ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL

Facultad de Ingeniería en Electricidad y Computación

Estimación de la velocidad de producción de una máquina mezcladora mediante la implementación de modelos de Machine Learning

PROYECTO DE TITULACIÓN

Previo la obtención del Título de:

Magister en Ciencias de Datos

Presentado por:

Robinson Diego Macias Sandoval

GUAYAQUIL - ECUADOR

Año: 2024

DEDICATORIA

A mis seres queridos.

En mi camino profesional, su amor y apoyo han sido mi faro y mi motivación. Cada página escrita, cada problema resuelto, cada logro alcanzado, ha sido posible gracias a la fortaleza y el apoyo que me han brindado.

AGRADECIMIENTOS

Quiero expresar mi sincero agradecimiento a todas las personas e instituciones que hicieron posible este trabajo. Su apoyo y contribuciones han sido esenciales en cada paso de este arduo proceso.

Declaración Expresa

Yo Robinson Diego Macias Sandoval acuerdo y reconozco que: La titularidad de los derechos patrimoniales de autor (derechos de autor) del proyecto de graduación corresponderá al autor o autores, sin perjuicio de lo cual la ESPOL recibe en este acto una licencia gratuita de plazo indefinido para el uso no comercial y comercial de la obra con facultad de sublicenciar, incluyendo la autorización para su divulgación, así como para la creación y uso de obras derivadas. En el caso de usos comerciales se respetará el porcentaje de participación en beneficios que corresponda a favor del autor o autores. El o los estudiantes deberán procurar en cualquier caso de cesión de sus derechos patrimoniales incluir una cláusula en la cesión que proteja la vigencia de la licencia aquí concedida a la ESPOL.

La titularidad total y exclusiva sobre los derechos patrimoniales de patente de invención, modelo de utilidad, diseño industrial, secreto industrial, secreto empresarial, derechos patrimoniales de autor sobre software o información no divulgada que corresponda o pueda corresponder respecto de cualquier investigación, desarrollo tecnológico o invención realizada por mí/nosotros durante el desarrollo del proyecto de graduación, pertenecerán de forma total, exclusiva e indivisible a la ESPOL, sin perjuicio del porcentaje que me corresponda de los beneficios económicos que la ESPOL reciba por la explotación de mi innovación, de ser el caso.

En los casos donde la Oficina de Transferencia de Resultados de Investigación (OTRI) de la ESPOL comunique al autor que existe una innovación potencialmente patentable sobre los resultados del proyecto de graduación, no se realizará publicación o divulgación alguna, sin la autorización expresa y previa de la ESPOL.

Guayaquil, lunes 09 de diciembre del 2024.

Firmado electrónicamente por:
ROBINSON DIEGO
MACIAS SANDOVAL

Robinson Diego Macias Sandoval

Evaluadores

Firmado electrónicamente por:
CHRISTIAN JAVIER
TUTIVEN GALVEZ

TUTIVEN GALVEZ CHRISTIAN JAVIER

PROFESOR TUTOR

Firmado electrónicamente por:
EDUARDO SEGUNDO
CRUZ RAMIREZ

EDUARDO SEGUNDO CRUZ RAMIREZ

PROFESOR REVISOR

RESUMEN

En este proyecto, se aborda el desafío de predecir la velocidad de una máquina mezcladora en una planta procesadora de alimentos balanceados, el propósito central del presente trabajo es implementar modelos de machine learning para estimar la velocidad de la máquina mezcladora. En primer lugar, se realizó el tratamiento del conjunto de datos, ejecutando tareas de limpieza y filtrado de valores atípicos para mejorar su calidad. Posteriormente, se implementó el modelado SARIMA (Seasonal Autoregressive Integrated Moving Average) para los pronósticos de series temporales. También, se ajustaron modelos LSTM (Long Short-Term Memory) de redes neuronales para predecir las velocidades de cada producto. Los resultados del proyecto demostraron que, aunque el modelo SARIMA fue adecuado debido a la estacionalidad en los datos, las redes LSTM demostraron un rendimiento superior en la predicción de la velocidad de la mezcladora. Esto se refleja en que los valores de MAPE fueron más bajos en aproximadamente el 84% de los productos. En consecuencia, el modelado LSTM fue escogido y se usó para la implementación del pipeline de MLOps. Además, se sugirió la exploración de una estrategia de modelado mixto que combine el enfoque SARIMA y las redes LSTM para maximizar la precisión. Este enfoque puede optimizar la eficiencia y elevar la calidad en las actividades de producción de la planta.

Palabras Clave: Modelos ARIMA, LSTM, Gobierno de Datos, Series Temporales

ABSTRACT

This project addresses the challenge of predicting the speed of a mixing machine in an animal feed processing plant. The primary objective of this work is to implement machine learning models to estimate the mixing machine's speed. Initially, data preprocessing was conducted, including cleaning and filtering outliers to enhance data quality. Subsequently, SARIMA (Seasonal Autoregressive Integrated Moving Average) modeling was applied for time series forecasting. Additionally, Long Short-Term Memory (LSTM) neural network models were adjusted to predict the speed for each product. The project results showed that, while the SARIMA model was suitable due to seasonality in the data, the LSTM networks demonstrated superior performance in predicting mixer speed, reflected by approximately 84% of products showing lower MAPE values. Consequently, the LSTM model was selected and used for implementing the MLOps pipeline. Additionally, exploring a mixed modeling strategy that combines SARIMA and LSTM approaches was recommended to maximize accuracy. This approach can optimize efficiency and enhance the quality of production activities at the plant.

Keywords: ARIMA Models, LSTM, Data Management, Time Series

ÍNDICE GENERAL

ABREVIATURAS

ESPOL	Escuela Superior Politécnica del Litoral
ARIMA	Auto Regressive Integrated Moving Average
SARIMA	Seasonal AutoRegressive Integrated Moving Average
MAPE	Mean Absolute Percentage Error
LSTM	Long Short-Term Memory
AWS	Amazon Web Services
IQR	Rango Intercuartil
ECR	Elastic Container Registry

SIMBOLOGÍA

t	Tonelada métrica
h	hora

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

ÍNDICE DE APÉNDICES

CAPÍTULO 1

1. INTRODUCCIÓN

1.1 Descripción del problema

La competitividad empresarial requiere que los procesos repetitivos sean analizados mediante el registro de información sobre su comportamiento, de este modo, es posible identificar oportunidades de mejora que optimicen el uso de recursos y promuevan una mejora continua, de esta forma el uso de herramientas prácticas en su proceso de producción se torna como una oportunidad para incrementar su competitividad en el mercado.

Para las empresas cuya actividad principal es la producción de balanceados para las industrias ganadera, avícola, acuícola y afines, deben cumplir ciertos estándares, como ISO 22000, Kosher, Buenas Prácticas de Manufactura, Análisis de Peligros y Puntos Críticos de Control (HACCP) para garantizar la inocuidad alimentaria, la norma International Food Standard (IFS), así como certificaciones entregadas por la Agencia Nacional de Regulación, Control y Vigilancia Sanitaria (ARCSA). Estos estándares el crecimiento y engorde de las diferentes especies, de esta forma requieren que exista disponibilidad de plantas, insumos y demás materia prima para su productividad, logrando cubrir la demanda existente en el mercado y la estacionalidad de la demanda de productos.

En este aspecto, las empresas ecuatorianas ejecutan los procesos orden de pedido, recepción, mezcla y demás de manera empírica, mediante hojas de cálculos las cuales son intercambiadas entre los diferentes departamentos, y van mejorando la calidad de su producto en base a su experiencia sin la integración de conocimientos teóricos-prácticos que promuevan su competitividad.

En torno a lo mencionado, estas máquinas permiten que los diferentes componentes de los balanceados puedan unificarse para lograr el producto a comercializar, sin embargo, la falta de un adecuado manejo y configuración de estas acarrea que su uso se torne ineficiente e incluso que se cometan falencias como su sobrecarga o déficit en su uso.

El registro de datos sobre la producción de balanceado se realiza de forma empírica, por ejemplo, en hojas de cálculos o en sistemas informáticos que no están interconectados y por consiguiente no tienen datos consolidados. Los tiempos, cantidades e insumos específicos no han sido definidos de forma predeterminada. Esto dificulta mejorar indicadores como calidad, eficiencia, eficacia y rendimiento. Además, impide la estandarización de procesos que promueve la mejora continua del producto.

Como consecuencia, no pueden delegar funciones y responsabilidades para garantizar que el proceso de producción continúe ante posibles cambios de personal. Esto crea un déficit en la operación. Además, se exponen falencias en la disponibilidad de materia prima, insumos y recursos según las necesidades de cada fase.

El balanceado comercializado debe cumplir con los requerimientos legales del mercado por ende son aptos para el consumo de animales de crianza, así que, el uso de las materias primas como plantas, vitaminas, y demás deben ser integradas de manera oportuna en el tiempo adecuado para que sean aptas para el consumo de los animales. En ello, al no contar con una determinación específica basado en sus antecedentes de operatividad acarrea vacíos que ocasionan un incremento en los costos y gastos.

Del mismo modo, al desconocer la capacidad de la mezcladora, los tiempos de mezcla y la forma de integrarlos, el producto puede no cumplir con los requerimientos del cliente. Esto acarrea una optimización deficiente de los recursos disponibles, lo que incrementa las problemáticas de la organización. Además, la productividad de las máquinas mezcladoras es fundamental, ya que son herramientas necesarias en la operatividad de la producción de empresas que elaboran balanceado.

En este punto, las máquinas mezcladoras cuentan con especificaciones técnicas que deben ser analizadas por los propietarios antes de ponerlas en funcionamiento. Esto evita que posibles daños generen retrasos o pérdidas en la producción. De esta forma, las organizaciones deben generar un registro histórico del proceso. Esto les permitirá determinar tiempos de operatividad y la necesidad de mantenimientos preventivos. Además, facilita el soporte mediante la adquisición de máquinas acorde al nivel de producción de las empresas.

En torno a lo mencionado se evidencia la necesidad de que las organizaciones integren sistemas de control que generen datos sobre la producción de balanceado con el propósito de lograr indagar los lapsos de tiempo de uso, mantenimiento, capacidad de

producción, entre otros elementos que necesitan ser evaluados para satisfacer la demanda actual del producto en el mercado.

Si bien, la empresa considerada en el estudio hace registro de su producción mediante un algoritmo el cual es utilizado en su planificación con la intención de conocer la disponibilidad del producto a comercializar, este no cuenta con un detalle de la velocidad de operatividad de la máquina mezcladora, en donde se determine el tiempo de uso, la velocidad baja, media o alta en la que se utilizó y la capacidad de la misma para cada una, así como, el requerimiento del tipo de mezclado según las características del producto.

Por ende, se evidencia la necesidad de analizar los datos con los que cuentan, e integrar herramientas de gestión de datos que favorezcan la mejora continua de la organización, en donde al integrar algoritmos haciendo uso de la metodología de machine learning la empresa cuente con el respaldo adecuado para desarrollar procesos de innovación.

1.2 Justificación

La planificación de una empresa debe incluir el análisis adecuado de sus condiciones actuales. A partir de este análisis, se pueden establecer metas realistas sobre sus alcances y oportunidades de mejora. Esto facilitará atender las necesidades de los clientes y aprovechar al máximo los recursos disponibles. Así, la producción será gestionada adecuadamente para satisfacer las necesidades del mercado. Además, influirá en la adquisición de materia prima, insumos, suministros, equipos y maquinaria.

La innovación desempeña un rol esencial en el ámbito empresarial, derivado de la necesidad de aprovechar las oportunidades existentes para favorecer la posición de una organización en el mercado, siendo que, sus productos cuenten con el respaldo de credibilidad respecto a la adherencia a los estándares de calidad, sumado al rendimiento de su personal, proveedores y demás interesados derivado de la capacidad de la empresa en cumplir y brindar oportunidades de mejora.

Sumado a ello, se tiene que la innovación tecnológica sigue factores de sostenibilidad y sustentabilidad en donde se tiene el cumplimiento de ciertos requerimientos para generar el menor impacto negativo posible derivado de actividades productivas, siendo que, de este modo se logre una mejor utilización de los recursos y

una disminución de los residuos generados en los distintos procesos, tomando en cuenta los Objetivos de Desarrollo Sostenible (ODS) en su objetivo número 11 que se centra en la promoción de acciones para favorecer el desarrollo de "*Ciudades y Comunidades sostenibles*" , en donde, invitan a las organizaciones sociales, empresas, y comunidad en general a tomar acción en favor del bienestar en común.

Con referencia en lo anteriormente descrito, el desarrollo del trabajo de investigación responde a la necesidad de que las empresas de este sector dispongan de una orientación de innovación, siendo que, la metodología de *machine learning* favorece a la estandarización de sus procesos y mejor aprovechamiento de los recursos disponibles mediante el uso de algoritmos que analicen su ejecución para irlo mejorando periódicamente, de tal manera que, las posibles problemáticas no afecten de forma significativa a la continuidad de la organización.

Al establecer algoritmos que logren integrar la metodología machine learning en la operatividad de la producción de las máquinas mezcladoras este puede ser replicado y adaptado en empresas de similares características, siendo el punto de partida para mejorar el mismo y promover un mercado más competitivo cuyos costos de producción sean menores, y reduzcan los residuos generados en esta fase, consolidándose como un referente de innovación tecnológica.

1.3 Objetivos

1.3.1 Objetivo General

Desarrollar modelos de machine learning para predecir la velocidad de procesamiento de una máquina mezcladora de una planta procesadora de alimentos balanceados.

1.3.2 Objetivos Específicos

1. Implementar modelos de pronóstico de series temporales, como ARIMA, SARIMA y redes neuronales LSTM, para la estimación de la velocidad de la mezcladora en una planta procesadora de alimentos balanceados.
2. Comparar los modelos implementados de series de tiempo y red LSTM para el cálculo de la velocidad estimada basados en su precisión.

3. Implementación de un pipeline de MLOps (machine learning operations) para la estimación de la velocidad de producción.

1.4 Marco teórico

Los fundamentos teóricos que se relacionan con la temática de investigación permiten determinar de forma previa un contexto sobre el objeto de estudio disminuyendo de esta forma improvisación y generando mayor asertividad en las actividades desarrolladas, de tal manera que los resultados analizados de investigaciones previas permitan identificar semejanzas y diferencias que puedan tomarse como referencia para el presente estudio.

En el trabajo desarrollado por Pérez-Zaldívar (2022), se aborda la importancia de estimar de manera precisa la eficiencia en el sector de la confección. Propone un procedimiento que utiliza técnicas de aprendizaje supervisado para predecir si se cumplirá la productividad en las líneas de producción, basándose en datos previos. Estas técnicas se aplicaron a registros de una empresa de confección textil en Bangladesh.

El estudio indica que los modelos de clasificación mejoran al equilibrar los datos de entrenamiento. En particular, el modelo Random Forest resulta ser el más eficiente según criterios como la sensibilidad , precisión, y el F1-Score. Se observó que el modelo Random Forest, entrenado con datos balanceados, obtuvo un alto AUC (75%) y fue útil para identificar casos en los que la productividad no se cumplía.

En concordancia, Abella-Miravet (2021) En este trabajo se aborda el tema de la clasificación en el aprendizaje supervisado, donde se predicen resultados categóricos, ya sean binarios o multiclase, dado que, aunque existen varios modelos de clasificación, como la regresión logística, los árboles de decisión y los bosques aleatorios, se ha observado que la presencia de clases desequilibradas en los conjuntos de datos representa un desafío significativo para la mayoría de estos algoritmos, que asumen una distribución más equilibrada.

Maisueche-Cuadrado (2019), analiza que los mayores avances económicos han sido impulsados a lo largo de la historia por importantes avances tecnológicos, como la máquina de vapor, la electricidad y el motor de combustión interna. Por ende, se ha buscado sacar provecho de estas tecnologías transformadoras para desarrollar modelos

de negocios innovadores y maximizar sus beneficios a través de la optimización de costos.

En el contexto actual, nos encontramos en lo que se denomina la cuarta revolución industrial. En esta era, la inteligencia artificial, y en particular el machine learning, desempeña un papel destacado. Este ámbito de la inteligencia artificial permite que las computadoras adquieran aprendizaje sin programación directa, lo cual genera nuevas oportunidades en la industria y diversos sectores.

Por su parte, Montes-Gallardo (2020), desarrolla un proyecto con el propósito de examinar los métodos de machine learning más comunes en la actualidad y su aplicación en la industria de control de procesos a través de un enfoque de control predictivo basado en modelos, conocido como MPC (Model Predictive Control). Los métodos de machine learning en cuestión son Kinky Inference (KI) y Gaussian Process (GP), y su implementación se lleva a cabo en el software de cálculo MATLAB 2017ª, por ende, ofrece una breve descripción de los métodos de ML utilizados y se aplican a un sistema estático creado con la función "peaks" de MATLAB, lo cual, permitió apreciar las ventajas y limitaciones de ambos métodos, resaltando la menor precisión de KI en comparación con GP, así como los mayores requisitos computacionales de este último.

1.4.1 Análisis de Series de Tiempo

El análisis de series temporales es un área de la estadística enfocada en el estudio de datos recolectados a intervalos de tiempo regulares, por ende, el objetivo es comprender las tendencias y patrones presentes en los datos, y utilizar esta información para hacer predicciones sobre el futuro (Oliva, 2019).

1.4.1.1 Modelos ARIMA

El modelo ARIMA (AutoRegressive Integrated Moving Average) es capaz de representar un valor como una combinación lineal de datos previos y términos de error aleatorio, y puede incorporar elementos cíclicos o estacionales si es necesario para describir completamente el fenómeno (Chávez-Quisbert, 1997), por ende, se integra este con la finalidad de hacer uso de los datos disponibles para generar análisis basados en

el comportamiento histórico de la máquina mezcladora, tomando en consideración la velocidad y tiempo de mezcla.

El modelo ARIMA es una herramienta estadística empleada en la predicción de series temporales, al ser "autorregresivo" señala que utiliza información previa para estimar el valor presente (González-Casimiro, 2008), de esta forma, puede incorporar un componente de integración, diseñado para transformar la serie en una forma estacionaria, es decir, que la serie temporal debe mantener una media, varianza y correlación constantes a lo largo del tiempo, ya que este es un requisito fundamental para la aplicación exitosa de los elementos de autorregresión y promedio móvil en el análisis de la serie temporal.

1.4.1.2 Modelos SARIMA

Representa el acrónimo de "Seasonal AutoRegressive Integrated Moving Average" en inglés, y se considera una ampliación del modelo ARIMA, en ello, esta metodología fue creada por George Box y Gwilym Jenkins, dos expertos destacados en la modelización de series temporales. El modelo SARIMA se utiliza de manera particular cuando los datos presentan patrones estacionales (Miranda-Chinlli, 2021).

Este modelo incluye parámetros adicionales destinados a capturar patrones estacionales en una serie de tiempo y se formula como SARIMA (p, d, q) x (P, D, Q, S), siendo que, en esta expresión, p, d y q representan los grados de los componentes no estacionales (autorregresión, integración y promedio móvil, respectivamente), mientras que P, D, Q se refieren a los grados de los componentes estacionales, y el símbolo S indica la periodicidad de los patrones estacionales.

1.4.2 Valor atípico (Outlier)

En el campo de la estadística, un valor atípico, comúnmente referido como "outlier," corresponde a una observación que se aleja considerablemente del resto en un conjunto de datos; la relevancia de estos valores radica en su capacidad para distorsionar e influir en los resultados e interpretaciones obtenidas en cualquier análisis estadístico, por ende, los valores atípicos pueden originarse por la variabilidad propia de los datos o por errores en los experimentos (Yepes-Piqueras, 2022).

Se especifica en la investigación que al organizar la información y revisar los datos históricos registrados en el algoritmo utilizado por la empresa, estos fueron evaluados para determinar la fuente y el margen de afectación a la investigación, en donde aquellos que no cumplían con los estándares de credibilidad y fiabilidad en su registro fueron excluidos para evitar alteraciones en la obtención de tendencias.

