para inversores y analistas que evalúan oportunidades en la industria o analizan su desempeño financiero.
- **Contribuir al conocimiento y comprensión del comportamiento del mercado de juguetes:** este estudio puede aportar una mejor comprensión del comportamiento del mercado y de los factores externos que pueden influir en la demanda, como tendencias económicas, cambios regulatorios y patrones estacionales de consumo. Se considera que este aspecto es clave para contextualizar los modelos predictivos y analizar los datos dentro de su entorno específico.

Debido a que las previsiones de ventas suelen ser en su mayoría incorrectas, es importante para las empresas hacer la previsión lo más precisa posible, para minimizar las posibles consecuencias como la compra incorrecta de materiales o la programación incorrecta de capital de trabajo (Arnold et al., 2007; Currie & Rowley, 2010; Herbig et al., 1993). Cuanto más precisa sea la información utilizada para hacer la previsión, más precisa será la información resultante que se obtenga (Lawrence et al., 2000).

Según Hoshmand (2010), muchas empresas solo utilizan técnicas cualitativas, donde se realizan juicios de ventas para crear un pronóstico de ventas. Las empresas no quieren hacer el esfuerzo o no ven el beneficio de las técnicas cuantitativas, que utilizan datos históricos,

como el historial de ventas, para crear un pronóstico de ventas. Moon (2013) menciona que depender únicamente de técnicas cualitativas puede ser costoso y llevar mucho tiempo al realizar pronósticos. Por lo tanto, las empresas deberían evaluar cómo adoptar diferentes técnicas de pronóstico para aumentar la precisión del pronóstico de ventas.

En síntesis, el enfoque del problema basado en modelos de series temporales para la predicción de ventas de juguetes en Argentina podría ser de gran valor para la industria del juguete, los inversores y analistas financieros, y para el conocimiento y comprensión del comportamiento del mercado de juguetes en general.

### **1.2. Objetivo, alcance y límites**

El objetivo general es desarrollar y comparar modelos predictivos de series temporales para pronosticar las ventas de juguetes por segmento de negocio en el mercado argentino, con un horizonte de predicción de 12 meses a partir del último dato mensual disponible. Con la finalidad de obtener uno o varios modelos que permitan tomar mejores decisiones empresariales.

Debido a la naturaleza confidencial de los datos, se ha optado por una estrategia de protección de la información. Esta estrategia incluye la modificación de los nombres originales de las variables y la transformación de los valores de las mismas a través de un factor de conversión fijo. Esta transformación asegura que las relaciones proporcionales entre los datos se mantengan intactas, permitiendo un análisis válido sin comprometer la confidencialidad.

El estudio se enfocará en la predicción de ventas para dos áreas geográficas específicas "M" y "C" y para el mercado argentino en su totalidad. Se utilizarán datos históricos de ventas y otras variables relevantes disponibles en las bases de datos internas de la compañía ficticia y fuentes externas.

Este trabajo se apoya en la información disponible en bases de datos internas de una compañía ficticia, desarrolladas y mantenidas por el Departamento de Planeamiento Económico y Financiero, así como en datos obtenidos de diversas fuentes gubernamentales y no gubernamentales. Se realizarán las transformaciones y correcciones correspondientes cuando sea oportuno y posible.

Si bien la información presentada en la sección 2 “Contexto organizacional” cuenta con información consolidada de la compañía hasta el año 2022, para el segmento de Argentina se cuenta con información de ventas mensuales desde enero de 2007 hasta junio de 2024 (210 observaciones) y datos de ventas presupuestadas desde julio de 2021 hasta junio de 2024 (36 observaciones). Adicionalmente se incorporaron 346 variables explicativas al modelo de fuentes externas.

Existen ciertos requerimientos y desafíos que deben considerarse al utilizar modelos de series temporales para la predicción de ventas de juguetes:

- **Datos suficientes y de calidad:** se considera que la cantidad y calidad de datos históricos es suficiente para lograr predicciones precisas.
- **Selección del modelo adecuado:** existen múltiples enfoques en la modelización de series temporales, por lo que es importante seleccionar el modelo que mejor se ajuste a los datos y al propósito de la predicción.
- **Consideración de factores externos:** Se considerarán factores económicos, estacionales y de consumo que puedan influir en las ventas de juguetes.
- **Gestión del error:** es necesario diseñar y gestionar un proceso de pronóstico eficiente, empleando diferentes técnicas de modelización y combinando metodologías para aumentar la precisión del análisis.
- **Cambios en los patrones:** los patrones en los datos de ventas de juguetes pueden cambiar con el tiempo debido a factores externos, como fluctuaciones económicas, cambios en políticas comerciales o regulaciones.

En resumen, existen desafíos que pueden afectar la precisión y confiabilidad de los modelos de series temporales, como cambios en los patrones de ventas, inconsistencias en los datos y la dificultad de incorporar ciertos factores externos. Sin embargo, a pesar de estos desafíos, se considera que sigue siendo posible obtener predicciones valiosas mediante el uso adecuado de modelos de series temporales y la gestión efectiva de estas limitaciones.

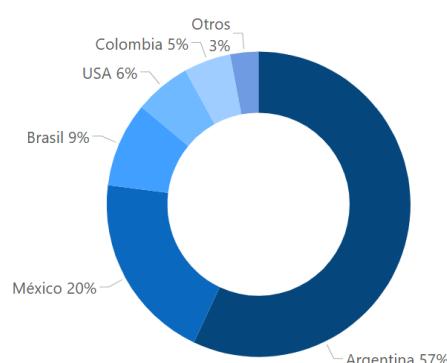
## 2. Contexto organizacional

MyC es una empresa ficticia de un fabricante de una amplia gama de productos de juguetería con la más alta tecnología. MyC y sus subsidiarias cuentan con centros de producción ubicados en Latinoamérica, consolidándose como una de las empresas más importantes del sector en la región.

Su capacidad de producción anual es de 4.1 millones de unidades de juguetes, y sus acciones cotizan en la Bolsa de Valores de Nueva York. Durante 2022, el volumen de ventas alcanzó los 3.9 millones de unidades, equivalentes a 5.3 mil millones de dólares, lo que representó un aumento del 2% en comparación con el período anterior y 83% con respecto a 2020 (año de la pandemia). Con un incremento de 3 millones de unidades en comparación con los niveles de ventas de 2020, MyC logró poner en marcha nuevas instalaciones en Brasil y Chile, impulsadas por la recuperación del mercado tras el impacto del brote de COVID-19 en la actividad económica y la demanda de juguetes.

MyC es una empresa líder en la fabricación de juguetes en América Latina, con aproximadamente 7,000 empleados al 31 de diciembre de 2022. Durante 2022, las ventas en el mercado argentino fueron de 2.2 millones de unidades, representando el 57% del total de envíos de la empresa. La demanda de juguetes creció en el año, reflejando un mercado más dinámico en 2022. Las ventas en la Región Sur alcanzaron el 71% de los envíos consolidados de MyC, con Argentina como principal destino dentro de la región. Las ventas en la región norte sumaron 1.0 millones de unidades en 2022, representando el 26% de las ventas consolidadas de MyC. Los principales destinos de exportación fueron Argentina, México y Brasil.

*Figura 1: Ventas de juguetes por país, 2022*



*Nota. Elaboración propia utilizando Jupyter Notebook en Visual Studio Code en base a Informe de Gestión MyC 2022.*

### 3. Series de tiempo, conceptos preliminares

Una serie temporal se define como una secuencia de observaciones recopiladas a intervalos de tiempo sucesivos (Chatfield, 2019). Estas series pueden originarse en diversas áreas, como la física, meteorología, demografía, o marketing. Es común encontrar series temporales en ámbitos económicos y financieros, donde se registran, por ejemplo, precios de acciones, volúmenes de exportación mensuales, o ingresos promedios mensuales.

Un modelo de serie temporal para un conjunto de datos  $\{X_t\}$  es una representación matemática que describe las distribuciones conjuntas (o en algunos casos solo las medias y covarianzas) de una secuencia de variables aleatorias  $\{X_t\}$ , siendo  $\{X_t\}$  una realización de estas (Brockwell, 2016). El análisis de series temporales implica la extracción de información estadística y significativa de puntos de datos dispuestos cronológicamente. Este análisis no solo ayuda a entender el comportamiento pasado, sino que también es fundamental para prever cómo puede evolucionar una serie en el futuro (Nielsen, 2020).

Al realizar pronósticos, el objetivo es estimar la evolución futura de una secuencia de observaciones. Dado que el futuro es incierto, la predicción puede verse como una variable aleatoria, y normalmente, la incertidumbre del pronóstico aumenta cuanto más se proyecta hacia el futuro (Hyndman & Athanasopoulos, 2021).

#### 3.1. Patrones de series de tiempo

Al analizar una serie temporal, es crucial identificar ciertos patrones que pueden caracterizarla y ofrecer información valiosa:

- **Tendencia:** Refleja un aumento o disminución a largo plazo en los datos, que no necesariamente es lineal.
- **Estacionalidad:** Se observa cuando una serie temporal se ve influenciada por factores recurrentes, como estaciones del año o días de la semana, con una frecuencia fija y conocida.
- **Ciclo:** Se refiere a fluctuaciones en los datos que no tienen una frecuencia fija (Hyndman & Athanasopoulos, 2021).

#### 3.2. Estacionariedad

Una serie temporal se considera estacionaria si no presenta cambios sistemáticos en la media (es decir, no tiene tendencia), si la varianza se mantiene constante, y si las variaciones periódicas han sido eliminadas (Chatfield, 2019). La teoría de series temporales se enfoca en gran medida en series estacionarias, lo que hace necesario, en ocasiones, transformar una serie no estacionaria en estacionaria para poder aplicar esta teoría. Esto puede implicar la eliminación de tendencias o patrones estacionales y modelar las variaciones residuales mediante un proceso estocástico estacionario (Chatfield, 2019).

Desde una perspectiva matemática, un proceso estocástico se define como una colección de variables aleatorias ordenadas en el tiempo y definidas en un conjunto de puntos temporales, que pueden ser continuos o discretos (Chatfield, 2019).

### **3.3. Función de Autocorrelación**

La autocorrelación es una herramienta esencial para evaluar la dependencia en los datos y seleccionar un modelo adecuado. Los coeficientes de autocorrelación miden la relación entre observaciones separadas por distintos intervalos de tiempo y proporcionan información descriptiva sobre la serie. Estos coeficientes varían entre -1 y 1, donde un valor cercano a cero indica independencia entre las variables (Chatfield, 2019). En una serie estacionaria, el ACF (Función de Autocorrelación) disminuye rápidamente a cero, mientras que en series no estacionarias, lo hace más lentamente (Hyndman & Athanasopoulos, 2021).

Las series que no muestran autocorrelación se conocen como ruido blanco. En estas series, los valores pasados no aportan información para predecir el futuro, ya que no tienen "memoria" (Peña, 2010).

### **3.4. Transformaciones**

Transformaciones simples, como la aplicación de logaritmos, pueden ayudar a estabilizar la varianza en una serie temporal, haciéndola estacionaria. La diferenciación, que consiste en transformar una serie en una secuencia de cambios entre valores consecutivos, es otra técnica común para eliminar tendencias y estacionalidades, estabilizando así la media (Hyndman & Athanasopoulos, 2021). En los modelos estadísticos, también es común suponer que las variables siguen una distribución normal; en caso contrario, transformaciones como la

de Box-Cox pueden hacer que los datos se aproximen más a una distribución normal (Nielsen, 2020).

### 3.5. Pruebas de Raíces Unitarias

Para determinar si es necesaria la diferenciación de una serie, se pueden utilizar pruebas de raíz unitaria, que son pruebas de hipótesis estadísticas diseñadas para evaluar la estacionariedad. Si el p-value es menor a 0.05, se rechaza la hipótesis nula de la prueba, que postula que la serie tiene una raíz unitaria (es decir, no es estacionaria). Entonces, si el p-value es menor a 0.05, se concluye que la serie es estacionaria. Si es mayor, la serie no es estacionaria. (Hyndman & Athanasopoulos, 2021).

### 3.6. Modelos Autorregresivos

En un modelo autorregresivo, se predice la variable de interés utilizando una combinación lineal de sus valores pasados. Estos modelos son flexibles y pueden capturar una amplia gama de patrones en series temporales. Un modelo autorregresivo de orden  $p$  se denota como AR( $p$ ) y sigue la fórmula:

$$Y_t = C + \phi_1 Y_{t-1} + \phi_2 Y_{t-2} + \dots + \phi_p Y_{t-p} + \epsilon_t \quad (3.6.1)$$

Donde:

- $Y_t$  es el valor de la variable en el tiempo  $t$ ,
- $C$  es una constante,
- $\phi_1, \phi_2, \dots, \phi_p$  son los coeficientes autorregresivos,
- $\epsilon_t$  es un término de error o ruido blanco con varianza  $\sigma^2$ .

Nota. Hyndman, R.J., & Athanasopoulos, G. (2021) *Forecasting: principles and practice*, 3rd edition.

### 3.7. Modelos de Media Móvil

Un modelo de media móvil, en lugar de utilizar valores pasados de la variable, emplea errores de pronósticos pasados. Un modelo de media móvil de orden  $q$  se denota como MA( $q$ ) y sigue la fórmula:

$$Y_t = C + \epsilon_t + \theta_1 \epsilon_{t-1} + \dots + \theta_q \epsilon_{t-q} \quad (3.7.1)$$

Donde:

- $Y_t$  es el valor de la variable en el tiempo t,
- $C$  es una constante,
- $\theta_1, \dots, \theta_q$  son los coeficientes de los errores pasados,
- $\epsilon_t$  es el término de error o ruido blanco con varianza  $\sigma^2$ .

### 3.8. Modelos ARIMA

Combinando diferenciación con modelos autorregresivos y de media móvil, se obtiene el modelo ARIMA, que se representa como ARIMA( $p, d, q$ ), donde  $p$  es el orden de la parte autorregresiva,  $d$  es el número de diferenciaciones necesarias y  $q$  es el orden de la parte de media móvil. Algunos casos especiales de los modelos ARIMA incluyen:

- ARIMA( $p, 0, 0$ ): Modelo AR( $p$ )
- ARIMA( $0, 0, q$ ): Modelo MA( $q$ )
- ARIMA( $0, 0, 0$ ): Ruido Blanco

La metodología ARIMA fue formalizada por Box y Jenkins y es ampliamente utilizada para ajustar modelos a series temporales con el fin de mejorar la precisión de los pronósticos.

### 3.9. Árboles de Decisión

Los árboles de decisión son una técnica de modelado predictivo que se utiliza tanto en problemas de clasificación como de regresión. Se basan en dividir repetidamente un conjunto de datos en subconjuntos más pequeños según una serie de pruebas condicionales que maximizan la ganancia informativa (en clasificación) o minimizan la varianza (en regresión). Este enfoque es intuitivo y fácil de interpretar, lo que los ha convertido en una herramienta popular en la ciencia de datos.

#### Funcionamiento y Estructura

Un árbol de decisión comienza en un nodo raíz, donde se realiza la primera división basada en una característica que optimiza un criterio específico, como la ganancia de información o el índice Gini. A partir de ahí, cada rama representa una decisión o una prueba basada en una característica, y cada hoja del árbol representa una predicción o clasificación.

final. A medida que se desciende en el árbol, los datos se van segmentando en grupos cada vez más homogéneos.

### **Ventajas y Desventajas**

Los árboles de decisión son fáciles de entender y visualizar, lo que facilita su interpretación por expertos no técnicos. Además, pueden manejar datos tanto numéricos como categóricos, y no requieren una suposición previa sobre la distribución de los datos. Sin embargo, los árboles de decisión pueden ser propensos a sobreajustarse si no se podan correctamente o si se permite que crezcan demasiado. Este problema puede mitigarse mediante técnicas como el "pruning" (poda) o el uso de conjuntos de árboles, como en el caso de los Random Forests.

### **Aplicación en Series Temporales**

Aunque los árboles de decisión no se utilizan comúnmente para modelar series temporales directamente, pueden ser útiles como componentes en modelos más complejos, como los Random Forests o en técnicas de ensamblado. Estos modelos pueden capturar relaciones no lineales y manejar grandes conjuntos de datos, lo que los hace útiles para ciertos tipos de análisis en series temporales, especialmente cuando se combinan con otras técnicas.

### **Ejemplo de Uso en Predicción**

En un contexto de predicción de series temporales, los árboles de decisión pueden ser aplicados para seleccionar características importantes que afectan el comportamiento de la serie a lo largo del tiempo. Por ejemplo, al incluir variables adicionales como indicadores económicos o factores estacionales, un árbol de decisión puede ayudar a determinar cuáles de estas variables tienen mayor impacto en la predicción de la serie.

*Nota. Hyndman, R.J., & Athanasopoulos, G. (2021) Forecasting: principles and practice, 3rd edition.*

### **3.10. Redes Neuronales LSTM**

Las redes neuronales de memoria a corto y largo plazo (LSTM) son un tipo de red neuronal recurrente (RNN) diseñada para manejar la dependencia temporal en secuencias de datos, lo que las hace especialmente adecuadas para la predicción de series temporales.

Las LSTM son capaces de aprender patrones a largo plazo gracias a su estructura interna de celdas de memoria, que les permite retener información durante períodos prolongados. A diferencia de los modelos tradicionales, las LSTM pueden capturar tanto la tendencia como la estacionalidad de los datos sin necesidad de realizar transformaciones adicionales.

Una ventaja importante de las LSTM es su capacidad para manejar datos no estacionarios y relaciones no lineales de manera efectiva. Sin embargo, entrenar estos modelos puede ser computacionalmente costoso y requiere un gran volumen de datos para obtener resultados precisos.

El desafío técnico que enfrentan los RNN es cómo entrenarlos efectivamente. Se ha mostrado que los modelos de RNN presentan inconvenientes en la actualización de los pesos resultando o bien en cambios a pesos muy pequeños (desvanecimiento del gradiente) o muy grandes (desbordamiento del gradiente). Los modelos LSTM superan este desafío por su diseño.

*Nota. Brownlee, J. (2017). Long Short-Term Memory Networks with Python. Develop Sequence Prediction Models with Deep Learning.*

### 3.11. XGBoost

XGBoost (Extreme Gradient Boosting) es un algoritmo de machine learning basado en árboles de decisión que utiliza una técnica de boosting para mejorar la precisión de las predicciones. Aunque originalmente no fue diseñado específicamente para series temporales, XGBoost se ha adaptado con éxito para esta tarea gracias a su capacidad para manejar datos tabulares y su flexibilidad para capturar relaciones no lineales y complejas en los datos.

El proceso de modelado de series temporales con XGBoost generalmente implica la transformación de los datos en una estructura supervisada, donde las características de las

observaciones pasadas se utilizan para predecir valores futuros. Una ventaja clave de XGBoost es su capacidad para manejar grandes volúmenes de datos y su eficiencia en términos de tiempo de entrenamiento. Además, XGBoost permite la inclusión de variables adicionales que pueden influir en las predicciones, como factores estacionales o eventos externos, mejorando la robustez del modelo.

Sin embargo, XGBoost puede no capturar tan fácilmente la estacionalidad o las dependencias a largo plazo en los datos, como lo harían modelos especializados en series temporales, a menos que estas características se incluyan explícitamente en las variables de entrada. A pesar de esto, XGBoost sigue siendo una herramienta poderosa en la predicción de series temporales, especialmente cuando se trata de datos con relaciones complejas y no lineales.

*Nota. Chen, T., & Guestrin, C. (2016, August). XGBoost: A scalable tree boosting system. In Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining (pp. 785-794).*

### **3.12. Prophet**

Prophet es una herramienta desarrollada por Facebook para la predicción de series temporales que se destaca por su simplicidad y efectividad al manejar datos con tendencias, estacionalidad y efectos de días festivos. Prophet utiliza un modelo aditivo, donde la tendencia, la estacionalidad y los efectos de eventos especiales se modelan como componentes separados que se suman para formar la predicción final.

Una de las principales ventajas de Prophet es su facilidad de uso, ya que requiere poca configuración por parte del usuario para producir pronósticos precisos. El modelo está diseñado para ser robusto frente a datos faltantes y cambios en la tendencia, lo que lo hace especialmente útil para aplicaciones en entornos comerciales y de negocios. Además, Prophet permite la incorporación de múltiples estacionalidades (diaria, semanal, anual), así como la inclusión de factores externos que puedan afectar las series temporales, como días festivos o eventos extraordinarios.

No obstante, aunque Prophet es eficaz para capturar patrones estacionales y tendencias a largo plazo, su simplicidad puede ser una limitación cuando se trata de modelar series temporales con relaciones no lineales complejas o cuando los datos presentan patrones que no se ajustan bien a un modelo aditivo. A pesar de esto, Prophet es una excelente opción para la predicción rápida y efectiva de series temporales, especialmente cuando se requiere un modelo interpretativo y fácil de ajustar.

*Nota. Sean J. Taylor and Benjamin Letham (2017). Forecasting at Scale.*

### **3.13. Algoritmos automáticos (TPOTRegressor)**

TPOT (Tree-based Pipeline Optimization Tool) es una herramienta de optimización automática de pipelines de machine learning que utiliza algoritmos genéticos para encontrar las mejores combinaciones de modelos y parámetros. TPOTRegressor es una extensión de TPOT que se enfoca en problemas de regresión, incluyendo la predicción de series temporales.

Algoritmos genéticos son una técnica de búsqueda y optimización basada en principios de la evolución biológica. Estos algoritmos trabajan creando una población de posibles soluciones (en este caso, pipelines de machine learning) y luego aplican operadores inspirados en la genética, como la selección, el cruce (crossover) y la mutación para evolucionar las soluciones hacia mejores resultados.

#### **Funcionamiento y Ventajas**

TPOTRegressor automatiza el proceso de selección y ajuste de modelos, explorando múltiples combinaciones de algoritmos y parámetros para identificar el mejor pipeline para los datos. Este enfoque puede incluir modelos de regresión lineales y no lineales, árboles de decisión, y técnicas avanzadas como XGBoost y Random Forest. La ventaja principal de TPOT es su capacidad para encontrar automáticamente el mejor modelo sin necesidad de intervención manual, lo que ahorra tiempo y esfuerzo en la fase de modelado.

Además, TPOTRegressor ofrece una solución robusta para problemas de predicción de series temporales al permitir la inclusión de técnicas de preprocesamiento, selección de características y modelado en un solo pipeline. Su uso de algoritmos genéticos permite una

exploración exhaustiva del espacio de modelos, lo que puede conducir a la identificación de combinaciones de modelos y parámetros que quizás no se considerarían en un enfoque manual.

### **Limitaciones y Consideraciones**

A pesar de sus ventajas, TPOTRegressor puede ser computacionalmente intensivo debido a la búsqueda exhaustiva de modelos y parámetros. El proceso de optimización puede llevar un tiempo considerable, especialmente con grandes conjuntos de datos o modelos complejos. Además, la interpretabilidad del modelo final puede verse comprometida, ya que TPOTRegressor puede combinar múltiples técnicas y parámetros que son difíciles de desglosar.

En el contexto de series temporales, es importante tener en cuenta que TPOTRegressor requiere una adecuada transformación de los datos en una estructura supervisada para realizar predicciones efectivas. Esto implica la creación de características basadas en observaciones pasadas y la incorporación de variables adicionales relevantes para mejorar la precisión del modelo.

TPOTRegressor es una herramienta poderosa para la automatización del proceso de modelado en problemas de predicción de series temporales. Su capacidad para optimizar automáticamente pipelines de machine learning ofrece una solución eficiente para identificar modelos precisos, aunque el costo computacional y la complejidad del modelo final deben ser considerados en su implementación.

*Nota. Randal S. Olson and Jason H. Moore (2016). TPOT: A Tree-based Pipeline Optimization Tool for Automating Machine Learning.*

#### **3.14. Separación de los datos**

Un modelo que simplemente se entrena y se evalúa sobre datos conocidos, tendría una puntuación perfecta pero no podría predecir nada útil en datos aún no vistos. Esta situación se llama sobreajuste (Overfitting). Además, de acuerdo con la documentación de la librería de Python Scikit Learn, al evaluar diferentes “hiperparámetros” en la búsqueda del mejor ajuste del modelo, existe el riesgo nuevamente de sobreajustar el mismo a los datos de prueba perdiendo

de esta forma el rendimiento deseado de generalización y devolviendo métricas de performance que no son realistas. Por este motivo es que surge el “conjunto de validación” (la tercera división del conjunto de datos). De esta manera, se entrena sobre el set de entrenamiento, se evalúa y se realizan ajustes sobre el set de validación y finalmente se evalúa en el conjunto de testeo.

### **3.15. Validación cruzada**

De todas formas, al particionar el conjunto original en tres partes, aún queda el efecto de que, según como se haya hecho la división, pueden variar los resultados. Por este motivo, se han generado distintos procedimientos como solución a este problema, llamados Validación Cruzada (Cross-Validation). En su versión más básica y tomada como ejemplo, el conjunto de entrenamiento se divide en  $k$  conjuntos más pequeños y se procede en primer lugar entrenando un modelo utilizando  $k-1$  de los conjuntos como set de entrenamiento y el modelo restante se valida con la parte restante de los datos.

### **3.16. Validación cruzada de datos de series temporales**

Los datos de series temporales se caracterizan por la secuencia entre observaciones cercanas en el tiempo. Por lo tanto, es muy importante evaluar nuestro modelo para datos de series temporales sobre las observaciones "futuras" menos parecidas a las que se utilizan para entrenar el modelo. Para lograr esto, existe una variación conocida como "TimeSeriesSplit" (en 33 la documentación de Scikit Learn de Python) la cual es una variación de la validación cruzada y proporciona una solución para los modelados de las series de tiempo. Debe considerarse que, en este tipo de validación, los conjuntos de entrenamiento sucesivos son superconjuntos de los anteriores.

### **3.17. Métricas de performance**

La evaluación del rendimiento de los modelos de predicción de series temporales es fundamental para medir su precisión y eficacia. A continuación, se presentan las métricas más comunes utilizadas para este propósito:

#### **3.17.1. Error Medio Absoluto (MAE)**

El Error Medio Absoluto (MAE, por sus siglas en inglés) mide la magnitud promedio de los errores en un conjunto de predicciones, sin considerar su dirección. Se calcula como la media de las diferencias absolutas entre los valores predichos y los valores reales.

$$MAE = \frac{1}{n} \sum_{i=1}^n |\gamma_i - \hat{\gamma}_i| \quad (3.17.1)$$

Donde  $\gamma_i$  es el valor real,  $\hat{\gamma}_i$  es el valor predicho y  $n$  es el número total de observaciones.

El MAE proporciona una medida clara de cuán lejos, en promedio, están las predicciones de los valores reales.

### 3.17.2. Error Cuadrático Medio (MSE)

El Error Cuadrático Medio (MSE, por sus siglas en inglés) cuantifica el promedio de los cuadrados de los errores, es decir, la diferencia entre los valores predichos y los valores reales. Se calcula como:

$$MSE = \frac{1}{n} \sum_{i=1}^n (\gamma_i - \hat{\gamma}_i)^2 \quad (3.17.2)$$

El MSE penaliza más los errores grandes debido a la cuadratura, proporcionando una medida que puede ser sensible a outliers. Esto lo hace útil para identificar modelos que podrían estar cometiendo grandes errores en la predicción.

### 3.17.3. Raíz del Error Cuadrático Medio (RMSE)

La Raíz del Error Cuadrático Medio (RMSE, por sus siglas en inglés) es la raíz cuadrada del MSE y proporciona una medida de la magnitud promedio de los errores en las mismas unidades que los datos originales. Se calcula como:

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (\gamma_i - \hat{\gamma}_i)^2} \quad (3.17.3)$$

El RMSE ofrece una interpretación intuitiva de los errores promedio y es especialmente útil para comparar modelos en situaciones donde las unidades de los datos son importantes.

### 3.17.4. Error Porcentual Absoluto Medio (MAPE)

El Error Porcentual Absoluto Medio (MAPE, por sus siglas en inglés) mide la precisión de las predicciones en términos porcentuales. Se calcula como:

$$MAPE = \frac{100\%}{n} \sum_{i=1}^n \left| \frac{\hat{y}_i - y_i}{y_i} \right| \quad (3.17.4)$$

El MAPE proporciona una perspectiva relativa del error, permitiendo una comparación fácil entre diferentes conjuntos de datos o modelos. Sin embargo, puede ser problemático en presencia de valores reales muy cercanos a cero, ya que puede generar valores extremadamente grandes.

## 4. Origen, exploración y descripción de los datos

### 4.1. Origen y consideraciones de los datos

Como se mencionó en los objetivos y alcances, debido a la naturaleza confidencial de los datos, se ha optado por una estrategia de protección de la información. Esta estrategia incluye la modificación de los nombres originales de las variables y la transformación de los valores de las mismas a través de un factor de conversión fijo. Esta transformación asegura que las relaciones proporcionales entre los datos se mantengan intactas, permitiendo un análisis válido sin comprometer la confidencialidad.

#### Datos internos:

- Variables objetivo ventas de juguetes:
  - Ventas totales en el mercado argentino.
  - Ventas por segmento geográfico "M"
  - Ventas por segmento geográfico "C"
- Frecuencia temporal de captura de los datos: mensual.
- Horizonte temporal: Enero 2007 - Junio 2024 (210 observaciones).

#### Datos externos (ver detalle en [Anexo I](#)):

- Información de agentes externos, consultoras, entes gubernamentales y diversas organizaciones que se dedican a la recopilación y divulgación de datos macroeconómicos. Variables externas de distintas fuentes gubernamentales y consultoras de Estados Unidos y de México (Banco Mundial, FMI, Reserva

Federal de St Louis, Bureau of Economic Analysis, INEGI México, Yahoo Finance, Thomson Reuters).

- Frecuencia temporal de captura de los datos: mensual / trimestral. Los datos trimestrales mencionados en Anexo I (GDP USA/MX y datos industriales) se transformaron a una serie mensual promediando las ventas del último trimestre conocido y asignando ese valor a cada mes. Esta aproximación se justifica por la falta de datos mensuales detallados y permite suavizar fluctuaciones a corto plazo, conservando la tendencia general para el análisis y predicción a 12 meses. Reconocemos que este método asume una distribución uniforme de las ventas dentro del trimestre, lo cual es una limitación. Sin embargo, proporciona una base razonable para el estudio.
- Horizonte temporal: Enero 2007 - Junio 2024 (210 observaciones).