La revisión de la información y la consolidación de datos se realizaron de manera manual, en donde, el investigador evaluó la procedencia y validez de los mismos, los siguientes métodos permitieron evaluar los datos por conglomerados e intervalos que permitieran identificar los valores atípicos, así como también establecer métricas estadísticas.

1.4.2.1 Z-Score

El "z-score" o puntaje z es una medida estadística que muestra la posición de una observación individual en relación con la media y la desviación estándar de un conjunto de datos. Este concepto se atribuye en gran parte a William S. Gosset, quien publicó su trabajo bajo el seudónimo de "Student". El z-score desempeña un papel es clave en la teoría de la probabilidad y forma una base esencial para la estadística inferencial (García, 2019).

El z-score de una observación *x* se calcula mediante la siguiente fórmula:

$$Z = \frac{x - \mu}{\sigma}$$

El uso del puntaje z es aplicado en diversas disciplinas, incluyendo psicología, medicina, empresas y, por supuesto, en el campo de la ciencia de datos, con el objetivo de normalizar la información y detectar observaciones atípicas, de esta forma la métrica es valiosa para comparar datos en diferentes campos y desempeña un papel crucial en la detección de valores atípicos en colecciones de datos, lo cual aporta de manera notable a la calidad y confiabilidad de los análisis efectuados en diversas disciplinas.

1.4.2.2 Rango Intercuartil

El rango intercuartílico (IQR, por su nombre en inglés Interquartile Range) es una medida estadística centrada en la 'zona central' de un conjunto de datos y, por ello, ha sido ampliamente discutido en la literatura estadística, incluso en los estudios de John

W. Tukey, un estadístico clave en el avance del análisis exploratorio de datos, estos son de particular utilidad para caracterizar la variabilidad en colecciones de datos que pueden incluir valores inusuales. Se aplica frecuentemente en relación con diagramas de caja (box plots), una representación gráfica que también fue promovida por Tukey (Rodriguez-Ojeda, 2017).

El rango intercuartílico se calcula restando el primer cuartil (Q1) al tercer cuartil (Q3):

$$\mathrm{IQR} = \mathrm{Q}_3 - \mathrm{Q}_1$$

1.4.3 Autocorrelación de series de tiempo

La noción de autocorrelación en el contexto de series temporales es esencial para comprender las relaciones temporales en un conjunto de datos, que están dispuestos en secuencia cronológica, dado que, este concepto ha sido exhaustivamente explorado por destacados estadísticos, incluyendo a Box y Jenkins, quienes son reconocidos por sus contribuciones al desarrollo del modelo ARIMA, que representa un concepto fundamental en el análisis de series temporales (Hernández, 2015).

La autocorrelación es una métrica esencial para evaluar la interdependencia temporal en una serie de datos a lo largo del tiempo, proporcionando información valiosa sobre la dinámica de la serie.

1.4.4 Red LSTM

Las Redes LSTM constituyen un tipo especializado de Redes Neuronales Recurrentes (RNN) diseñadas para abordar dependencias de largo alcance en secuencias de datos, fue creado por Sepp Hochreiter y Jürgen Schmidhuber en 1997, representando una contribución significativa al aprendizaje profundo y al análisis de series temporales (Frąckiewicz, 2023), consiste en unidades de memoria llamadas células LSTM, que incluyen puertas para regular el flujo de información, determinando qué datos retener o descartar. Estas operaciones en una célula LSTM, se describen mediante ecuaciones que involucran estas puertas y estados de celda, de esta forma, se han consolidado como una de las arquitecturas más populares en la modelación de secuencias en ciencia de datos y aprendizaje automático, por lo tanto, se utilizan

extensamente en entornos de aprendizaje profundo como Keras, TensorFlow y PyTorch, mostrando su efectividad en múltiples aplicaciones prácticas.

1.4.4.1 ETL (Extract, Transform, Load)

El término ETL, acrónimo en inglés de Extract, Transform, Load, describe un proceso fundamental en la gestión de datos que implica tres pasos fundamentales (Barahama & Wardani, 2021).

1.4.5 Almacenamiento y Gestión de Datos

1.4.5.1 AWS (Amazon Web Services)

Amazon Web Services (AWS) es una plataforma de nube que proporciona diversos servicios, entre ellos opciones de capacidad de cómputo, áreas de almacenamiento, bases de datos, capacidades de aprendizaje automático, respaldo para videojuegos y una amplia gama de soluciones adicionales (Gimenes, 2020), con el propósito de simplificar la implementación y expansión de una diversidad de aplicaciones. este es adoptado por organizaciones de diversos tamaños, que abarcan desde incipientes startups hasta grandes corporaciones, consolidándose como la plataforma de computación en la nube preeminente en uso en la actualidad.

1.4.6 Pipeline de MLOps

MLOps es una metodología que combina prácticas de ML y DevOps para mejorar el desarrollo, despliegue y mantenimiento de modelos de aprendizaje automático. MLOps comparte varias características clave con DevOps, que incluyen: CI/CD, automatización, transparencia y colaboración, infraestructura como código, testing, monitoreo, flexibilidad y agilidad (Llosa, 2020).

La Fig. 1.1 ilustra un flujo de trabajo de MLOps implementado mediante diversos servicios ofrecidos por Amazon Web Services (AWS). Inicialmente, el orquestador del sistema MLOps opta por diversas herramientas y servicios según las necesidades del proyecto; estos pueden incluir Amazon SageMaker para el entrenamiento y despliegue del modelo, el almacenamiento de datos en Amazon S3, y la automatización de tareas mediante AWS Lambda. Subsecuentemente, en la segunda etapa del proceso, el orquestador transfiere los recursos esenciales para el 'pipeline', que pueden ser el modelo de aprendizaje automático previamente ajustado o los conjuntos de datos

almacenados en un 'bucket' de Amazon S3, o bien recurriendo al registro de modelos de Amazon SageMaker.

En la tercera fase, se provisiona una instancia de AWS CodePipeline mediante la confirmación de un archivo de configuración en un repositorio de Git. La especificidad de esta etapa varía en función del tipo de 'pipeline' y del método de invocación, ya sea mediante la API o a través del archivo de configuración. En este punto, el orquestador encapsula el modelo objetivo y sus respectivas configuraciones en un paquete de AWS CloudFormation, el cual actuará como base para la instancia de AWS CodePipeline.

La fase subsecuente, denominada 'DeployPipeline', se encarga de desplegar el 'pipeline' tomando como base el modelo y sus parámetros o configuraciones encapsuladas en AWS CloudFormation. Finalmente, una vez que se ha completado el aprovisionamiento del 'pipeline' de destino, los usuarios obtienen acceso a sus funcionalidades.

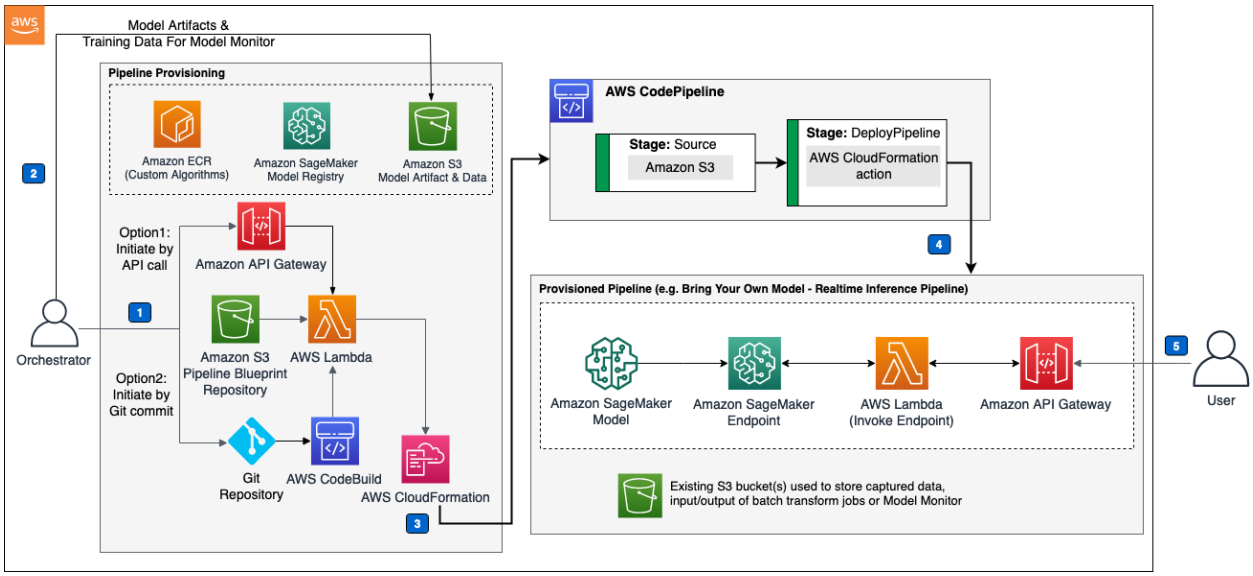


Figura 1.1. Arquitectura de un pipeline de MLOps de AWS

1.4.7 Marco de trabajo CYNEFIN

El modelo CYNEFIN divide los contextos de toma de decisiones en cinco dominios: claro, complicado, complejo, caótico y desordenado. Cada dominio representa una categoría diferente de situaciones de decisión que requieren estrategias específicas:

- **Claro (Obvio)**: En este dominio, la relación entre causa y efecto es evidente para todos. Las mejores prácticas son la guía principal, y las decisiones se basan en la categorización y la aplicación de soluciones probadas y establecidas. Un ejemplo podría ser la manufactura de productos en una línea de producción estandarizada.
- **Complicado**: Este dominio es más difícil que el claro porque puede haber múltiples respuestas correctas, aunque la relación causa-efecto aún es clara, pero requiere de análisis experto. Las decisiones se basan en el análisis y el uso de buenas prácticas. Un ejemplo sería el diagnóstico médico donde se necesitan especialistas para interpretar los síntomas y pruebas.
- **Complejo**: En este dominio, la relación entre causa y efecto solo puede entenderse en retrospectiva y no es predecible. Es necesario adoptar un enfoque innovador en el cual las decisiones se fundamenten en la experimentación continua y en un proceso de aprendizaje que se ajusta y mejora de forma iterativa. Un ejemplo sería la gestión de proyectos innovadores donde los resultados no pueden predecirse completamente.
- **Caótico**: En este dominio, no hay una relación discernible entre causa y efecto en ningún nivel, por lo que la acción debe ser inmediata para estabilizar la situación. Las decisiones se centran en actuar rápidamente para imponer orden, luego detectar las áreas de estabilidad y de desorden. Un ejemplo es la gestión de crisis o desastres.
- **Desordenado**: Este dominio representa una situación en la que no está claro en qué categoría cae el problema. El primer paso es descomponer el problema en partes y determinar el dominio apropiado para cada parte.

1.4.8 Marco de Trabajo Ágil SCRUM

SCRUM es un marco ágil diseñado principalmente para la gestión y desarrollo de software, aunque su uso se extiende a otros proyectos y procesos complejos. Concebido inicialmente por Ken Schwaber y Jeff Sutherland en los años 90, SCRUM fomenta un método iterativo e incremental que busca mejorar la previsión y mitigar riesgos. Mediante un sistema de roles, eventos y artefactos específicos, SCRUM facilita la colaboración y el perfeccionamiento constante dentro de equipos multifuncionales y autoorganizados.

SCRUM se fundamenta en la teoría empírica del control de procesos, la cual plantea que el conocimiento se adquiere a través de la experiencia, y que las decisiones deben tomarse en función de la información disponible. Los principios esenciales que sostienen SCRUM son la **transparencia**, la **inspección** y la **adaptación**. Estos pilares permiten a los equipos gestionar y organizar el trabajo de forma eficaz y eficiente, especialmente en contextos complejos y en constante cambio.

1.4.9 Prueba de Ljung-Box

La prueba de Ljung-Box es un método estadístico para evaluar si una serie temporal exhibe independencia o dependencia en sus datos sucesivos. Desarrollado en 1978 por Greta M. Ljung y George E. P. Box, este método es esencial en el análisis de series temporales, especialmente para comprobar la existencia de autocorrelaciones significativas en los distintos rezagos de una serie de datos.

La prueba de Ljung-Box evalúa si una serie de datos tiene autocorrelación en los primeros "m" rezagos, lo que podría indicar una dependencia temporal entre los valores de la serie. La hipótesis nula (H_0) de esta prueba plantea que los datos son independientes y no presentan correlación, es decir, que son ruido blanco, mientras que la hipótesis alternativa (H_1) sugiere la presencia de autocorrelación significativa en al menos uno de los rezagos considerados.

El estadístico de prueba de Ljung-Box se define como:

$$Q = n(n+2)\sum_{k=1}^{m}\frac{\hat{\rho}_k^2}{n-k}$$

donde:

- Q es el estadístico de Ljung-Box,
- n representa el número de elementos en la muestra,
- m indica la cantidad de rezagos que se están evaluando,
- $\hat{\rho}_k$ es la autocorrelación muestral en el rezago k

El valor de Q se ajusta aproximadamente a una distribución X^2 (chi-cuadrado) con m grados de libertad, especialmente cuando n es grande. Para un nivel de significancia

α, Si el valor calculado de Q supera el valor crítico de la distribución X^2 , se rechaza la hipótesis nula, lo que sugiere una autocorrelación significativa en la serie temporal.

La prueba de Ljung-Box es frecuentemente empleada para validar modelos de series temporales, como los modelos ARIMA. En este contexto, la validación de un modelo implica verificar que los residuos se comporten como ruido blanco. Si los residuos no muestran autocorrelación significativa, el modelo se considera adecuado para representar los datos. De lo contrario, puede ser necesario ajustar el modelo para capturar mejor las dependencias temporales presentes en los datos.

1.4.10 Criterio de Información de Akaike (AIC)

El Criterio de Información de Akaike (AIC) es una herramienta estadística muy utilizada para elegir modelos en el análisis de datos. Propuesto por el estadístico japonés Hirotugu Akaike en 1974, El AIC facilita la evaluación de la calidad relativa entre varios modelos estadísticos candidatos aplicados al mismo conjunto de datos. Su principal objetivo es encontrar un modelo que equilibre adecuadamente la precisión y la complejidad.

El AIC está basado en la entropía y proporciona una estimación de la pérdida de información cuando un modelo dado es utilizado para describir la realidad. Este criterio evalúa la calidad de un modelo considerando cuánta información se pierde en el proceso de ajuste, considerando tanto el ajuste del modelo a los datos como el número de parámetros utilizados. Matemáticamente, el AIC se define como:

$$AIC = 2k - 2\ln(L)$$

donde:

- k corresponde a la cantidad de parámetros que se estiman dentro del modelo.
- L es la función de verosimilitud máxima del modelo.

1.4.11 Error de Porcentaje Medio Absoluto (MAPE)

El **MAPE** (Mean Absolute Percentage Error, por sus siglas en inglés) es utilizado para evaluar la precisión de un modelo. El MAPE calcula el error promedio porcentual de los valores medidos y los valores predichos por el modelo, lo que permite evaluar qué tan preciso es el modelo en términos relativos.

Su definición matemática es:

$$MAPE = \frac{1}{n}\sum_{t=1}^{n}\left|\frac{\mathrm{yt} - \hat{y}t}{yt}\right| \times 100$$

donde:

- n es el número de observaciones,
- yt son los valores reales,
- $\hat{y}t$ son los valores pronosticados por el modelo.

El MAPE se representa en porcentaje y ofrece una medida del error promedio en comparación con el valor real. Un MAPE menor sugiere un modelo más preciso, ya que significa que el error medio de predicción es pequeño en relación con los valores reales. Por ejemplo, un MAPE del 4.45% indica que el modelo, en promedio, tiene un error de predicción del 4.45% en relación con los valores reales. Por otro lado, un MAPE del 33.57% sugiere que las predicciones del modelo son menos precisas.

1.4.12 Docker

Es una plataforma de contenerización de aplicaciones que permite crear, desplegar y ejecutar aplicaciones en entornos aislados llamados contenedores. A diferencia de las máquinas virtuales, Docker emplea el mismo núcleo del sistema operativo, lo cual disminuye el consumo de recursos y aumenta la eficiencia, haciendo que las aplicaciones sean más portables entre diferentes entornos.

La arquitectura de Docker se compone de imágenes y contenedores, donde las imágenes son plantillas que contienen las dependencias y configuraciones necesarias para la ejecución de una aplicación y, los contenedores son versiones en ejecución de dichas imágenes.

Entre sus principales ventajas se encuentran la portabilidad, la eficiencia en el uso de recursos y el amplio soporte para servicios en la nube. En particular, se empleará esta tecnología sobre AWS para la creación del pipeline.

1.4.13 Artefactos

En el contexto de MLOps y DevOps, los artefactos son archivos generados en distintas etapas del pipeline que contienen los resultados intermedios o finales de un proceso. En un pipeline de MLOps, los artefactos pueden incluir modelos entrenados, archivos de datos preprocesados, logs de entrenamiento, métricas de evaluación y configuraciones de hiperparámetros. Estos artefactos son esenciales para asegurar la trazabilidad, reproducibilidad y despliegue de los modelos, ya que permiten almacenar y versionar los resultados de cada fase del proceso, facilitando su reutilización y auditoría en futuros desarrollos o en producción.

CAPÍTULO 2

2. METODOLOGÍA

En esta sección, se detallan los pasos y enfoques implementados para el desarrollo del proyecto. Se describe el diseño metodológico adoptado, incluyendo la aplicación de técnicas de machine learning como ARIMA y LSTM para la predicción de la velocidad de la máquina mezcladora, así como el uso de metodologías ágiles mediante sprints para gestionar y ejecutar el proyecto de manera eficiente. Además, se examinan el tipo y enfoque de investigación, el proceso de procesamiento y análisis de datos, así como la detección de valores atípicos, y la implementación de un pipeline de MLOps, lo cual permitió consolidar los modelos de predicción más eficientes.

2.1 Introducción

El diseño metodológico bajo el que se rige el proyecto está centrado en la integración de la metodología machine learning, para predecir la velocidad de la máquina mezcladora de una empresa de balanceados, siendo que, a partir de los algoritmos diseñados la empresa incremente su eficiencia, dado que reducen la incertidumbre en la ejecución de procesos en la operatividad y mejora la rentabilidad. Se utilizó los antecedentes históricos del algoritmo utilizado por la empresa para diseñar una propuesta de mejora continua.

Podemos definir seis fases clave para el desarrollo de este proyecto, cada una diseñada para abordar aspectos específicos del objetivo general de estimar la velocidad de la máquina mezcladora mediante métodos de machine learning. Estas fases permiten organizar el trabajo de manera estructurada, desde la definición inicial del proyecto hasta la entrega de resultados. A continuación, se describen las fases y sus respectivas actividades principales:

1. Definición del proyecto:
    a. Identificación de los objetivos de la investigación.
    b. Definición del alcance del proyecto, considerando el uso de ARIMA, SARIMA y redes neuronales LSTM.
    c. Selección del marco metodológico (CYNEFIN y Scrum).

2. Recopilación y preparación de datos (Sprint 1):
    a. Extracción de registros históricos de la máquina mezcladora.
    b. Limpieza, filtrado y estandarización de datos para su uso en los modelos.
    c. Implementación de un flujo ETL para organizar la información.
3. Análisis exploratorio y modelado inicial (Sprints 2 y 3):
    a. Análisis de series temporales para identificar patrones y estacionalidad.
    b. Configuración de entornos de cómputo en la nube para modelado avanzado.
    c. Implementación de los modelos ARIMA y SARIMA y análisis de sus resultados.
4. Desarrollo de redes neuronales LSTM (Sprint 4 y 5):
    a. Diseño y entrenamiento del modelo LSTM.
    b. Generación de conjuntos de datos derivados para entrenamiento y validación.
    c. Comparación de resultados entre modelos SARIMA y LSTM.
5. Implementación del pipeline de MLOps (Sprint 6):
    a. Despliegue de los modelos seleccionados en un pipeline automatizado utilizando AWS.
    b. Pruebas y validación de la solución final.
6. Revisión y entrega del proyecto:
    a. Análisis final de resultados.
    b. Documentación y presentación de conclusiones.
    c. Revisión de los objetivos alcanzados.