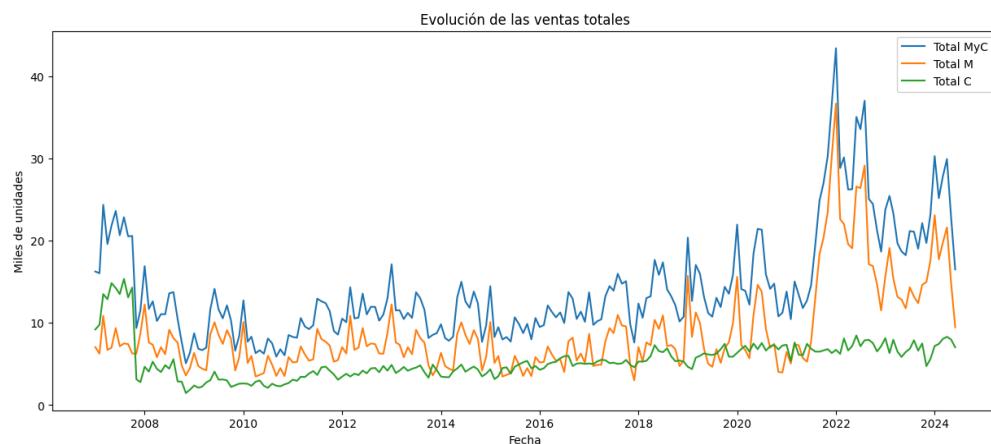
#### **4.2. Exploración y descripción de los datos**

El desarrollo del análisis correspondiente se realizó utilizando como soporte el entorno de Jupyter Notebook a través de Visual Studio Code, empleando Python como lenguaje de programación. El mismo puede ser consultado y reproducido en el siguiente repositorio Github ([Tesis-cdibono](#)).

Los gráficos y tablas presentados a continuación se muestran con periodicidad mensual y en cantidad de unidades vendidas.

En la fase inicial de exploración de datos, se realizó un análisis exhaustivo para identificar posibles valores faltantes en el conjunto de datos. Los resultados revelaron la ausencia de datos faltantes, asegurando la integridad y completitud de la información para el análisis posterior.

A continuación observamos la evolución de las cantidades vendidas en el período Enero 2007 - Junio 2024 a nivel total y por regiones “M” y “C”.

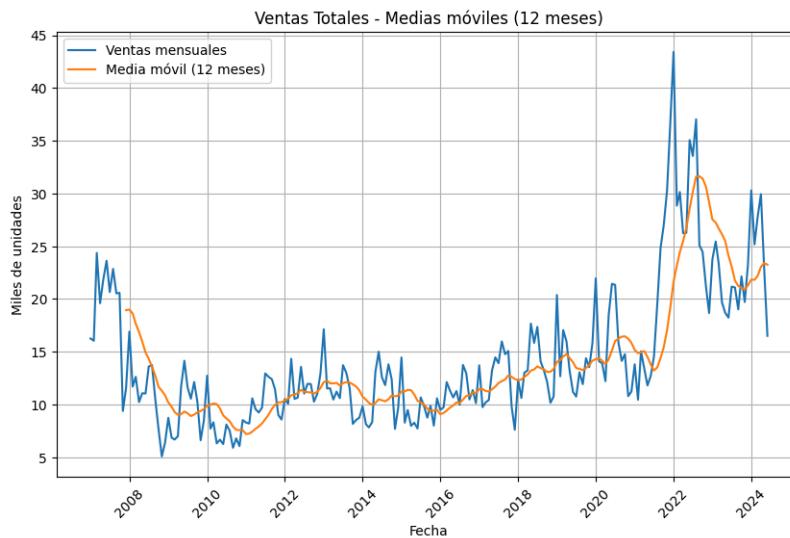


**Figura 2:** Ventas mensuales desde enero 2007 hasta junio 2024, segregando por regiones "M" y "C".

*Nota. Elaboración propia utilizando Jupyter Notebook en Visual Studio Code.*

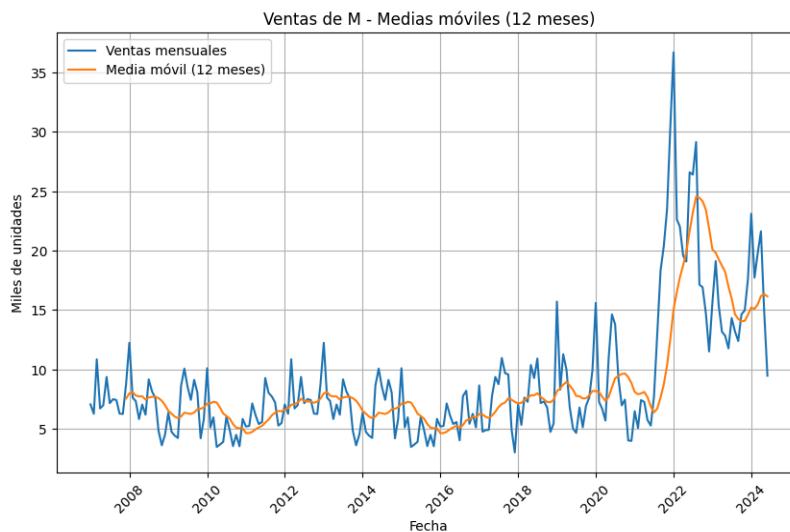
También se observan las medias móviles con el objetivo de suavizar las fluctuaciones de corto plazo y poder observar la tendencia subyacente en la serie. La idea es destacar el comportamiento a largo plazo de la serie sin el ruido de las variaciones aleatorias.

**Figura 3:**



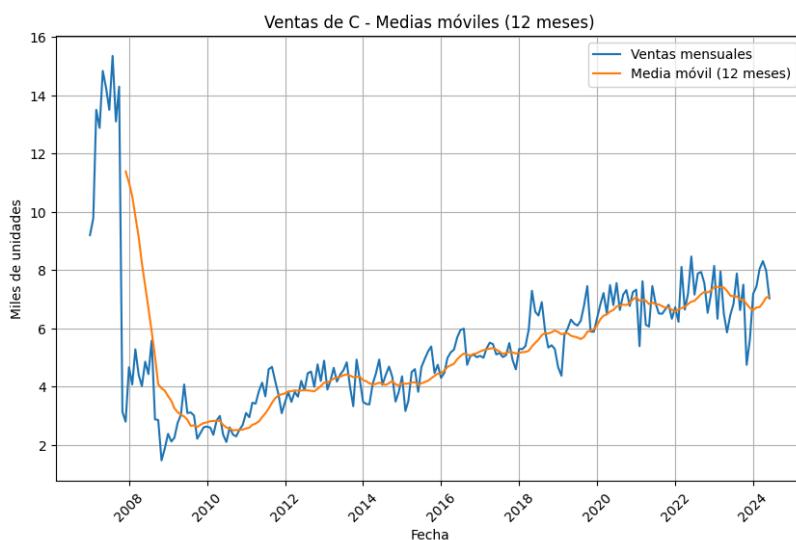
*Nota. Elaboración propia utilizando Jupyter Notebook en Visual Studio Code.*

**Figura 4:**



*Nota. Elaboración propia utilizando Jupyter Notebook en Visual Studio Code.*

*Figura 5:*

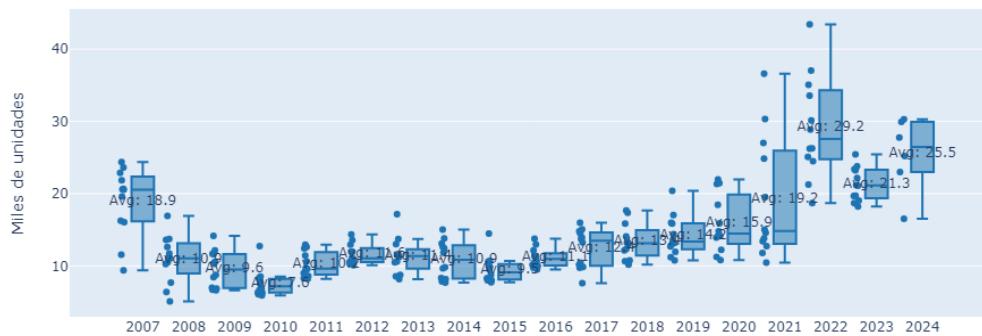


*Nota. Elaboración propia utilizando Jupyter Notebook en Visual Studio Code.*

En la inspección inicial de los datos correspondientes a la región C, se identifica una desviación significativa en el año 2007 en comparación con el resto del período analizado. Tras consultar con expertos en la materia, se determinó que esta anomalía se debe a la consolidación de ventas de múltiples unidades de negocio en la región C durante ese año, en contraste con la estructura de una sola unidad de negocio que prevalece en la actualidad. A pesar de esta singularidad, se decidió incluir estos datos en el conjunto de datos final, ya que los patrones estacionales inherentes a la serie temporal pueden ser aprovechados por los algoritmos de predicción implementados.

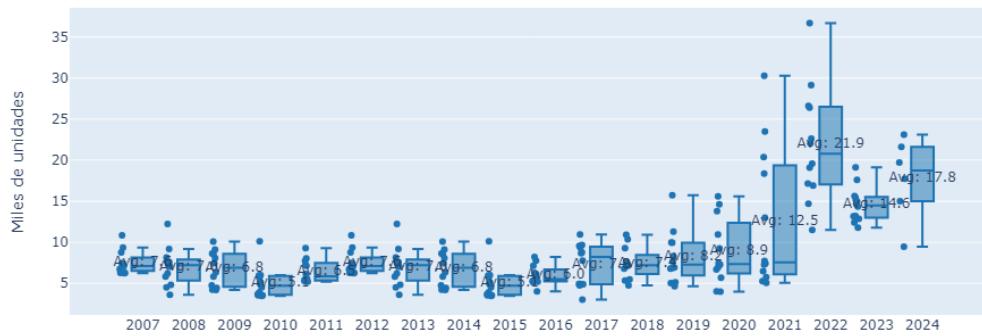
Con el fin de proporcionar una visión detallada de la distribución y variabilidad de las ventas a lo largo del tiempo, se presentarán gráficos de tipo boxplot. Estos gráficos mostrarán la evolución anual de cada serie temporal desde 2007 hasta 2024. Los boxplots permitirán identificar la mediana, los cuartiles, los valores atípicos y la dispersión de las ventas para cada año, facilitando la detección de cambios en la distribución y la identificación de posibles anomalías o tendencias a lo largo del período analizado.

Figura 6: Distribución de las ventas totales por año desde 2007-2024.



Nota. Elaboración propia utilizando Jupyter Notebook en Visual Studio Code.

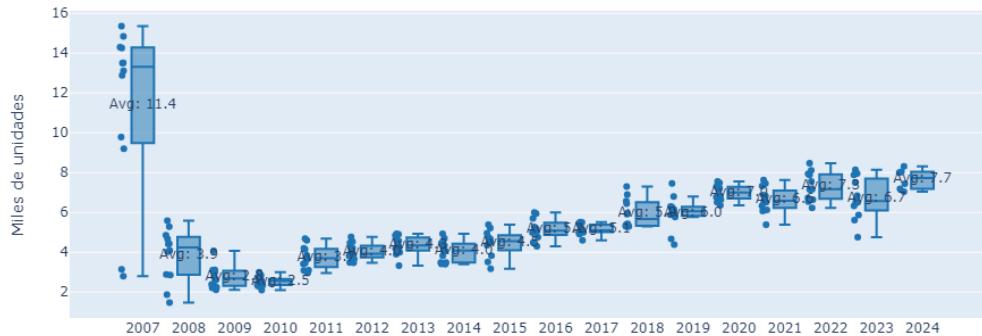
Figura 7: Distribución de las ventas de M por año desde 2007-2024.



Nota. Elaboración propia utilizando Jupyter Notebook en Visual Studio Code.

Se observa un incremento significativo en la media, así como en los valores máximos y el tercer cuartil de las cantidades vendidas durante los años 2021 y 2022. Este aumento sustancial puede atribuirse a diversos factores interrelacionados, incluyendo la recuperación post-pandemia de COVID-19 en el sector industrial y un incremento notable en los precios del mercado local, impulsado por una demanda insatisfecha.

Figura 8: Distribución de las ventas de C por año desde 2007-2024.



Nota. Elaboración propia utilizando Jupyter Notebook en Visual Studio Code.

Como se comentó previamente, para el caso de C se observa una anomalía en las ventas del año 2007, donde la compañía tenía 6 unidades productivas, desde el mes de noviembre 2007 solo queda la unidad que luego se mantendrá hasta la actualidad.

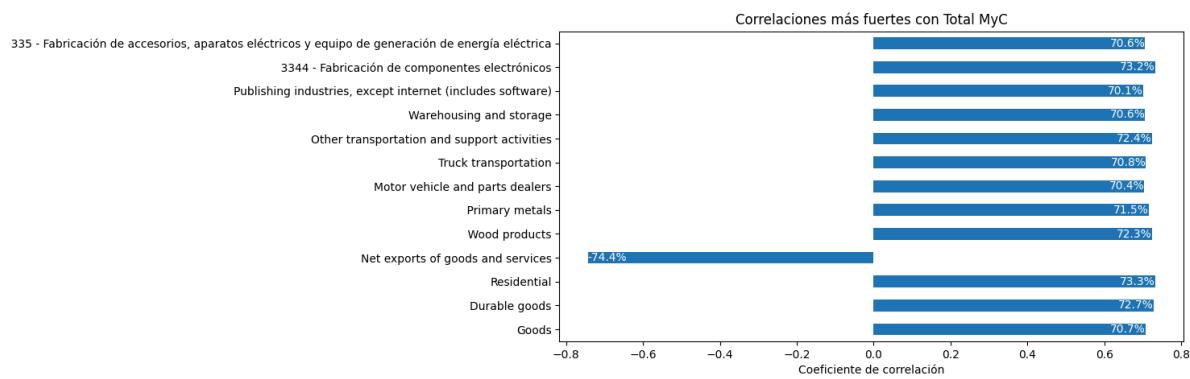
Las observaciones iniciales derivadas del análisis exploratorio de los datos revelan las siguientes características distintivas:

- Variabilidad de las series temporales: los datos de ventas correspondientes a la unidad productiva C exhiben una notable estabilidad a lo largo del período analizado, mostrando una menor dispersión en comparación con la unidad de negocio M. Por el contrario, la unidad de negocio M presenta una mayor variabilidad en sus datos de ventas, indicando fluctuaciones más pronunciadas a lo largo del tiempo.
- Medidas de tendencia central: se observa un incremento sustancial en las medidas de tendencia central (media, mediana) durante los últimos cinco años, tanto para la unidad de negocio M como para la unidad productiva C. Este aumento sugiere una tendencia alcista en las ventas de ambas entidades durante el período reciente.

El conjunto de datos incluye 346 variables explicativas, lo que requiere un análisis exhaustivo para determinar su relevancia en la predicción de las variables objetivo. Con el fin de simplificar la presentación y enfocarnos en los factores más influyentes, se seleccionaron las variables con los coeficientes de correlación más elevados. Para calcular estos coeficientes, se utilizó la función `corr()` de la biblioteca pandas en Python.

A continuación, se presentan los resultados de este análisis (variables con mayor correlación comparadas en valores absolutos) para el total de ventas de la compañía.

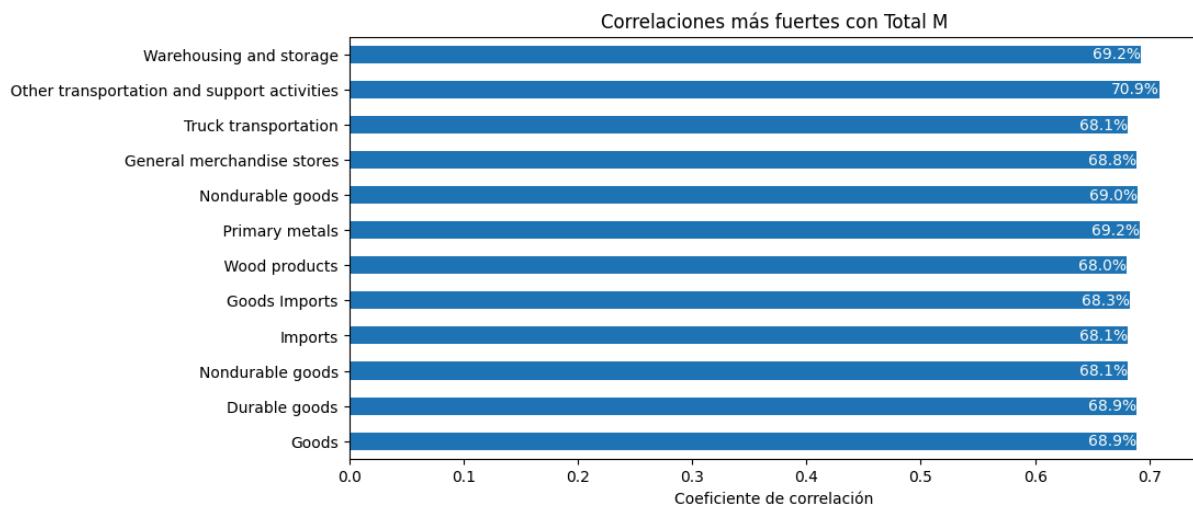
*Figura 9:*



*Nota. Elaboración propia utilizando Jupyter Notebook en Visual Studio Code.*

Variables con mayor correlación (comparadas en valores absolutos) para el total de ventas de la compañía en la unidad de negocio M.

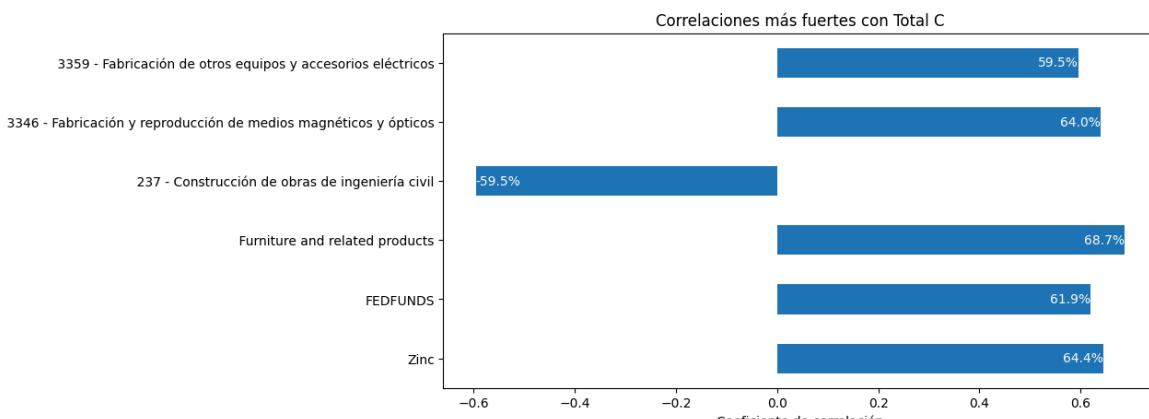
*Figura 10:*



*Nota. Elaboración propia utilizando Jupyter Notebook en Visual Studio Code.*

Variables con mayor correlación (comparadas en valores absolutos) para el total de ventas de la compañía en la unidad productiva C.

*Figura 11:*



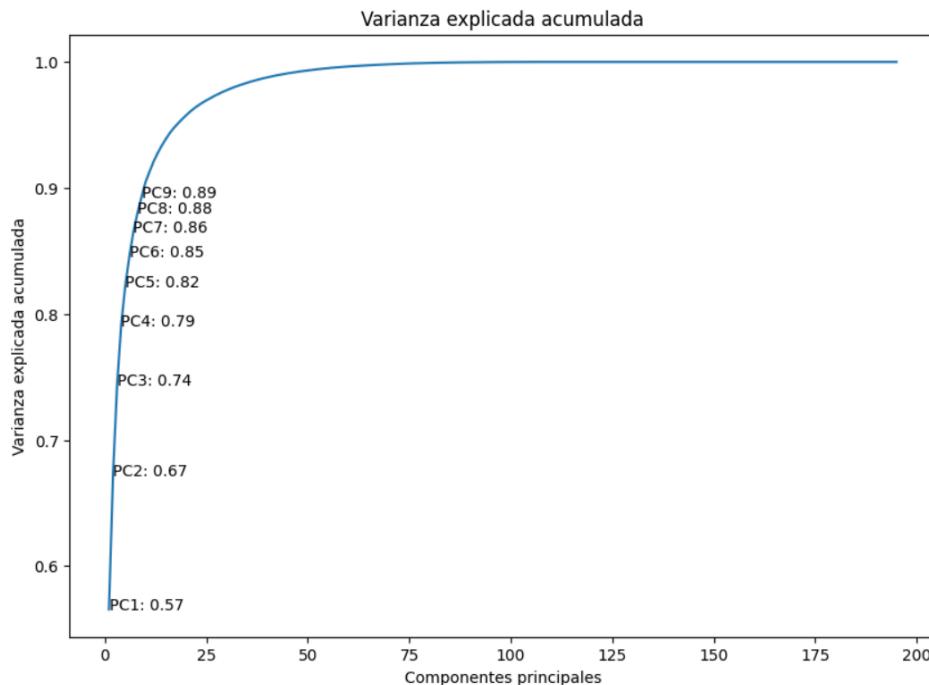
*Nota. Elaboración propia utilizando Jupyter Notebook en Visual Studio Code.*

La evaluación de las relaciones lineales entre las variables explicativas y las variables objetivo, mediante el cálculo de coeficientes de correlación, demostró que no existen asociaciones fuertes en el conjunto de datos. Los valores máximos de correlación observados, cercanos a 0.70, sugieren una influencia moderada, pero no una dependencia lineal significativa.

#### 4.3. Análisis de Componentes Principales (PCA)

Siendo que se cuenta con una cantidad de variables explicativas elevada (346), se decide aplicar PCA para intentar reducir la dimensionalidad.

Figura 12:



*Nota. Elaboración propia utilizando Jupyter Notebook en Visual Studio Code.*

El Análisis de Componentes Principales (PCA) se incluyó en la tesis como parte de una exploración exhaustiva de las técnicas de reducción de dimensionalidad. Dado el alto número de variables explicativas (346), se consideró fundamental evaluar la posibilidad de simplificar el modelo sin perder información relevante.

El PCA se realizó para:

- Evaluar la viabilidad de reducir la complejidad del modelo.
- Identificar patrones y relaciones ocultas en los datos.

- Comprobar la cantidad de varianza que se podía mantener con una menor cantidad de variables.

Los gráficos presentados demostraron que las primeras 9 componentes capturaron aproximadamente el 90% de la variabilidad, lo que inicialmente sugería una reducción significativa de la dimensionalidad.

A pesar de los resultados prometedores del PCA en términos de variabilidad capturada, se decidió no incluirlo en el modelo final debido a la pérdida de interpretabilidad que conlleva.

En un contexto de predicción de ventas, la interpretabilidad del modelo es crucial para la toma de decisiones comerciales. Los gerentes y analistas necesitan comprender cómo las variables influyen en las predicciones para poder actuar en consecuencia. Las componentes principales, al ser combinaciones lineales de las variables originales, dificultan la interpretación de los resultados.

Esta pérdida de interpretabilidad se consideró un factor crítico, especialmente en un entorno empresarial donde la transparencia y la comprensión son esenciales.

#### 4.4. Descomposición de la serie

Con el fin de comprender las características subyacentes de las series temporales, se desarrolla en esta sección la descomposición para la serie total de ventas en mercado argentino (MyC).

Se utilizó la técnica de descomposición estacional aditiva mediante el método Seasonal Decompose de la librería `statsmodels` en Python. Este método descompone la serie temporal en tres componentes principales:

1. **Tendencia:** Representa el comportamiento general de la serie a lo largo del tiempo, eliminando fluctuaciones a corto plazo.
2. **Estacionalidad:** Muestra los patrones recurrentes que se repiten a intervalos regulares.
3. **Residuos:** Son las irregularidades o el "ruido" que no se explican por la tendencia ni la estacionalidad.

El modelo aditivo supone que la serie temporal puede ser descompuesta como la suma de estas tres componentes:

$$Y_{(t)} = T_{(t)} + S_{(t)} + R_{(t)} \quad (4.1.1)$$

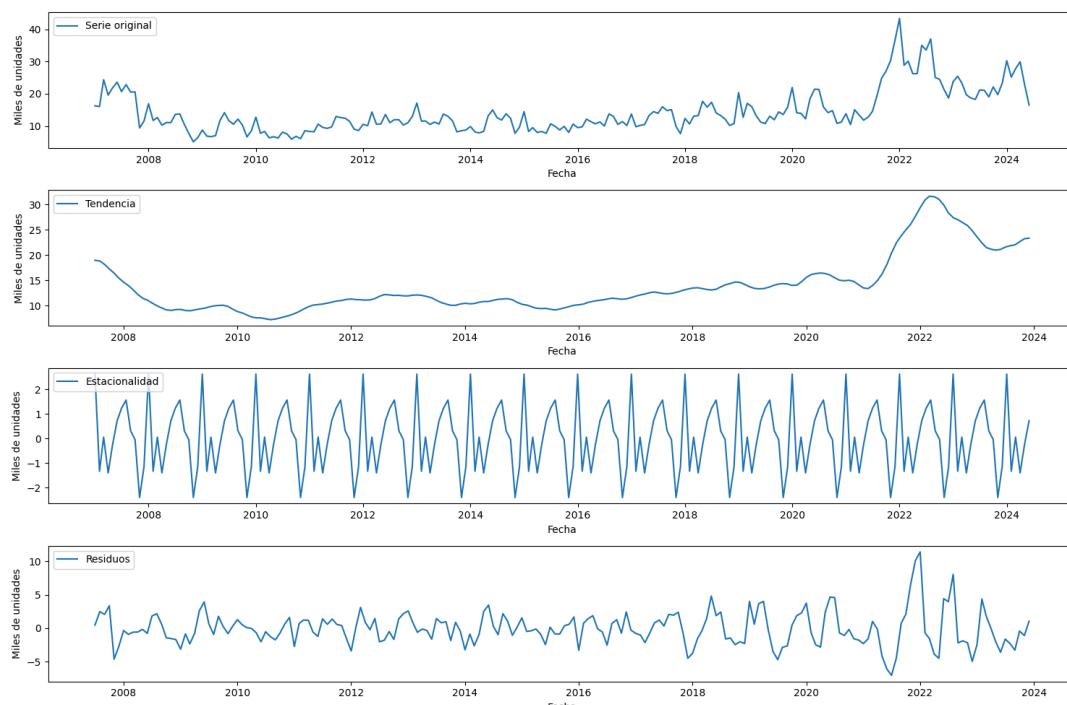
Donde  $Y_{(t)}$  es el valor de la serie en el tiempo  $t$ ,  $T_{(t)}$  es la tendencia,  $S_{(t)}$  es la estacionalidad, y  $R_{(t)}$  son los residuos.

*Nota. Hyndman, R.J., & Athanasopoulos, G. (2021) Forecasting: principles and practice, 3rd edition.*

El periodo de estacionalidad se estableció en 12 meses, ya que los datos corresponden a series temporales mensuales, y se busca capturar los patrones estacionales anuales.

Esta descomposición permite observar separadamente cómo se comporta la tendencia, la estacionalidad y las fluctuaciones no explicadas (residuos), lo que facilita la interpretación de la serie temporal.

*Figura 13:*



*Nota. Elaboración propia utilizando Jupyter Notebook en Visual Studio Code.*

La serie original muestra una fluctuación considerable a lo largo del tiempo, con un incremento notable en la variabilidad y el nivel de ventas a partir de aproximadamente 2021.

Este aumento sugiere un cambio significativo en el comportamiento de las ventas durante este período.

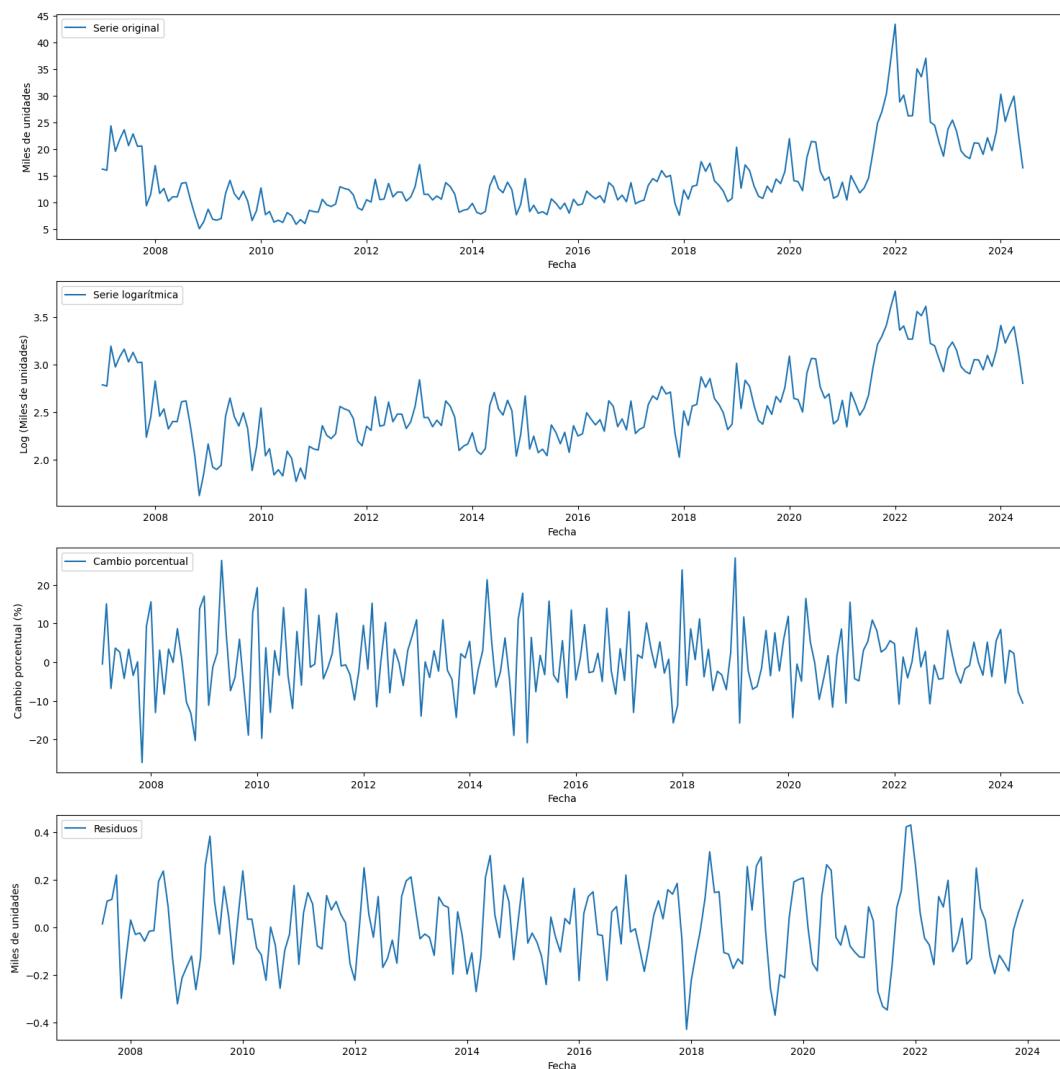
La componente de tendencia revela un patrón de crecimiento gradual desde 2008 hasta aproximadamente 2021, seguido de un aumento abrupto y pronunciado. Este cambio en la tendencia indica una aceleración significativa en las ventas a partir de 2021, lo que podría estar asociado con factores externos o internos específicos de la empresa.

La componente de estacionalidad muestra un patrón repetitivo y consistente a lo largo del tiempo, con ciclos anuales claros. Esta estacionalidad sugiere la presencia de factores recurrentes que afectan las ventas, como variaciones en la demanda relacionadas con estaciones del año, eventos promocionales o ciclos económicos.

La componente de residuos representa la variabilidad no explicada por la tendencia y la estacionalidad. Se observa una dispersión relativamente uniforme de los residuos a lo largo del tiempo, con algunos picos aislados. El incremento en la magnitud de los residuos a partir de 2021 coincide con el aumento en la variabilidad de la serie original y el cambio en la tendencia. Esto sugiere que factores no modelados están contribuyendo a la variabilidad de las ventas durante este período.

A continuación se observa la serie con cambio logarítmico (serie original, serie logarítmica, cambio porcentual y residuos).

Figura 14:



Nota. Elaboración propia utilizando Jupyter Notebook en Visual Studio Code.

La transformación logarítmica ha suavizado las fluctuaciones en la serie, especialmente durante el período de alta variabilidad a partir de 2021. Las observaciones son similares en cuanto a la tendencia y los residuos.

La magnitud de los residuos es relativamente pequeña en comparación con la escala de la serie original y la serie logarítmica, lo que sugiere que el modelo (implícito en estas transformaciones) captura una parte significativa de la variabilidad.

## 5. Implementación de algoritmos

A continuación se presentarán los modelos desarrollados con las características propias de cada uno de ellos, parametrizaciones, ajustes, métricas de performance, comparaciones, problemas, soluciones y mejoras encontradas. Finalmente se presentará un resumen y futuras aplicaciones o mejoras.

Como premisa se trabajará con los datos de las series temporales de ventas totales (“Total MyC”) y con las ventas segmentadas por región geográfica, M (“Total M”) y C (“Total C”).

### 5.1. Test de raíces unitarias

En la presente sección, se aborda el análisis de la estacionariedad de las series temporales mediante la aplicación de pruebas de raíces unitarias. La decisión de iniciar la implementación de algoritmos con este análisis se fundamenta en la relevancia crítica de la estacionariedad para la modelización y predicción de series temporales.

La estacionariedad, definida como la constancia de las propiedades estadísticas de una serie a lo largo del tiempo, es una condición fundamental para la aplicación de muchos modelos de series temporales, como los modelos ARIMA. La presencia de raíces unitarias, indicativa de no estacionariedad, puede llevar a predicciones espurias y a la identificación de relaciones aparentes que no reflejan la dinámica real de los datos.

El análisis de estacionariedad de las series temporales se inició mediante la aplicación de la prueba de Dickey-Fuller aumentada (ADF), una técnica estadística fundamental para determinar la presencia de raíces unitarias. Esta prueba evalúa la hipótesis nula de que la serie temporal posee una raíz unitaria, lo que implica no estacionariedad.

La decisión de rechazar o no la hipótesis nula se basa en el valor p (p-value) obtenido. Específicamente: Si el valor p es inferior al nivel de significancia predefinido ( $\alpha = 0.05$ ), se rechaza la hipótesis nula. Esto indica que la serie temporal es estacionaria, ya que proporciona evidencia estadística suficiente para descartar la presencia de una raíz unitaria. Por el contrario, si el valor p es superior a 0.05, no se rechaza la hipótesis nula, lo que sugiere que la serie temporal no es estacionaria.

Resultados para '**Total MyC**':

- Estadístico ADF: -3.074277
- Valor p: 0.028516
- Valores críticos:
  - 1%: -3.462,
  - 5%: -2.876,
  - 10%: -2.574

Conclusión: La serie es estacionaria ( $p < 0.05$ ).

Resultados para '**Total M**':

- Estadístico ADF: -4.114483
- Valor p: 0.000916
- Valores críticos:
  - 1%: -3.462,
  - 5%: -2.875,
  - 10%: -2.574

Conclusión: La serie es estacionaria ( $p < 0.05$ ).

Resultados para '**Total C**':

- Estadístico ADF: -1.476946
- Valor p: 0.544884
- Valores críticos:
  - 1%: -3.464,
  - 5%: -2.876,
  - 10%: -2.575

Conclusión: La serie no es estacionaria ( $p > 0.05$ ).

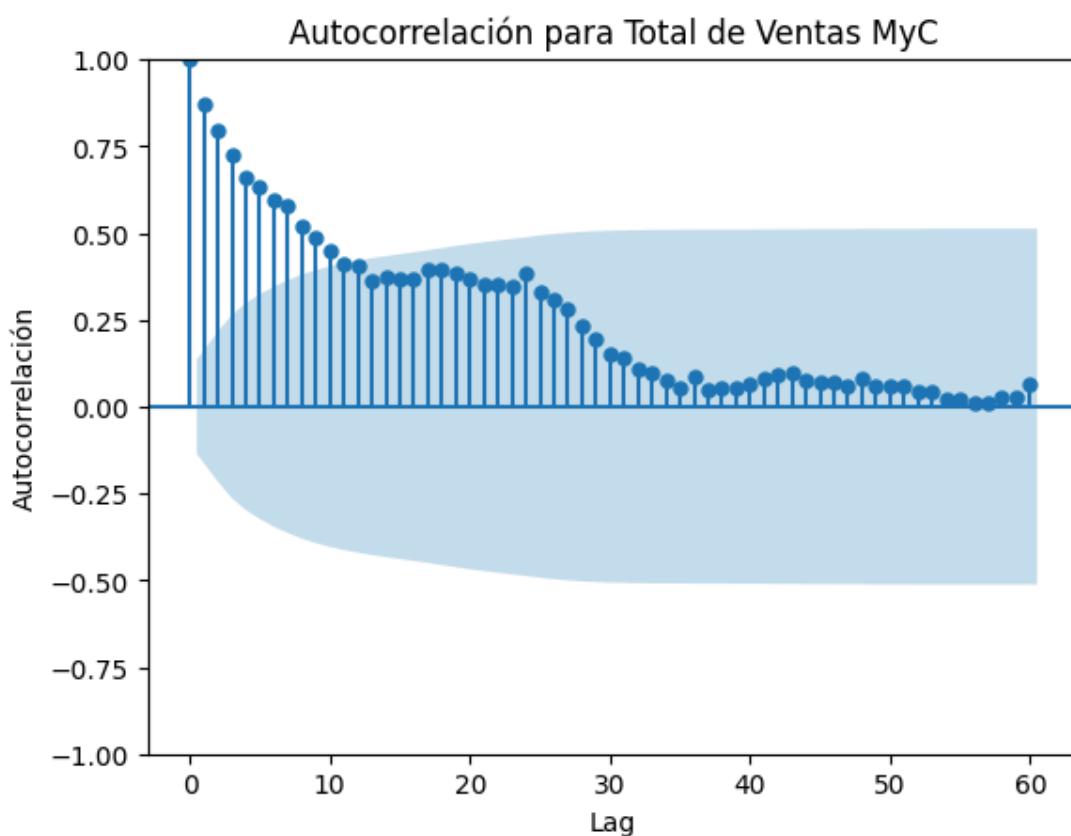
**5.2. Función de Autocorrelación (ACF):**

Propósito principal de la función de autocorrelación es identificar patrones de autocorrelación en la serie.

Determinar el orden de los componentes autorregresivos (AR) y de media móvil (MA) en modelos ARIMA.

Muestra las correlaciones entre una serie y sus valores rezagados (lags). En este caso, al tener una frecuencia mensual, el gráfico de ACF ayuda a identificar patrones de correlación a lo largo del tiempo.

Figura 15:



Nota. Elaboración propia utilizando Jupyter Notebook en Visual Studio Code.

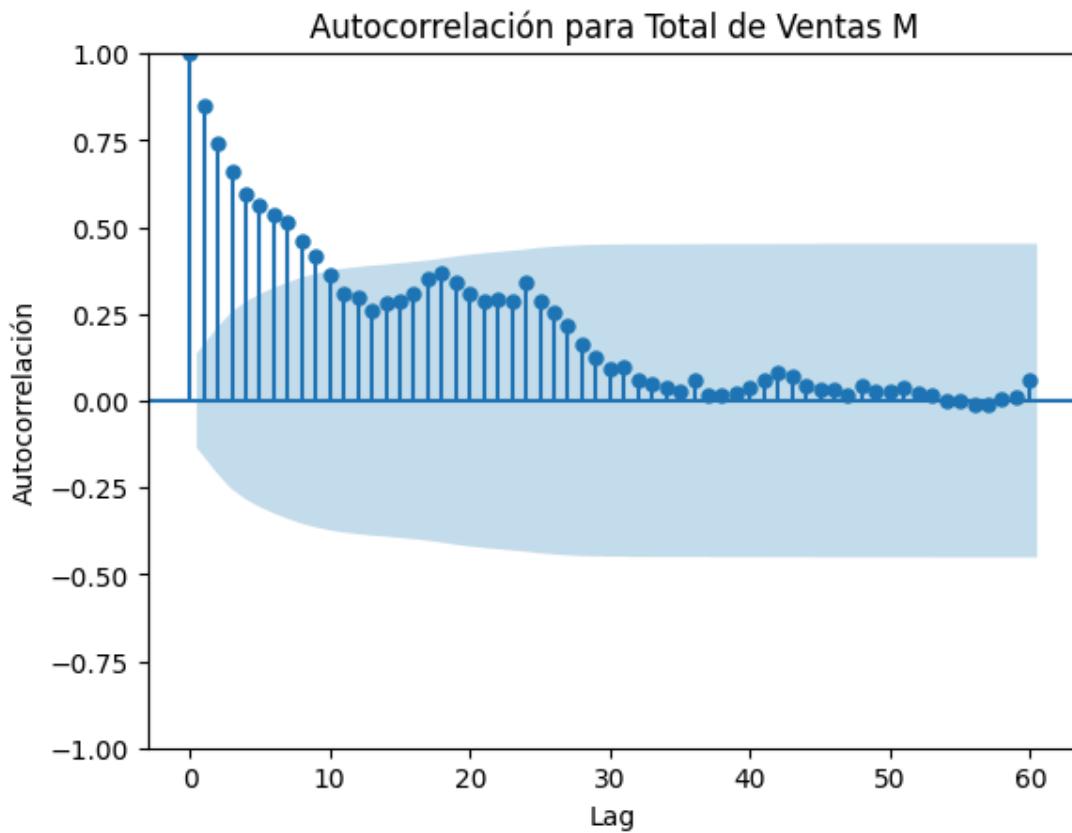
Para la serie Total MyC se observa un decaimiento gradual y lento de los coeficientes de autocorrelación a medida que aumenta el rezago (lag). Esto indica que existe una fuerte dependencia temporal en la serie, con correlaciones significativas presentes incluso en rezagos relativamente grandes.

Varios rezagos iniciales muestran coeficientes de autocorrelación que exceden los límites de confianza (región sombreada azul). Esto sugiere que las observaciones pasadas tienen una influencia significativa en las observaciones presentes. La persistencia de coeficientes significativos en rezagos más largos indica que la dependencia temporal se extiende a lo largo de varios períodos.

La ACF de la serie Total MyC revela una fuerte dependencia temporal y un decaimiento lento, lo que sugiere no estacionariedad o una fuerte tendencia. Según la prueba de raíces unitarias realizada previamente la serie es estacionaria.

Conclusión los modelos AR podrían ser adecuados para capturar la autocorrelación presente en la serie.

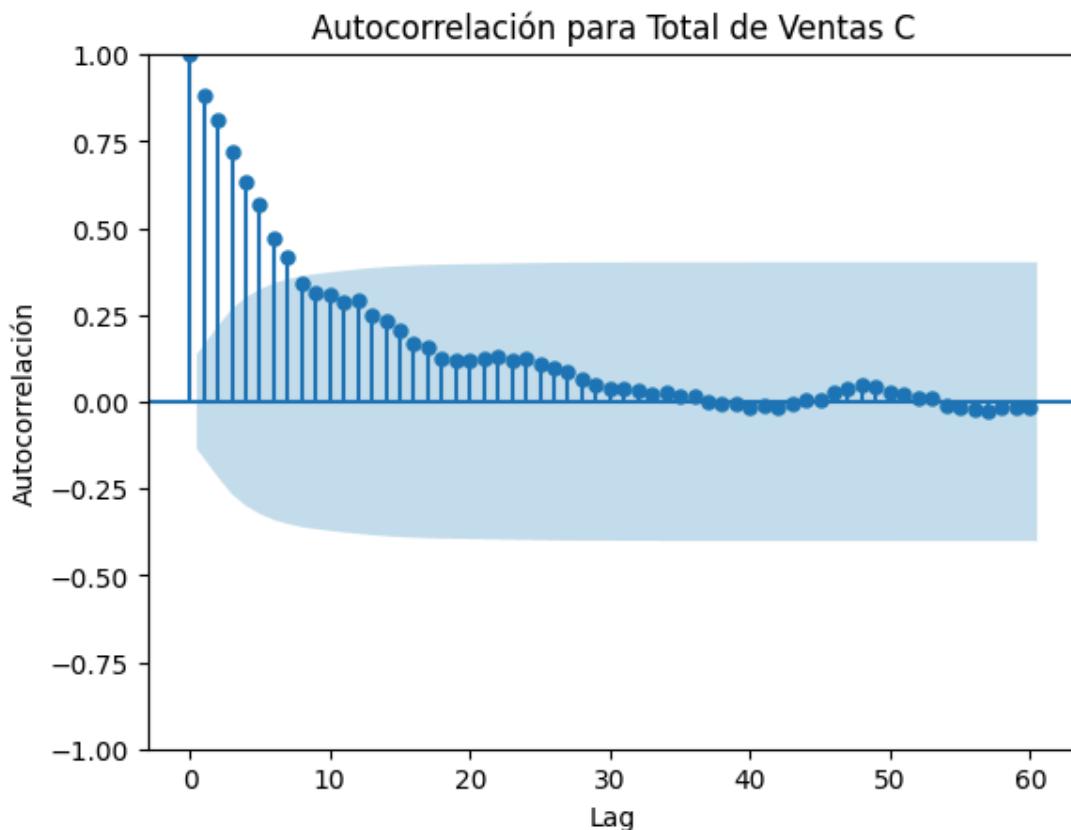
Figura 16:



*Nota. Elaboración propia utilizando Jupyter Notebook en Visual Studio Code.*

Las observaciones visuales para la serie Total M son las mismas que las indicadas para la serie Total MyC.

Figura 17:



Nota. Elaboración propia utilizando Jupyter Notebook en Visual Studio Code.

Las observaciones visuales para la serie Total C son las mismas que las indicadas para la serie Total MyC y Total M. La particularidad es que según la prueba de raíces unitarias realizada previamente la serie es no estacionaria. Esto significa que la serie tiene una tendencia estocástica, lo que implica que sus propiedades estadísticas (media y varianza) cambian con el tiempo.

Se debe realizar una transformación de la misma antes de modelar con el fin de convertirla de no estacionaria a estacionaria.

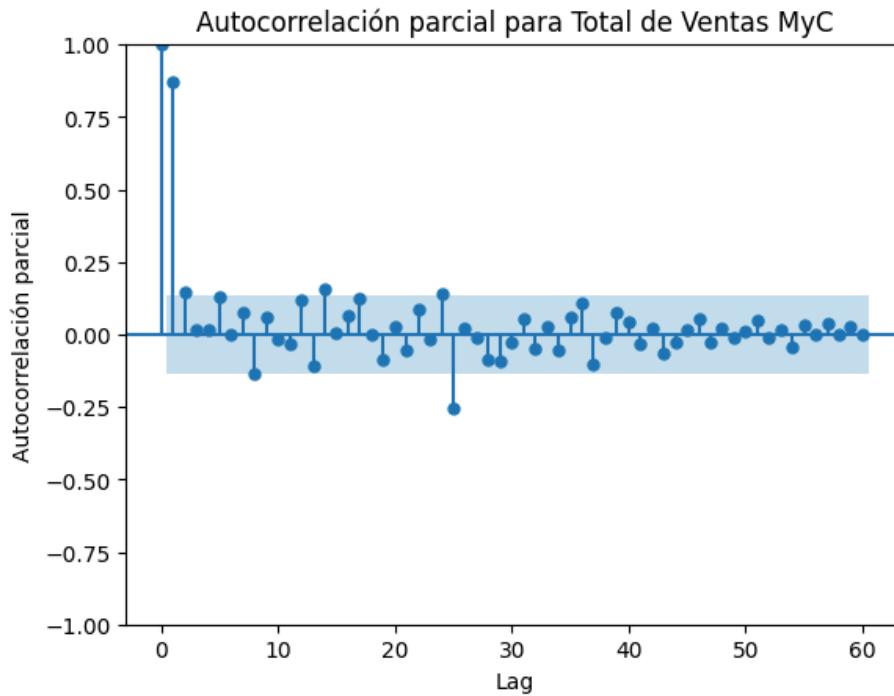
Aplicar la diferenciación para eliminar la tendencia implica restar la observación anterior de la observación actual. Si la primera diferenciación no es suficiente, se debe aplicar una segunda diferenciación.

### 5.3. Gráfico de la función de autocorrelación parcial (PACF):

Muestra las correlaciones parciales entre una serie y sus valores rezagados, controlando los efectos de los rezagos intermedios. Si hay una correlación parcial significativa en el primer lag (lag 1) y las correlaciones parciales de los lags anteriores son cercanas a cero,

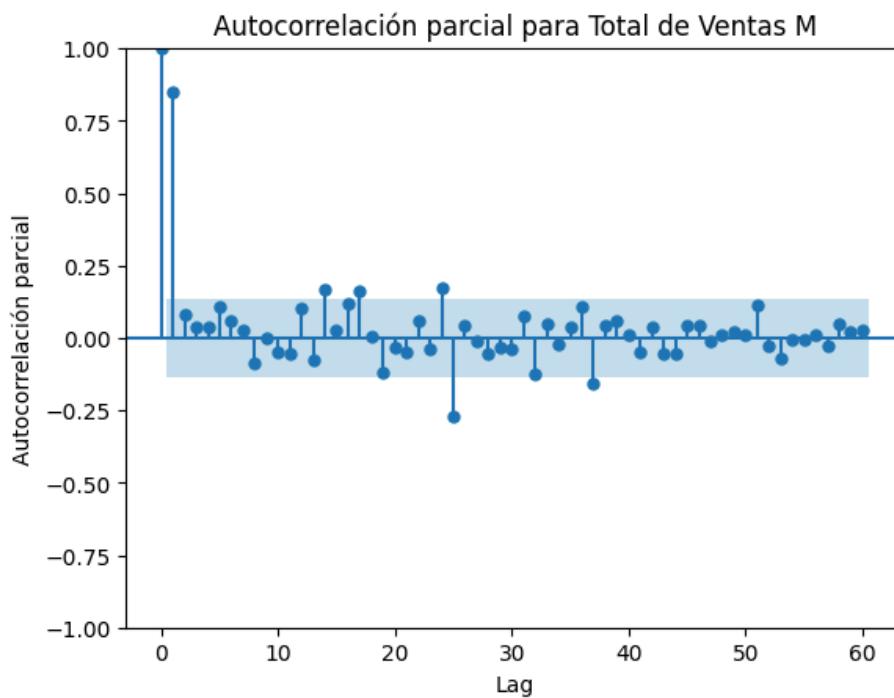
esto puede indicar una fuerte correlación directa entre las ventas de juguetes de un mes y las ventas del mes siguiente, sin una correlación fuerte con los rezagos anteriores.

Figura 18:



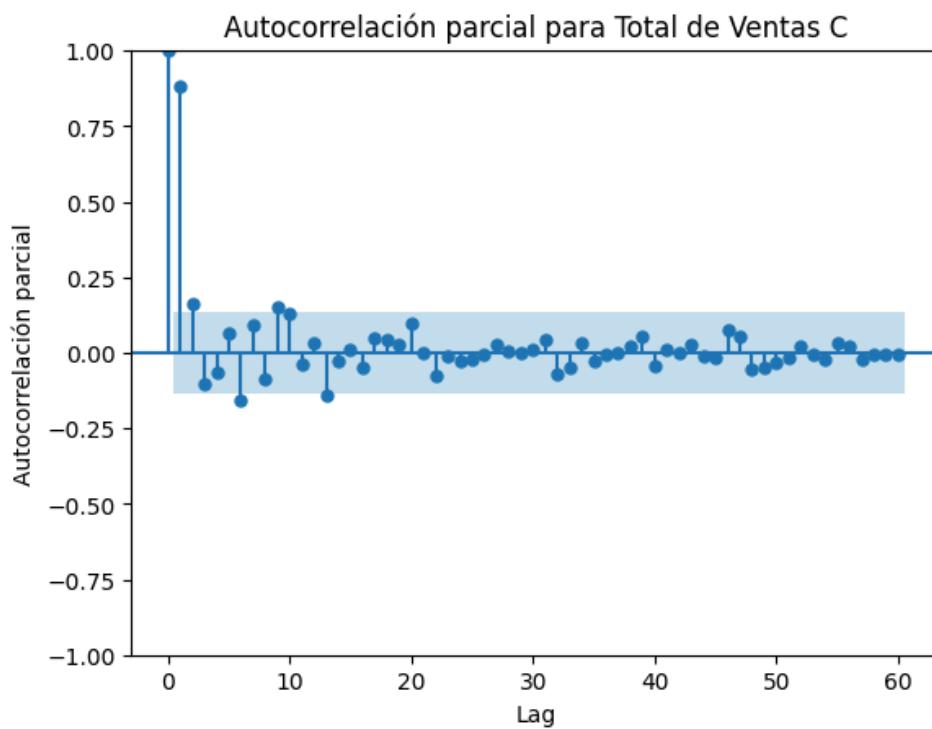
Nota. Elaboración propia utilizando Jupyter Notebook en Visual Studio Code.

Figura 19:



Nota. Elaboración propia utilizando Jupyter Notebook en Visual Studio Code.

Figura 20:



*Nota. Elaboración propia utilizando Jupyter Notebook en Visual Studio Code.*

Las tres series muestran en sus gráficas de PACF un rezago significativo en el rezago 1. Esto indica una fuerte correlación directa entre cada observación y la observación inmediatamente anterior en cada serie. Esto sugiere que el valor actual de las ventas está fuertemente influenciado por el valor de las ventas del mes anterior.

Después del rezago 2, los coeficientes de autocorrelación parcial caen rápidamente dentro de los límites de confianza. Esto sugiere que no hay correlaciones parciales significativas más allá del rezago 2 en ninguna de las series. El patrón de corte abrupto en las tres PACF es característico de modelos autorregresivos (AR).

#### 5.4. Proceso Autorregresivo (AR) y Proceso de Media Móvil (MA)

De acuerdo a lo mencionado previamente se pueden considerar modelos AR para analizar y predecir la serie temporal basándose en los valores pasados.

Limitaciones: un modelo AR asume que solo los valores pasados de la serie influyen en los futuros. No considera otros factores externos (como cambios en el mercado, eventos inesperados, etc.) que podrían influir en las ventas. También, un modelo AR asume que la relación entre los valores pasados y presentes es lineal.

Que las series sigan un proceso de media móvil significa que se pueden modelar como una combinación de un término constante y los errores cometidos en las estimaciones de las ventas en los períodos anteriores.

Predicción: En un modelo MA, los valores futuros no dependen directamente de los valores pasados de la serie, sino de los errores cometidos en los períodos anteriores. Por lo tanto, para hacer predicciones, se necesita conocer los errores anteriores.

## 5.5. Modelos Univariados

### 5.5.1. Modelos ARIMA

Se utiliza la función `auto_arima` librería `pmdarima` para desarrollar modelos automatizando el proceso de selección de los mejores parámetros del modelo.

Encontrándose los siguientes parámetros para cada serie:

#### Serie: Total MyC

$p = 2$ : El modelo usa los dos valores anteriores de la serie para predecir el valor actual. Esto indica que hay una dependencia de hasta dos períodos pasados (en línea con los análisis visuales realizados previamente con ADF/ACF/PACF).

$d = 1$ : Se ha aplicado una diferencia para hacer que la serie sea estacionaria, es decir, se ha eliminado una tendencia lineal o no estacionaria de la serie temporal (fue lo comentado previamente con ACF/PACF no así con el test de ADF que arrojó que no era necesario diferenciar).

$q = 1$ : El modelo usa el error del período anterior en la predicción actual. Esto sugiere que el error en el período pasado tiene un efecto en el valor actual.

$P = 0, D = 0, Q = 0, s = 0$ : No se ha utilizado ninguna componente estacional.

### Serie: **Total M**

$p = 1$ : El modelo usa solo el valor anterior de la serie para hacer predicciones. Esto indica una dependencia más simple en comparación con  $p = 2$  (en línea con los análisis visuales realizados previamente con ADF/ACF/PACF).

$d = 1$ : Se ha aplicado una diferencia para hacer la serie estacionaria (fue lo comentado previamente con ACF/PACF no así con el test de ADF que arrojó que no era necesario diferenciar).

$q = 1$ : El modelo considera el error del período anterior en la predicción actual, similar a la serie Total MyC.

$P = 0, D = 0, Q = 0, s = 0$ : No se ha utilizado ninguna componente estacional.

### Serie: **Total C**

$p = 2$ : Al igual que en la serie Total MyC, el modelo usa los dos valores anteriores de la serie para la predicción (en línea con los análisis visuales realizados previamente con ADF/ACF/PACF).

$d = 1$ : Se ha aplicado una diferencia para la estacionarización de la serie (en línea con lo analizado y comentado previamente con ADF/ACF/PACF).

$q = 2$ : El modelo utiliza los errores de los dos períodos anteriores para predecir el valor actual. Esto indica una dependencia más compleja en comparación con  $q = 1$ .

$P = 0, D = 0, Q = 0, s = 0$ : No se ha utilizado ninguna componente estacional.

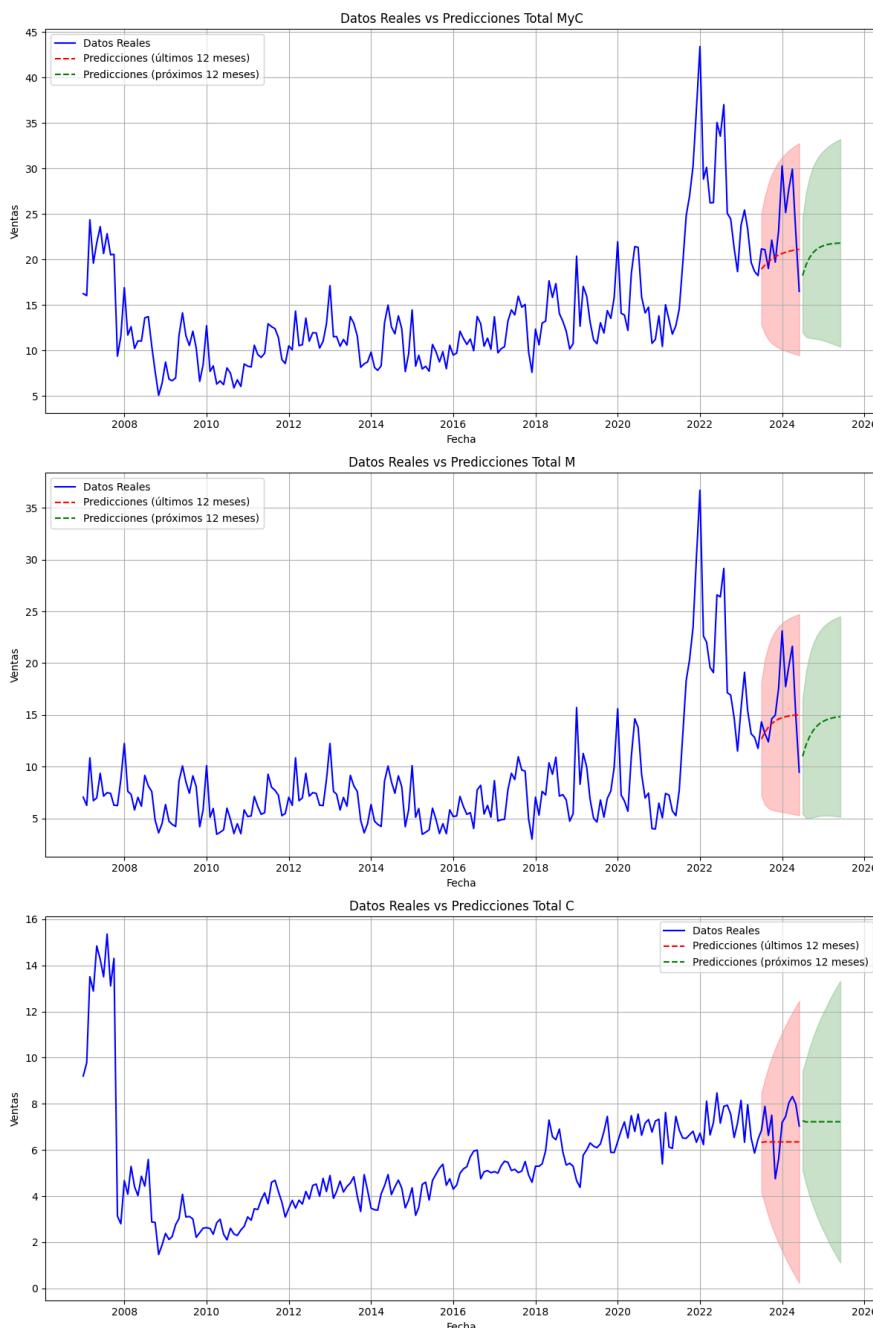
Interpretación General Dependencia de Valores Pasados ( $p$ ): En las series con  $p = 2$ , el modelo considera dos períodos pasados para hacer predicciones, lo cual es útil si los datos muestran una relación que se extiende más allá de un período.