Utilizando el marco de trabajo CYNEFIN, se establece que el presente proyecto se encuentra en el dominio de los proyectos complejos, dado que, no solo implica la implementación de los métodos mencionados previamente, sino que también incluye un componente de revisión retrospectiva que se llevó a cabo mediante revisión semanal, adaptándose a los requerimientos de cambio en medida que avanzó la materia integradora, los cuales presentaron un alto grado de incertidumbre, ya que, en su mayoría, se implementaron por primera vez en el contexto de este proyecto, por ende, se aplicaron los lineamientos del marco de trabajo Scrum, dividiendo los avances del proyecto en sprints con una duración de dos semanas, considerando que, cada uno de

los sprints cuentan con una retroalimentación y gestión del cambio para la mejora, de la siguiente manera:

- **Sprint 1:** En la primera fase de trabajo, se abordó la implementación de un flujo ETL para los datos, lo cual, implicó la extracción de información de registros y reportes del procesamiento, siendo que los mismos fueron filtrados con la finalidad de obtener datos codificados y estandarizados bajo parámetros semanales.
- **Sprint 2:** Se realizó un análisis de las series de tiempo con el fin de encontrar el mejor camino para tratar los diferentes productos y encontrar los valores de parámetros necesarios para el modelado ARIMA. También debido a las limitaciones inherentes en la capacidad de cómputo de los recursos locales, se optó por la utilización de servicios de procesamiento en la nube. Esta estrategia permitió la implementación de un método de búsqueda exhaustiva de los parámetros en cuestión, proporcionando así un entorno más adecuado para la realización de cálculos computacionalmente intensivos.
- **Sprint 3:** consistió en entrenar un modelo SARIMA, graficar los valores obtenidos y analizar sus resultados.
- **Sprint 4:** se desarrolló un análisis para implementar una red LSTM a fin de con los datos procesados anteriormente poder generar los distintos conjuntos de datos necesarios para el entrenamiento.
- **Sprint 5**: se realizó el análisis de los resultados alcanzados por el entrenamiento de la red LSTM y ya con los dos métodos de machine learning se procedió a comparar sus resultados a fin de elegir el más conveniente.
- **Sprint 6:** en esta fase se desarrolló la implementación del pipeline de MLOps, el cual se basó en los servicios de AWS.

Se debe considerar que la metodología ágil de proyectos scrum se integra al estudio considerando que las retroalimentaciones de cada sprint permitieron tomar medidas preventivas para satisfacer y mejorar el rendimiento de los procesos, mediante

lo cual, se cumplieron los objetivos de investigación, así mismo, permitió mejorar la toma de decisiones en la gestión de cambios, reduciendo la incertidumbre en la realización de los procesos.

2.2 Tipo de Investigación

Al desarrollarse un procesamiento de datos se realizó una investigación cuantitativa, que involucra el análisis de información medible y cuantificable que puede ser comparada con una escala, cuyos resultados brindan un aporte exacto que permite establecer rangos específicos sobre el estado de un contexto específico (Hernández-Sampieri, et al., 2014), de realizar lo mencionado, dado que se procesó los datos registrados del algoritmo actual utilizado por la empresa, para procesarlos con otros modelos e integrar el de mayor precisión.

El procesamiento de los datos se realizó en diversas etapas denominadas sprint, en donde, se procede a ejecutar cada paso hasta la integración de los diferentes modelos y medir la precisión de los mismos, esto se desarrolló mediante el análisis de series de tiempo que pasaron de brindar datos cada media hora, a emitir informes diarios, esto con la intención de conocer la variación de la velocidad en periodos de tiempo más largos, permitiendo elaborar diversos análisis en torno al comportamiento de los datos.

Del mismo modo, se hizo uso de la metodología machine learning para encontrar el modelo con mayor precisión en torno a la predicción de la velocidad de la máquina mezcladora, considerando que, este permite que la empresa pueda preestablecer su capacidad de producción y coordinar lo recursos necesarios para cumplir con la planificación, en donde, puedan cubrir con la demanda existente manteniendo los estándares de calidad.

2.3 Enfoque de Investigación

El enfoque será exploratorio-comparativo dado que integra diversos modelos de análisis de series de tiempo en torno a la metodología machine learning para consolidar una propuesta en específico, siendo que, se comparó los resultados obtenidos en las diferentes fases, y con los diferentes métodos para definir el que emana mayor precisión, exponiendo ventajas para la organización, y como consecuencia de ello, una mejor gestión de los costos.

Dentro del campo del análisis predictivo de datos, la veracidad de la información existente es muy importante para crear pronósticos con mayor precisión, las cuales sirvan en el entorno de la toma de decisiones, de tal manera, al comparar los modelos determinados en los objetivos, y compararlos mediante su integración en periodos de tiempo y condiciones similares, exponiendo resultados verificables en su aplicación.

2.4 Objeto de Estudio

La mezcla es un procedimiento fundamental en la producción de alimentos balanceados, y es particularmente crítico para asegurar la calidad nutricional, por ende, es esencial llevar a cabo un procesamiento adecuado de los ingredientes y garantizar la uniformidad en la mezcla (Yuque-Mendoza, 2019). Dentro del ámbito industrial, una máquina mezcladora de productos balanceados hace referencia a un dispositivo diseñado con el propósito de unir diversos ingredientes en proporciones exactas para obtener un producto final uniforme, por ende, desempeñan un papel fundamental en una variedad de sectores, que incluyen la industria alimentaria, farmacéutica, química y agroindustrial, entre otros. Su principal función radica en garantizar que la mezcla resultante cumpla con los criterios particulares de calidad, cantidad y uniformidad establecidos.

En la presente investigación, la máquina mezcladora se consolidó como el objeto de estudio considerando que los datos históricos en torno a su velocidad se procesaron para favorecer la producción de balanceado, de tal manera que se pudo definir el modelo con mayor precisión en torno a la predicción de la velocidad, disminuyendo la incertidumbre en la operatividad.

2.5 Diseño metodológico

El diseño metodológico de este proyecto se estructuró en seis fases, cada una diseñada para abordar aspectos específicos del objetivo general de estimar la velocidad de la máquina mezcladora mediante métodos de machine learning. Estas fases proporcionaron un enfoque ordenado y sistemático que permitió garantizar la calidad de los datos, el desarrollo efectivo de los modelos y la implementación exitosa de la solución.

En la primera fase, definición del proyecto, se establecieron los objetivos de la investigación y se definió el alcance del proyecto, considerando la implementación de los modelos ARIMA,

SARIMA y redes neuronales LSTM. Además, se seleccionó el marco metodológico, combinando el enfoque CYNEFIN para clasificar la naturaleza compleja del proyecto con la metodología ágil Scrum, dividiendo el trabajo en sprints quincenales.

La segunda fase, recopilación y preparación de datos, incluyó la extracción de registros históricos de la máquina mezcladora, seguida de una limpieza exhaustiva para garantizar la calidad de los datos. En esta etapa, se aplicó el método Z-Score para identificar valores atípicos, calculando desviaciones que pudieran afectar los resultados. Además, los datos fueron filtrados, estandarizados y organizados mediante un flujo ETL, asegurando que estuvieran listos para el análisis exploratorio y el entrenamiento de los modelos.

En la tercera fase, análisis exploratorio y modelado inicial, se utilizó el análisis de series de tiempo para identificar patrones y tendencias en las variables predictoras. Los modelos ARIMA y SARIMA fueron desarrollados para analizar estacionalidades y tendencias a corto, mediano y largo plazo. Esta etapa también incluyó el uso de entornos de cómputo en la nube para realizar cálculos intensivos y optimizar los parámetros de los modelos.

Durante la cuarta fase, desarrollo de redes neuronales LSTM, se implementó un enfoque ETL para recolectar, transformar y cargar los datos necesarios para entrenar la red neuronal. En esta etapa, los datos procesados previamente fueron utilizados para entrenar y ajustar el modelo LSTM, generando conjuntos de datos derivados para garantizar la precisión de los resultados.

La quinta fase, implementación del pipeline de MLOps, incluyó el despliegue del modelo seleccionado en un entorno automatizado utilizando AWS. Este pipeline permitió integrar el modelo de manera eficiente, asegurando su reproducibilidad y escalabilidad para la producción.

Finalmente, en la sexta fase, revisión y entrega del proyecto, se analizaron los resultados obtenidos por los modelos implementados, comparando su rendimiento mediante métricas como el MAPE. Se seleccionó el modelo más eficiente, y los resultados fueron documentados y presentados como parte de las conclusiones del proyecto.

CAPÍTULO 3

3. DISEÑO E IMPLEMENTACIÓN

3.1 Proceso para Elaboración de Mezclado

Las empresas avícolas, porcinas y ganaderas, dependen de la alimentación proporcionada por diversos tipos de balanceados, por ende, requieren un suministro ininterrumpido de estos, de esta manera, las instalaciones encargadas de la producción de piensos operan con sistemas complejos que involucran múltiples etapas en la producción de alimentos destinados a los animales de granja, y entre estos procesos, la máquina mezcladora se destaca como un recurso fundamental, como se observa en la Fig. 3.1 de proceso.

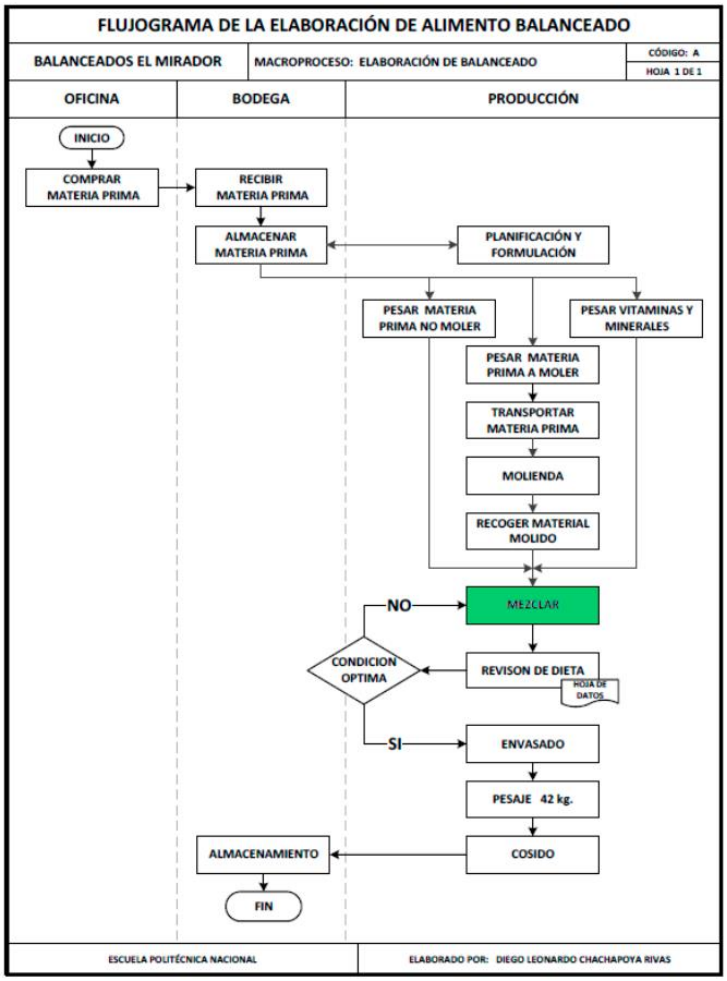


Figura 3.1 Flujograma Elaboración de Balanceado

La función central de la mezcladora es combinar diversos ingredientes, que son las materias primas componentes de un producto final. Por ejemplo, en la elaboración de productos como "Engorde 1 Mig", que consta de 34 ingredientes, la máquina mezcladora es esencial en el proceso ya que asegura la homogeneidad de la mezcla.

3.2 Análisis de los conjuntos de datos

Se realizó un análisis de los datos recolectados en los registros de la empresa, no obstante, en la fase de exploración de los datos se identificaron algunas discrepancias, encontrándose variaciones notorias entre las cantidades procesadas en los primeros años de registro y los datos más recientes, acorde a la Fig. 3.2:

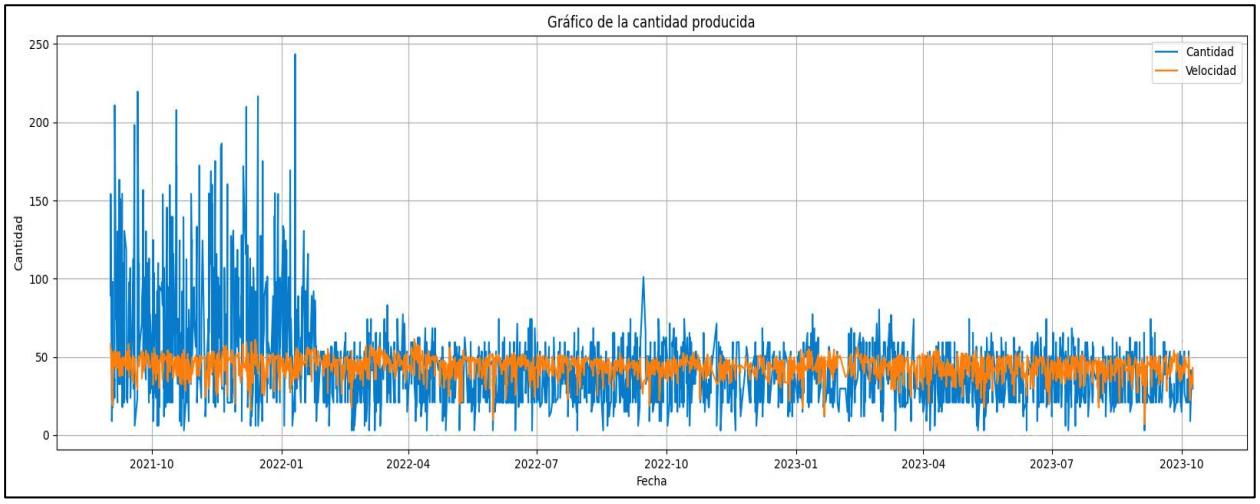


Figura 3.2 Serie de tiempo graficada-Cantidad Producida

Luego de un análisis exhaustivo y tras revisar el ciclo de vida de los datos, se determinó que la empresa carece de una sólida gobernanza de datos. Esto resultó en la alteración de la información, debido a un fallo en uno de los sistemas que la transmite al sistema de planificación. Sin embargo, se identificó una fuente adicional para obtener estos datos: el sistema de producción que supervisa la mezcladora. Este sistema también contenía la información necesaria para la investigación. Para procesarla, fue necesario depurar los datos, ya que se detectaron valores atípicos, como se observa en la Fig. 3.3:

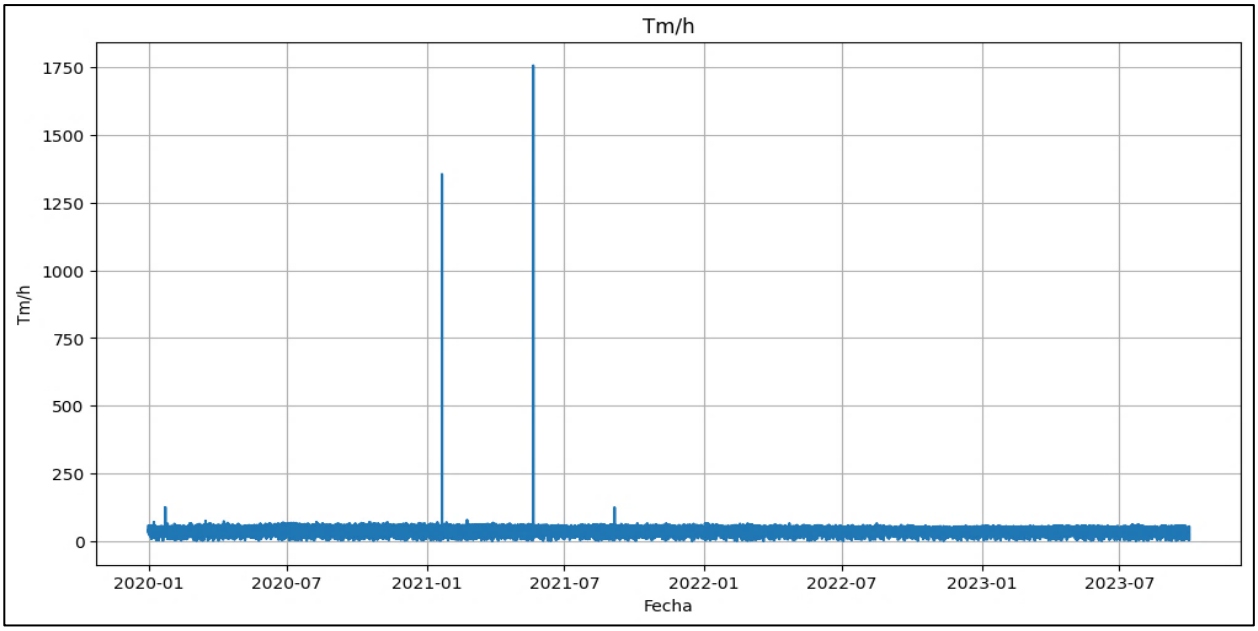


Figura 3.3 Detección Datos Atípicos

En la Figura 3.3 se observan los datos atípicos detectados en la serie temporal de producción de la máquina mezcladora. Se observa que hay varios picos pronunciados que indican valores extremadamente altos de toneladas por hora (Tm/h), especialmente en los primeros meses de 2021. Estos picos, que superan significativamente los valores normales del resto de la serie, sugieren la presencia de posibles errores de registro o condiciones operativas excepcionales que no son representativas del comportamiento habitual de la máquina.

Para analizar más detalladamente estos datos atípicos, se elaboró un diagrama de caja mostrado en la Fig. 3.4. En este diagrama, los puntos que se encuentran alejados del rango intercuartílico (representado por los límites de la caja) confirman la existencia de múltiples valores atípicos. Estos puntos, situados considerablemente por encima de la mayoría de las observaciones, refuerzan la sospecha de anomalías en el registro de datos o condiciones operativas fuera de lo común. Comparando ambas figuras, se concluye que los valores atípicos detectados son notablemente extremos y que su presencia podría influir negativamente en la calidad del análisis predictivo si no se tratan o eliminan adecuadamente del conjunto de datos. Este análisis sugiere la necesidad de

aplicar métodos robustos de limpieza y preprocesamiento para mejorar la calidad de los modelos de predicción utilizados en el estudio.