Diferenciación (d): En todos los casos, se ha utilizado  $d = 1$ , lo que indica que se ha realizado una diferencia para tratar con la no estacionariedad en la serie.

Dependencia de Errores Pasados (q): Para las series con  $q = 1$  o  $q = 2$ , el modelo ajusta el error de uno o dos períodos anteriores, lo que puede capturar la influencia de errores pasados en el valor actual.

Componentes Estacionales (P, D, Q, s): Ninguna de las series usa componentes estacionales ( $P, D, Q, s$  todos en 0), lo que indica que no se han considerado efectos estacionales en el modelo, ya sea porque no se observaron estacionalidades o porque no se incluyó estacionalidad en el análisis.

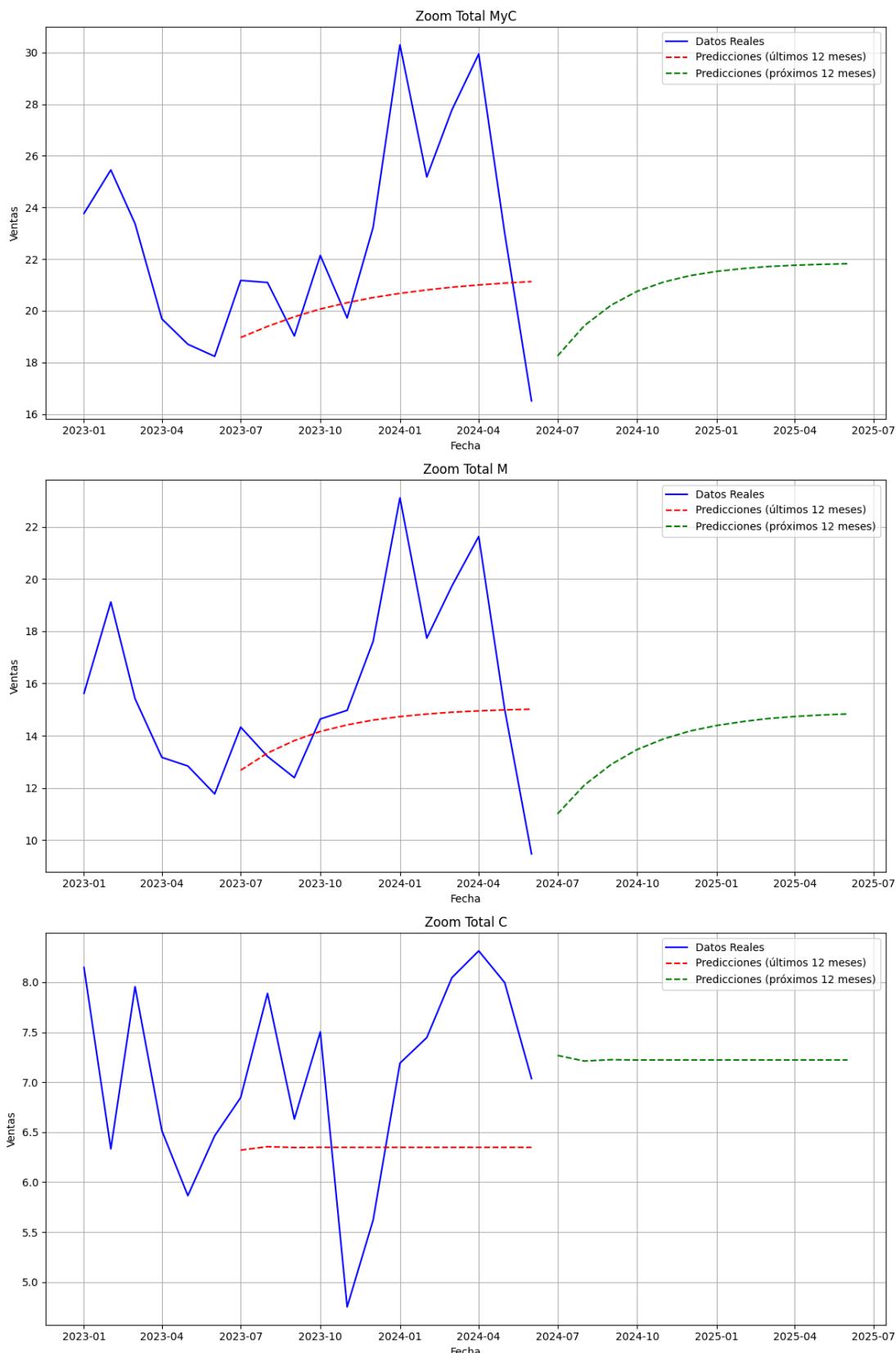
Figura 21:



Nota. Elaboración propia utilizando Jupyter Notebook en Visual Studio Code.

Se hace un zoom sobre las predicciones para observar mejor las diferencias entre datos reales y predicciones.

Figura 22:



Nota. Elaboración propia utilizando Jupyter Notebook en Visual Studio Code.

Se observan y se guardan tanto las predicciones como las métricas de performance para cada serie de modelos ARIMA.

### 5.5.2. Modelo de Redes Neuronales Long Short-Term Memory (LSTM)

En la presente sección, se aborda la implementación y evaluación de modelos de Redes Neuronales Long Short-Term Memory (LSTM) para la predicción de series temporales. Las Redes LSTM, una variante de las Redes Neuronales Recurrentes (RNN), han demostrado ser particularmente efectivas en el modelado de datos secuenciales debido a su capacidad para capturar dependencias temporales a largo plazo. La implementación de los modelos LSTM se ha llevado a cabo utilizando la librería [Keras](#), un marco de trabajo de alto nivel para la construcción y entrenamiento de modelos de aprendizaje profundo. Se han desarrollado diversas versiones de modelos LSTM, cada una con configuraciones y arquitecturas específicas, con el objetivo de explorar y optimizar el rendimiento de las predicciones. No obstante, todas las versiones de modelos LSTM comparten un conjunto de procedimientos comunes en el preprocesamiento de los datos. Específicamente, se ha aplicado una transformación de escalado a los datos, utilizando la función [MinMaxScaler](#) de la librería [scikit-learn](#) (`sklearn`), con el fin de normalizar los valores en un rango de 0 a 1. Esta transformación tiene como objetivo mejorar la convergencia del modelo y la estabilidad del entrenamiento, al tiempo que facilita la interpretación de los resultados en la escala original. A continuación se separaron los datos entre *train* y *test* para que el modelo realice los cálculos de métricas de performance sobre datos no vistos previamente.

A continuación, se detallan las especificaciones de cada versión del modelo LSTM, incluyendo la arquitectura de la red, los parámetros de entrenamiento y las métricas de rendimiento obtenidas.

El primer modelo LSTM propuesto se configuró de la siguiente manera:

- **Longitud de la Secuencia (SEQ\_LENGTH):** longitud 12. Esto significa que el modelo toma en cuenta 12 períodos previos (por ejemplo, 12 meses en el caso de datos mensuales) para predecir el siguiente valor en la serie temporal.
- **Capas del Modelo:**
  - **Primera Capa LSTM:** Esta capa LSTM tiene 50 unidades y está configurada con `return_sequences=True`. Esto permite que la capa

devuelva la secuencia completa de salida para cada paso de tiempo, lo cual es necesario para que la siguiente capa LSTM pueda procesar las secuencias completas y aprender mejor las relaciones temporales.

- **Segunda Capa LSTM:** La segunda capa LSTM también tiene 50 unidades, pero no devuelve secuencias (`return_sequences=False`). Esta capa se encarga de resumir la información de la secuencia completa en un único vector que captura la información relevante para la predicción.
- **Capa Densa:** Finalmente, la capa densa con una sola unidad produce la salida final del modelo, que en este caso es un valor continuo correspondiente a la predicción de la serie temporal.
- **Optimizador:** se utilizó Adam con una tasa de aprendizaje de 0.01. Adam es un optimizador eficiente que ajusta adaptativamente las tasas de aprendizaje para cada parámetro del modelo, lo cual facilita el entrenamiento.
- **Función de Pérdida:** Se utiliza la función de pérdida de error cuadrático medio (MSE), que mide la diferencia promedio al cuadrado entre los valores predichos y los valores reales. Esta métrica es común en problemas de regresión y ayuda a ajustar los pesos del modelo durante el entrenamiento.

La estructura del modelo LSTM con dos capas permite al modelo capturar tanto patrones a corto como a largo plazo en los datos de series temporales. La primera capa LSTM captura las relaciones temporales inmediatas, mientras que la segunda capa sintetiza esta información para hacer una predicción más precisa. La capa densa final ajusta la salida del modelo a un valor específico.

Una vez obtenidas las predicciones se realiza el proceso inverso al escalado para volver a tener las variables en su escala original y con esto poder realizar una mejor interpretación de los resultados obtenidos.

Se calculan las métricas de performance para cada modelo de cada variable bajo estudio, se guardan tanto las métricas como las predicciones en dos dataframes para luego ser comparadas con los demás modelos LSTM y el resto de modelos planteados previamente.

En un segundo modelo, lo que se realiza es un seteo de semilla fija para la inicialización de la parametrización del modelo.

Al realizar este procedimiento se observa que hay una diferencia notable entre las predicciones del modelo 1 con las del modelo 2, por lo cual se entiende que la semilla elegida aleatoriamente puede generar una variabilidad demasiado grande y por consecuencia una distorsión de los valores que se buscan estimar.

Por lo mencionado previamente en un tercer modelo lo que se incorpora es la posibilidad de que se elijan aleatoriamente una cantidad “X” de semillas para lograr estabilizar mediante promedios los valores que parametriza el modelo LSTM y luego utiliza para las predicciones de las variables objetivo.

Se realizaron varias pruebas desde la utilización de 23 semillas (tiempo de ejecución de alrededor de 10 minutos) hasta la utilización de 5 semillas (tiempo de ejecución aproximado de 4 minutos), encontrando un punto óptimo entre estabilización de los parámetros y velocidad de performance de los modelos en el promedio de ejecución con 5 semillas que fue lo utilizado en los modelos subsiguientes.

Definición del cuarto modelo LSTM Una vez definida la cantidad de semillas a utilizar y promediar sus resultados, se procedió a modificar los parámetros utilizados en los modelos previos aumentando las capas 1 y 2 del modelo de 50 a 100 cada una de ellas respectivamente y disminuyendo el learning rate de 0.01 a 0.001. Los resultados fueron que el modelo se sobreajustó y perdió capacidad de generalizar de manera correcta sobre los datos nunca vistos.

En el quinto modelo LSTM se plantea la posibilidad de asignar un mayor peso a los datos más recientes, teniendo en cuenta tanto el dominio intrínseco de los datos (expertos en el dominio del área de planeamiento económico de la compañía) como lo analizado previamente en los gráficos de Función de Autocorrelación y Función de Autocorrelación Parcial donde se podía observar y se comentó la dependencia de los valores futuros con una alta correlación en los valores pasados más cercanos.

En el sexto y último modelo LSTM se modifica el parámetro de longitud de la secuencia (SEQ\_LENGTH): de longitud 12 a longitud 48. Esto significa que el modelo pasa de tomar en cuenta 12 periodos previos para predecir el siguiente valor en la serie temporal a tomar 48

períodos previos, lo cual lo convierte en un modelo más complejo con la capacidad de modelar procesos más complejos como por ejemplo estacionalidades puntuales.

Comparaciones de métricas de performance dentro de los modelos LSTM:

**Modelo 1:** Capas 1 y 2 de 50 unidades y learning rate de 0.01 (optimizador ADAM)

Total MyC\_LSTM - MAE: 12.17, MSE: 195.51, RMSE: 13.98, MAPE: 16.75%

Total M\_LSTM - MAE: 15.62, MSE: 318.88, RMSE: 17.86, MAPE: 29.61%

Total C\_LSTM - MAE: 3.41, MSE: 15.69, RMSE: 3.96, MAPE: 15.67%

**Modelo 2:** fijación de semilla

Total MyC\_LSTM - MAE: 12.39, MSE: 209.05, RMSE: 14.46, MAPE: 15.94%

Total M\_LSTM - MAE: 16.04, MSE: 357.25, RMSE: 18.90, MAPE: 29.90%

Total C\_LSTM - MAE: 2.97, MSE: 12.30, RMSE: 3.51, MAPE: 14.20%

**Modelo 3:** promedio de 23 semillas aleatorias

Total MyC\_LSTM - MAE: 11.31, MSE: 172.65, RMSE: 13.14, MAPE: 15.46%

Total M\_LSTM - MAE: 9.95, MSE: 130.77, RMSE: 11.44, MAPE: 20.05%

Total C\_LSTM - MAE: 2.87, MSE: 11.78, RMSE: 3.43, MAPE: 13.92%

**Modelo 4:** promedio de 5 semillas aleatorias y aumento de capas 1 y 2 de 50 a 100 unidades cada una y disminución del learning rate de 0.01 a 0.001.

Total MyC\_LSTM - MAE: 12.26, MSE: 225.27, RMSE: 15.01, MAPE: 16.68%

Total M\_LSTM - MAE: 15.14, MSE: 300.21, RMSE: 17.33, MAPE: 28.72%

Total C\_LSTM - MAE: 3.38, MSE: 14.73, RMSE: 3.84, MAPE: 15.61%

**Modelo 5:** promedio de 5 semillas aleatorias y capas 1 y 2 de 50 unidades cada una y learning rate de 0.01 con la incorporación de mayores pesos a períodos cercanos.

Total MyC\_LSTM - MAE: 12.44, MSE: 196.56, RMSE: 14.02, MAPE: 16.58%

Total M\_LSTM - MAE: 15.86, MSE: 328.34, RMSE: 18.12, MAPE: 30.34%

Total C\_LSTM - MAE: 2.81, MSE: 11.44, RMSE: 3.38, MAPE: 13.69%

**Modelo 6:** modelo anterior con aumento de longitud de secuencia a 48 períodos

Total MyC\_LSTM\_Weighting - MAE: 9.74, MSE: 152.98, RMSE: 12.37, MAPE: 14.18%

Total M\_LSTM\_Weighting - MAE: 12.92, MSE: 232.20, RMSE: 15.24, MAPE: 24.65%

Total C\_LSTM\_Weighting - MAE: 3.01, MSE: 11.99, RMSE: 3.46, MAPE: 14.45%

Se almacenan en un dataframe las métricas de performance por Modelo y por Variable.

### I. Resultados para Total MyC

El análisis de la variable **Total MyC** muestra que el **Modelo 6**, que incorpora un aumento en la longitud de la secuencia a 48 períodos, ofrece las mejores métricas de rendimiento en todas las categorías evaluadas. Específicamente, este modelo logró los valores más bajos de MAE (9.74), MSE (152.98), RMSE (12.37) y MAPE (14.18%). Esto sugiere que la longitud de la secuencia es un factor crítico para mejorar la precisión del modelo en la predicción de esta serie temporal.

El **Modelo 3**, que realiza el promedio con 23 semillas aleatorias, también muestra un desempeño sólido, especialmente en MAPE (15.46%) y MAE (11.31), destacándose como el segundo mejor modelo en estas métricas. Por otro lado, los modelos 1, 2, 4 y 5, que varían en la cantidad de capas, unidades, y la aplicación de pesos, presentan métricas superiores a las de los modelos 3 y 6, lo que indica un menor rendimiento.

### II. Resultados para Total M

Para la variable **Total M**, el **Modelo 3** (promedio de 23 semillas aleatorias) sobresale con un MAE de 9.95, MSE de 130.77, RMSE de 11.44, y MAPE de 20.05%. Estos resultados indican que el modelo es particularmente efectivo en la predicción de esta serie temporal, logrando la menor magnitud de errores en comparación con las otras configuraciones.

El **Modelo 6** también proporciona buenos resultados, aunque no supera al Modelo 3, con un MAPE de 24.65%, RMSE de 15.24, MAE de 12.92 y MSE de 232.20. Sin embargo, su desempeño es notablemente mejor que el de los otros modelos, lo que subraya la importancia del aumento de la longitud de la secuencia.

Los modelos 1, 2, 4 y 5 muestran un rendimiento inferior, con errores más altos en todas las métricas, lo que sugiere que las variaciones en las capas, unidades y pesos no son tan efectivas para esta variable como lo son los promedios de múltiples semillas.

### **III. Resultados para Total C**

En cuanto a la variable **Total C**, el **Modelo 5** (promedio de 5 semillas aleatorias y capas con 50 unidades cada una, con un learning rate de 0.01 y mayor peso a períodos cercanos) produce las mejores métricas en MAE (2.81), MSE (11.44), y RMSE (3.38), con un MAPE de 13.69%. Esto sugiere que la incorporación de pesos adicionales a períodos recientes es particularmente beneficiosa para este modelo en la predicción de la serie temporal.

El **Modelo 6** también muestra un rendimiento competitivo, aunque ligeramente inferior al del Modelo 5, con un MAPE de 14.45% y RMSE de 3.46. Esto indica que la estrategia de ponderar más los datos recientes es ventajosa para esta variable.

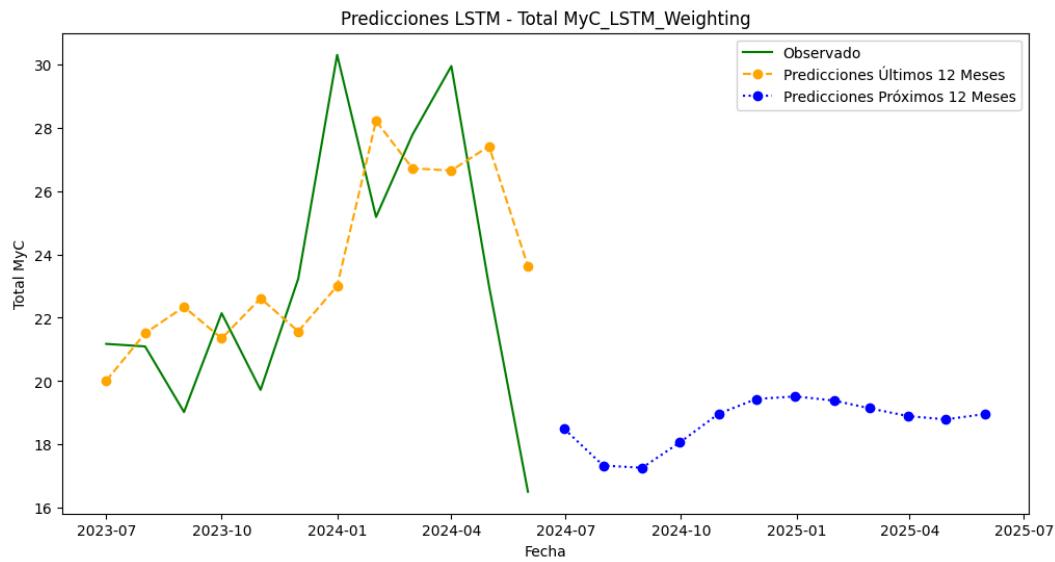
En general, los resultados para **Total C** destacan la eficacia del enfoque basado en la ponderación de períodos recientes y la longitud de la secuencia en la mejora de la precisión del modelo LSTM para esta serie temporal.

**A continuación se muestran las gráficas para el modelo 6.**

Parámetros:

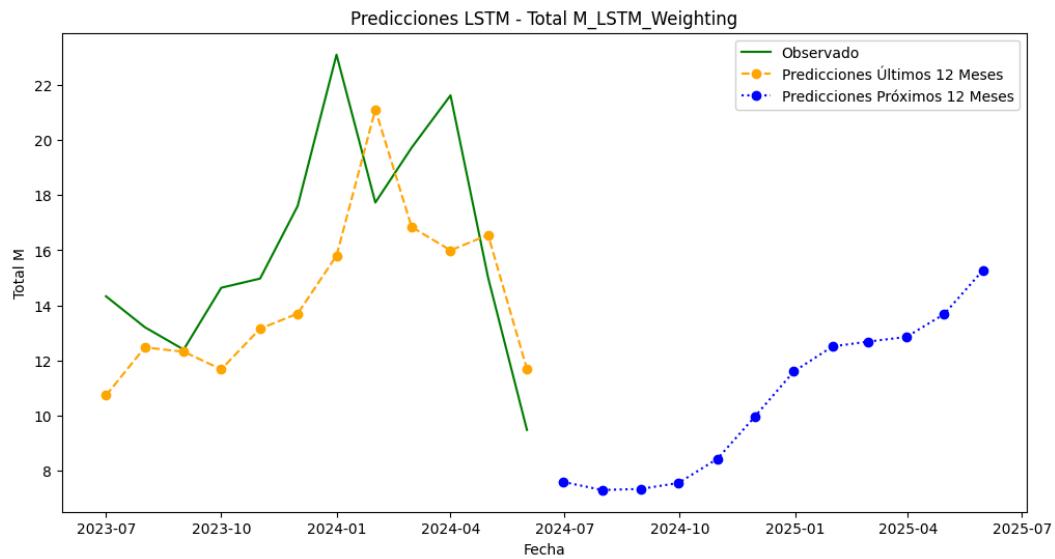
- Capa 1 y 2: 50 unidades cada una
- Learning rate: 0.01 (optimizador Adam)
- Promedio de 5 semillas aleatorias
- Utilización de mayor ponderación de pesos en últimos períodos.
- Longitud de secuencia: 48 períodos.

Figura 23:



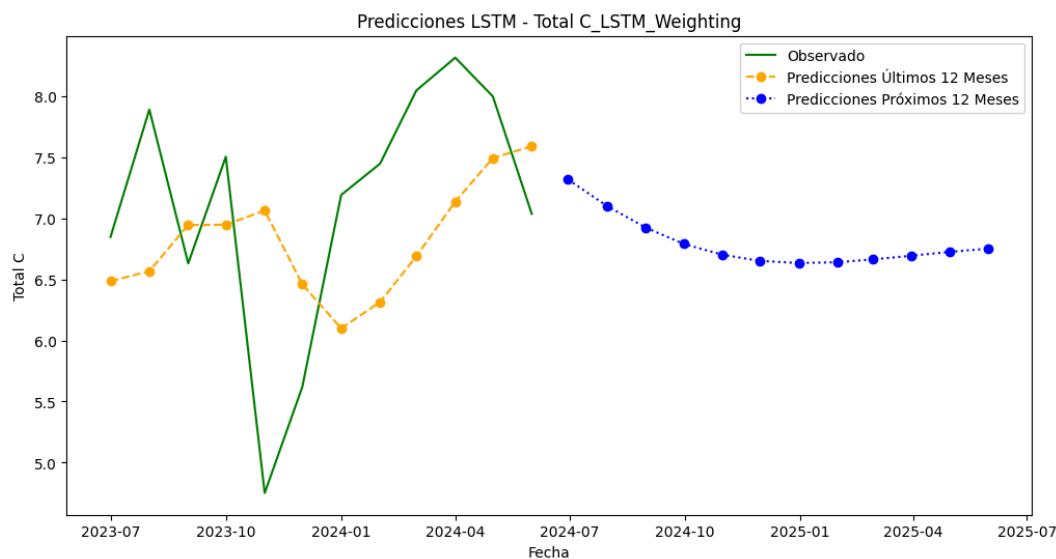
Nota. Elaboración propia utilizando Jupyter Notebook en Visual Studio Code.

Figura 24:



Nota. Elaboración propia utilizando Jupyter Notebook en Visual Studio Code.

Figura 25:



Nota. Elaboración propia utilizando Jupyter Notebook en Visual Studio Code.

Conclusiones de los diferentes modelos LSTM:

- **La longitud de la secuencia y la ponderación de los períodos recientes** son factores críticos para mejorar el rendimiento de los modelos LSTM en la predicción de series temporales.
- **El uso de promedios de múltiples semillas** contribuye significativamente a la estabilidad y precisión de los modelos.
- **La personalización del modelo LSTM** en términos de capas y unidades, aunque importante, debe ser cuidadosamente ajustada en conjunto con la longitud de la secuencia y los pesos asignados a períodos recientes para maximizar la precisión del modelo.

### 5.5.3. Modelos Random Forest Regressor

En esta sección, se aborda la implementación y evaluación de modelos Random Forest Regressor para la predicción de series temporales. Los modelos Random Forest, un método de aprendizaje automático de conjunto, han demostrado ser eficaces en la captura de relaciones no lineales y la generación de predicciones robustas. La implementación de los modelos Random Forest se ha llevado a cabo utilizando la librería `scikit-learn` (`sklearn`), un marco de trabajo ampliamente utilizado para el aprendizaje automático en Python. Se han desarrollado

diversas versiones de modelos Random Forest, cada una con configuraciones y parámetros específicos, con el objetivo de explorar y optimizar el rendimiento de las predicciones.

## Preparación de los Datos

Antes de entrenar los modelos Random Forest, los datos fueron preprocesados de la siguiente manera:

- **Escalado de Datos:** Los datos se normalizaron utilizando la técnica de Min-Max Scaling con la librería `sklearn`, que ajusta los datos a un rango de 0 a 1. Este paso es importante para asegurar que todas las características contribuyan de manera equitativa al entrenamiento del modelo.
- **Separación en datos de entrenamiento y test,** quitando los últimos 12 meses de datos reales observados y usándolos para luego testear con una predicción de esos 12 meses.

El primer modelo Random Forest propuesto se configuró de la siguiente manera:

Se establecieron 100 árboles de decisión y una semilla fija para obtener resultados consistentes y reproducibles y solo se predicen datos para los últimos 12 meses de datos reales observados.

En un segundo modelo, se agregan predicciones a futuro sobre datos no observados con reentrenamiento progresivo durante el proceso de predicción para los próximos 12 meses.

Para el tercer modelo se utiliza promedio de 23 semillas diferentes buscando estabilizar la variabilidad en las predicciones.

Un cuarto modelo donde se utilizaron 5 semillas para promediar los resultados obtenidos, confirmando la estabilidad de la cantidad elegida.

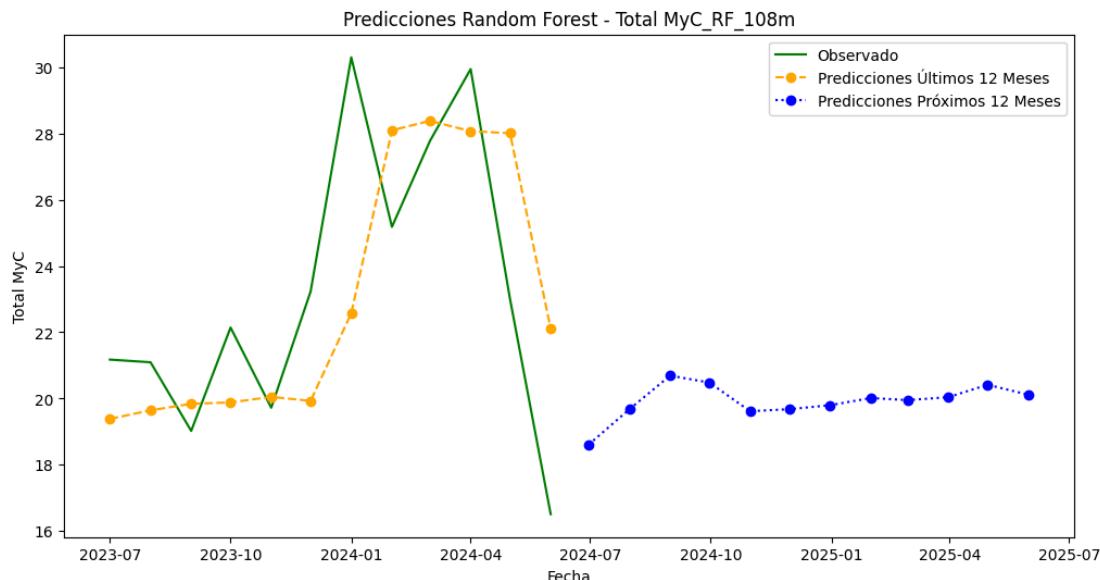
Quinto modelo similar al cuarto modelo pero se aplica reentrenamiento progresivo para la proyección de los datos futuros.

En el sexto modelo la variación viene dada por la cantidad de secuencias utilizadas en el mismo, mientras que todos los anteriores tenían una secuencia de 12 períodos mensuales, en este último se modificó a 48 períodos, buscando con ello capturar patrones más complejos como por ejemplo estacionalidad en los datos.

En el séptimo modelo se busca profundizar sobre las ventanas temporales creando una grilla y probando diferentes espacios temporales (6, 12, 18, 24, 36 y 48 meses).