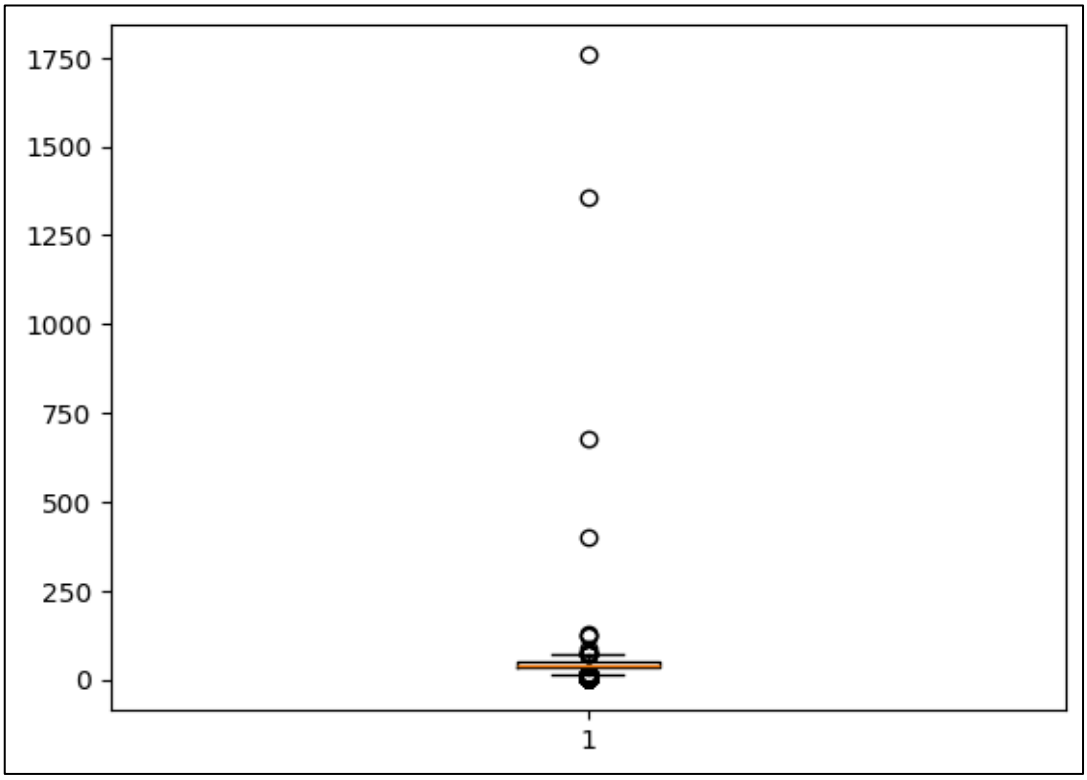


Figura 3.4 Diagramas de Caja de Datos

3.3 Eliminación de valores atípicos

Para la eliminación de los valores atípicos, se evaluaron dos alternativas en la selección del método: aplicar la técnica de Z-Score o el Rango Intercuartil (IQR). A pesar de que en la prueba de normalidad se obtuvo p = 0.35, indicando la posibilidad de una distribución normal en los datos y, por lo tanto, la idoneidad de cualquiera de los dos métodos, se optó por utilizar el método IQR, considerando la robustez del IQR ante valores atípicos extremadamente elevados. Se tuvo en cuenta que la media de los datos es de 39.93 t/h, mientras que el valor máximo alcanza 1756.67 t/h, lo cual podría afectar considerablemente al método Z-Score. Tras aplicar el método elegido para suprimir los valores aberrantes, los datos se presentan de la siguiente manera en la Fig. 3.5.

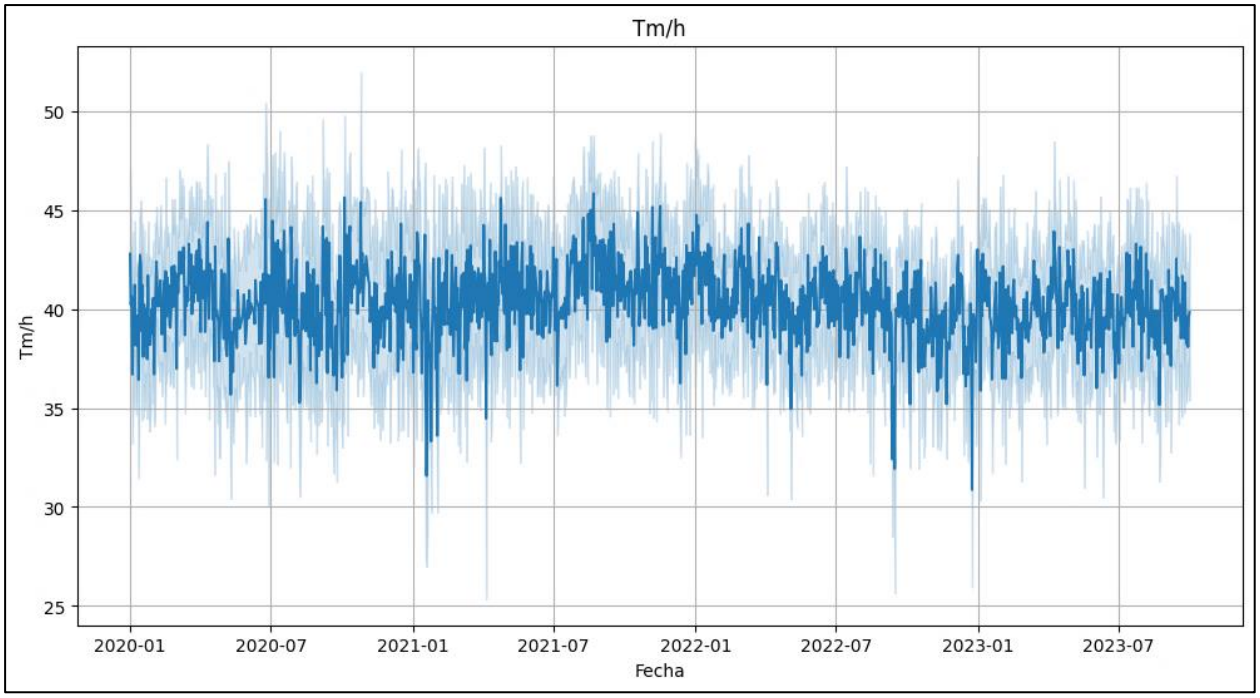


Figura 3.5 Datos sin Valores Atípicos

Como se puede observar el comportamiento de los datos al extraer los valores atípicos disminuyo las variaciones que se salían del rango evaluado, permitiendo comprobar las alteraciones producidas por los valores atípicos, los cuales generaban cambios en la veracidad de los resultados.

A fin de corroborar visualmente, se muestra nuevamente el gráfico de caja de los datos sin valores atípicos, el cual se observa en la Fig. 3.6.

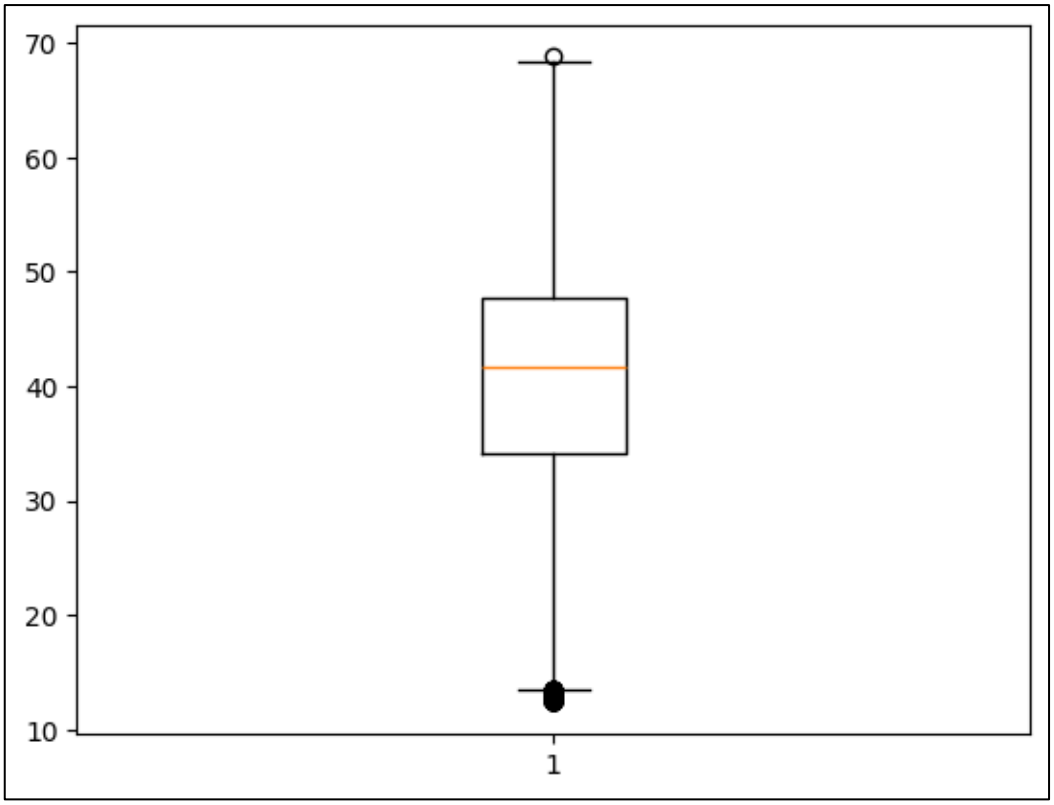


Figura 3.6 Diagrama de Caja sin valores atípicos

En la Fig. 3.6 Se puede observar que, al excluir los datos atípicos expuestos, estos ya no se marcan en el diagrama y del mismo modo la media se recalculó en alrededor de 40t/h.

El cambio en la media después de eliminar los valores atípicos indica que los datos extremos o anómalos estaban afectando significativamente la medida central de los datos originales. En la versión inicial de los datos, con los valores atípicos incluidos, la media era más alta debido a la influencia de estos valores extremadamente elevados, como los picos que superan las 1750 t/h.

Al suprimir estos datos atípicos utilizando la técnica del Rango Intercuartil (IQR), los datos resultantes tienen una media recalculada de aproximadamente 40 t/h, lo que es más representativo del comportamiento típico del sistema bajo condiciones normales de operación. Este cambio sugiere que los valores atípicos distorsionaban las conclusiones que se podían obtener sobre la capacidad de producción de la máquina mezcladora, haciendo que los resultados sean menos fiables y precisos.

Por lo tanto, la nueva media ofrece una representación más precisa del desempeño promedio de la máquina mezcladora sin la influencia de eventos anómalos

o errores de registro. Esto mejora la veracidad de cualquier análisis posterior y asegura que las decisiones basadas en estos datos sean más confiables.

3.4 Criterios de selección y filtrado de productos para el modelado predictivo

La atención en los productos se centra en aquellos que cuentan con una mayor cantidad de observaciones de velocidad en los datos, dado que, algunos productos pueden tener apariciones ocasionales y podrían tener un comportamiento distinto en comparación con aquellos que son mezclados con mayor frecuencia, por ende, un mapa de calor ilustra los productos con el mayor número de mediciones en el conjunto de datos.

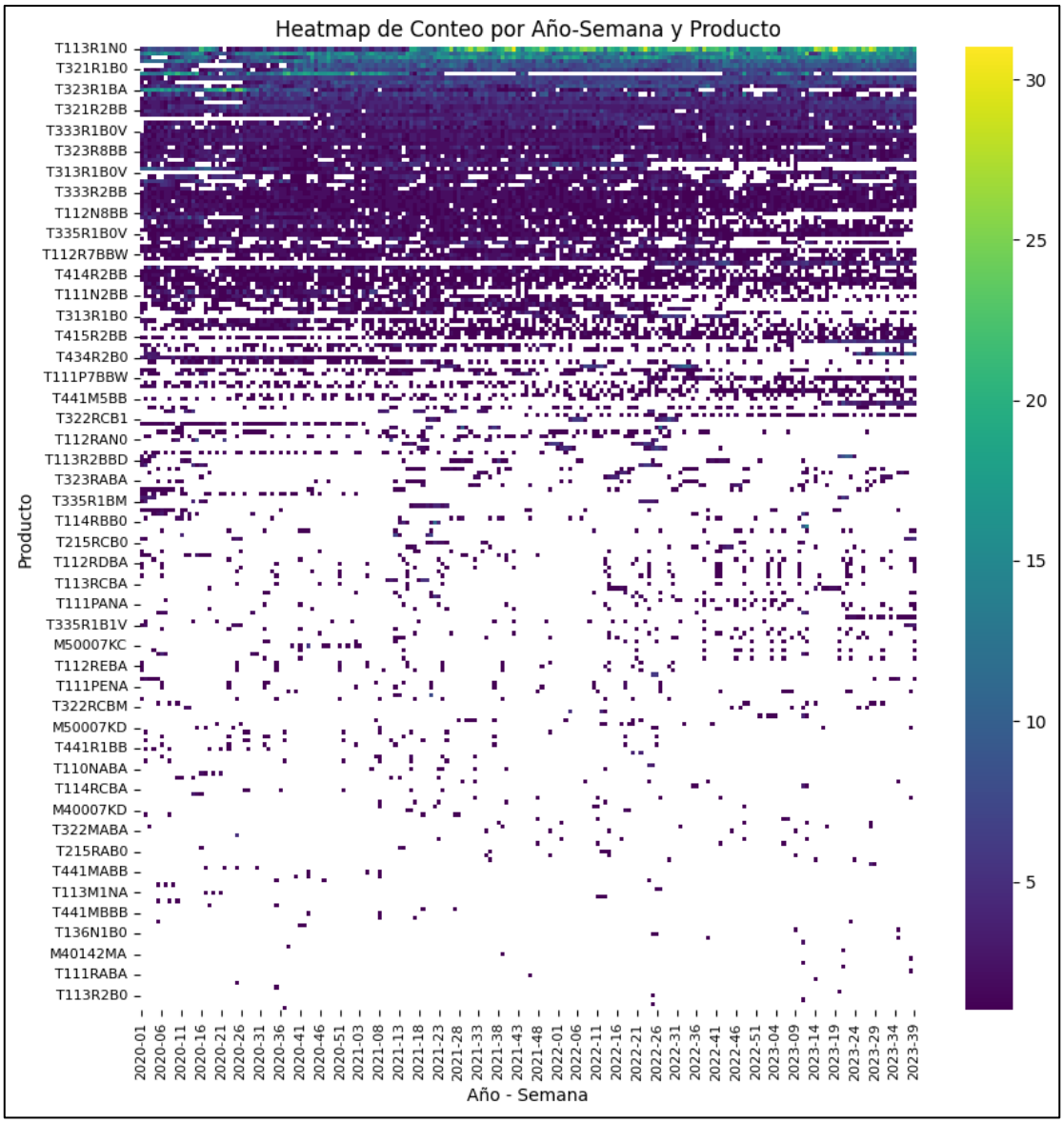


Figura 3.7 Mapa de calor de los productos mezclados

Se tuvo un conjunto de 234 productos que cuentan con observaciones en el registro de datos del estudio, a lo cual, aplicando el principio de Pareto 80-20, se seleccionaron aquellos que abarcan el 80% de las observaciones, ya que es probable que los demás presenten datos faltantes y, por ende, no sea adecuado aplicar los mismos métodos que a la mayoría.

Después de haber identificado y elegido los productos que comprenden el 80% de las observaciones, se determinó que pertenecían a 46 productos. Se procede a presentar la Fig. 3.8, que muestra un histograma de los productos seleccionados.

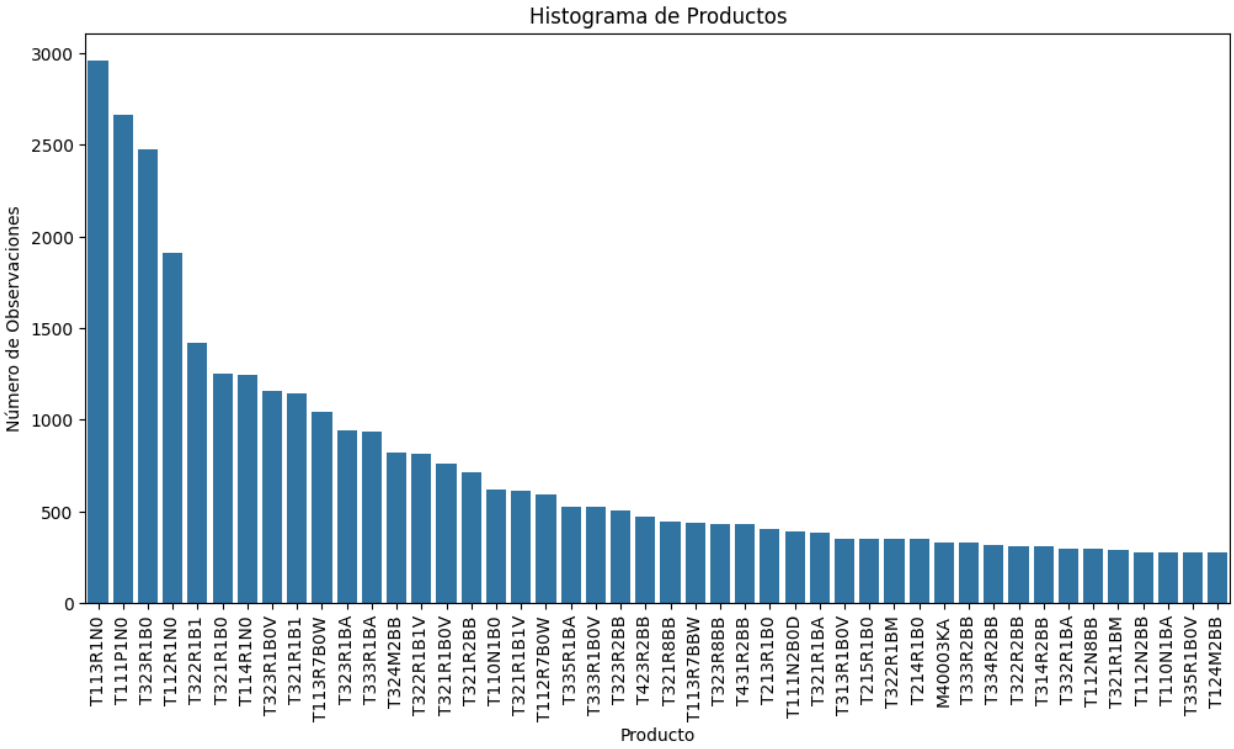


Figura 3.8 Histograma de Productos

A partir de la Figura 3.8, se observa que un pequeño número de productos concentra la mayoría de las observaciones, lo que justifica la aplicación del principio de Pareto (80-20) para seleccionar los productos a modelar.

Los productos con más observaciones, como T113R1N0 y T111P1N0, son los más frecuentemente mezclados, mientras que otros tienen menos datos disponibles. Esta distribución sugiere que es más eficiente y preciso centrar el modelado en los 46 productos seleccionados, ya que cuentan con suficientes datos para generar

predicciones fiables, evitando problemas derivados de la falta de datos en productos menos frecuentes.

3.5 Conversión a frecuencia semanal de la serie de tiempo

Al determinar los productos se toma el producto con mayor número de observaciones como referencia para realiza el seguimiento respectivo por un periodo de dos semanas, lo cual, se procede a exponer en la Fig. 3.9:

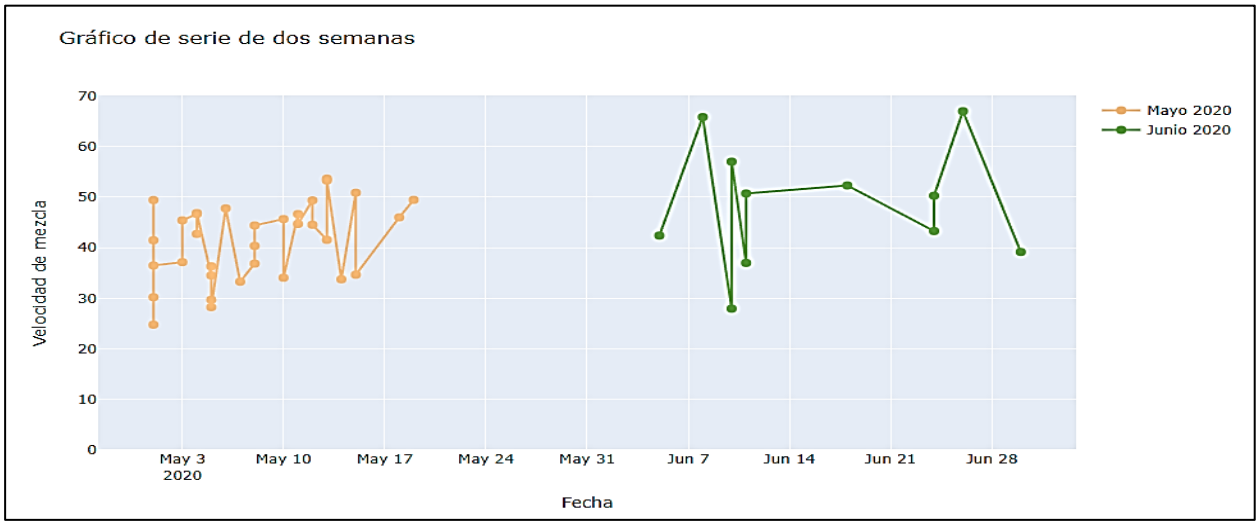


Figura 3.9 Observación de un producto- Periodo de dos semanas

En la Fig. 3.9 se observan las mediciones de velocidad tomadas en un día, y se identificó la presencia de días sin observaciones, específicamente desde el 20 de mayo al 5 de junio, y otros con múltiples observaciones, como el 1 y el 3 de mayo. Por ende, se optó por normalizar los datos y obtener un valor semanal único.