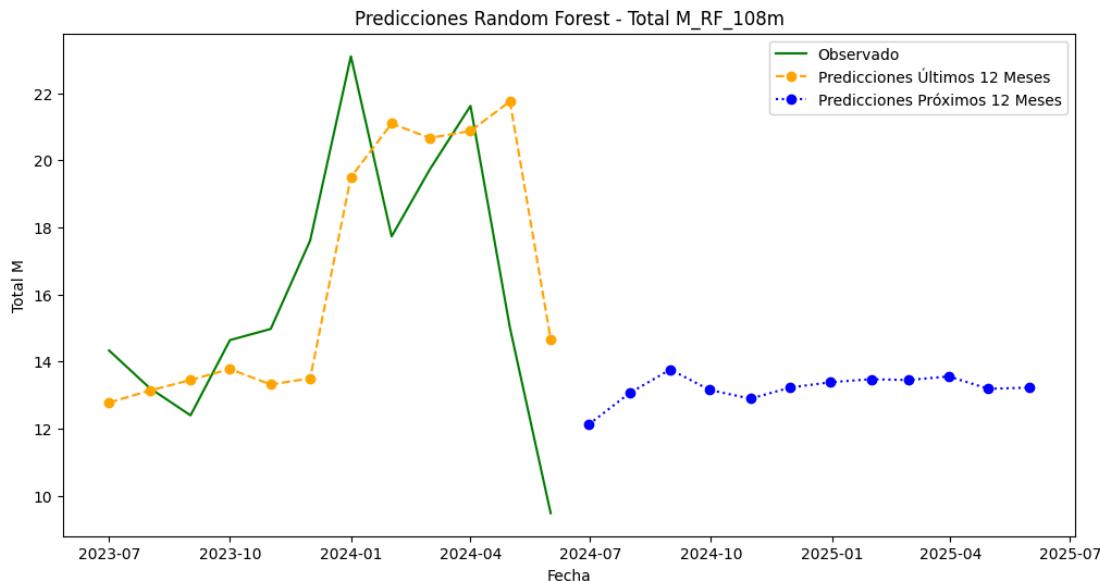
Finalmente en un modelo 7 bis se realizaron pruebas ampliando aún más las ventanas temporales.

*Figura 26:*



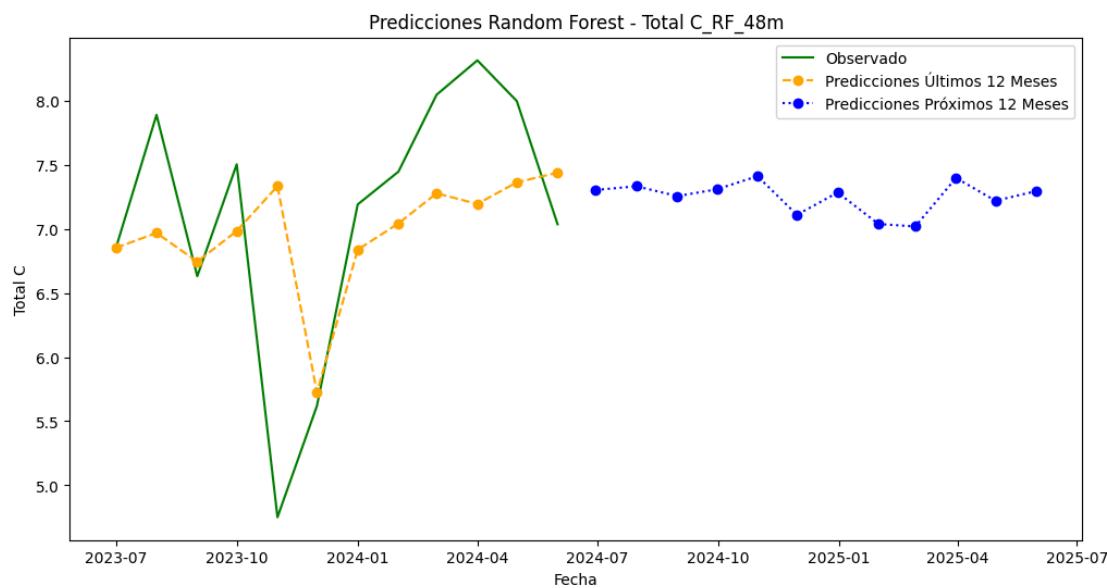
*Nota. Elaboración propia utilizando Jupyter Notebook en Visual Studio Code.*

*Figura 27:*



*Nota. Elaboración propia utilizando Jupyter Notebook en Visual Studio Code.*

*Figura 28:*



*Nota. Elaboración propia utilizando Jupyter Notebook en Visual Studio Code.*

El análisis comparativo de los modelos Random Forest Regressor de la versión 7 ha revelado que aquellos configurados con ventanas temporales de entre 18 y 24 meses exhiben el rendimiento predictivo más robusto, según lo evidenciado por las métricas de evaluación. Estos resultados sugieren que la captura de patrones temporales dentro de este rango específico optimiza la capacidad del modelo para generar predicciones precisas.

En la versión 7 Bis de los modelos Random Forest Regressor, se ha observado un comportamiento diferencial en función de la variable objetivo. Para las series temporales Total MyC y Total M, se ha constatado que la ampliación de la ventana temporal a 108 meses conduce a una mejora significativa en las métricas de rendimiento. Este hallazgo sugiere que estas series presentan dependencias temporales de largo alcance que son capturadas de manera efectiva por ventanas temporales extensas. Por el contrario, la serie Total C ha mostrado un rendimiento óptimo con ventanas temporales de menor duración, específicamente entre 24 y 48 meses. Esto indica que la dinámica temporal de Total C se caracteriza por patrones de corto a mediano plazo, los cuales son mejor modelados con ventanas temporales más acotadas.

#### 5.5.4. Modelos Xgboost

En la presente sección, se aborda la implementación y evaluación de modelos XGBoost (Extreme Gradient Boosting) para la predicción de series temporales. XGBoost, un algoritmo de aprendizaje automático de gradiente boosting, ha demostrado ser altamente eficaz en la resolución de problemas de regresión y clasificación, destacando por su rendimiento y eficiencia computacional.

Se han desarrollado dos versiones distintas de modelos XGBoost, cada una con un enfoque particular en la configuración de hiperparámetros y la validación del modelo, con el objetivo de analizar el impacto de estas estrategias en la precisión de las predicciones.

##### Modelo 1:

Se caracteriza por la utilización de un conjunto predefinido de hiperparámetros:

- n\_estimators=100,
- learning\_rate=0.01,
- max\_depth=5,
- Fijación de la semilla aleatoria (seed=run) en cada iteración.

Este enfoque se distingue por su simplicidad y velocidad de implementación, al evitar la optimización exhaustiva de hiperparámetros. No obstante, el Modelo 1 prescinde de la validación cruzada, evaluando el rendimiento del modelo directamente sobre los conjuntos de entrenamiento y prueba. Si bien esta estrategia reduce el tiempo de cómputo, puede limitar la capacidad del modelo para generalizar adecuadamente a datos no vistos, especialmente si los hiperparámetros predefinidos no son óptimos para el conjunto de datos específico.

##### Modelo 2:

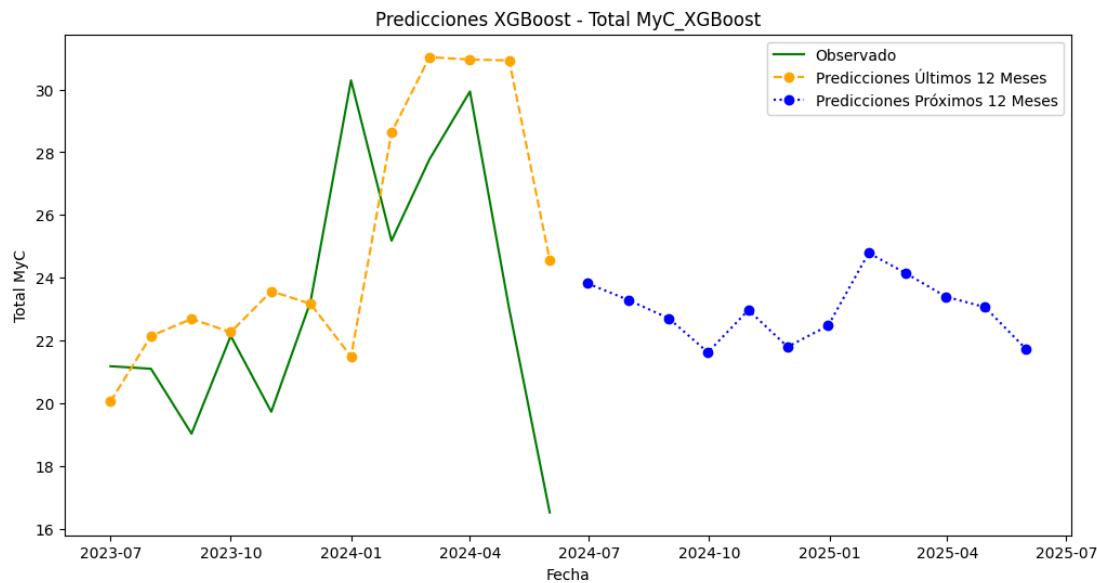
- Optimización de Hiperparámetros con GridSearchCV

El Modelo 2 adopta un enfoque más riguroso mediante la implementación de una búsqueda exhaustiva de hiperparámetros utilizando la técnica GridSearchCV. Este proceso permite explorar diversas combinaciones de parámetros, incluyendo n\_estimators, learning\_rate y max\_depth, con el fin de identificar la configuración óptima que maximice el rendimiento del modelo. La validación cruzada de 3-fold se integra dentro del proceso

GridSearchCV, lo que permite evaluar la robustez de cada combinación de hiperparámetros y seleccionar la configuración que ofrezca el mejor rendimiento promedio. Este enfoque dinámico de ajuste de hiperparámetros permite al Modelo 2 adaptarse de manera más efectiva a las características específicas del conjunto de datos, lo que potencialmente conduce a predicciones más precisas. A pesar de su mayor demanda computacional, el Modelo 2 ofrece un enfoque más robusto y potencialmente más preciso, al optimizar la configuración del modelo mediante la búsqueda de hiperparámetros y la validación cruzada.

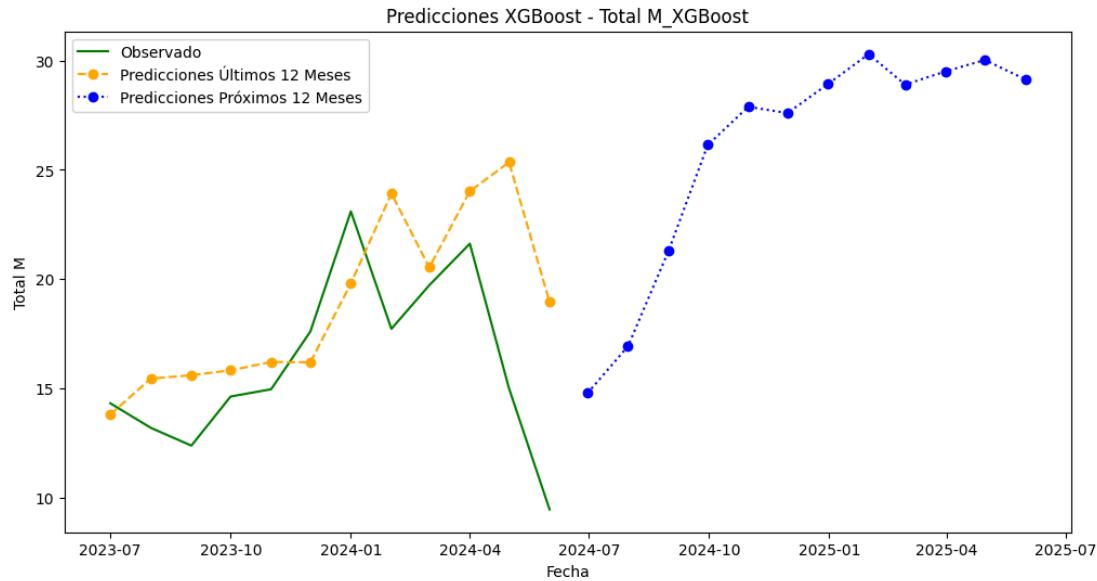
Gráficas Modelo 2:

Figura 29:



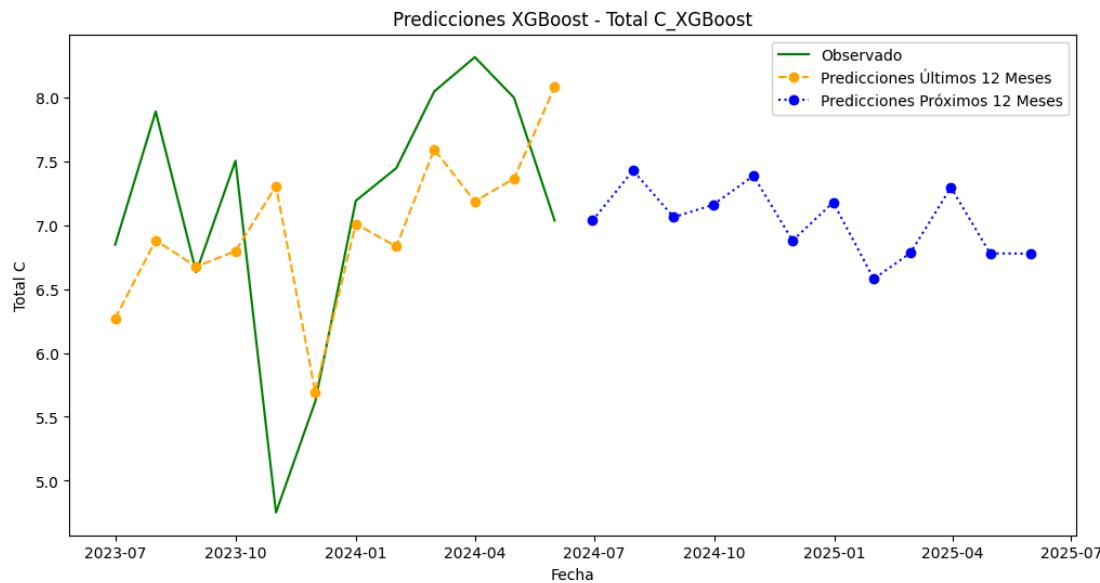
Nota. Elaboración propia utilizando Jupyter Notebook en Visual Studio Code.

Figura 30:



Nota. Elaboración propia utilizando Jupyter Notebook en Visual Studio Code.

Figura 31:



Nota. Elaboración propia utilizando Jupyter Notebook en Visual Studio Code.

Modelo 2 tiene un rendimiento superior en la variable Total MyC y es considerablemente mejor en Total C. En ambos casos, presenta menores errores y una mayor consistencia en las predicciones.

La optimización de hiperparámetros con GridSearchCV ha permitido ajustar los modelos XGBoost a las características específicas de cada serie temporal. La serie 'Total C' ha mostrado el mejor rendimiento general con XGBoost, indicando que este modelo es particularmente adecuado para capturar su dinámica.

#### 5.5.5. Modelos Prophet (Facebook)

A continuación se desarrollará la implementación y evaluación de modelos Prophet, una herramienta de predicción de series temporales desarrollada por Facebook, diseñada para modelar series con fuertes patrones estacionales y tendencias. La librería Prophet, basada en un modelo aditivo, permite descomponer las series temporales en sus componentes de tendencia, estacionalidad y días festivos, facilitando la interpretación y la predicción.

Se han desarrollado cuatro modelos Prophet distintos, cada uno con configuraciones específicas, con el objetivo de explorar el impacto de la personalización de parámetros y la validación en el rendimiento de las predicciones. A continuación, se detallan las características de cada modelo:

- Modelo 1:

- Estacionalidad: predeterminada de Prophet.
- Tendencia: Lineal por defecto.
- Validación Cruzada: no se aplicó.
- Sin personalización en los ajustes de estacionalidad, tendencia, o cambio de puntos de inflexión.

- Modelo 2:

- Estacionalidad: Se desactivan las estacionalidades anual, semanal y diaria predeterminadas.
- Escala de Estacionalidad: Se configura el *seasonality\_prior\_scale* para ajustar la importancia de la estacionalidad
- Tendencia: se ajusta el parámetro *changepoint\_prior\_scale* para controlar la flexibilidad de la tendencia.
- Validación Cruzada: Se incluye un proceso de validación cruzada para evaluar el rendimiento del modelo en diferentes horizontes temporales.

- Modelo 3:

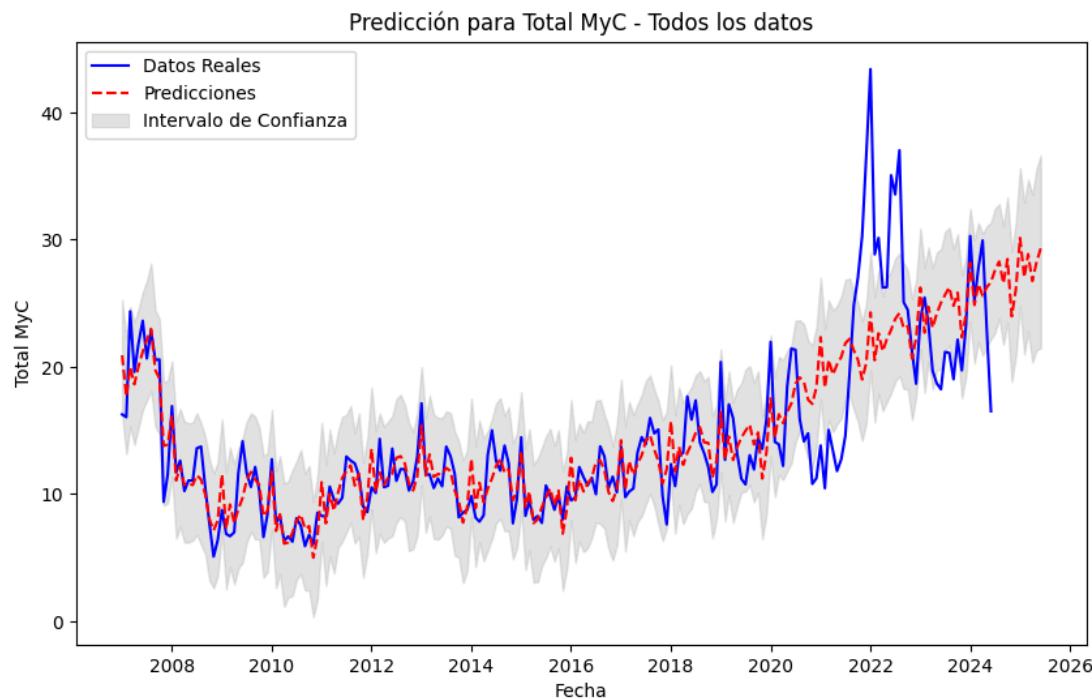
- Estacionalidad: Se mantiene la estacionalidad anual predeterminada y se añade una estacionalidad mensual personalizada.
- Tendencia: Lineal por defecto.
- Validación Cruzada: no se aplicó.

- Modelo 4: se agrega un grilla de búsqueda de hiperparámetros.

- Estacionalidad: se prueba con y sin estacionalidad anual y con el mejor valor para el parámetro *seasonality\_prior\_scale*.
- Tendencia: se busca el mejor valor para el parámetro *changepoint\_prior\_scale* para controlar la flexibilidad de la tendencia.
- Validación Cruzada: no se aplicó.

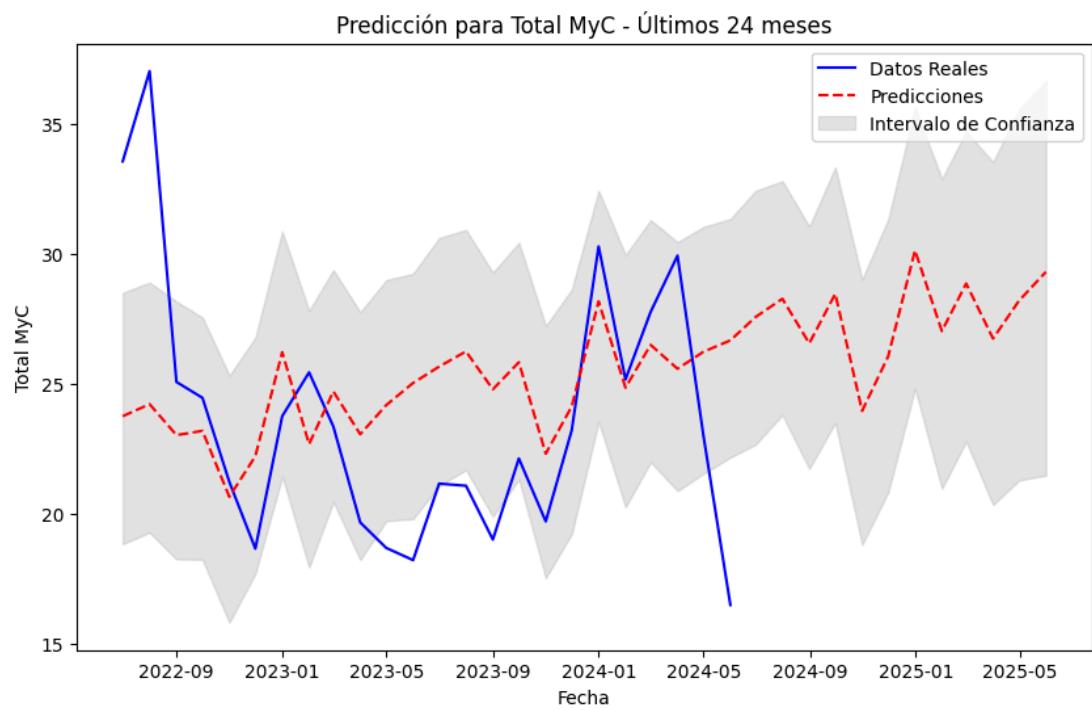
A continuación se presentarán las gráficas del “*Modelo 4*” en dos versiones, una con la totalidad de períodos y otra haciendo zoom en los últimos 24 meses.

Figura 32:



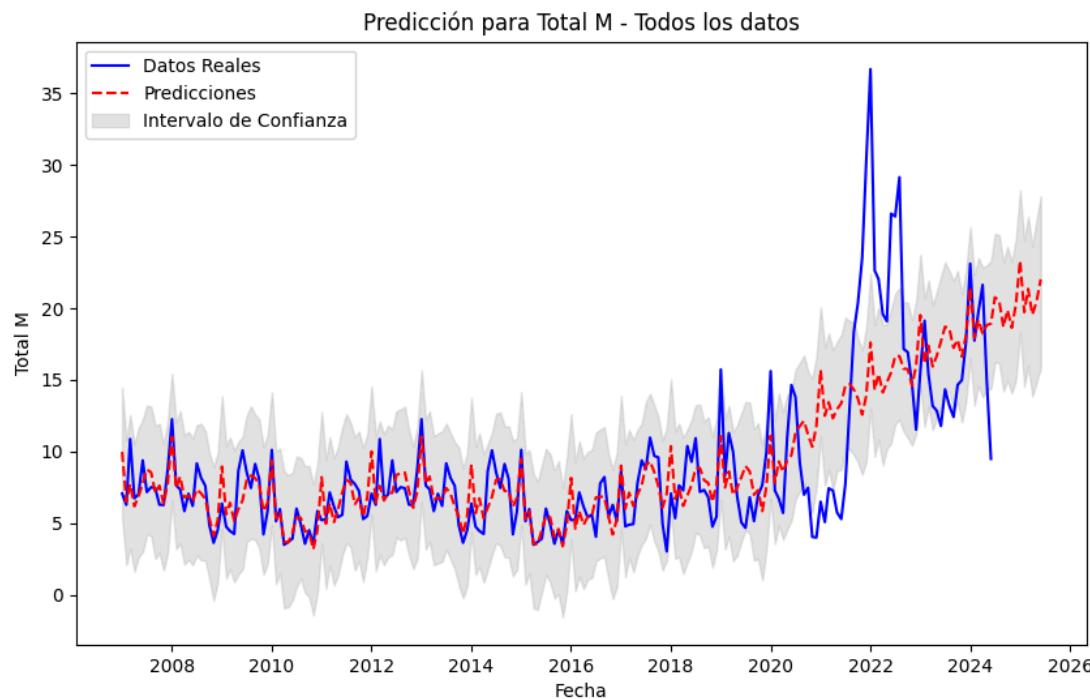
Nota. Elaboración propia utilizando Jupyter Notebook en Visual Studio Code.

Figura 33:



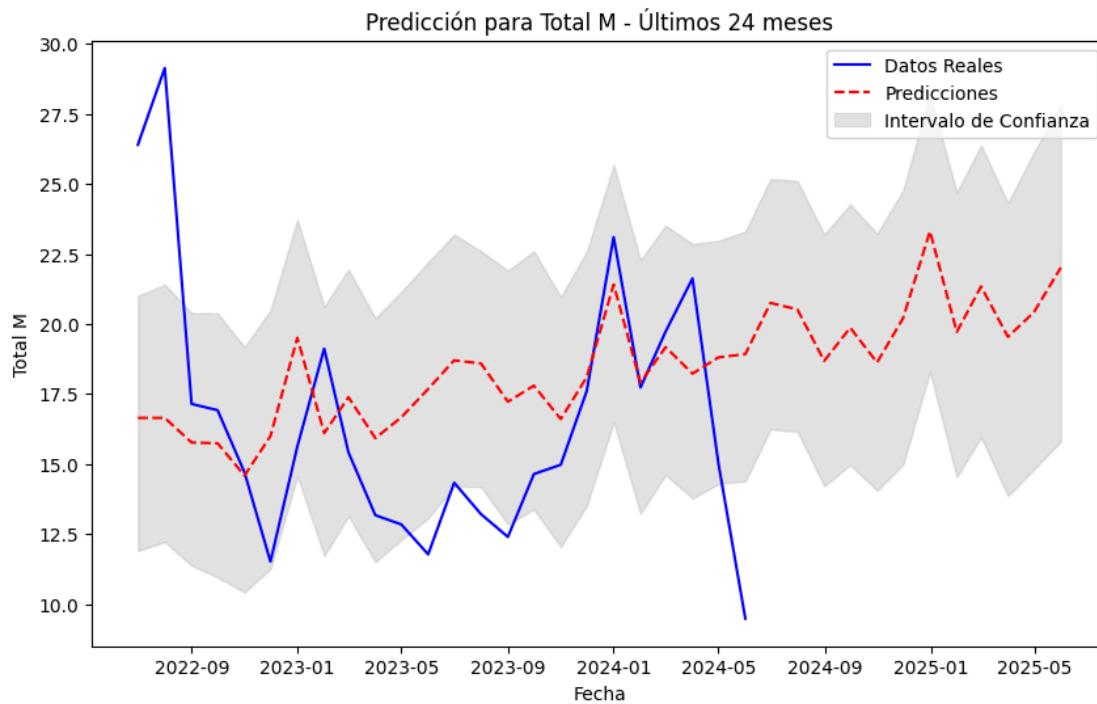
Nota. Elaboración propia utilizando Jupyter Notebook en Visual Studio Code.

Figura 34:



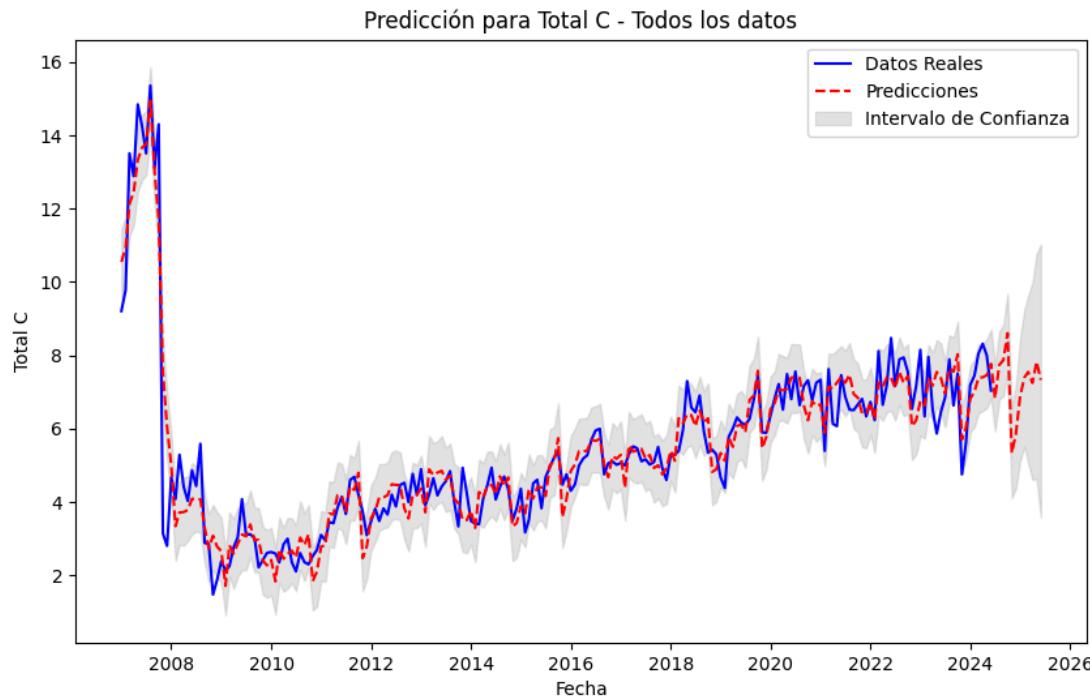
Nota. Elaboración propia utilizando Jupyter Notebook en Visual Studio Code.

Figura 35:



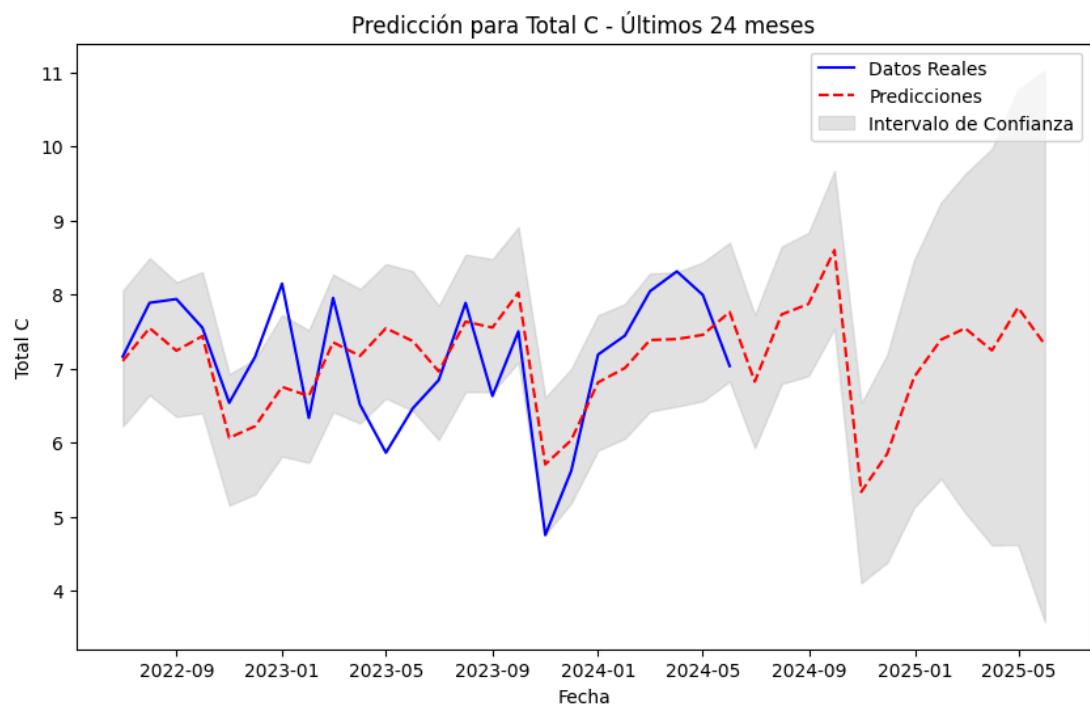
Nota. Elaboración propia utilizando Jupyter Notebook en Visual Studio Code.