La decisión de tomar un valor semanal único fue dada por el jefe de planificación, ya que la planta de procesamiento sigue un ciclo de producción semanal, al igual que sus métricas e indicadores.

El valor semanal se calculó como el promedio de las velocidades registradas durante la semana para cada producto, por ende, se tuvo que implementar una imputación temporal hacia adelante (Forward Fill) de valores para las semanas que no tenían datos registrados. La serie en una frecuencia semanal se refleja en la Fig. 3.10.

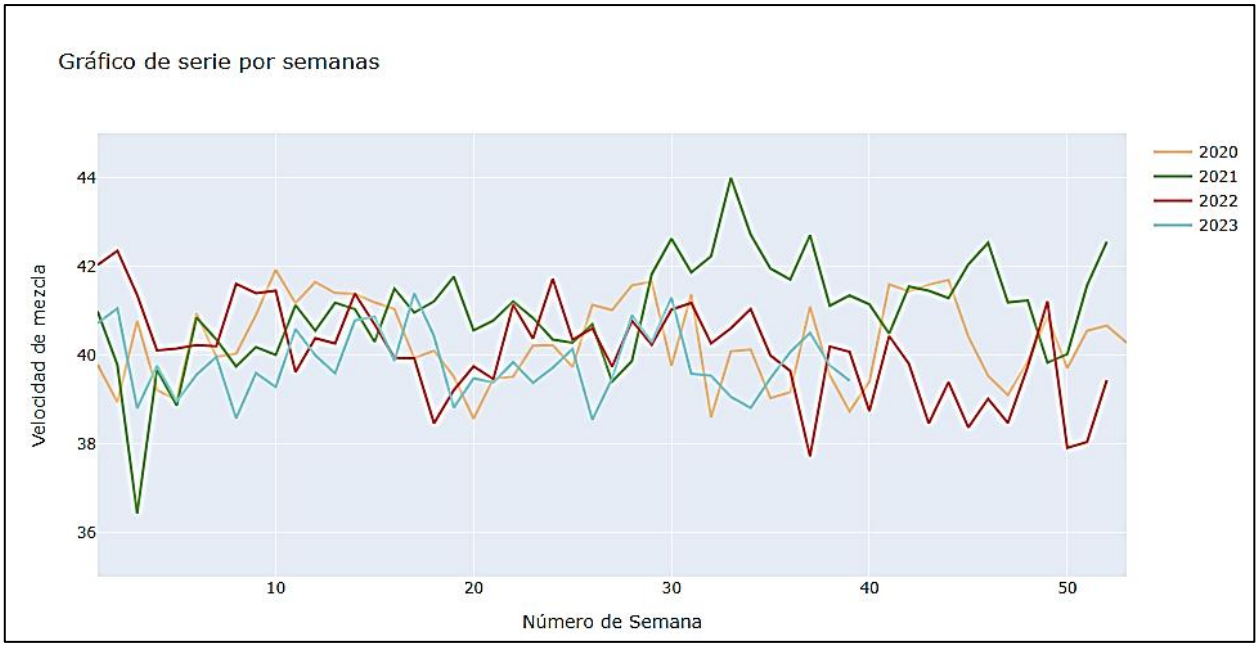


Figura 3.10 Promedio Velocidad/semana del Producto seleccionado

En la Fig. 3.10 se ilustra una serie de tiempo particular para un producto (T113R1N0) en lapsos semanales, en la cual se observa que no existen cortes en los gráficos de línea, manteniendo la serie temporal completa con frecuencia semanal, lo cual es el objetivo a fin de ajustar un modelo de machine learning.

Al considerar los 46 productos seleccionados se incrementa la complejidad de su visualización gráfica, por ende, el análisis se consolidó de los cinco productos con mayor incidencia en la producción, lo cual, permite observar cómo varían las series de tiempo según el producto, como se observa en la Fig. 3.11:

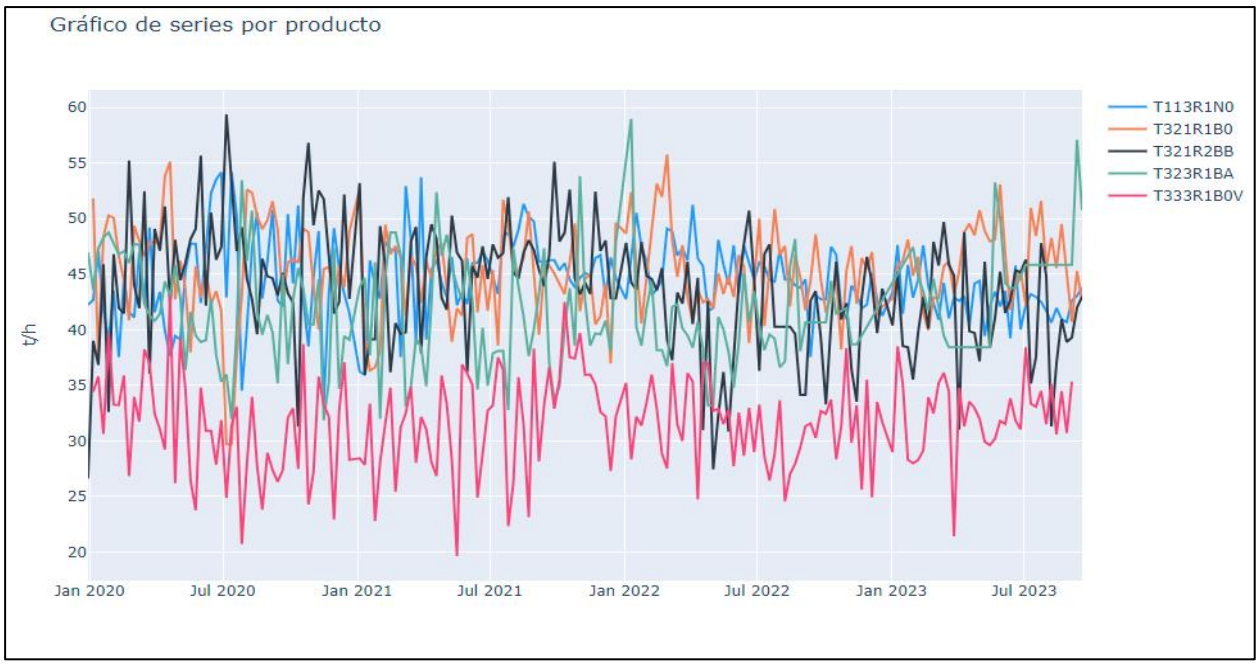


Figura 3.11 Series por Producto

De la Figura 3.11, se puede concluir que las series temporales de velocidad de producción varían significativamente según el producto. Cada línea representa un producto diferente y muestra fluctuaciones a lo largo del tiempo, indicando variabilidad en la velocidad de mezcla. Algunos productos, como T113R1N0 y T321R1B0, muestran patrones más estables con menor variación en la velocidad, mientras que otros, como T333R1B0V, exhiben más fluctuaciones, lo que puede ser indicativo de diferentes procesos operativos o condiciones de producción. Estas diferencias sugieren que cada producto puede requerir un enfoque de modelado específico para ajustar la precisión de las velocidades estimadas, por ende, se toma la decisión de ajustar modelos para cada producto.

3.6 Verificación de autocorrelación de la serie temporal

Con el propósito de evaluar la existencia de correlación en la serie, se ha generado un gráfico de autocorrelación utilizando los datos, lo cual se refleja en la Fig. 3.12.

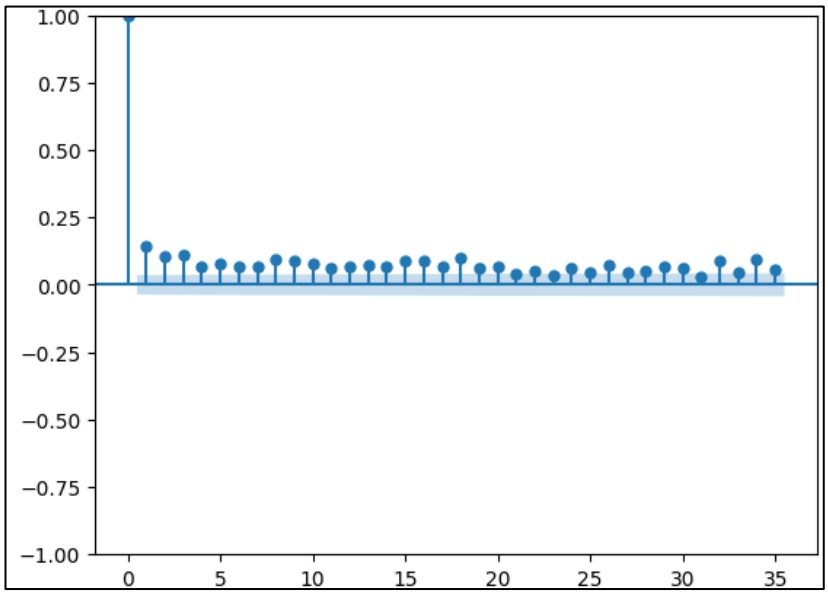


Figura 3.12 Autocorrelación

Al examinar el gráfico, se aprecia la ausencia de ruido blanco, lo que sugiere la presencia de autocorrelación, por ende, como complemento a esta observación, se realiza la prueba de Ljung-Box, con H_0 (hipótesis nula) de que la serie corresponde a ruido blanco, esta prueba arrojó un valor de p extremadamente bajo, calculado en $1.2280e^{-49}$, lo que conlleva al rechazo de la hipótesis H_0 e indica que los datos probablemente exhiben autocorrelación significativa. De esta forma, la significancia estadística de este resultado respalda la presunción de que hay relaciones temporales en la serie.

Esta misma prueba se aplicó a los 46 productos teniendo los valores p que se observa en la siguiente tabla.

Tabla 1 Prueba Ljung-Box aplicada a 46 productos

Producto	Valor p	Conclusión
T111P1N0	4.11e-56	
T112N8BB	0.06	Ruido Blanco
T322R1BM	2.93e-18	
T321R8BB	0.03	
T113R7B0W	2.38e-06	

T214R1B0	1.44e-28	
T323R1BA	8.84e-06	
T323R1B0V	0.0047	
T114R1N0	0.0017	
T112R1N0	4.36e-05	
T324M2BB	3.98-08	
T113R1N0	1.30e-05	
T333R2BB	0.04	
T111N2B0D	0.00040	
T321R1BA	4.12e-10	
T334R2BB	0.51	Ruido Blanco
T314R2BB	0.24	Ruido Blanco
T423R2BB	8.33e-05	
T323R2BB	1.73e-22	
T213R1B0	1.94e-15	
T321R1B0V	0.83	Ruido Blanco
T332R1BA	0.09	Ruido Blanco
T323R1B0	2.49e-09	
T322R2BB	0.00036	
T323R8BB	4.62e-18	
T110N1B0	9.79e-05	
T431R2BB	1.42e-19	
T124M2BB	0.00046	
T322R1B1V	0.00033	
T321R1B1	7.66e-103	
T113R7BBW	2.03e-05	
T321R2BB	4.48e-11	
T321R1BM	0.61	Ruido Blanco
T112N2BB	0.09	Ruido Blanco
T333R1BA	9.44e-06	
M40003KA	7.38e-18	
T333R1B0V	0.01	
T321R1B0	9.96e-05	
T335R1BA	2.40e-08	
T112R7B0W	0.001029	
T110N1BA	0.006668	
T215R1B0	0.44	Ruido Blanco
T335R1B0V	0.0145	
T322R1B1	8.21e-35	
T313R1B0V	0.24	Ruido Blanco
T321R1B1V	0.037	

Según la tabla, hay un conjunto de nueve productos cuyas series temporales muestran un comportamiento de ruido blanco. Debido a esta característica, no es viable aplicar métodos de modelado de series temporales para hacer pronósticos. Por lo tanto, estos productos no serán considerados en el procesamiento del estudio. El análisis se enfocará exclusivamente en un conjunto de 37 productos para los análisis posteriores.

3.7 Modelado con SARIMA

Después de confirmar que las series temporales no exhiben un comportamiento de ruido blanco y presentan estacionalidad, se procedió a estimar un modelo SARIMA. Primero, se realizó una búsqueda en cuadrícula para determinar los parámetros p, d, q, P, D, Q y S necesarios para configurar el modelo SARIMA. Este proceso es automático, pero computacionalmente intensivo. La búsqueda se centró en minimizar el criterio AIC para identificar el mejor ajuste. En la Tabla 3.2 se muestran los resultados de la búsqueda de parámetros por cada serie correspondiente a los distintos productos.

Tabla 2 Parámetros por Productos

Producto	AIC	p	d	q	P	D	Q	S
T111P1N0	902,73	0	1	1	2	0	2	12
T322R1BM	12	1	2	0	2	0	2	12
T321R8BB	10	0	0	1	2	0	1	12
T113R7B0W	8	0	0	1	2	0	0	12
T214R1B0	10	2	0	0	2	0	0	12
T323R1BA	12	0	0	2	2	0	1	12
T323R1B0V	14	2	2	0	2	2	2	12
T114R1N0	12	0	0	2	2	0	1	12
T112R1N0	14	2	2	2	2	2	0	12
T324M2BB	901,92	2	1	1	0	1	1	12
T113R1N0	992,72	1	0	1	2	2	2	12
T333R2BB	39,73	2	2	2	1	1	2	12
T111N2B0D	915,99	0	1	1	0	2	2	12
T321R1BA	1199,53	1	0	0	0	2	2	12
T423R2BB	14	2	0	2	1	0	1	12
T323R2BB	14	0	0	2	2	0	2	12
T213R1B0	1236,04	0	1	2	0	2	2	12
T323R1B0	12	1	2	2	1	1	1	12
T322R2BB	607,24	0	0	0	2	2	2	12
T323R8BB	18	2	2	2	2	2	2	12
T110N1B0	1170,99	1	0	1	0	2	2	12
T431R2BB	12	0	0	2	2	0	1	12

T124M2BB	112,67	2	2	2	1	0	0	12
T322R1B1V	1084,56	1	0	1	2	2	2	12
T321R1B1	1090,38	2	1	2	0	2	2	12
T113R7BBW	1194,28	0	0	2	0	2	2	12
T321R2BB	12	0	1	1	2	2	2	12
T333R1BA	12	1	0	1	1	0	2	12
M40003KA	14	2	0	0	2	2	2	12
T333R1B0V	249,94	1	2	2	1	2	0	12
T321R1B0	16	2	0	2	1	0	2	12
T335R1BA	10	0	0	2	2	0	0	12
T112R7B0W	1237,51	1	0	0	0	2	2	12
T110N1BA	1193,62	1	0	0	0	2	2	12
T335R1B0V	1224,8	0	0	1	2	2	1	12
T322R1B1	12	2	0	1	0	0	2	12
T321R1B1V	860,73	1	0	0	0	2	2	12

Una vez determinados los parámetros necesarios, se procede a la configuración del modelo SARIMA. En la Fig. 3.13 se muestra el correspondiente modelo ajustado del producto T111P1N0 el cuál fue previamente elegido porqué es el producto con mayor número de observaciones.

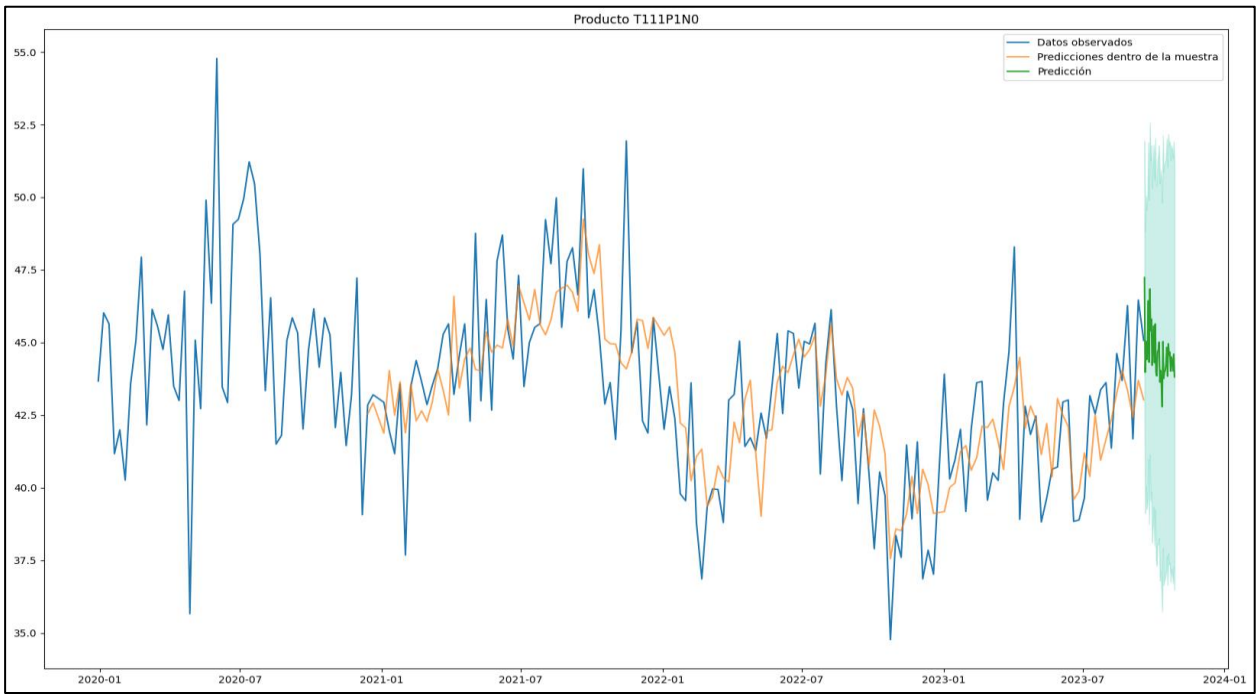


Figura 3.13 Modelo SARIMA Ajustado

El gráfico de la Fig. 3.13 muestra que el modelo está ajustado adecuadamente a la serie temporal del producto T111P1N0, lo cual, se respalda con el cálculo del error MAPE, que

revela un valor del 4.45%, indicando una aproximación precisa. No obstante, existen modelos ajustados para otros productos con un MAPE más elevado, alcanzando un 33.57%, como se muestra en la Fig. 3.14 del modelo ajustado.