Figura 36:



*Nota. Elaboración propia utilizando Jupyter Notebook en Visual Studio Code.*

Figura 37:



*Nota. Elaboración propia utilizando Jupyter Notebook en Visual Studio Code.*

El Modelo 4, que incorpora una búsqueda de hiperparámetros mediante grid search, ha demostrado el mejor desempeño general en comparación con los otros tres modelos. Esto sugiere que la optimización de los parámetros `changepoint_prior_scale`, `seasonality_prior_scale` y `yearly_seasonality` mediante grid search ha permitido ajustar el modelo de manera más precisa a las características específicas de cada serie. La optimización de los parámetros `changepoint_prior_scale` y `seasonality_prior_scale` ha permitido ajustar la

flexibilidad de la tendencia y la importancia de la estacionalidad, respectivamente, mejorando la capacidad del modelo para capturar los patrones subyacentes en los datos. La exploración de la inclusión y exclusión de la estacionalidad anual (yearly\_seasonality) ha permitido determinar la configuración más adecuada para cada serie temporal.

#### 5.5.6. Modelos Automáticos (TPOTRegressor)

En la presente sección, se aborda la implementación y evaluación de modelos de regresión automática generados mediante la herramienta TPOTRegressor (Tree-based Pipeline Optimization Tool). TPOTRegressor, una librería de Python basada en algoritmos genéticos, permite la optimización automática de pipelines de aprendizaje automático, incluyendo la selección de modelos, el preprocesamiento de datos y la optimización de hiperparámetros.

El objetivo de esta sección es explorar la capacidad de TPOTRegressor para identificar modelos de regresión efectivos para las series temporales sin la necesidad de una intervención manual exhaustiva en la selección y configuración de modelos.

Se encontraron los siguientes modelos con sus métricas de performance:

##### Variable Total MyC:

Best pipeline: LinearSVR(input\_matrix, C=5.0, dual=True, epsilon=0.0001, loss=epsilon\_insensitive, tol=0.1)  
MAE: 13.57, MSE: 370.90, RMSE: 19.26, MAPE: 21.71%

##### Variable Total M

Best pipeline: ElasticNetCV(input\_matrix, l1\_ratio=0.9500000000000001, tol=0.01)  
MAE: 12.81, MSE: 326.15, RMSE: 18.06, MAPE: 31.19%

##### Variable Total C

Best pipeline: AdaBoostRegressor(ZeroCount(input\_matrix), learning\_rate=0.1, loss=exponential, n\_estimators=100)  
MAE: 2.09, MSE: 7.56, RMSE: 2.75, MAPE: 10.14%

Estos resultados indican un rendimiento moderado, con errores absolutos y cuadráticos significativos, sugiriendo que la relación lineal modelada podría no capturar completamente la complejidad de la serie.

En un trabajo a futuro se podrían profundizar los algoritmos propuestos, tratando de mejorar hiperparámetros de cada uno de los modelos seleccionados por el algoritmo *TPOTRegressor*.

## **5.6. Modelos Multivariados (para variable “Total MyC”)**

### **5.6.1. Random Forest Multivariado**

A diferencia de los modelos univariados, que consideran únicamente la variable objetivo en su historia pasada, los modelos multivariados incorporan información de múltiples variables explicativas, permitiendo capturar relaciones más complejas y potencialmente mejorar la precisión de las predicciones.

Para la sección de modelos multivariados se decidió acotar el desarrollo de modelos, análisis y comparaciones de los mismos solo para la variable Total MyC.

Se han desarrollado dos versiones de modelos Random Forest multivariados, cada una con un enfoque distinto en la selección y transformación de variables explicativas, con el objetivo de evaluar el impacto de estas estrategias en el rendimiento del modelo.

#### Versión 1: Modelo Multivariado con Variables Explicativas Originales.

Utiliza el conjunto completo de variables explicativas disponibles en el conjunto de datos, sin realizar transformaciones o adiciones. Este enfoque permite evaluar el rendimiento del modelo utilizando la información original proporcionada por las variables explicativas.

#### Versión 2: Modelo Multivariado con Variables Explicativas Extendidas.

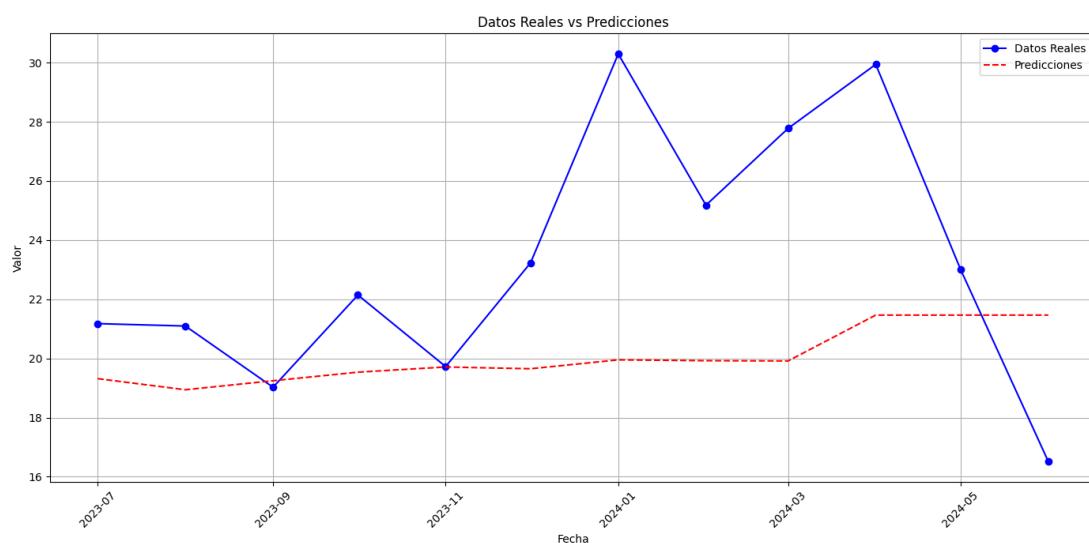
La Versión 2 del modelo multivariado extiende el conjunto de variables explicativas mediante la inclusión de rezagos (lags) de la variable objetivo Total MyC, específicamente los rezagos 1 y 2, así como la media móvil de 3 meses de la misma variable. La inclusión de estas

variables derivadas tiene como objetivo capturar patrones temporales adicionales y mejorar la capacidad del modelo para predecir la dinámica de la serie Total MyC.

A continuación, se presentará un análisis comparativo de las dos versiones de modelos Random Forest multivariados, basado en las métricas de rendimiento definidas previamente, con el fin de identificar la versión que ofrece el mejor equilibrio entre precisión y generalización.

*V1. Utilizando el total de variables explicativas y sin agregar nuevas variables al modelo.*

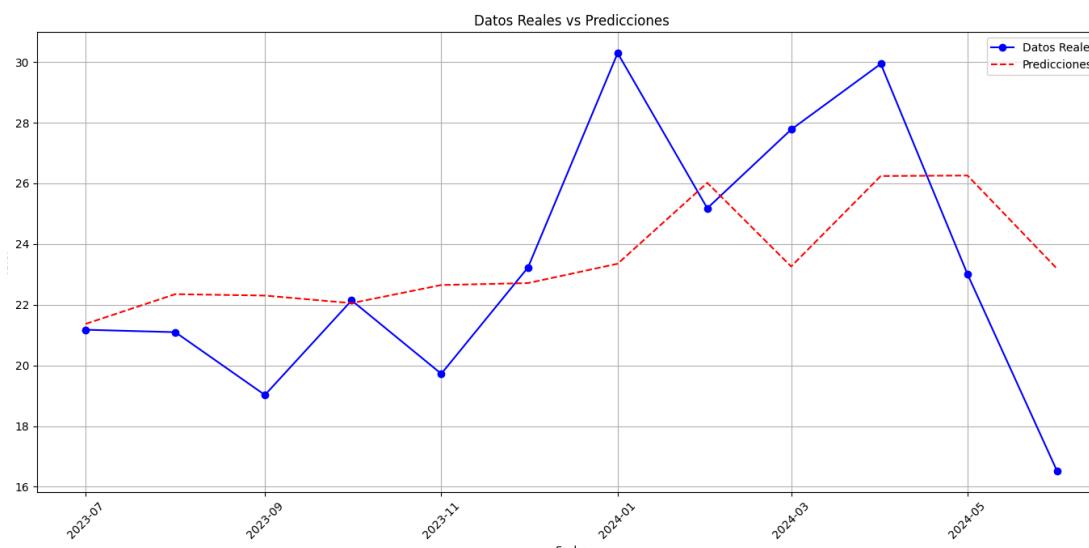
*Figura 38:*



*Nota. Elaboración propia utilizando Jupyter Notebook en Visual Studio Code.*

*V2. Utilizando el total de variables explicativas y agregando nuevas variables al modelo (Lag 1 y 2 y media móvil de 3 meses de la variable target).*

*Figura 39:*



*Nota. Elaboración propia utilizando Jupyter Notebook en Visual Studio Code.*

### 5.6.2. Gradient Boosting Regressor Multivariado

En la sección actual se han desarrollado dos versiones de modelos Gradient Boosting Regressor multivariados, cada una con un enfoque distinto en la ingeniería de características y la optimización de hiperparámetros, con el objetivo de evaluar el impacto de estas estrategias en el rendimiento de los modelos.

#### Versión 1: Modelo Multivariado con Características Extendidas (Lags y Media Móvil)

La Versión 1 del modelo Gradient Boosting Regressor multivariado utiliza el conjunto completo de variables explicativas disponibles, extendiéndose con la inclusión de rezagos (lags) de la variable objetivo, específicamente los rezagos 1 y 2, así como la media móvil de 3 meses de la misma variable. Este enfoque busca capturar patrones temporales adicionales y mejorar la capacidad del modelo para predecir la dinámica de la variable objetivo.

#### Versión 2: Modelo Multivariado con Ingeniería de Características Avanzada y Optimización de Hiperparámetros

La Versión 2 del modelo multivariado incorpora una ingeniería de características más avanzada, incluyendo:

- Rezagos (Lags): Se incluyen los rezagos 1, 3 y 12 de la variable objetivo, buscando capturar patrones temporales de corto, mediano y largo plazo.
- Medias Móviles: Se calculan las medias móviles de 3 y 6 meses de la variable objetivo, suavizando las fluctuaciones y resaltando las tendencias.
- Diferenciación Estacional: se calcula la diferencia entre el valor actual y el valor 12 meses atrás, con el objetivo de modelar la estacionalidad presente en la serie temporal.
- Búsqueda de hiperparámetros mediante grid search, explorando el siguiente espacio de parámetros:
  - ◆ param\_grid = { 'n\_estimators': [100, 200, 300],
  - ◆ 'learning\_rate': [0.01, 0.05, 0.1],
  - ◆ 'max\_depth': [3, 4, 5],

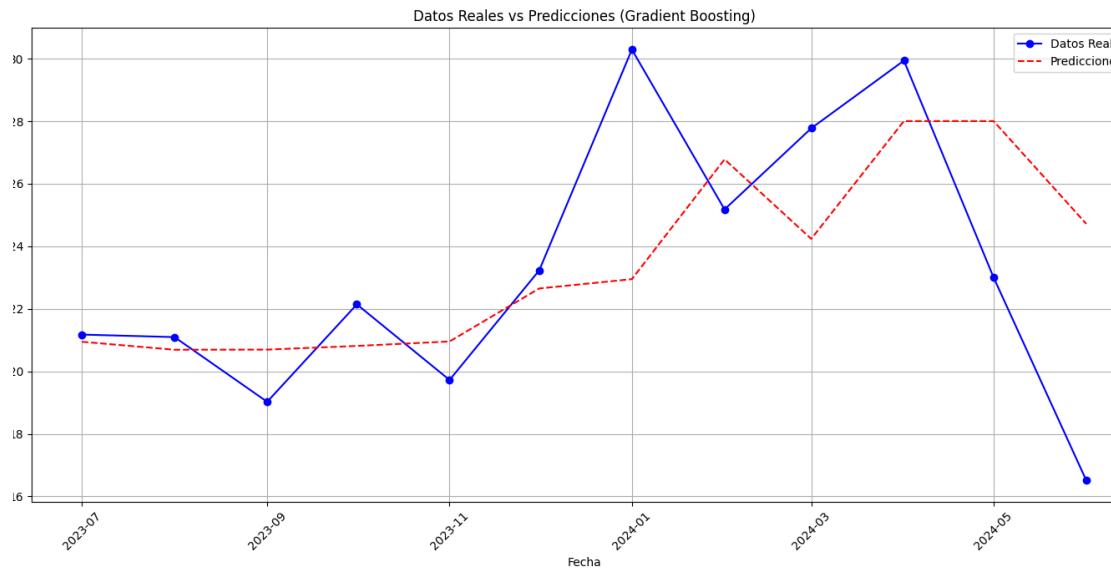
◆ 'min\_samples\_leaf': [1, 2, 4] }

Este proceso de optimización busca identificar la combinación de hiperparámetros que maximice el rendimiento del modelo.

A continuación, se presentará un análisis comparativo de las dos versiones de modelos Gradient Boosting Regressor multivariados, basado en las métricas de rendimiento definidas previamente, con el fin de identificar la versión que ofrece el mejor equilibrio entre precisión y generalización.

*V1. Utilizando el total de variables explicativas y se agregan nuevas variables al modelo (Lag 1 y 2 y media móvil de 3 meses de la variable target).*

Figura 40:



*Nota. Elaboración propia utilizando Jupyter Notebook en Visual Studio Code.*

*V2. Utilizando el total de variables explicativas y se agregan nuevas variables al modelo:*

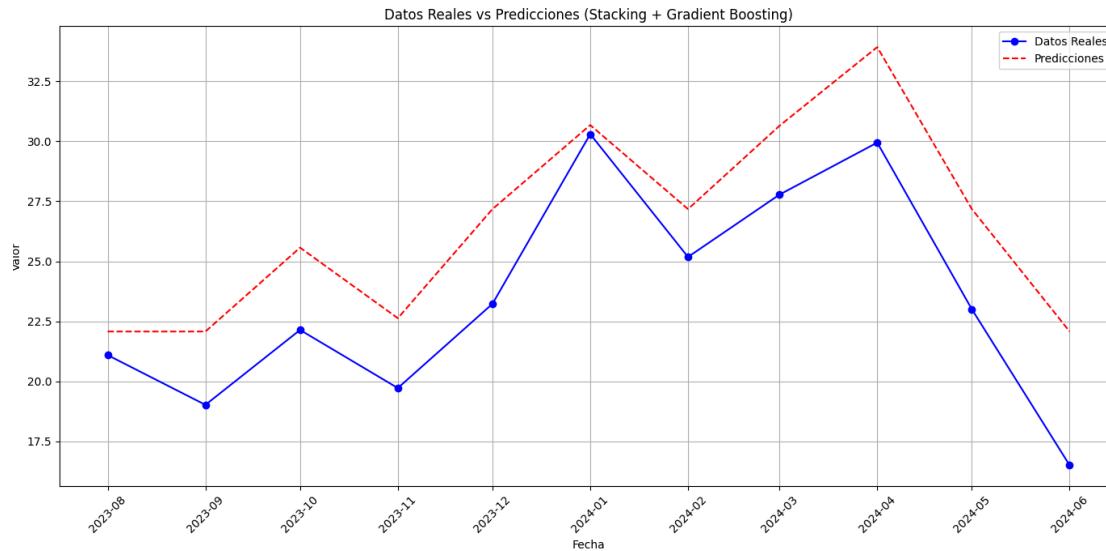
*Lag 1, 3 y 12,*

*Media móvil de 3 y 6 meses*

*Diferenciación estacional: se calcula la diferencia entre el valor actual y el valor 12 meses atrás, lo cual puede ayudar a modelar la estacionalidad.*

*Búsqueda de hiperparámetros a través de param grid:*

Figura 41:



Nota. Elaboración propia utilizando Jupyter Notebook en Visual Studio Code.

### 5.6.3. LSTM Multivariado

Se ha desarrollado una única versión del modelo LSTM multivariado, que se caracteriza por la inclusión de un conjunto extendido de variables explicativas, diseñado para capturar patrones temporales de diversa naturaleza.

Específicamente, se han incorporado las siguientes variables derivadas:

- Rezagos (Lags): Se incluyen los rezagos 1, 3 y 12 de la variable objetivo, buscando capturar patrones temporales de corto, mediano y largo plazo.
- Medias Móviles: Se calculan las medias móviles de 3 y 6 meses de la variable objetivo, suavizando las fluctuaciones y resaltando las tendencias.
- Diferenciación Estacional: Se calcula la diferencia entre el valor actual y el valor 12 meses atrás, con el objetivo de modelar la estacionalidad presente en la serie temporal.

Este enfoque de ingeniería de características busca enriquecer la información disponible para el modelo LSTM, permitiendo una mejor captura de la dinámica subyacente en los datos.

A continuación, se presentará un análisis detallado de la arquitectura del modelo LSTM y los resultados obtenidos en la evaluación de su rendimiento.

La arquitectura del modelo LSTM implementado se puede describir en los siguientes componentes y pasos:

1. Creación de Secuencias de Tiempo:

→ Función `create_sequences(X, y, time_steps=1)`: para transformar los datos en un formato adecuado para las redes LSTM, que procesan secuencias de tiempo.

→ Entrada:

X: La matriz de características (variables explicativas).

y: La serie temporal objetivo que se va a predecir.

→ Time\_steps: el número de pasos de tiempo (lag) que se utilizarán para crear cada secuencia. La función itera a través de los datos y crea ventanas deslizantes de longitud `time_steps`. Para cada ventana, se crea una secuencia de características (elementos de X) y la etiqueta correspondiente es el valor de y en el siguiente paso de tiempo después de la ventana.

→ Salida:

Xs: Un array NumPy que contiene las secuencias de características. La forma de Xs será (`número_de_secuencias, time_steps, número_de_características`).

ys: Un array NumPy que contiene las etiquetas correspondientes a cada secuencia.

Se define `time_steps = 24`, lo que significa que cada secuencia de entrada para el modelo LSTM consistirá en 24 pasos de tiempo. Se aplica la función `create_sequences` a los datos escalados (`X_scaled` y `y_scaled`) para generar los datos de entrada `X_lstm` y las etiquetas `y_lstm` en el formato requerido.

División de Datos en *train* y *test*: se divide el conjunto de datos secuencial `X_lstm` y `y_lstm` en conjuntos de entrenamiento y prueba. Se utiliza un índice de división para reservar el final de los datos como conjunto de prueba. `X_train, X_test`: Los conjuntos de secuencias de entrenamiento y prueba. `y_train, y_test`: Las etiquetas de entrenamiento y prueba.

Definición del Modelo LSTM: se crea un modelo secuencial utilizando la librería Keras.

Capa 1: con 100 unidades (celdas de memoria). `return_sequences=True`: Indica que esta capa devuelve todas las salidas de la secuencia, lo cual es necesario para apilar otra capa LSTM encima.

Capa 2: Dropout `Dropout(0.3)`: Una capa de dropout con una tasa de dropout del 30% (técnica de regularización que ayuda a prevenir el sobreajuste al desactivar aleatoriamente una fracción de las neuronas durante el entrenamiento).

Capa 3: una segunda capa LSTM con 50 unidades. `return_sequences=False`: Indica que esta capa solo devuelve la última salida de la secuencia.

Capa 4: Dropout `Dropout(0.2)`: Una capa de dropout con una tasa de dropout del 20%.

Capa 5: `Dense(units=25, activation='relu')`: una capa densa (totalmente conectada) con 25 unidades y la función de activación ReLU (Rectified Linear Unit).

Capa 6: `Dense (Salida) Dense(units=1)`: la capa de salida, que tiene una sola unidad, ya que se trata de un problema de regresión (predicción de un valor único).

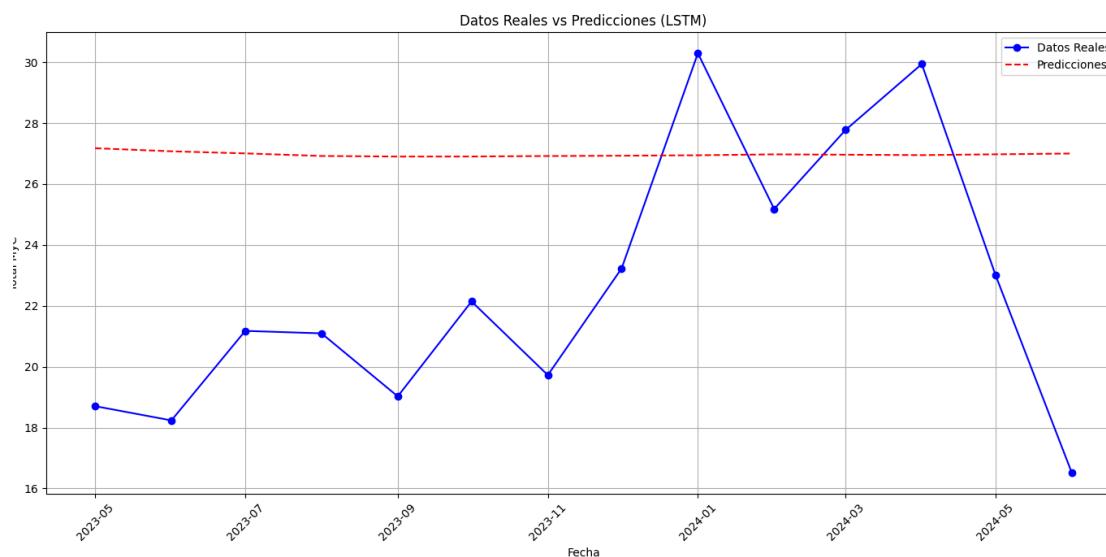
Se compila el modelo antes del entrenamiento con `optimizer='adam'`: se utiliza el optimizador Adam, un algoritmo de optimización eficiente y popular para el entrenamiento de redes neuronales y `loss='mean_squared_error'`: se utiliza la función de pérdida del error cuadrático medio (MSE), que es apropiada para problemas de regresión.

Entrenamiento del Modelo: se define un mecanismo de early stopping para detener el entrenamiento cuando el rendimiento en el conjunto de validación deja de mejorar. `monitor='val_loss'`: Se monitorea la pérdida en el conjunto de validación. `patience=10`: el entrenamiento se detendrá si la pérdida en el conjunto de validación no mejora durante 10 épocas consecutivas. `restore_best_weights=True`: se restauran los pesos del modelo correspondientes a la época con el mejor rendimiento en el conjunto de validación.

Ajuste del Modelo: history = model.fit(...): se entrena el modelo utilizando los datos de entrenamiento. El modelo se entrena durante un máximo de 100 épocas. Se utiliza un tamaño de lote de 16.

Se realizan predicciones en el conjunto de prueba. Las predicciones iniciales (y\_pred\_scaled) están en la escala normalizada. Inversión de la escala: y\_pred = scaler\_y.inverse\_transform(y\_pred\_scaled). Se invierte la escala de las predicciones para volver a la escala original de los datos. y\_test\_inv = scaler\_y.inverse\_transform(y\_test): Se invierte la escala de las etiquetas del conjunto de prueba.

Figura 42:



Nota. Elaboración propia utilizando Jupyter Notebook en Visual Studio Code.

#### 5.6.4. Híbrido Random Forest con residuos de LSTM

Este enfoque híbrido como su nombre lo indica busca aprovechar la capacidad del Random Forest para capturar relaciones no lineales en los datos, complementándola con la habilidad del LSTM para modelar patrones secuenciales en los errores de predicción del Random Forest.

##### Entrenamiento del Modelo Random Forest:

1. **Modelo:** Se utiliza `RandomForestRegressor`, un modelo de aprendizaje automático basado en árboles de decisión que opera en múltiples subconjuntos del conjunto de datos y combina los resultados para mejorar la precisión y controlar el sobreajuste.

2. **Entrenamiento:** El modelo se entrena con el conjunto de datos de entrenamiento (`X_train`, `y_train`) que ha sido previamente escalado. Durante el entrenamiento, el Random Forest aprende a predecir los valores de la variable objetivo (`y_train`) a partir de las variables explicativas (`X_train`).
3. **Predicciones del Modelo:** Una vez entrenado, el modelo se utiliza para predecir los valores en el conjunto de prueba (`X_test`), generando `y_pred_rf`, que son las predicciones del modelo Random Forest para la variable objetivo.
4. **Cálculo de Residuos:** se calculan como la diferencia entre los valores reales (`y_test`) y las predicciones del modelo Random Forest (`y_pred_rf`). Estos residuos indican cuán lejos están las predicciones del modelo de los valores reales y reflejan errores que el modelo no ha podido capturar.
5. **Uso de Residuos:** Los residuos calculados se usarán en el siguiente paso para que el modelo LSTM intente modelar estos errores y mejorar las predicciones finales.

### Creación de Secuencias de Tiempo para LSTM

1. **Motivación:** El modelo LSTM (Long Short-Term Memory) es un tipo de red neuronal recurrente diseñada para capturar patrones en secuencias de tiempo, lo que lo hace especialmente útil para modelar relaciones temporales que el modelo Random Forest podría no capturar.
2. **Creación de Secuencias:** para que el LSTM pueda aprender patrones en los residuos, los datos de entrada (`X_test`) y los residuos (`residuals`) se organizan en secuencias de tiempo utilizando la función `create_sequences`. Esta función divide los datos en pequeñas secuencias temporales de longitud `time_steps` (en este caso, 12). Esto significa que para predecir el residuo en un punto de tiempo dado, el modelo utilizará información de los 12 períodos anteriores.
3. **Salida:** La función devuelve `X_lstm`, que es un array 3D donde cada muestra contiene 12 pasos de tiempo de las variables explicativas, y `y_lstm`, que son los residuos correspondientes a esos pasos de tiempo.

### Entrenamiento del Modelo LSTM para Predecir Residuos

1. **Definición del Modelo LSTM:** Se define un modelo secuencial LSTM con múltiples capas. Las capas LSTM son capaces de aprender dependencias a largo plazo en secuencias de tiempo.

- La primera capa tiene 200 unidades y está configurada para devolver secuencias, lo que permite que las siguientes capas LSTM también trabajen con secuencias.
- La segunda capa tiene 100 unidades y también devuelve secuencias.
- La tercera capa tiene 50 unidades y no devuelve secuencias, lo que significa que esta capa procesa toda la secuencia de entrada y devuelve un solo valor para cada muestra.
- **Dropout:** Se añade dropout para prevenir el sobreajuste, desconectando aleatoriamente neuronas durante el entrenamiento.
- **Salida:** La capa final es una capa densa (fully connected) con una sola unidad, que predice el residuo para cada muestra.

## 2. Entrenamiento del Modelo LSTM:

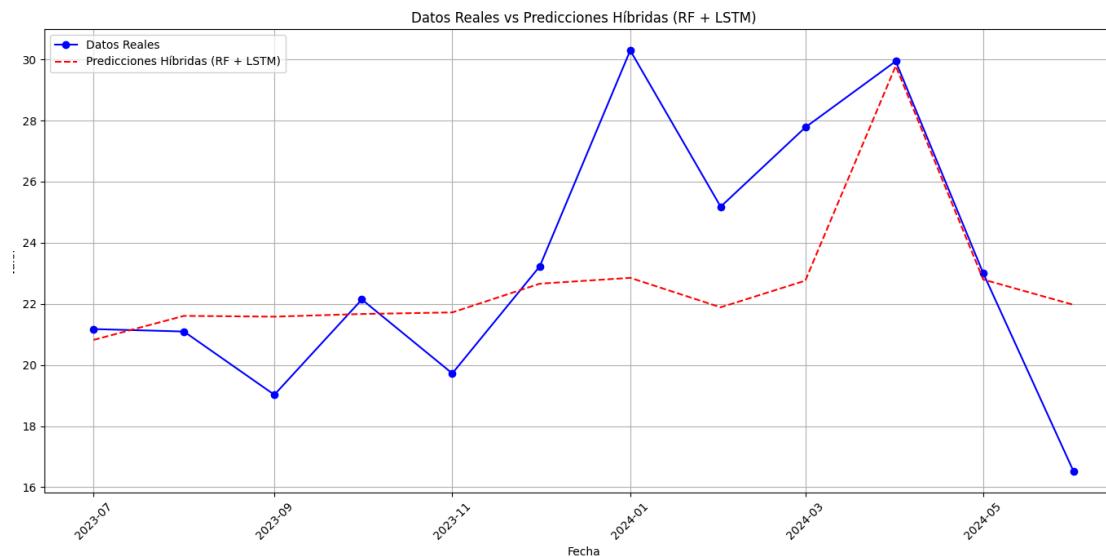
- **Compilación:** El modelo se compila con el optimizador 'adam' y se utiliza la función de pérdida `mean_squared_error`, que es adecuada para problemas de regresión.
- **Early Stopping:** Se usa `EarlyStopping`, una técnica que detiene el entrenamiento si la pérdida de validación no mejora después de varias épocas, para evitar el sobreentrenamiento.
- **Entrenamiento:** El modelo se entrena utilizando `X_lstm` y `y_lstm` con un conjunto de validación para ajustar el modelo.

3. **Predicción de Residuos:** después del entrenamiento, el modelo LSTM se utiliza para predecir los residuos (`residuals_pred`). Estos residuos predichos son el intento del modelo LSTM de capturar errores que el modelo Random Forest no pudo corregir.