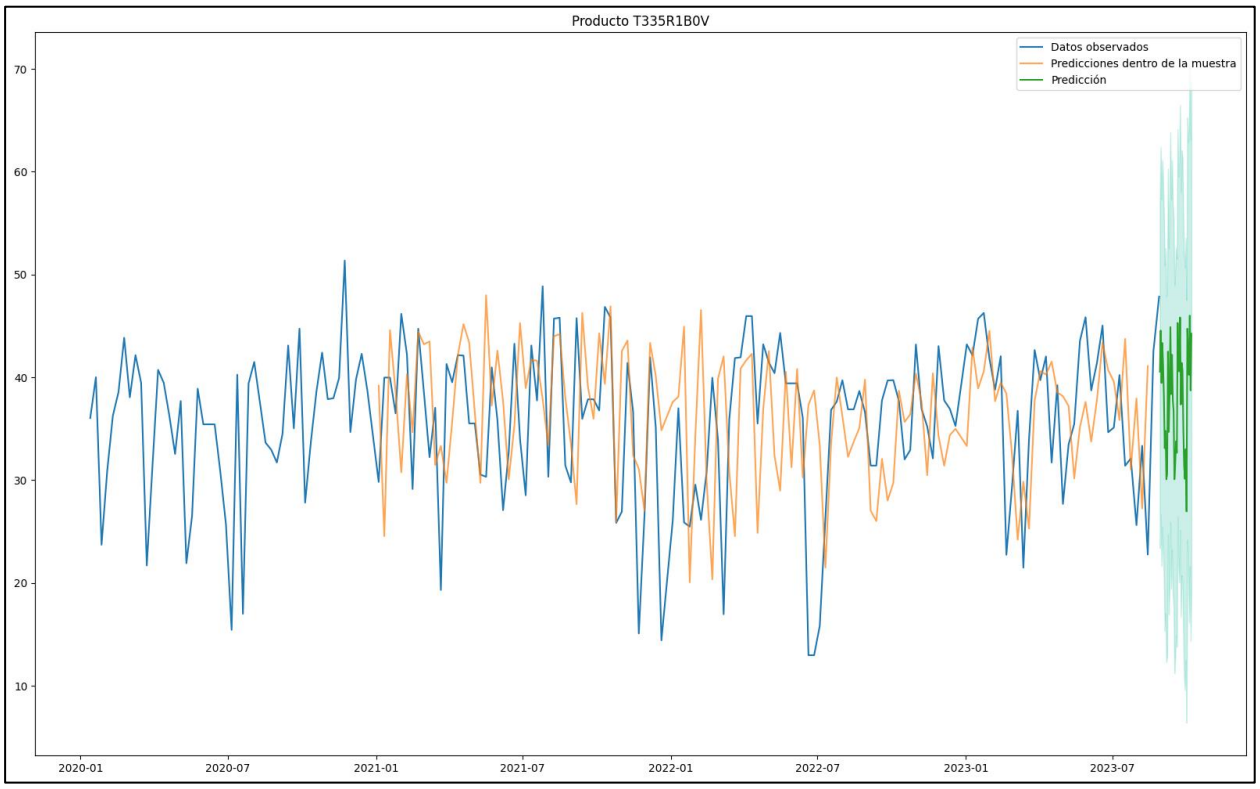


Figura 3.14 Modelado SARIMA

Al aplicar los parámetros encontrados en la búsqueda en cuadrícula, se logró modelos con MAPE en el rango del 4.45% al 33.57%, de esta forma, las gráficas de los modelos ajustados se muestran en el Apéndice 3.

3.8 Modelado de LSTM

Para el modelado de la Red LSTM se tomó en cuenta los mismos datos consolidados en el modelado SARIMA, con la intención de comparar los modelos acordes a los objetivos específicos planteados previamente. Se resalta que se entrenó una red neuronal para cada producto, ya que cada uno representa una serie de tiempo única, cómo se concluyó en el apartado 3.5 de este trabajo. Para lograr ajustar la red LSTM, el conjunto de datos que consta de 6,888 registros se dividirá en conjuntos de entrenamiento, validación y prueba, en proporciones de 70%, 15% y 15%, respectivamente, lo cual, se puede observar en la Tabla 3.3:

Tabla 3 Conjunto de Datos de Series Temporales por Producto

Producto	Observaciones
T111P1N0	186
T321R8BB	186
T113R7B0W	186
T214R1B0	186
T323R1B0V	186
T112R1N0	186
T324M2BB	186
T113R1N0	186
T333R2BB	186
T423R2BB	186
T323R2BB	186
T213R1B0	186
T323R1B0	186
T323R8BB	186
T110N1B0	186
T431R2BB	186
T322R1B1V	186
T321R1B1	186
T321R2BB	186
T322R2BB	185
T124M2BB	185
T321R1B0	185
T112R7B0W	185
M40003KA	184
T113R7BBW	183
T333R1B0V	183
T335R1BA	183
T333R1BA	181
T335R1B0V	181
T322R1BM	180
T322R1B1	180
T110N1BA	179
T321R1BA	178
T323R1BA	176
T111N2B0D	145
T321R1B1V	143
T114R1N0	116

Con estas observaciones, se organizaron conjuntos de entrenamiento, asegurando un orden cronológico, ya que se ajustan a un modelo de series temporales, utilizando un enfoque de cinco pasos en el tiempo para el conjunto de entrenamiento, empleando la biblioteca de redes neuronales Keras, se entrenó el modelo en 50,000 épocas con 32 de tamaño de lote, también se integró la técnica de parada temprana (early stopping, en inglés) para finalizar el entrenamiento en el caso de que no hay mejoras en el modelo durante 100 épocas, con estos parámetros, se calculó la métrica MAPE y obtuvo los siguientes valores para cada producto.

Tabla 4 Métrica MAPE

Producto	MAPE
T111N2B0D	3,26
T113R1N0	3,52
T112R1N0	3,65
T114R1N0	4,82
T323R1B0	5,36
T322R1B1	5,36
T323R1B0V	5,65
T111P1N0	5,73
T321R1B0	6
T321R1B1	6,05
T321R8BB	6,47
T324M2BB	6,77
T323R1BA	6,87
T322R1B1V	7,75
T110N1B0	8,09
T333R1B0V	8,34
T333R1BA	8,88
T321R1B1V	9,55
T323R8BB	10,05
T321R2BB	10,68
T321R1BA	11,44
T423R2BB	11,59
T431R2BB	12,26
T213R1B0	12,61
T214R1B0	13,05
T124M2BB	13,05
T113R7B0W	13,49
T322R1BM	13,79
T323R2BB	13,88

T322R2BB	14,95
T110N1BA	15,23
T113R7BBW	17,04
T112R7B0W	17,46
T335R1B0V	18,31
M40003KA	20,58
T333R2BB	20,74
T335R1BA	26,1

Siguiendo el enfoque utilizado en el método SARIMA, se representa gráficamente la serie temporal del producto T111N2B0D que tiene el valor de MAPE más bajo, el cual es 3.26%. Esta representación se observar en la Fig. 3.15, en la cual nos muestra que los valores reales son constantes entre los pasos de tiempo cero a 19, y que la gráfica de las predicciones tiene una diferencia mayor en dos intervalos.

El primer intervalo en tener mayor diferencia es entre el paso de tiempo cero y tres, el segundo intervalo es para los valores entre 19 y 23. También muestra que la diferencia entre los valores reales y los predichos entre el intervalo 3 y 19 es cercano a ser constante, y por eso su valor de MAPE relativamente bajo.

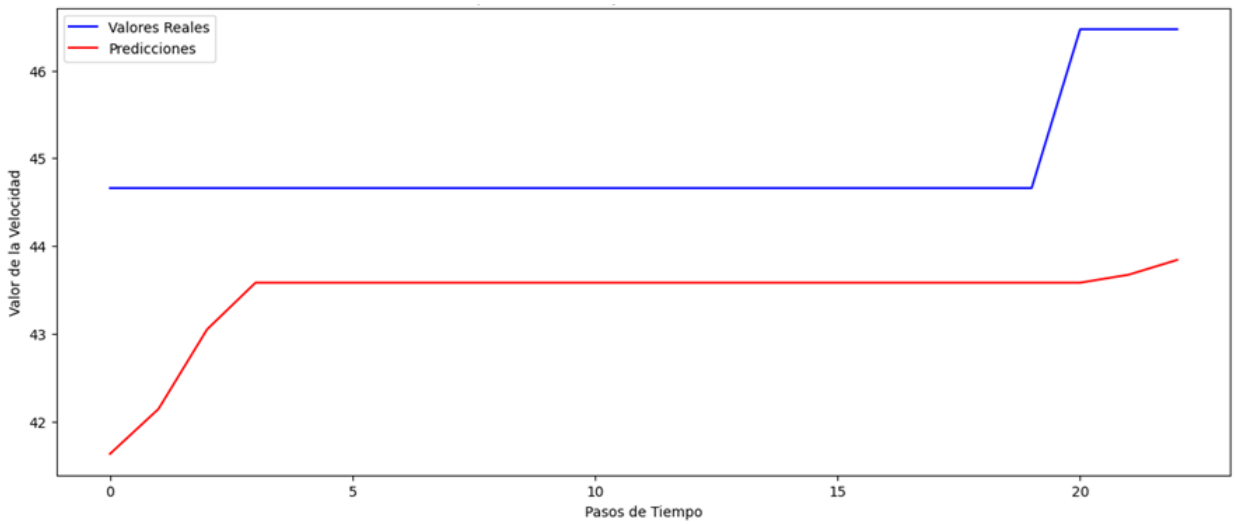


Figura 3.15 Comparativo Valores Reales y Predicciones con MAPE 3.26%

De la Fig. 3.15, podemos concluir que la red LSTM es bastante precisa para predecir velocidades de mezcla de productos cuyos valores no presentan fluctuación considerable en los pasos del tiempo.

Posteriormente la Fig. 3.16 muestra la gráfica de los valores reales y las predicciones del producto que obtuvo el MAPE más alto (26.1%). Este producto es el T335R1BA.

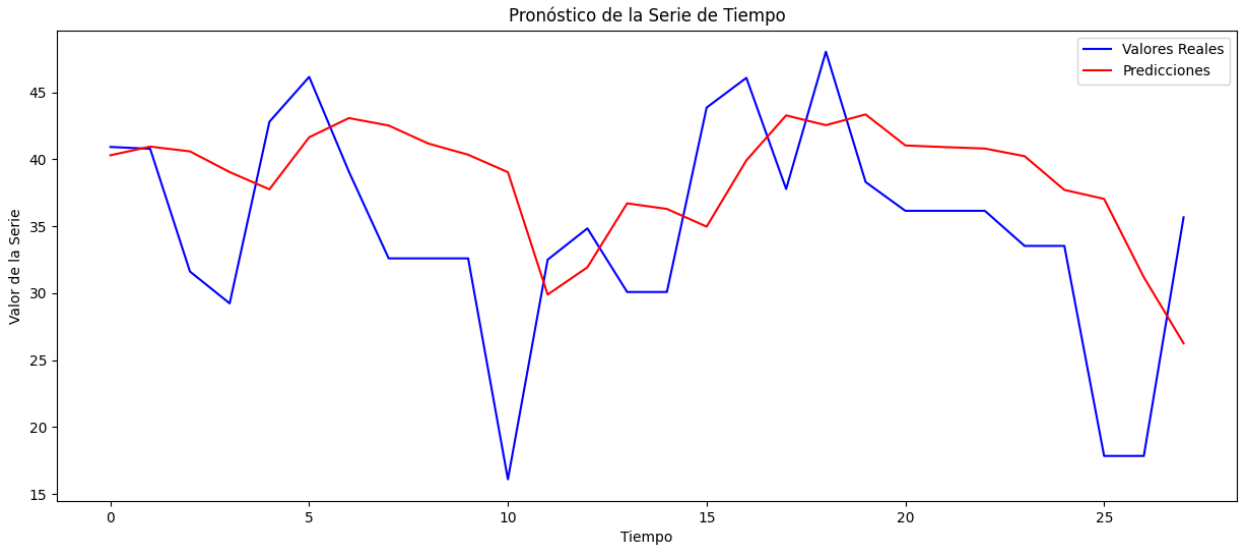


Figura 3.16 Comparativo Valores Reales y Predicciones con MAPE 26.1%

Podemos observar en la Fig. 3.16 que los valores medidos de velocidad son mucho más cambiantes que los de la Fig. 3.15, pero a pesar de contener mayores fluctuaciones, las predicciones se ajustan aceptablemente a los valores reales. De esto se concluye que el modelado de una red LSTM para este conjunto de datos es viable.

3.9 Valores de MAPE

Luego del ajuste de los modelos en estudio, se procede a evaluar su precisión mediante el cálculo del MAPE para cada modelo correspondiente a los distintos productos analizados. La Fig. 3.17 ofrece una comparativa detallada entre los valores de MAPE obtenidos en los modelos ajustados de SARIMA y LSTM.

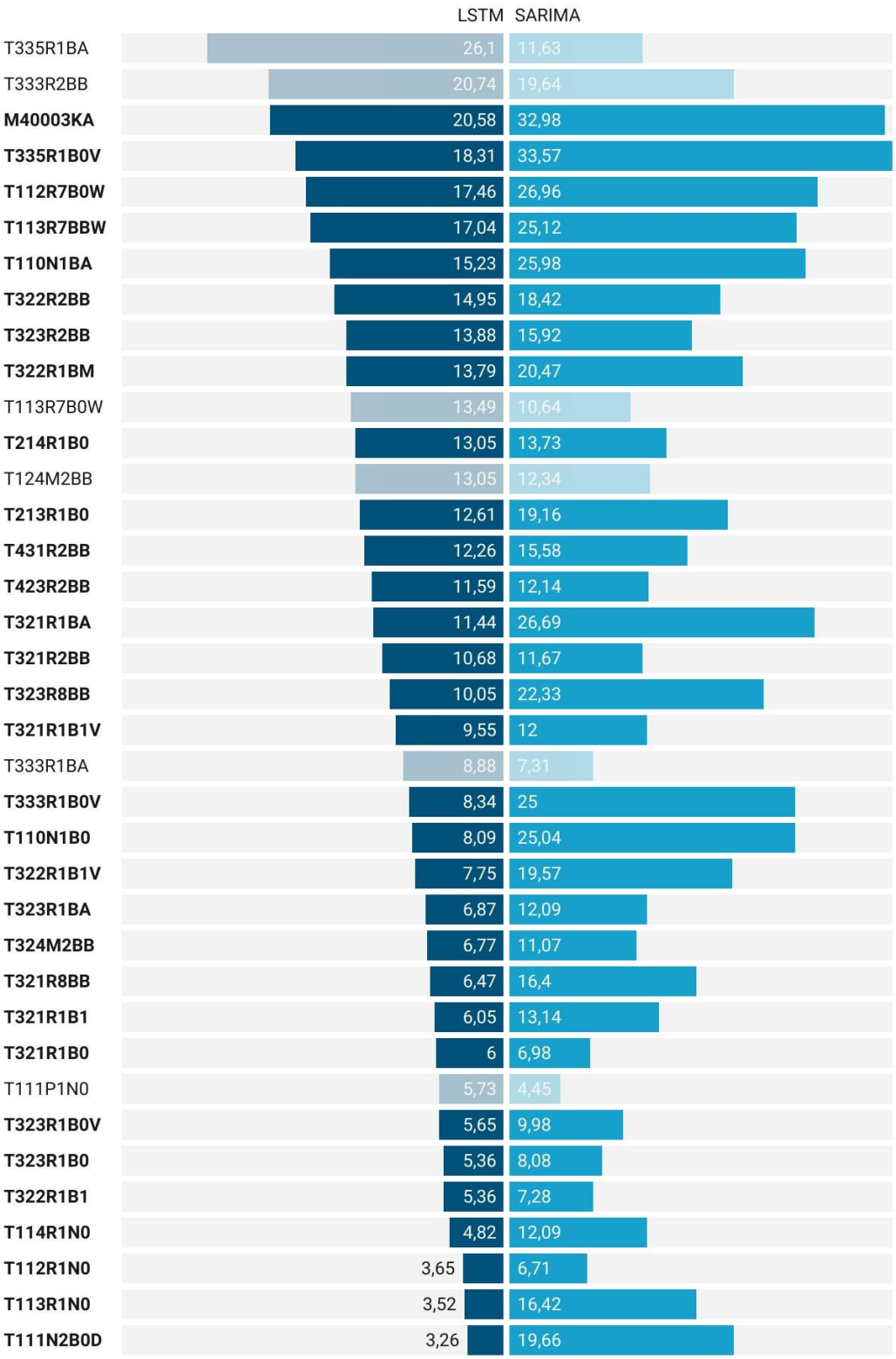


Figura 3.17 Comparación de valores de MAPE

En la Fig. 3.17 se muestra que el modelo ajustado con el método SARIMA supera al de la red LSTM en solo 6 productos. De estos, únicamente el producto T335R1BA presenta una diferencia significativa de 14.47 puntos porcentuales en el MAPE con respecto al valor obtenido por la red LSTM.

En consecuencia, el modelo ajustado con la red LSTM presenta mejores valores de MAPE en 31 productos, con diferencias significativas, como en el caso del producto T1110N1B0, que alcanza una diferencia de 16.95 puntos porcentuales.

Al analizar la información de la Fig. 3.17, podemos concluir que el modelo LSTM es más adecuado para la implementación, ya que ofrece mejores valores de precisión en la mayoría de los productos.

3.10 Creación de un Pipeline de MLOps con Amazon SageMaker

En el apartado se describen el proceso seguido para la creación de un pipeline de MLOps haciendo uso de Amazon SageMaker, lo indicado, incluye la preparación del entorno con la definición de cada paso y su ejecución.

3.10.1 Diseño del Pipeline

En la fase de diseño se establecen los pasos a seguir, los cuales, se exponen en la siguiente Fig. 3.18:

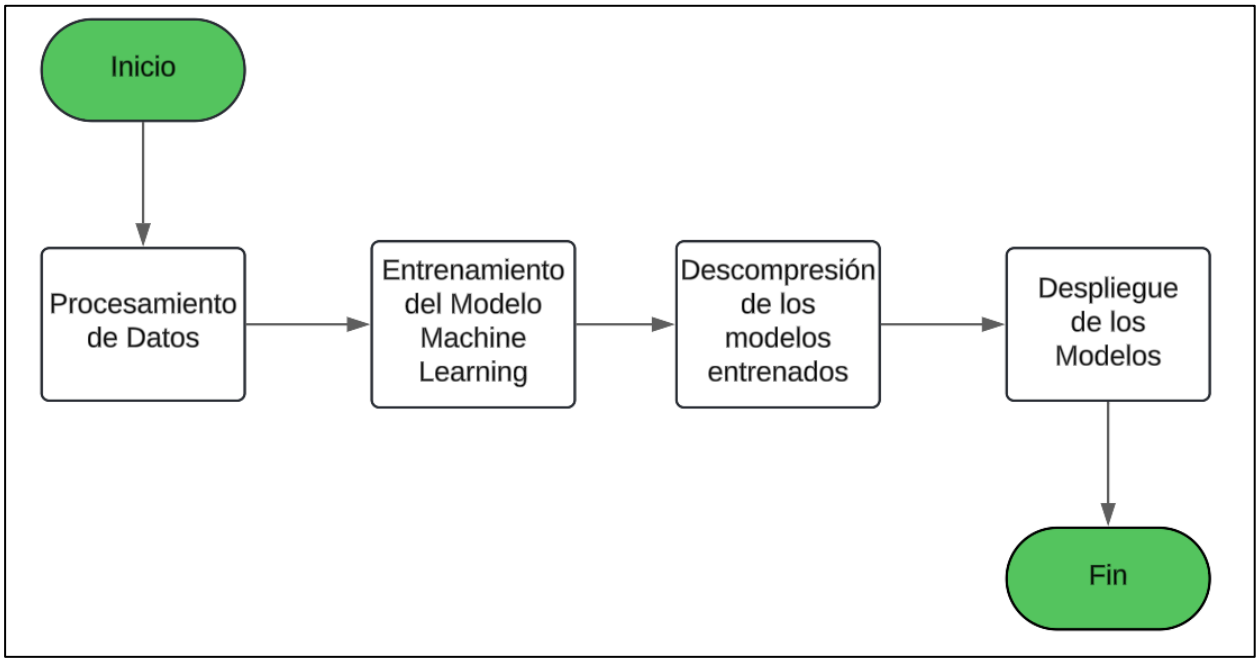


Figura 3.18 Proceso Pipeline

Este Pipeline consta de las fases expuestas en la Fig. 3.22, en donde, se tiene en primer lugar el procesamiento de datos, posterior a ello, el entramiento del modelo machine learning, seguido de la descompresión de los modelos entrenados y separado en artefactos individuales, finalizando con el despliegue de los modelos.

3.10.2 Preparación del Entorno

Se usa Docker en el proceso de creación de imágenes personalizadas, que integran librerías para ejecutar tareas en los diferentes pasos. Estas se subirán como respaldo al servicio de AWS llamado Elastic Container Registry (ECR), en ello, destaca el uso de Python, SageMaker 2.117.0, Keras 2.11.0 y Tensorflow 2.11.0 siendo las principales librerías. Así mismo se estableció la configuración del ambiente (Apéndice 5).

3.10.2.1 Creación de las imágenes personalizadas

Las imágenes personalizadas se desarrollaron utilizando Docker, tomando como referencia una imagen base disponible en el repositorio de Amazon ECR, que contiene la instalación del paquete sagemaker-scikit-learn (versión 0.23-1 para CPU). Al instalar las librerías adicionales sobre esta imagen base, se generó la imagen final utilizada en este proyecto, denominada pipeline1-training-step-v3.

Los detalles completos de los nombres de las imágenes y la construcción se encuentran en el Apéndice 5, junto con los archivos Dockerfile correspondientes.

3.10.3 Creación del paso de procesamiento de datos

Para llevar a cabo el procesamiento de los datos, se tiene que desarrollar un script el cual denominaremos "procesamiento_script.py", cuyos detalles completos se encuentran en el Apéndice 6. En la Fig. 3.19, se presentan los fragmentos más relevantes del código, destacando los elementos clave del paso de procesamiento del pipeline.

```python
script_processor = ScriptProcessor(
    command=['python3'],
    image_uri='221551376986.dkr.ecr.us-east-1.amazonaws.com/pipeline1:latest',
    role=sagemaker_role,
    instance_count=1,
    instance_type='ml.m5.xlarge',
)

processing_step = ProcessingStep(
    name="PasoDeProcesamiento",
    processor=script_processor,
    inputs=[ProcessingInput(source=input_data, destination='/opt/ml/processing/input')],
    outputs=[ProcessingOutput(output_name='train', source='/opt/ml/processing/train',
                              destination='s3://sagemaker-us-east-1-221551376986/data_procesada/')],
    code='procesamiento.py'
)
```

Figura 3.19 código fuente en Python del script de procesamiento

Como se muestra en la Fig. 3.19, el procesamiento de datos utiliza una imagen personalizada alojada en el servicio ECR, ejecutada en una instancia *ml.m5.xlarge*. Esta configuración garantiza tiempos de ejecución adecuados para el procesamiento de los datos.

3.10.4 Creación del paso de entrenamiento de los modelos

Al revisar los apartados anteriores, los registros de datos de cada producto se separaron, el encargado de este proceso es el script de entrenamiento ejecutado de la siguiente forma:

```python
estimator = Estimator(
    entry_point='entrenamiento.py',
    image_uri='221551376986.dkr.ecr.us-east-1.amazonaws.com/pipeline1-training-step-v3:latest',
    role=sagemaker_role,
    instance_count=1,
    instance_type='ml.m5.xlarge',
    output_path='s3://sagemaker-us-east-1-221551376986/modelosTF/'
)

training_input = TrainingInput(s3_data=processing_step.properties.ProcessingOutputConfig.Outputs['train'].S3Output.S3Uri)

training_step = TrainingStep(
    name="PasoDeEntrenamiento",
    estimator=estimator,
    inputs={'train': training_input}

)
```

Figura 3.20 Entrenamiento de Red

Este paso es el encargado de entrenar la red neuronal de tipo LSTM y guardar los modelos en formato SavedModel, el cual, contiene la estructura observada en la Fig. 3.20.

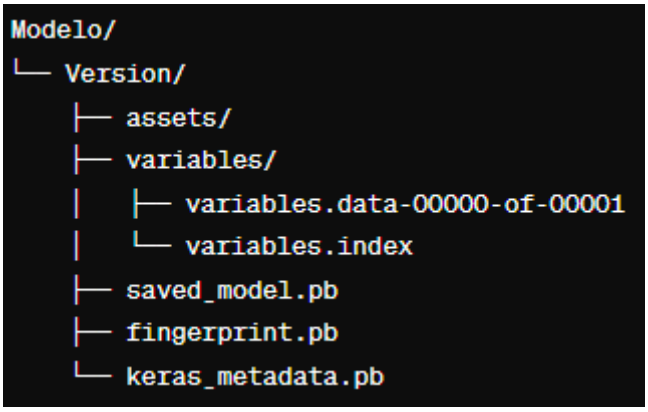


Figura 3.21 Estructura de Ficheros

Los directorios deben ser estrictamente como se muestra en la Fig. 3.21, una variación en estos generaría incompatibilidad con las versiones de las librerías usadas para el posterior despliegue. La variable de entrada de este paso es la URI de un directorio en S3, en el cual se guardan artefactos generados en el paso de procesamiento, siendo, la variable de salida los datos procesados. Al igual que en el paso anterior, se usa una instancia *ml.m5.xlarge* y una de las imágenes personalizadas creadas para ejecutar este paso.

3.10.5 Creación del paso de descompresión

Al finalizar la fase de entrenamiento, se genera un artefacto en formato comprimido denominado "model.tar.gz", que contiene todos los detalles de los modelos entrenados. Sin embargo, para cumplir con el objetivo de crear un endpoint único para cada modelo, es necesario descomprimir este artefacto y asignar una nueva identificación a cada modelo. Este proceso se lleva a cabo mediante un script de descompresión desarrollado específicamente para esta tarea, el cual se ejecuta en el tercer paso del pipeline.

```python
script_decompress = ScriptProcessor(
    command=['python3'],
    image_uri='763104351884.dkr.ecr.us-east-1.amazonaws.com/tensorflow-inference:2.3.0-cpu-py37-ubuntu18.04',
    role=sagemaker_role,
    instance_count=1,
    instance_type='ml.m5.xlarge',
)

decompress_step = ProcessingStep(
    name="PasoDeDescompresion",
    processor=script_decompress,
    inputs=[
        ProcessingInput(
            source=training_step.properties.ModelArtifacts.S3ModelArtifacts,
            destination='/opt/ml/processing/model'
        )
    ],
    outputs=[
        ProcessingOutput(
            output_name='models',
            source='/opt/ml/processing/extracted_models',
            destination='s3://sagemaker-us-east-1-221551376986/extracted_modelsTF/'
        )
    ],
    code='descompresion.py'
)
```

Figura 3.22 Script de Descompresión

Es importante destacar que, como se observa en la Fig. 3.22, no fue necesario utilizar una imagen personalizada para esta fase. Las imágenes estándar disponibles en ECR cumplieron con los requisitos necesarios. Como resultado de esta fase, se genera un directorio en S3 que contiene los modelos individuales en formato comprimido.

3.10.6 Creación del paso de despliegue

En esta fase, se procede con el despliegue de los endpoints individuales de los modelos generados en la etapa anterior. Para ello, se desarrolla un script llamado despliegue.py, disponible en el apéndice 9, encargado de configurar los endpoints y registrar los modelos según las especificaciones definidas. Este proceso es esencial para desplegar los modelos en SageMaker de manera efectiva. En la Fig. 3.23, se observa el

código utilizado para implementar el cuarto paso del pipeline, que ejecuta el despliegue de los modelos.

```
127  script_processor_deploy = ScriptProcessor(
128      command=["python3"],
129      image_uri='221551376986.dkr.ecr.us-east-1.amazonaws.com/deploy-models-v1:latest',
130      role=sagemaker_role,
131      instance_count=1,
132      instance_type="ml.m5.large"
133  )
134
135  deploying_step = ProcessingStep(
136      name="PasoDeDespiegue",
137      processor=script_processor_deploy,
138      inputs=[
139          ProcessingInput(
140              source=decompress_step.properties.ProcessingOutputConfig.Outputs['models'].S3Output.S3Uri,
141              destination='/opt/ml/processing/model'
142          )
143      ],
144      outputs=[],
145      job_arguments=[
146          "--bucket-name", bucket_name_param,
147          "--s3-model-directory", s3_model_directory_param,
148          "--role", role_param,
149          "--framework-version", framework_version_param,
150          "--region", region_param
151      ],
152      code="despliegue.py"
153  )
```

Figura 3.23 Código de Despliegue

En este caso se ejecuta de forma similar a las fases anteriores, no obstante, se enfatiza que se hizo uso de una imagen personalizada creada específicamente para este proceso, y también, un punto importante por resaltar está en la línea 31 de la Fig. 3.24.

```
22      # Iterar sobre cada archivo de modelo y desplegarlo
23      for model_file in model_files:
24          model_url = f's3://{bucket_name}/{model_file}'
25          tensorflow_model = TensorFlowModel(model_data=model_url,
26                                             role=role,
27                                             framework_version=framework_version,
28                                             sagemaker_session=sagemaker_session)
29
30          # Configurar la inferencia serverless
31          serverless_config = ServerlessInferenceConfig(memory_size_in_mb=1024, max_concurrency=1)
32
33          # Desplegar el modelo como endpoint serverless
34          predictor = tensorflow_model.deploy(serverless_inference_config=serverless_config)
35
36          print(f"Modelo serverless desplegado desde {model_url}")
37
```

Figura 3.24 Configuración tipo serverless

Podemos observar que la configuración es de tipo *"serverless"*, esto debido a que AWS limita la cantidad de instancias a 8 y expandir esta capacidad no sería viable en términos monetarios.

Al completar este paso, se puede verificar en la sección de Pipelines de SageMaker Studio, como se muestra en la Fig. 3.25, que todos los pasos se han ejecutado con éxito.

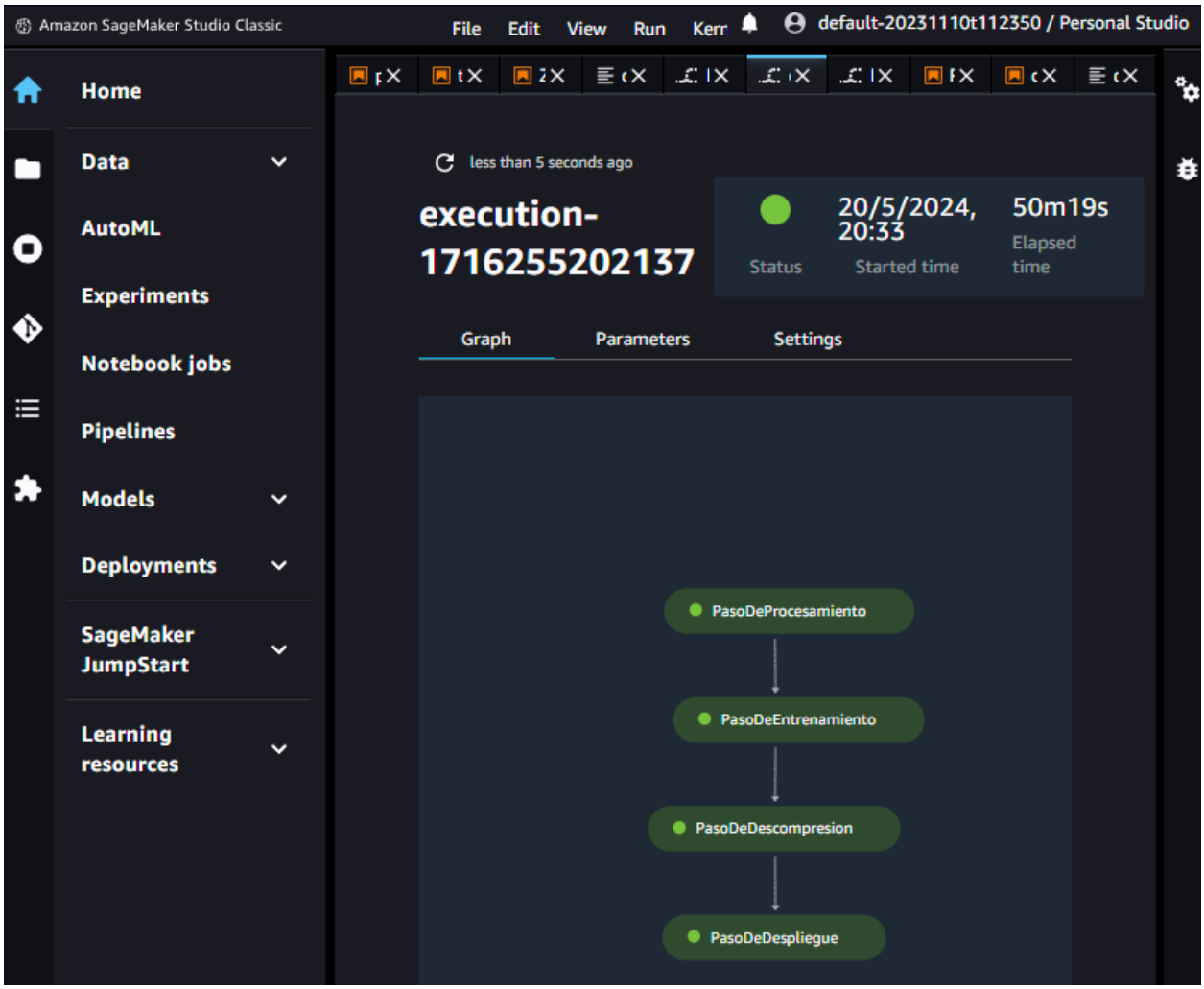


Figura 3.25 Verificación de Procesos Completados

En la Fig. 3.25 se observa los cuatro pasos que constituyen el pipeline de MLOps. Aunque la visualización en SageMaker Studio parece sencilla, el pipeline involucra procesos más complejos y detallados, como la creación de imágenes personalizadas y

su carga en el servicio ECR. Estos pasos son fundamentales para ejecutar los scripts necesarios y transformar los datos en modelos listos para realizar inferencias.

CAPÍTULO 4

4. ANÁLISIS DE RESULTADOS

Este proyecto destaca la importancia de contar con una variedad de herramientas y enfoques para abordar la predicción de series de tiempo, ya que diferentes productos pueden requerir diferentes modelos, la identificación de valores atípicos es esencial para asegurar la calidad de los datos antes de aplicar los modelos. Además, el uso de modelos SARIMA y redes neuronales LSTM resulta eficaz para manejar la estacionalidad y complejidad de las series temporales.

En primer lugar, se procedió a la evaluación de las dos opciones disponibles para el tratamiento de datos atípicos: la aplicación de la técnica Z-Score y el uso del Rango Intercuartil (IQR), aunque la prueba de normalidad mostró un valor p igual a 0.35, lo que sugiere la posibilidad de una distribución normal del conjunto de datos y, consecuentemente, la posibilidad de aplicar cualquiera de los dos métodos, se tomó la decisión de emplear el método IQR. Esta elección se basó en la robustez demostrada por el método IQR en situaciones con valores atípicos extremadamente elevados.

Se consideró crucial este enfoque debido a la amplia disparidad entre la media de los datos, que se sitúa en 39.93 toneladas por hora (t/h), y el valor máximo alcanzado de 1756.67 t/h. Esta disparidad podría tener un impacto significativo en la aplicación del método Z-Score, ya que este último puede ser influenciado de manera desproporcionada por valores atípicos extremadamente elevados.

En ello, se disponía de un amplio conjunto de 234 productos, cada uno con sus respectivas observaciones en el registro de datos bajo estudio. Siguiendo el principio conocido como la regla de Pareto 80-20, se procedió a la selección de un subconjunto de productos que abarcara el 80% de las observaciones disponibles. Esta elección se basó en la premisa de que los productos restantes podrían tener datos faltantes o poco representativos, lo que haría inapropiado el uso de los mismos métodos de análisis aplicados al conjunto principal.

Después de realizar este proceso de selección, se identificaron y eligieron los productos que conformaban el 80% de las observaciones totales. Como resultado, se encontró que este subconjunto constaba de 46 productos. Esto representaba una proporción significativa de las observaciones disponibles y se consideraba representativo

de la mayoría de los productos en términos de sus datos de velocidad de producción. De esta manera, se logró centrar el análisis en un conjunto de productos que proporcionaba una visión sólida y representativa de la producción, evitando la inclusión de productos con datos insuficientes o poco confiables.

Se llevó a cabo la prueba de Ljung-Box, en la cual se planteó una hipótesis nula (H_0) que afirmaba que la serie de datos se comportaba como ruido blanco. Los resultados arrojaron un valor de p extremadamente bajo, calculado en 1.2280e-49. Este valor de p es inferior a 0.05 y está muy próximo a cero, lo cual implica un rechazo definitivo de la hipótesis nula. En otras palabras, la evidencia estadística sustenta que la serie de datos no se comporta como ruido blanco. Más bien, esta serie exhibe autocorrelación significativa, lo que sugiere relaciones temporales entre las observaciones.

Estos resultados respaldan con firmeza la presunción inicial de que la serie de datos no es independiente y que las observaciones están interconectadas en el tiempo. La baja probabilidad de obtener estos resultados si la serie fuera ruido blanco es una señal clara de la existencia de autocorrelación en los datos. Por lo tanto, podemos concluir que la serie de datos muestra dependencias temporales, lo que es un hallazgo fundamental para el análisis subsiguiente y el desarrollo de modelos de pronóstico.

Un grupo de nueve productos dentro del conjunto de datos exhibe patrones de series temporales que se asemejan a un comportamiento de ruido blanco. A causa de esta particularidad, no resulta viable aplicar métodos de modelado de series temporales con fines de pronóstico en estos productos. Por lo tanto, se ha decidido excluirlos del proceso de análisis subsiguiente. El enfoque se centra exclusivamente en el conjunto restante de 37 productos que presentan patrones más adecuados para el modelado y la generación de pronósticos.

Este proceso de selección se basa en la observación de que estos nueve productos no siguen patrones temporales discernibles y, en cambio, presentan fluctuaciones aleatorias que dificultan la aplicación de técnicas de pronóstico precisas. Al excluirlos del análisis, se simplifica el estudio y se concentra en los productos que muestran patrones de series temporales más adecuados para el modelado y la predicción. Esto permitirá un enfoque más efectivo en la generación de pronósticos precisos y útiles para la velocidad de producción de estos productos, lo que a su vez contribuirá al logro de los objetivos del estudio.

Para evaluar el desempeño de los modelos implementados, se utilizó la métrica Mean Absolute Percentage Error (MAPE), la cual permite medir el error porcentual promedio entre las predicciones y los valores reales, teniendo en cuenta que un MAPE menor indica un mejor modelo. Los resultados obtenidos se resumen en la Tabla 4.1:

Tabla 4.1 Mejor modelo del producto

Producto	MAPE LSTM	MAPE SARIMA	Mejor Modelo
T335R1BA	26,1	11,63	SARIMA
T333R2BB	20,74	19,64	SARIMA
M40003KA	20,58	32,98	LSTM
T335R1B0V	18,31	33,57	LSTM
T112R7B0W	17,46	26,96	LSTM
T113R7BBW	17,04	25,12	LSTM
T110N1BA	15,23	25,98	LSTM
T322R2BB	14,95	18,42	LSTM
T323R2BB	13,88	15,92	LSTM
T322R1BM	13,79	20,47	LSTM
T113R7B0W	13,49	10,64	SARIMA
T214R1B0	13,05	13,73	LSTM
T124M2BB	13,05	12,34	SARIMA
T213R1B0	12,61	19,16	LSTM
T431R2BB	12,26	15,58	LSTM
T423R2BB	11,59	12,14	LSTM
T321R1BA	11,44	26,69	LSTM
T321R2BB	10,68	11,67	LSTM
T323R8BB	10,05	22,33	LSTM
T321R1B1V	9,55	12	LSTM
T333R1BA	8,88	7,31	SARIMA
T333R1B0V	8,34	25	LSTM
T110N1B0	8,09	25,04	LSTM
T322R1B1V	7,75	19,57	LSTM
T323R1BA	6,87	12,09	LSTM
T324M2BB	6,77	11,07	LSTM
T321R8BB	6,47	16,4	LSTM
T321R1B1	6,05	13,14	LSTM
T321R1B0	6	6,98	LSTM
T111P1N0	5,73	4,45	SARIMA

T323R1B0V	5,65	9,98	LSTM
T323R1B0	5,36	8,08	LSTM
T322R1B1	5,36	7,28	LSTM
T114R1N0	4,82	12,09	LSTM
T112R1N0	3,65	6,71	LSTM
T113R1N0	3,52	16,42	LSTM
T111N2B0D	3,26	19,66	LSTM

La cuantificación MAPE emerge como un indicador crítico para la evaluación comparativa de los modelos implementados. Los resultados obtenidos señalan una eficacia superior de las redes LSTM en relación con el modelo SARIMA, ya que los valores de MAPE del modelado LSTM fueron mejores en 31 de los 37 productos analizados, es decir que los modelos LSTM son mejor en casi el 84% de los productos. Sin embargo, no se deben obviar las potencialidades del enfoque SARIMA, dado que la integración de ambas metodologías podría facilitar pronósticos más adecuados en aquellos casos donde SARIMA muestra un mejor indicador.