4. **Corrección de Predicciones del Random Forest:** las predicciones finales se obtienen sumando los residuos predichos por el LSTM a las predicciones originales del modelo Random Forest (`y_pred_rf`). Esto ajusta las predicciones del Random Forest con la información adicional capturada por el LSTM.

Este proceso mejora las predicciones al combinar la capacidad del Random Forest para capturar relaciones no lineales con la habilidad del LSTM para modelar patrones secuenciales en los errores.

Figura 43:



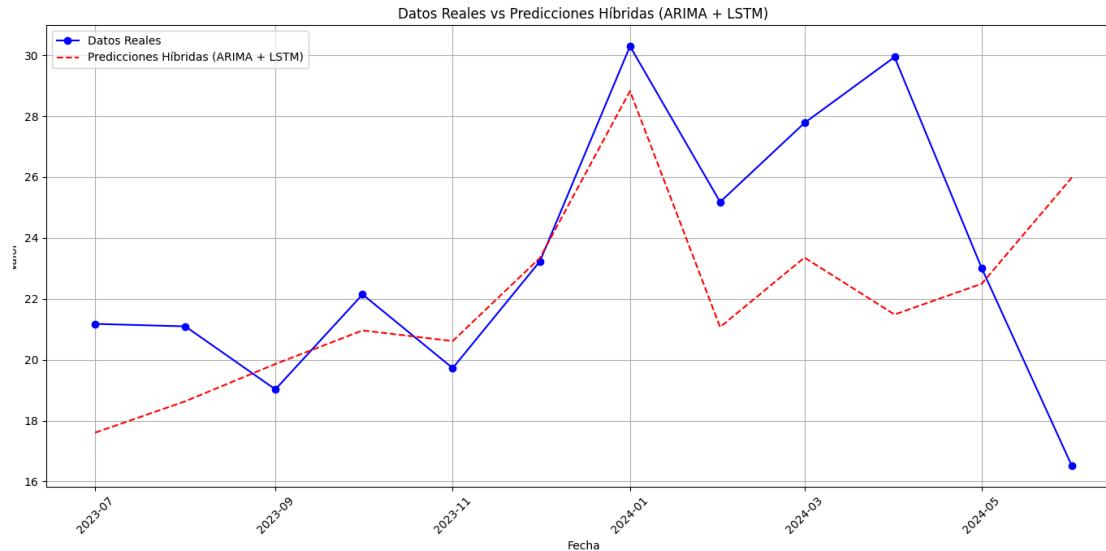
Nota. Elaboración propia utilizando Jupyter Notebook en Visual Studio Code.

#### 5.6.5. Híbrido Autoarima con residuos de LSTM

Este enfoque híbrido sigue una lógica similar al modelo anterior, que combinaba Random Forest Regressor con LSTM, pero en este caso, se utiliza un modelo AutoARIMA como base para la predicción inicial, complementándolo con un modelo LSTM para modelar los residuos.

Se utiliza `auto_arima` para ajustar un modelo ARIMA en los datos de entrenamiento. `auto_arima` encuentra automáticamente los mejores parámetros  $p$ ,  $d$ ,  $q$  para un modelo ARIMA, teniendo en cuenta la estacionalidad.

Figura 44:



Nota. Elaboración propia utilizando Jupyter Notebook en Visual Studio Code.

#### **5.6.6. Random Feature Addition / Random Noise Injection (“pajaritos”)**

Random Feature Addition / Random Noise Injection, también conocida como 'pajaritos', como método para identificar variables irrelevantes en un conjunto de datos. Esta técnica, de carácter intuitivo, consiste en la adición de una o más columnas de datos aleatorios (ruido) al conjunto de datos original, seguida de la observación del impacto de estas variables aleatorias en el rendimiento del modelo predictivo. El fundamento de esta técnica radica en la hipótesis de que, si el rendimiento del modelo no se ve significativamente afectado por la inclusión de variables aleatorias, es probable que existan variables originales en el conjunto de datos que no aportan valor predictivo relevante. En otras palabras, la presencia de variables irrelevantes puede enmascarar la verdadera importancia de las variables predictivas relevantes. Para la implementación de esta técnica, se ha utilizado la librería scikit-learn (sklearn) y el módulo Recursive Feature Elimination (RFE).

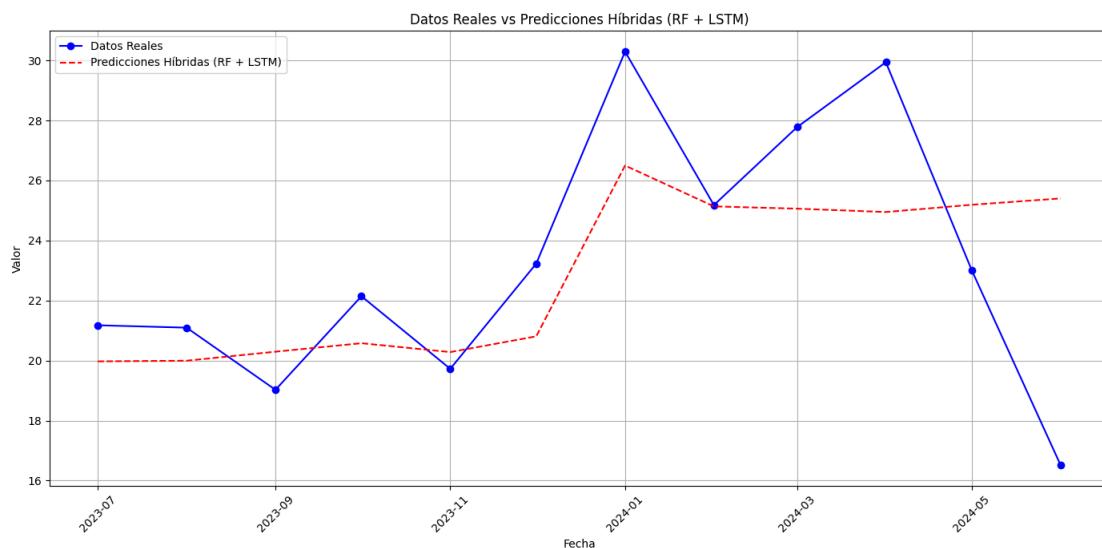
Inicialmente, se ha realizado un análisis con la selección de 50 variables, seguido de un análisis refinado con la selección de las 10 mejores variables identificadas en el primer paso. Este enfoque iterativo permite identificar de manera progresiva las variables más relevantes para el modelo predictivo.

#### **5.6.7. Híbrido Random Forest con residuos de LSTM (usando las 10 variables seleccionadas con RFE)**

Se retoman los modelos híbridos mencionados previamente pero utilizando solo las 10 variables explicativas de mayor relevancia según el RFE aplicado previamente.

*En V1 sin lags ni medias móviles ni diferenciación estacional.*

Figura 45:



Nota. Elaboración propia utilizando Jupyter Notebook en Visual Studio Code.

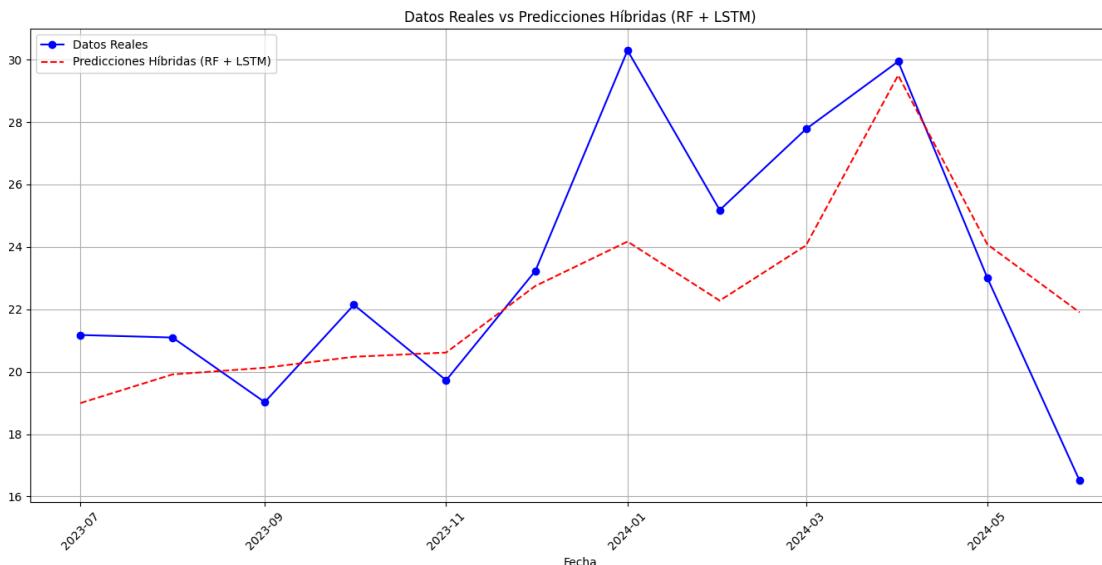
V2. Idem anterior agregando:

Lags de 1, 3 y 12 meses

Medias móviles 3 y 6 meses

Diferenciación estacional 12 meses.

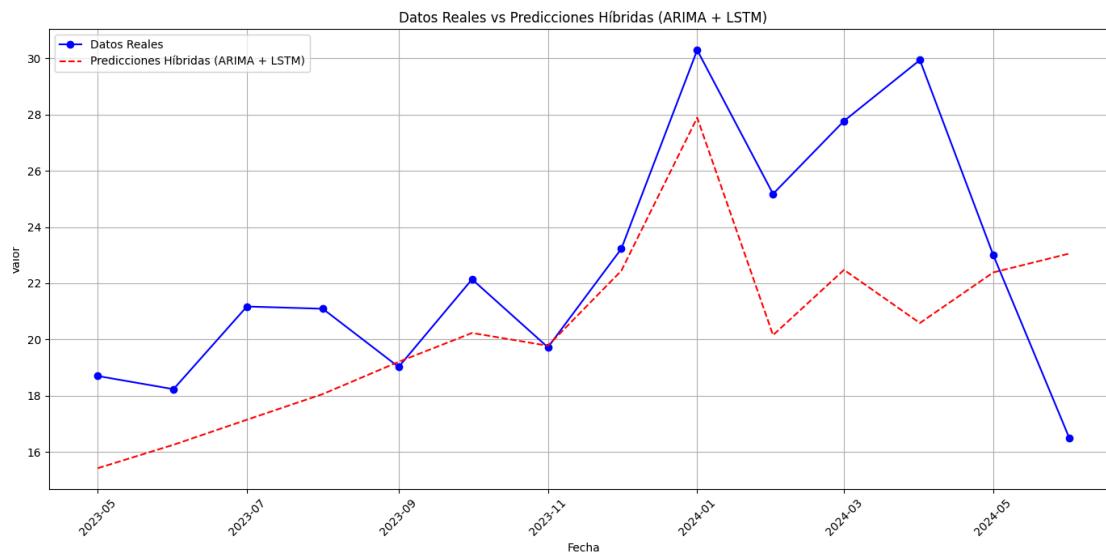
Figura 46:



Nota. Elaboración propia utilizando Jupyter Notebook en Visual Studio Code.

### 5.6.8. Híbrido Autoarima con residuos de LSTM (usando las 10 variables seleccionadas con RFE)

Figura 47:



Nota. Elaboración propia utilizando Jupyter Notebook en Visual Studio Code.

Comentarios sobre los modelos multivariados y sus resultados de performance:

**Ventajas:**

- Captura de interacciones complejas:** Los modelos multivariados pueden capturar interacciones entre múltiples variables, permitiendo que el modelo comprenda relaciones más complejas en los datos. Esto es especialmente útil en conjuntos de datos con múltiples factores que afectan la variable objetivo.
- Mejora de la precisión:** Al incluir múltiples variables explicativas, estos modelos tienden a mejorar la precisión de las predicciones en comparación con modelos univariados. Como se observa en los resultados, los modelos que utilizan más variables explicativas (y técnicas como lags, medias móviles y diferenciación estacional) suelen tener un mejor desempeño en términos de MAE, MSE, RMSE y MAPE.
- Flexibilidad:** Los modelos multivariados permiten la incorporación de diversas transformaciones de datos, como lags y medias móviles, lo que puede mejorar la modelación de fenómenos como la estacionalidad y las tendencias.

**Desventajas:**

- Mayor complejidad:** A medida que se incorporan más variables y transformaciones, los modelos se vuelven más complejos y difíciles de interpretar. La selección adecuada de variables y la prevención del sobreajuste se vuelven cruciales.

2. **Mayor requerimiento computacional:** Modelos multivariados, especialmente los que combinan técnicas como Random Forest y LSTM, requieren más recursos computacionales tanto para el entrenamiento como para la predicción, lo que puede ser un desafío en entornos con recursos limitados.
3. **Riesgo de multicolinealidad:** La inclusión de múltiples variables explicativas puede introducir multicolinealidad, donde algunas variables están altamente correlacionadas entre sí, lo que puede afectar la estabilidad y la interpretabilidad del modelo.
4. **Estimación de variables explicativas:** Para predecir valores futuros, los modelos multivariados no solo requieren el valor pasado de la variable objetivo, sino también las futuras observaciones de todas las variables explicativas incluidas en el modelo. Si estos datos no están disponibles, se deben estimar, lo que introduce una capa adicional de incertidumbre y potencial error en las predicciones.
5. **Complejidad en la recolección de datos:** A medida que se incrementa el número de variables explicativas en un modelo multivariado, también aumenta la necesidad de recopilar datos para cada una de esas variables. Esto puede ser un desafío, especialmente en entornos donde la disponibilidad de datos es limitada o donde la calidad de los datos puede variar.

### **Random Forest Multivariado**

- **V1:**
  - **MAE: 12.80, RMSE: 16.14**
  - Desempeño moderado, pero claramente mejorado en la versión V2.
- **V2:**
  - **MAE: 8.81, RMSE: 11.41**
  - La incorporación de lags y medias móviles resultó en una mejora significativa en la precisión del modelo.

### **Gradient Boosting Regressor Multivariado**

- **V1:**
  - **MAE: 8.39, RMSE: 11.18**
  - Muestra un buen desempeño con las nuevas variables introducidas.
- **V2:**
  - **MAE: 8.76, RMSE: 10.22**

- Ligeramente inferior en MAE pero mejora en RMSE. El ajuste de hiperparámetros no condujo a una mejora significativa.

### **LSTM Multivariado**

- **Solo una versión:**
  - **MAE: 12.93, RMSE: 17.19**
  - Aunque captura bien las relaciones no lineales y dependencias temporales, su desempeño es inferior al de otros modelos como Gradient Boosting y Random Forest.

### **Híbrido Random Forest con Residuos de LSTM**

- **V1:**
  - **MAE: 8.48, RMSE: 11.24**
  - Mejora sobre el Random Forest puro, aprovechando las capacidades de LSTM para modelar residuos no capturados.
- **V2 (con RFE):**
  - **MAE: 7.51, RMSE: 10.46**
  - Utilizar RFE para seleccionar variables redujo significativamente el error, demostrando la importancia de la selección de características.

### **Híbrido AutoARIMA con Residuos de LSTM**

- **V1:**
  - **MAE: 10.17, RMSE: 13.72**
  - Un enfoque sólido, pero con un rendimiento inferior al híbrido basado en Random Forest.
- **V2 (con RFE):**
  - **MAE: 10.05, RMSE: 12.91**
  - La selección de variables mejoró el rendimiento, pero sigue siendo menos eficiente que el modelo híbrido Random Forest con RFE.

### **Conclusiones Modelos Multivariados.**

Los modelos multivariados ofrecen una mejora significativa en la precisión de las predicciones al capturar interacciones complejas entre variables. Sin embargo, el incremento en la complejidad y los recursos computacionales puede ser un desafío, como así también la

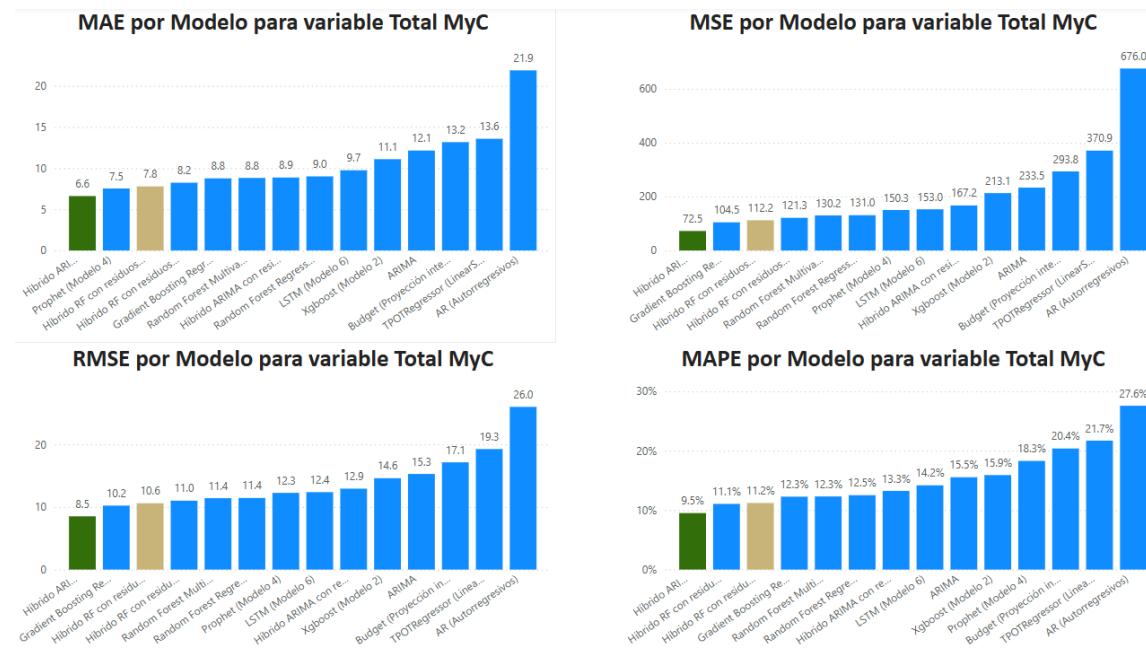
recopilación de datos. Entre los modelos evaluados, las versiones que combinan Random Forest con la corrección de residuos mediante LSTM, especialmente aquellas que aplican técnicas de selección de características como RFE, mostraron un rendimiento superior en términos de MAE, RMSE y MAPE, destacándose como las más efectivas para la predicción multivariada en este contexto.

## 6. Comparaciones de mejores modelos y conclusiones generales

A continuación se presentan los resultados comparando los diferentes modelos y sus métricas de performance.

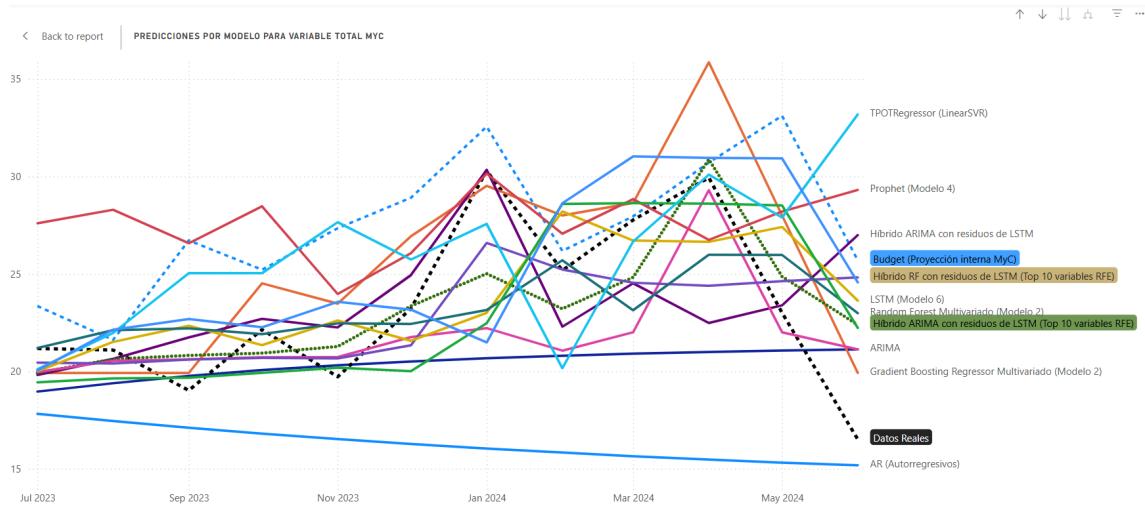
### 6.1. Conclusiones para variable “Total MyC”

*Figura 48:*



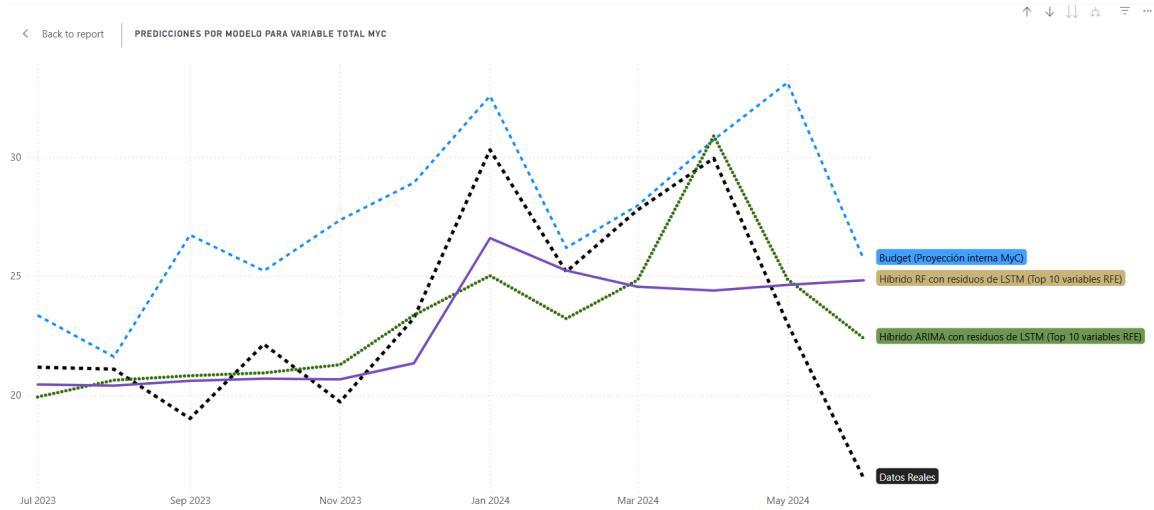
*Nota. Elaboración propia utilizando Power BI.*

Figura 49:



Nota. Elaboración propia utilizando Power BI.

Figura 50:



Nota. Elaboración propia utilizando Power BI.

Cuadro 1:

Modelo	MAE	MSE	RMSE	MAPE
Híbrido ARIMA con residuos de LSTM (Top 10 variables RFE)	6.6	72.5	8.5	9.5%
Híbrido RF con residuos de LSTM	8.2	121.3	11.0	11.1%
Híbrido RF con residuos de LSTM (Top 10 variables RFE)	7.8	112.2	10.6	11.2%
Gradient Boosting Regressor Multivariado (Modelo 2)	8.8	104.5	10.2	12.3%
Random Forest Multivariado (Modelo 2)	8.8	130.2	11.4	12.3%
Random Forest Regressor (Ventana 108 meses)	9.0	131.0	11.4	12.5%
Híbrido ARIMA con residuos de LSTM	8.9	167.2	12.9	13.3%
LSTM (Modelo 6)	9.7	153.0	12.4	14.2%
ARIMA	12.1	233.5	15.3	15.5%
Xgboost (Modelo 2)	11.1	213.1	14.6	15.9%
Prophet (Modelo 4)	7.5	150.3	12.3	18.3%
Budget (Proyección interna MyC)	13.2	293.8	17.1	20.4%
TPOTRegressor (LinearSVR)	13.6	370.9	19.3	21.7%
AR (Autorregresivos)	21.9	676.0	26.0	27.6%

Nota. Elaboración propia utilizando Power BI.

## Análisis Comparativo

### Mejores Modelos:

- i. **Híbrido ARIMA con residuos de LSTM (Top 10 variables RFE):** Este modelo presenta los mejores resultados en todas las métricas, con MAE de 6.61, MSE de 72.45, RMSE de 8.51 y MAPE de 9.51%. Esto indica que es el modelo más preciso y confiable para predecir la variable "Total MyC".
- ii. **Híbrido RF con residuos de LSTM (Top 10 variables RFE):** También muestra un rendimiento destacado, con MAE de 7.78 y MSE de 112.24, sugiriendo que es una excelente opción para mejorar las predicciones.

La implementación de modelos multivariados ha demostrado su capacidad para mejorar significativamente la precisión predictiva mediante la captura de interacciones complejas entre variables. No obstante, esta mejora en el rendimiento predictivo se acompaña de un incremento en la complejidad del modelo y los requerimientos computacionales, así como de desafíos relacionados con la gestión de la multicolinealidad y la disponibilidad de datos.

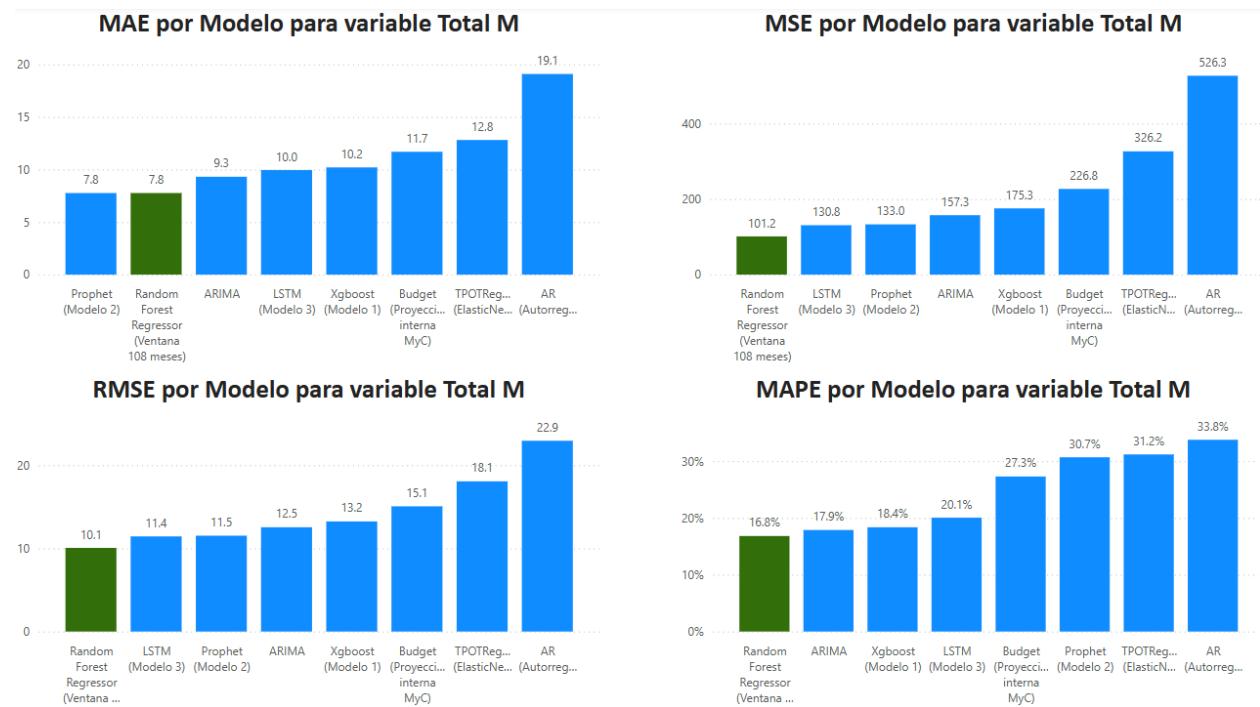
La inclusión de múltiples variables explicativas y la aplicación de técnicas de ingeniería de características, como la incorporación de rezagos (lags), medias móviles y diferenciación estacional, han demostrado ser estrategias efectivas para mejorar las métricas de rendimiento.

Específicamente, los modelos que combinan Random Forest o ARIMA con la corrección de residuos mediante LSTM, especialmente aquellos que implementan Recursive Feature Elimination (RFE) para la selección de características, han sobresalido en términos de precisión predictiva.

La necesidad de recopilar datos para todas las variables explicativas representa un desafío logístico, especialmente en entornos con limitaciones de datos.

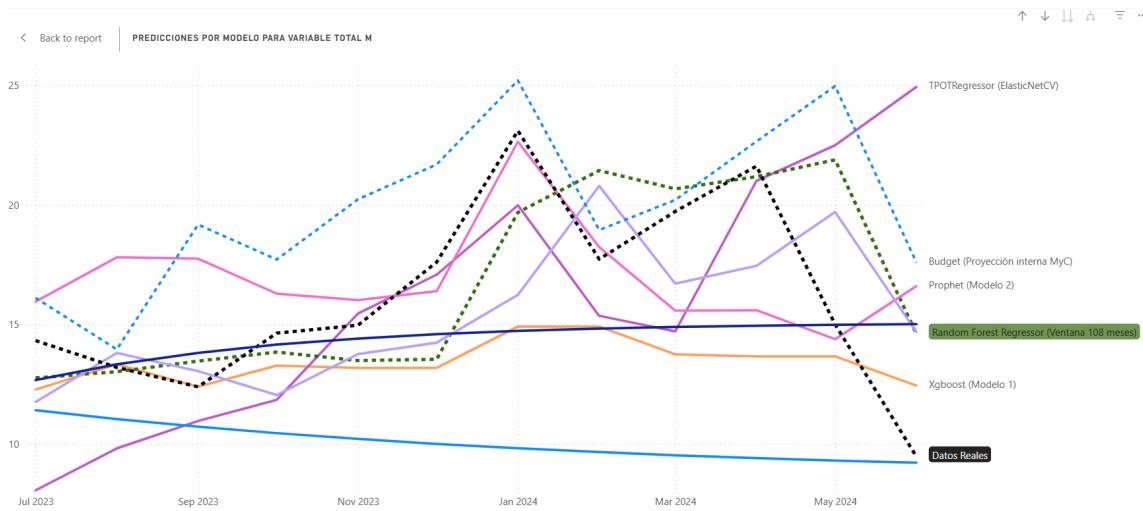
## 6.2. Conclusiones para variable “Total M”

Figura 51:



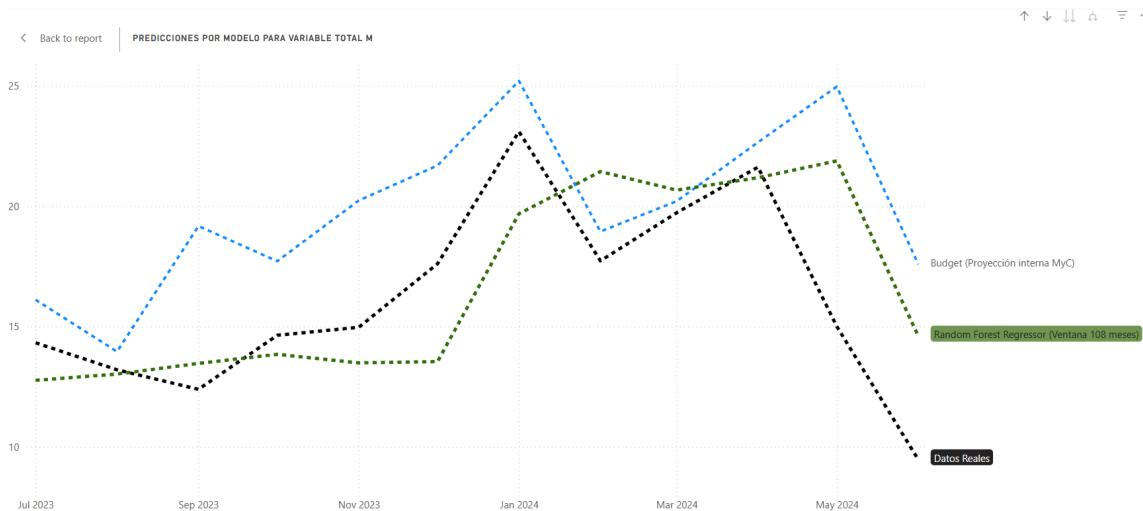
Nota. Elaboración propia utilizando Power BI.

**Figura 52:**



*Nota. Elaboración propia utilizando Power BI.*

**Figura 53:**



*Nota. Elaboración propia utilizando Power BI.*

**Cuadro 2:**

Modelo	MAE	MSE	RMSE	MAPE
Random Forest Regressor (Ventana 108 meses)	7.8	101.2	10.1	16.8%
ARIMA	9.3	157.3	12.5	17.9%
Xgboost (Modelo 1)	10.2	175.3	13.2	18.4%
LSTM (Modelo 3)	10.0	130.8	11.4	20.1%
Budget (Proyección interna MyC)	11.7	226.8	15.1	27.3%
Prophet (Modelo 2)	7.8	133.0	11.5	30.7%
TPOTRegressor (ElasticNetCV)	12.8	326.2	18.1	31.2%
AR (Autorregresivos)	19.1	526.3	22.9	33.8%

*Nota. Elaboración propia utilizando Power BI.*

## Análisis Comparativo

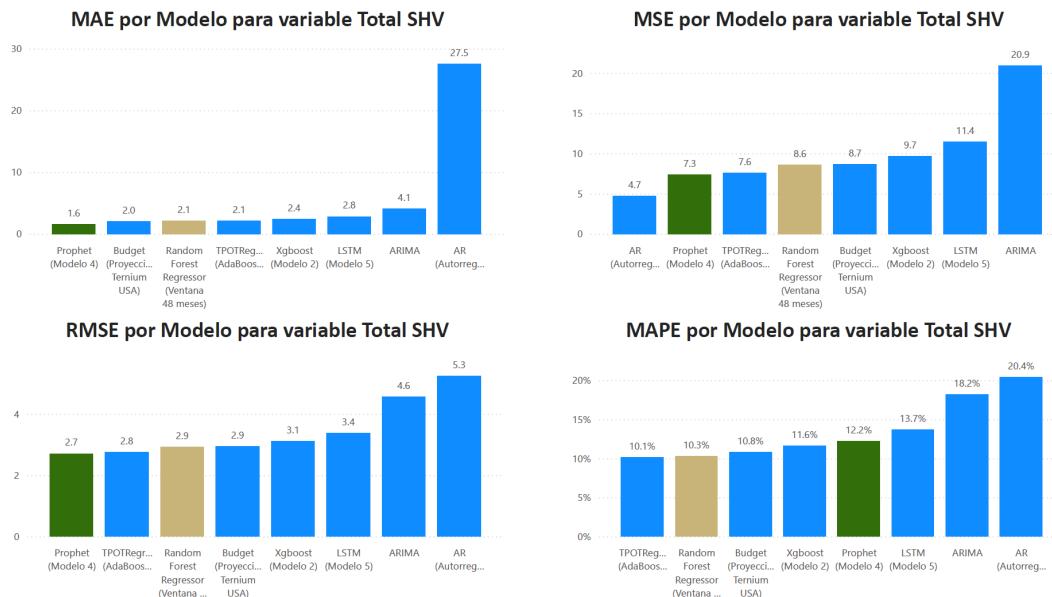
El modelo Random Forest Regressor, configurado con una ventana temporal de 108 meses, ha demostrado ser el modelo univariado con el rendimiento predictivo más robusto para la serie temporal Total M. Este modelo ha alcanzado un Error Absoluto Medio (MAE) de 7.76, un Error Cuadrático Medio (MSE) de 101.16, una Raíz del Error Cuadrático Medio (RMSE) de 10.06 y un Error Porcentual Absoluto Medio (MAPE) de 16.82%.

El éxito del modelo Random Forest en este contexto se atribuye, en gran medida, a su capacidad para capturar patrones estacionales mediante la utilización de ventanas temporales extensas. La inclusión de rezagos (lags) de largo alcance, implícita en la ventana de 108 meses, ha permitido al modelo aprender y extraer la dinámica estacional de la serie temporal de manera efectiva. La selección de una ventana temporal de 108 meses sugiere que la serie temporal analizada exhibe dependencias temporales de largo alcance, con patrones estacionales que se manifiestan en un horizonte temporal amplio.

El modelo Random Forest, al incorporar esta información temporal extensa, ha logrado superar a otros modelos univariados en términos de precisión predictiva.

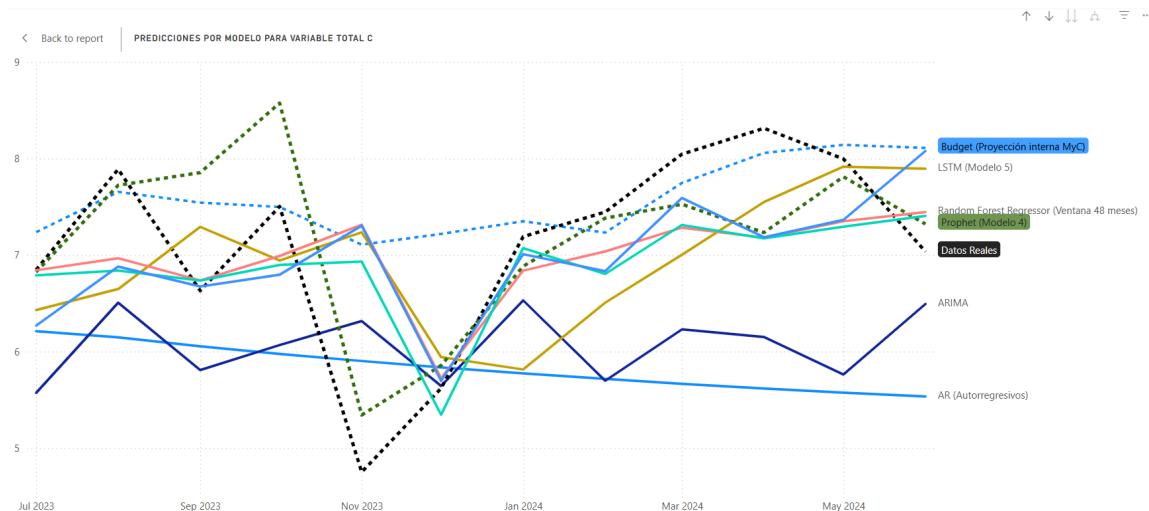
### 6.3. Conclusiones para variable “Total C”

*Figura 54:*



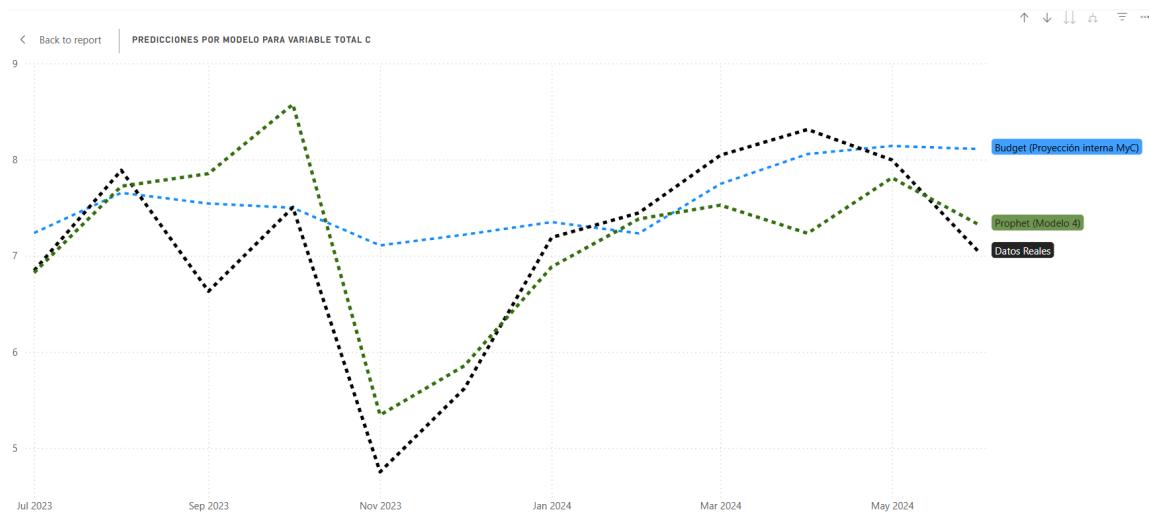
*Nota. Elaboración propia utilizando Power BI.*

**Figura 55:**



*Nota. Elaboración propia utilizando Power BI.*

**Figura 56:**



*Nota. Elaboración propia utilizando Power BI.*

**Cuadro 3:**

Modelo	MAE	MSE	RMSE	MAPE
Prophet (Modelo 4)	1.6	7.3	2.7	12.2%
Budget (Proyección interna MyC)	2.0	8.7	2.9	10.8%
Random Forest Regressor (Ventana 48 meses)	2.1	8.6	2.9	10.3%
TPOTRegressor (AdaBoostRegressor)	2.1	7.6	2.8	10.1%
Xgboost (Modelo 2)	2.4	9.7	3.1	11.6%
LSTM (Modelo 5)	2.8	11.4	3.4	13.7%
ARIMA	4.1	20.9	4.6	18.2%
AR (Autorregresivos)	27.5	4.7	5.3	20.4%

*Nota. Elaboración propia utilizando Power BI.*

## Análisis Comparativo

El modelo Prophet (Modelo 4) ha demostrado ser el modelo univariado con el rendimiento predictivo más sobresaliente para la serie temporal "Total C". Este modelo ha alcanzado métricas de rendimiento excepcionales, con un Error Absoluto Medio (MAE) de 1.56, un Error Cuadrático Medio (MSE) de 7.34, una Raíz del Error Cuadrático Medio (RMSE) de 2.71 y un Error Porcentual Absoluto Medio (MAPE) de 12.19%.

Estos resultados consolidan al Modelo 4 de Prophet como el modelo más preciso y confiable para la predicción de la serie temporal "Total C". El éxito se atribuye a su capacidad inherente para modelar series temporales con patrones complejos, específicamente a través de:

- Flexibilidad en la Modelización de Tendencias: Prophet permite la modelización de tendencias no lineales y la identificación de puntos de cambio en la tendencia, lo que resulta crucial para series temporales que experimentan variaciones significativas en su dirección y magnitud.
- Manejo Eficaz de la Estacionalidad: Prophet está diseñado para modelar múltiples patrones estacionales, incluyendo estacionalidad anual, semanal y diaria, así como estacionalidades personalizadas. Esta capacidad es fundamental para capturar la dinámica cíclica presente en la serie temporal "Total C".
- Optimización de Hiperparámetros: la implementación de GridSearch en el modelo 4 permitió optimizar los hiperparámetros del modelo, mejorando de esta manera la exactitud de las predicciones.

La capacidad de Prophet para adaptarse a cambios en la tendencia y modelar patrones estacionales complejos ha permitido superar a otros modelos univariados en la predicción de la serie temporal "Total C".

### 6.4. Conclusiones generales

La presente tesis ha cumplido con el objetivo de desarrollar y comparar distintos modelos de predicción de series temporales para pronosticar las ventas de juguetes por segmento de negocio en el mercado argentino, con un horizonte de predicción de 12 meses.

Los hallazgos obtenidos no solo proporcionan una evaluación comparativa del rendimiento de los modelos, sino que también ofrecen información valiosa para la toma de decisiones empresariales estratégicas.

A partir de un enfoque riguroso y sistemático, se obtuvieron resultados que no solo permiten evaluar la eficacia de cada modelo, sino que también proporcionan aprendizajes significativos para la modelización en contextos similares.

Los modelos predictivos desarrollados permiten a las empresas del sector anticipar la demanda futura con un alto grado de precisión, facilitando la optimización de la gestión de inventarios, la planificación de la producción y la asignación de recursos. La capacidad de pronosticar las ventas por segmento de negocio permite una segmentación más efectiva del mercado y la personalización de estrategias de marketing y ventas.

La aplicabilidad de los modelos desarrollados se extiende a otros contextos empresariales con características similares, como la estacionalidad marcada y la presencia de tendencias complejas.

Sin embargo, es importante considerar las siguientes limitaciones:

- Sensibilidad a ventanas temporales: la selección de la ventana temporal óptima es crucial para el rendimiento de los modelos. Los resultados obtenidos sugieren que la ventana temporal óptima varía según las características de cada segmento de negocio, lo que requiere un análisis individualizado.
- Pruebas Retrospectivas: se realizaron pruebas retrospectivas para validar la capacidad de los modelos para predecir ventas pasadas con precisión. Estas pruebas, que simulan escenarios de predicción, respaldan la robustez y la confiabilidad de los modelos desarrollados.

Uno de los hallazgos más destacados fue la identificación y selección de variables explicativas, donde técnicas como el RFE (Recursive Feature Elimination) demostraron ser cruciales para mejorar el rendimiento predictivo. La focalización en variables clave permitió simplificar los modelos y, simultáneamente, mejorar su precisión. Esto subraya la importancia de una adecuada selección de variables como un paso fundamental en la modelización de series temporales.

Además, el enfoque de modelos híbridos aprovechó las fortalezas del modelo ARIMA para captar tendencias lineales y estacionales y, a su vez, integró la capacidad de LSTM para identificar relaciones no lineales complejas, resultando en un modelo robusto y preciso.

Asimismo, el uso de ventanas temporales amplias fue clave para capturar la estructura de la serie “Total M”, destacando la necesidad de ajustar la longitud de estas ventanas según las características de cada variable.

En el caso de la serie “Total C,” el modelo Prophet sobresalió por su flexibilidad para manejar cambios en tendencias y estacionalidades complejas. La capacidad adaptable de Prophet permitió ajustarse de manera efectiva a los patrones observados, mostrando su potencial aplicabilidad en series con características similares.

Los resultados de este trabajo subrayan la importancia de adoptar un enfoque multidimensional y flexible en la predicción de series temporales. Los modelos híbridos que combinan técnicas estadísticas tradicionales con machine learning y deep learning demostraron ser especialmente efectivos, permitiendo no solo una mejor captura de la complejidad inherente de las series temporales, sino también proporcionando un marco adaptable para posibles futuras variaciones en los datos.

## 7. Trabajos Futuros

Con miras a futuras investigaciones, se recomienda continuar explorando la integración de nuevas técnicas de selección de variables y el desarrollo de enfoques híbridos que se adapten a contextos y variables de distinta naturaleza. El creciente volumen de datos históricos disponible ofrece la posibilidad de construir modelos cada vez más precisos y robustos, especialmente mediante la incorporación de nuevas fuentes de información y técnicas avanzadas de optimización.

Asimismo, los resultados obtenidos sugieren la importancia de seguir investigando la combinación de modelos tradicionales y modernos en la predicción de series temporales. Enfoques como ARIMA y Prophet, que proporcionan una base sólida y comprensible, pueden verse fortalecidos al integrarse con técnicas de machine learning y deep learning, generando

mejoras significativas, especialmente para series con estructuras complejas o no lineales. Este camino hacia una mayor precisión está, sin duda, impulsado por la innovación constante y la experimentación.

El campo de la predicción de series temporales se perfila como un ámbito vasto y en constante evolución, donde la convergencia entre métodos estadísticos tradicionales y herramientas de machine learning y deep learning abre nuevos horizontes. Cada modelo construido y cada variable seleccionada representa un avance en la construcción de un conocimiento integral sobre la dinámica de las series temporales, en el que la búsqueda de precisión se entrelaza con una reflexión sobre la esencia y el comportamiento de los datos.

En esta línea, el trabajo futuro no se limitará únicamente a optimizar los modelos existentes, sino también a cuestionar los supuestos analíticos que guían nuestras decisiones. ¿Hasta qué punto un modelo puede capturar la complejidad inherente a la realidad que representa? ¿De qué forma se puede equilibrar la búsqueda de precisión con la presencia inevitable de incertidumbre? Estas cuestiones, de índole filosófica además de técnica, nos recuerdan que la predicción es, en última instancia, una interpretación en la que convergen el rigor matemático y la experiencia humana, y donde el pasado y el futuro se entrelazan en un proceso de descubrimiento continuo y abierto a nuevas posibilidades.

**Hyndman, R. J., & Athanasopoulos, G. (2018). *Forecasting: Principles and Practice* (2nd ed.). OTexts.**

- **Cita:** "The importance of understanding the underlying patterns in time series data cannot be overstated. Forecasting is as much an art as it is a science, requiring careful consideration of model selection and the context in which it is applied."

## Bibliografía

1. Bartolomé, K., & Zambrano, A. (2021, April 25). Workflowsets in time series. Recuperado de: <https://karbartolome-blog.netlify.app/posts/workflowsets-timeseries/>
2. Box, G. E., Jenkins, G. M., & Reinsel, G. C. (2015). *Time series analysis: Forecasting and control* (Vol. 534). John Wiley & Sons.
3. Brownlee, J. (2017). *Long Short-Term Memory Networks with Python. Develop Sequence Prediction Models with Deep Learning*.
4. Chatfield, C. (2016). *The analysis of time series: An introduction* (Vol. 493). CRC Press.
5. Chen, T., & Guestrin, C. (2016, August). XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 785-794).
6. Fan, J., & Yao, Q. (2003). *Nonlinear time series: Nonparametric and parametric methods*. Springer Science & Business Media.
7. Génieys, S. (2015). Time series forecasting using artificial neural networks: A systematic review. *Expert Systems with Applications*, 42(22), 9143–9162.
8. Granger, C. W. J., & Hyndman, R. J. (2008). *Forecasting financial time series*. John Wiley & Sons.
9. Hamilton, J. D. (1994). *Time series analysis*. Princeton University Press.
10. Hyndman, R. J., & Athanasopoulos, G. (2021). *Forecasting: Principles and practice* (3rd ed.). OTexts. Recuperado de: <https://otexts.com/fpp3>
11. Hyndman, R. J., & Khandakar, Y. (2008). Automatic time series forecasting: The forecast package for R. *Journal of Statistical Software*, 26(3), 1–22.
12. Lütkepohl, H. (2005). *New introduction to multiple time series analysis*. Springer Science & Business Media.
13. Randal S. Olson and Jason H. Moore (2016). *TPOT: A Tree-based Pipeline Optimization Tool for Automating Machine Learning*.
14. Sean J. Taylor and Benjamin Letham (2017). *Forecasting at Scale*.
15. Shumway, R. H., & Stoffer, D. S. (2017). *Time series analysis and its applications: With R examples* (Vol. 4). Springer.
16. Wei, W. W. S. (2006). *Time series analysis: Univariate and multivariate methods* (Vol. 7). Pearson Education India.

## Anexo I

Diccionario de datos principales utilizados para los modelos predictivos (se destacan con un color y se ordenan inicialmente las variables que tuvieron mayor importancia en los modelos multivariados).

Variable	Agrupación	País	Medición	Fuente	Concepto amplio
<b>Crude Oil</b>	Índice de precio	Internacional	Mensual	<a href="#">Index Mundi</a>	Crude oil, average spot price of Brent, Dubai and West Texas Intermediate, equally weighed (World Bank)
<b>Residential</b>	Indicador macroeconómico	USA	Trimestral	<a href="#">Bureau of Economic Analysis</a>	Inversión en construcción y mejoras de viviendas residenciales. Incluye gastos en nuevas viviendas unifamiliares, edificios multifamiliares, y renovaciones o remodelaciones de viviendas existentes. Este indicador es una parte clave del gasto en inversión privada y puede influir en el crecimiento económico, ya que refleja el estado del mercado inmobiliario y la demanda de vivienda.
<b>Goods Imports</b>	Indicador macroeconómico	USA	Trimestral	<a href="#">Bureau of Economic Analysis</a>	Compra de bienes físicos desde el extranjero. Esto incluye una amplia gama de productos, como maquinaria, vehículos, alimentos, combustibles, y otros artículos manufacturados. Las importaciones de bienes son una parte esencial del comercio internacional y pueden afectar la balanza comercial de un país, ya que representan un gasto hacia el exterior que puede influir en el producto interno bruto (PIB) y en las relaciones económicas internacionales.
<b>Primary metals</b>	Índice por industria	USA	Trimestral	<a href="#">Bureau of Economic Analysis</a>	Metales básicos producidos a partir de minerales o materiales reciclados que han pasado por procesos

Variable	Agrupación	País	Medición	Fuente	Concepto amplio
				<a href="#">Analysis</a>	de refinación o fundición para ser convertidos en formas utilizables, como lingotes, placas, barras o láminas. Estos metales incluyen, entre otros, acero, aluminio, cobre, y zinc, que son fundamentales en la fabricación de productos finales en diversas industrias, como la construcción, la automotriz, y la manufactura de maquinaria.
<b>Furniture and related products</b>	Índice por industria	USA	Trimestral	<a href="#">Bureau of Economic Analysis</a>	Categoría de bienes que incluye la fabricación y producción de muebles y otros productos relacionados para el hogar, la oficina y otros espacios. Esto abarca una amplia variedad de artículos como sofás, mesas, sillas, camas, muebles de almacenamiento, colchones, y decoraciones. Esta categoría también puede incluir productos como persianas, alfombras y otros elementos que complementan la funcionalidad y el diseño de los espacios interiores. La producción y venta de estos productos es un indicador del consumo en el sector del mobiliario y la decoración.
<b>Funds, trusts, and other financial vehicles</b>	Índice por industria	USA	Trimestral	<a href="#">Bureau of Economic Analysis</a>	Instrumentos financieros y entidades que gestionan y administran activos en nombre de inversores. Esto incluye fondos de inversión, fideicomisos (trusts), fondos de pensiones, fondos mutuos, fondos de cobertura (hedge funds), y otras estructuras similares que agrupan capital de múltiples inversores para invertir en una variedad de activos como acciones, bonos, bienes raíces, y derivados. Estos vehículos financieros son esenciales para diversificar riesgos,

Variable	Agrupación	País	Medición	Fuente	Concepto amplio
					facilitar el acceso a mercados financieros, y ofrecer soluciones de inversión a individuos e instituciones.
<b>3359 - Fabricación de otros equipos y accesorios eléctricos</b>	Índice por industria	México	Trimestral	<a href="#">INEGI (Instituto Nacional de Estadística y Geografía)</a>	Fabricación de productos como cables eléctricos, conectores, interruptores, paneles de control, baterías, equipos de iluminación, transformadores, y otros accesorios eléctricos utilizados en aplicaciones residenciales, comerciales, industriales, y de infraestructura. Este sector es crucial para el suministro de componentes esenciales en la construcción, la energía, y diversas industrias tecnológicas.
<b>Índice de volumen físico: Extracción de petróleo y gas. 213 - Servicios relacionados con la minería</b>	Índice por industria	México	Trimestral	<a href="#">INEGI (Instituto Nacional de Estadística y Geografía)</a>	Actividad en la extracción de petróleo y gas natural, así como en los servicios auxiliares relacionados con la minería. Esto incluye la perforación, la exploración, y otros servicios de apoyo a las operaciones de extracción. El índice refleja cambios en la producción física en este sector, ajustando por inflación y variaciones de precios, lo que proporciona una medida del crecimiento o disminución en la actividad de extracción y los servicios asociados a la minería, esenciales para la industria energética y el suministro de combustibles fósiles.
<b>Índice de volumen físico Total . Construcción. Edificación. 237 - Construcción de obras de ingeniería civil</b>	Índice por industria	México	Trimestral	<a href="#">INEGI (Instituto Nacional de Estadística y Geografía)</a>	Actividad en la construcción de grandes infraestructuras y proyectos de ingeniería civil. Esto incluye obras como carreteras, puentes, túneles, presas, sistemas de alcantarillado, y otras infraestructuras públicas y privadas. El índice ajusta por inflación y cambios en los precios, proporcionando una medida del crecimiento o contracción en la

Variable	Agrupación	País	Medición	Fuente	Concepto amplio
					actividad física de construcción en este sector. Es un indicador clave del desarrollo económico y de inversión en infraestructura de un país o región.
aB.1bP - Producto interno bruto	Indicador macroeconómico	México	Trimestral	<a href="#">INEGI (Instituto Nacional de Estadística y Geografía)</a>	El Producto Interno Bruto trimestral ofrece, en el corto plazo, una visión oportuna, completa y coherente de la evolución de las actividades económicas del país, proporcionando información oportuna y actualizada, para apoyar la toma de decisiones.
CPI (Commodity Price Index)	Índice de precio	Internacional	Mensual	<a href="#">Index Mundi</a>	All Commodity Price Index, 2016 = 100, includes both Fuel and Non-Fuel Price Indices (International Monetary Fund)
Iron Ore	Índice de precio	Internacional	Mensual	<a href="#">Index Mundi</a>	Iron ore (any origin) fines, spot price, c.f.r. China, 62% Fe beginning December 2008; previously 63.5% (Thomson Reuters Datastream, World Bank.)
Metals Price Index	Índice de precio	Internacional	Mensual	<a href="#">Index Mundi</a>	Commodity Metals Price Index, 2005 = 100, includes Copper, Aluminum, Iron Ore, Tin, Nickel, Zinc, Lead, and Uranium Price Indices (International Monetary Fund)
Aluminum	Índice de precio	Internacional	Mensual	<a href="#">Index Mundi</a>	Aluminum (LME) London Metal Exchange, unalloyed primary ingots, high grade, minimum 99.7% purity, settlement price beginning 2005; previously cash price (World Bank)
Natural Gas	Índice de precio	Internacional	Mensual	<a href="#">Index Mundi</a>	Natural Gas (U.S.), spot price at Henry Hub, Louisiana (Thomson Reuters Datastream; The Wall Street Journal; World Bank.)
Zinc	Índice de precio	Internacional	Mensual	<a href="#">Index Mundi</a>	Zinc (LME), high grade, minimum 99.95% purity, settlement price beginning April 1990; previously special high grade, minimum 99.995%, cash prices (Platts Metals Week, Engineering and Mining

Variable	Agrupación	País	Medición	Fuente	Concepto amplio
					Journal; Thomson Reuters Datastream; World Bank.)
Coal South African	Índice de precio	Internacional	Mensual	<a href="#">Index Mundi</a>	Coal (South Africa), thermal NAR netback assessment f.o.b. Richards Bay 6,000 kcal/kg from February 13, 2017; during 2006-February 10, 2017 thermal NAR; during 2002-2005 6,200 kcal/kg (11,200 btu/lb), less than 1.0%, sulfur 16% ash; years 1990-2001 6390 kcal/kg (11,500 btu/lb) (Bloomberg; International Coal Report; Coal Week International; Coal Week; World Bank.)
Coal Australia	Índice de precio	Internacional	Mensual	<a href="#">Index Mundi</a>	Coal (Australia), thermal GAR, f.o.b. piers, Newcastle/Port Kembla from 2002 onwards , 6,300 kcal/kg (11,340 btu/lb), less than 0.8%, sulfur 13% ash; previously 6,667 kcal/kg (12,000 btu/lb), less than 1.0% sulfur, 14% ash (International Coal Report; Coal Week International; Coal Week; Bloomberg; IHS McCloskey Coal Report; World Bank.)
FEDFUND\$	Tasa % de interés	Internacional	Mensual	<a href="#">FEDFUNDS</a>	The federal funds rate is the interest rate at which depository institutions trade federal funds (balances held at Federal Reserve Banks) with each other overnight
Moody's Seasoned AAA Corporate Bond Yield	Tasa % de interés	Internacional	Mensual	<a href="#">FEDFUNDS</a>	These instruments are based on bonds with maturities 20 years and above.
NASDAQ 100	Índice de precio	Internacional	Mensual	<a href="#">Finance Yahoo</a>	The observations for the NASDAQ 100 Index represent the daily index value at market close. The market typically closes at 4 PM ET, except for holidays when it sometimes closes early.
S&P 500	Índice de precio	Internacional	Mensual	<a href="#">Finance Yahoo</a>	The observations for the S&P 500 represent the daily index value at market close. The market typically closes at 4 PM ET, except for

Variable	Agrupación	País	Medición	Fuente	Concepto amplio
					holidays when it sometimes closes early.
EUR/USD	Índice de monedas	Internacional	Mensual	<a href="#">Finance Yahoo</a>	Exchange rate
Gross domestic product	Indicador macroeconómico	EEUU	Trimestral	<a href="#">Bureau of Economic Analysis</a>	Gross domestic product (GDP), the featured measure of U.S. output, is the market value of the goods and services produced by labor and property located in the United States. For more information, see the Guide to the National Income and Product Accounts of the United States (NIPA) and the Bureau of Economic Analysis.