Las redes LSTM son capaces de capturar relaciones no lineales complejas, lo que les permitió adaptarse mejor al escenario dinámico y multifactorial de la producción de alimento balanceado. Esto se debe a que múltiples componentes afectan la planificación y la velocidad de la máquina mezcladora. En contraste, SARIMA, al ser un modelo lineal, está limitado por su naturaleza autorregresiva y depende de una configuración adecuada de sus hiperparámetros. Esta limitación dificulta que SARIMA se ajuste de manera efectiva a un entorno tan caótico y variable como el de la producción.

CAPÍTULO 5

5. CONCLUSIONES Y RECOMENDACIONES

5.1. Conclusiones

- A lo largo de este proyecto enfocado en el análisis de series temporales dentro de la industria de producción de alimentos balanceados para animales, se realizaron diversas etapas críticas para comprender y pronosticar las velocidades de producción de diferentes productos, siendo que, en primer lugar, se identificaron 46 productos que fueron objeto de análisis, de los cuales 9 productos mostraron ser series de ruido blanco y se excluyeron del estudio considerando las alteraciones de estos en los resultados y su fiabilidad.
- La aplicación de varios métodos de modelado, incluyendo el modelo SARIMA y redes neuronales LSTM, para realizar pronósticos sobre las velocidades de producción de los 37 productos restantes, como resultado, se obtuvieron valores de error MAPE que variaron ampliamente, oscilando desde 4.45% a 33.57% para SARIMA y 3.26% a 26.1% para las redes LSTM. Por ejemplo, el producto T111P1N0 mostró un MAPE del 4.45%, lo que indicó una buena precisión en el modelo, mientras que otro producto tuvo un MAPE del 33.57%, lo que sugirió un rendimiento menos preciso.
- Este proyecto verificó la capacidad de implementar y comparar diferentes modelos de pronóstico para estimar la velocidad de procesamiento de la mezcladora, puesto que los resultados resaltan la relevancia de elegir el modelo adecuado en función de las características de los datos, en donde la implementación proporcionó una base sólida para futuros desarrollos y despliegues de modelos en la industria de producción de alimentos balanceados, en donde, los MAPE variaron entre productos, oscilando entre 3.26% y 26.1%, en el método seleccionado, lo que resalta la necesidad de un enfoque personalizado en el análisis de series temporales.
- La identificación y gestión de datos atípicos son aspectos críticos en el análisis de datos, considerando que estos pueden influir significativamente en los resultados de un análisis estadístico o de series temporales, de tal manera que, al no identificarlos de forma preventiva se incrementará la incertidumbre, por ende, es importante abordarlos adecuadamente para evitar sesgos en las conclusiones.

- Contar con una adecuada gestión de los datos es esencial en proyectos que involucran la manipulación y análisis de datos.
- El uso de estadísticas robustas y la selección de técnicas apropiadas, como el rango intercuartil (IQR) o el puntaje Z, para detectar y tratar datos atípicos, favorecen la gestión de datos atípicos, y esto puede mejorar la precisión y la validez de los resultados, lo que es de alta importancia en la toma de decisiones basada en datos.
- Las variables que actualmente reportan los sistemas de monitoreo resultan insuficientes para implementar adecuadamente un modelo de ML basado en árboles de decisión.
- Es indispensable aplicar métodos de eliminación de datos atípicos ya que los sistemas de producción reportan datos erróneos producto de fallas en los instrumentos de medición.
- En este proyecto, el método SARIMA demostró ser altamente eficaz para modelar y pronosticar datos con una fuerte estacionalidad, lo que permitió obtener estimaciones adecuadas para la velocidad de la mezcladora en el proceso de producción de alimentos balanceado.
- Las redes neuronales de memoria a largo-corto plazo (LSTM, por sus siglas en inglés), se destacaron como una poderosa herramienta para el pronóstico de la velocidad de la mezcladora. La capacidad de estas redes para captar dependencias a largo plazo en series temporales resultó ventajosa, y, aunque algunos modelos mostraron MAPE más elevados que otros, se posicionaron como una alternativa viable para estimar la velocidad de producción.
- La combinación de métodos tradicionales como SARIMA y enfoques de vanguardia como las redes neuronales LSTM proporcionó un enfoque integral para la predicción de la velocidad de la mezcladora en la planta de procesamiento de alimentos balanceados. Estos resultados respaldan la importancia de seleccionar y comparar diferentes enfoques de modelado para lograr la precisión deseada en la estimación de series temporales en un entorno de producción.

5.2. Recomendaciones

- Se recomienda continuar fortaleciendo la gobernanza de datos en la empresa para evitar futuras alteraciones no deseadas de los datos, lo que podría afectar la calidad y confiabilidad de los resultados del análisis.
- Es recomendable realizar una supervisión regular de la calidad de los datos y establecer procesos efectivos de detección y corrección de valores atípicos, especialmente en aquellas series de tiempo que presenten observaciones anómalas.
- Se recomienda considerar la implementación de una estrategia de modelado mixto que combine enfoques tradicionales como SARIMA con técnicas de aprendizaje profundo, como las redes neuronales LSTM, dado que, esto permitirá aprovechar las fortalezas de cada método y mejorar la precisión de las predicciones.
- Es recomendable hacer la detección y corrección automática de datos atípicos, esto a fin de tener los datos disponibles a la brevedad posible.
- Se recomienda continuar con el análisis de procesos que pueden ser automatizados a fin de cada vez depender menos del criterio experto y tener procesos estadísticos definidos.
- Se recomienda investigar el potencial de las herramientas para automatizar y estandarizar el desarrollo, implementación y mantenimiento de modelos de aprendizaje automático, lo cual puede acelerar el ciclo de vida de los proyectos de análisis de series temporales.
- Es recomendable tener en cuenta el impacto de la estacionalidad en la producción y demanda de productos, ajustando las estrategias operativas en función de ello para asegurar una gestión eficiente de los recursos y una producción óptima.
- Se recomienda analizar las tendencias de las velocidades de los diferentes productos, ya que acorde con los gráficos del Apéndice 2, se ve fluctuaciones que no denotan una clara tendencia, pero se podrían encontrar bajo qué parámetros se logró los valores más altos a fin de replicarlos y poder optimizar el tiempo de la máquina mezcladora.
- Es recomendable llevar a cabo un análisis exhaustivo de las tendencias asociadas con las velocidades de procesamiento de los diversos productos en estudio. Tal como se evidencia en los gráficos incluidos en el Apéndice 2, las fluctuaciones observadas

no presentan una tendencia discernible. Sin embargo, un estudio detallado podría permitir la identificación de las condiciones bajo las cuales se alcanzan valores de velocidad óptimos, con el objetivo ulterior de replicar dichas condiciones para maximizar la eficiencia de la máquina mezcladora.

6 REFERENCIAS

Observatorio Social del Ecuador, OSE. (2018). *Situación de la niñez y adolescencia en el Ecuador, una mirada a través de los ODS. .* Quito: UNICEF Ecuador.

Abella-Miravet, B. (2021). *Mejora de las predicciones en muestras desbalanceadas.* España: Universidad Autónoma de Madrid.

Alonso, J. (2008). *TUTORIAL PARA LA ESTIMACIÓN DE UN MODELO CON PRESENCIA DE AUTOCORRELACIÓN EN EASYREG .* Colombia: ICESI.

Barahama, A., & Wardani, R. (2021). Utilization Extract, Transform, Load For Developing Data Warehouse In Education Using Pentaho Data Integration. *Journal of Physics: Conference Series*. doi:doi:10.1088/1742-6596/2111/1/012030

Chávez-Quisbert, N. (1997). MODELOS ARIMA. *Revista Ciencia y Cultura*(1), 23-30.

Frąckiewicz, M. (7 de julio de 2023). *Explorando el poder de las redes de memoria a corto plazo (LSTM) en aplicaciones de IA*. Obtenido de TS2: https://ts2.space/es/ia-y-redes-de-memoria-a-largo-plazo-lstm/

García, A. (15 de agosto de 2019). *El indicador Z-Score*. Obtenido de Quant Space: https://quantspace.es/2019/08/15/el-indicador-z-score/

Gimenes, M. (20 de julio de 2020). *Amazon Web Services (AWS): ¿qué es y qué ofrece?* Obtenido de Hiberus Blog: https://www.hiberus.com/crecemos-contigo/amazon-web-services-aws-que-es-y-que-ofrece/

González-Casimiro, M. (2008). *Análisis de series temporales: Modelos ARIMA.* España: Universidad del País Vasco (UPV-EHU).

Heizer, J., & Render, B. (2022). *Operations Management: Sustainability and Supply Chain Management* (Décimo cuarta ed.). Washington : Pearson.

Hernández, S. (2015). *Análisis de Series de Tiempo.* México: CEPAL.

Hernández-Sampieri, R., Hernandéz-Collado, C., & Baptista-Lucio, P. (2014). *Metodología de la Investigación* (6ta ed.). México: McGraw-Hill.

Lambert, D., Stock, J., & Ellram, L. (1998). *Fundamentals of logistics management.* Bóston: McGraw-Hill.

Llisterri, J., Angelelli, P., Painter, F., Chrisney, M., Nieder, F., Mico, A., & Wilson, S. (2002). *Guía Operativa para programas de Competitividad para la pequeña y mediana empresa.* Washintong. Obtenido de https://docplayer.es/10840117-Guia-

operativa-para-programas-de-competitividad-para-la-pequena-y-mediana-empresa.html

Llosa, A. (5 de marzo de 2020). *Qué es DataOps y MLOps?* Obtenido de Medium: https://anllogui.medium.com/qu%C3%A9-es-dataops-y-mlops-e07ef8281416

Maisueche-Cuadrado, A. (2019). *Utilización del Machine-Learning en la Industria 4.0.* Valladolid: Universidad de Valladolid.

Mentzer, J., Stank, T., & Esper, T. (2011). SUPPLY CHAIN MANAGEMENT AND ITS RELATIONSHIP TO LOGISTICS, MARKETING, PRODUCTION, AND OPERATIONS MANAGEMENT. *Journal of Business Logistics, 29*(1), 31-46. doi:https://doi.org/10.1002/j.2158-1592.2008.tb00067.x

Miranda-Chinlli, C. (2021). *Modelización de Series Temporales modelos clásicos y SARIMA.* España: Universidad de Granada.

Montes-Gallardo, A. (2020). *Control Predictivo basado en Machine Learning de una planta de laboratorio.* Sevilla: Universidad de Sevilla.

Montoya, A., Montoya, I., & Castellanos, O. (2010). Situación de la competitividad de las Pyme en Colombia: elementos actuales y retos. *Agronomía Colombiana, 28*(1). Obtenido de https://revistas.unal.edu.co/index.php/agrocol/article/view/17600

Oliva, B. (2019). *Análisis de Series de Tiempo.* México: UNAM.

Pérez-Zaldivar, L. (2022). *Algoritmos de Aprendizaje Autómatico para la predicción de la productividad de la confección textil.* España: Universidad Internacional de Andalucía.

Rodriguez-Ojeda, L. (2017). *Probabilidad y Estadística Básica para Ingenieros.* Guayaquil: Escuela Superior Politécnica del Litoral.

Stank, T., Mentzer, J., & Esper, T. (2008). Supply Chain Management and its Relationship to Logistics, Marketing, Production, and Operations Management. *Journal of Business Logistics, 29*(1), 31-46. doi:10.1002/j.2158-1592.2008.tb00067.x

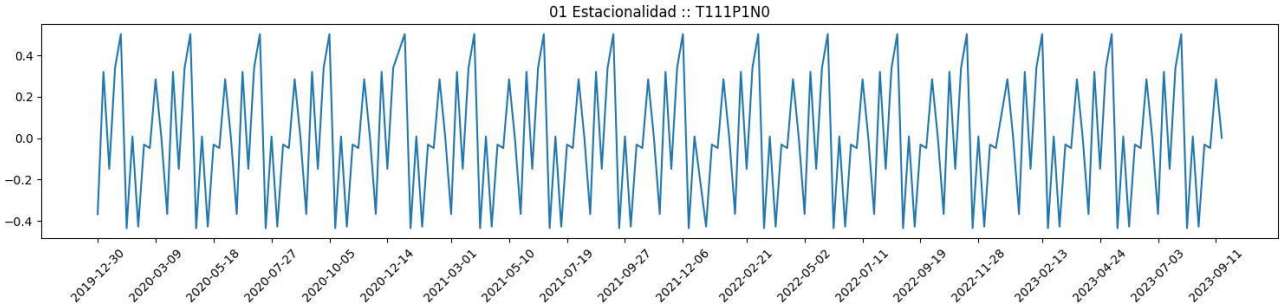
Thompson, A., Peteraf, M., Gamble, J., & Strickland, A. (2022). *Crafting & Executing Strategy: The Quest for Competitive Advantage: Concepts and Cases* (23 ed.). Washintong: McGrawHill.

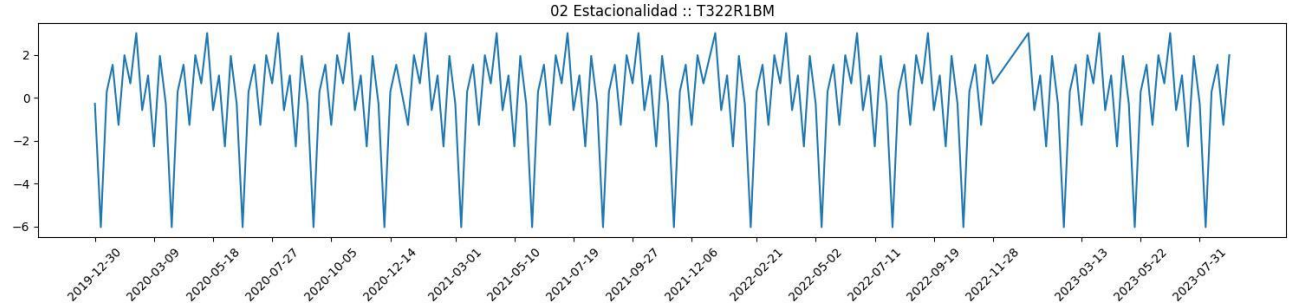
Yepes-Piqueras, V. (2022). *¿Qué hacemos con los valores atípicos (outliers)?* España: Universidad Politénica de Valencia.

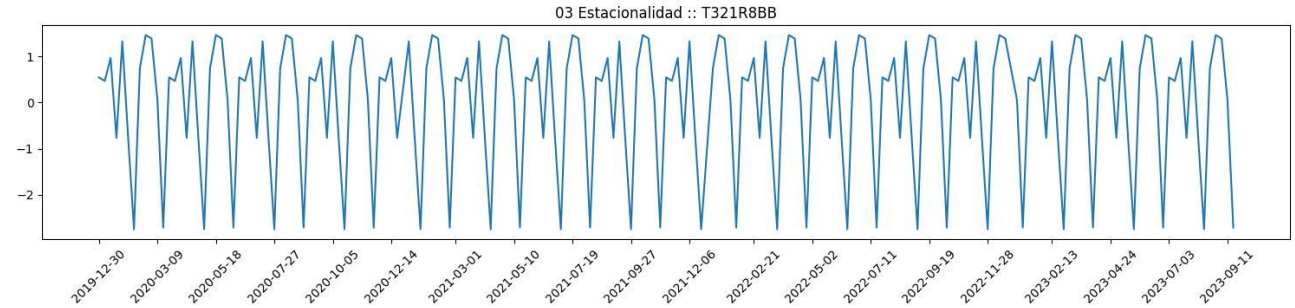
Yuque-Mendoza, E. (2019). *Diseño de una máquina mezcladora de alimento balanceado para pequeñas granjas ganaderas .* Lima: Universidad Nacional Mayor "San Marcos".

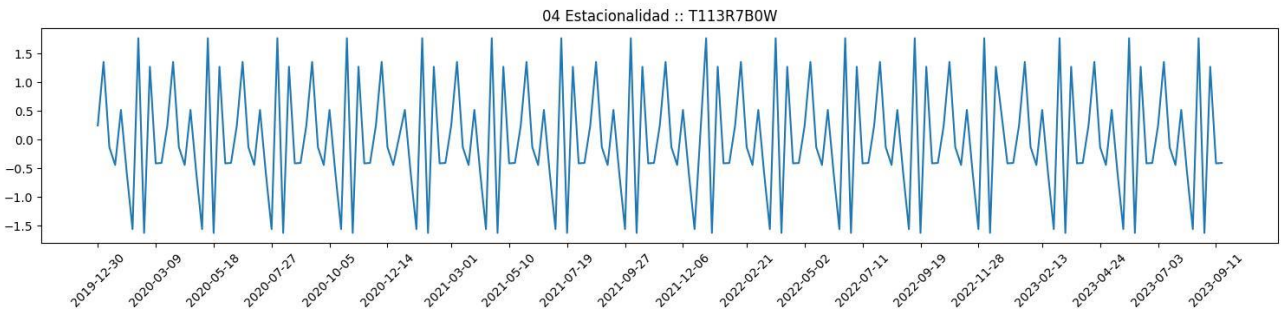
7 APÉNDICES

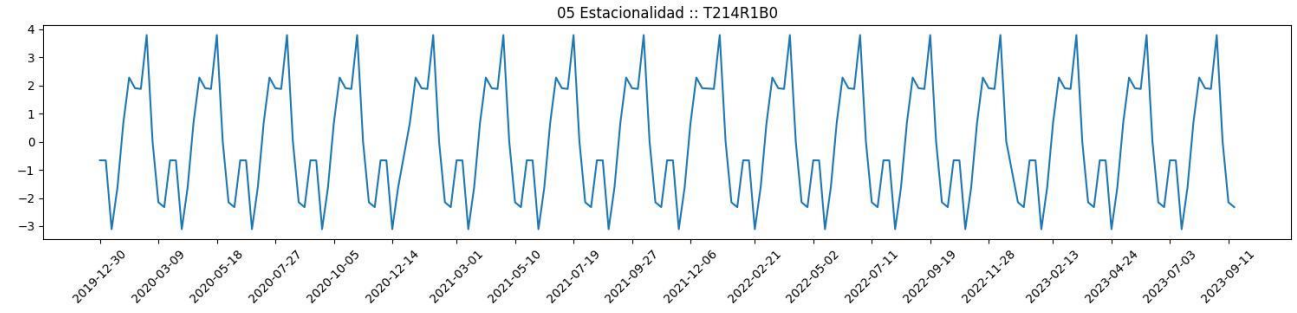
Apéndice 1 Descomposición Serie Temporal

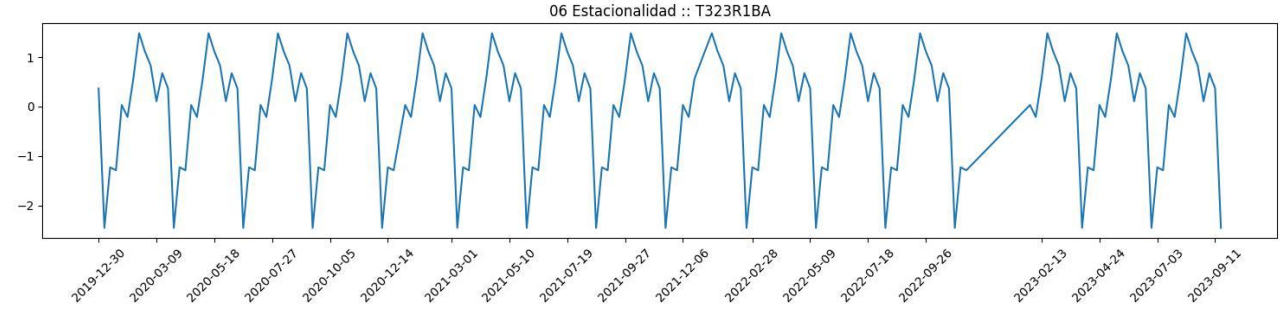


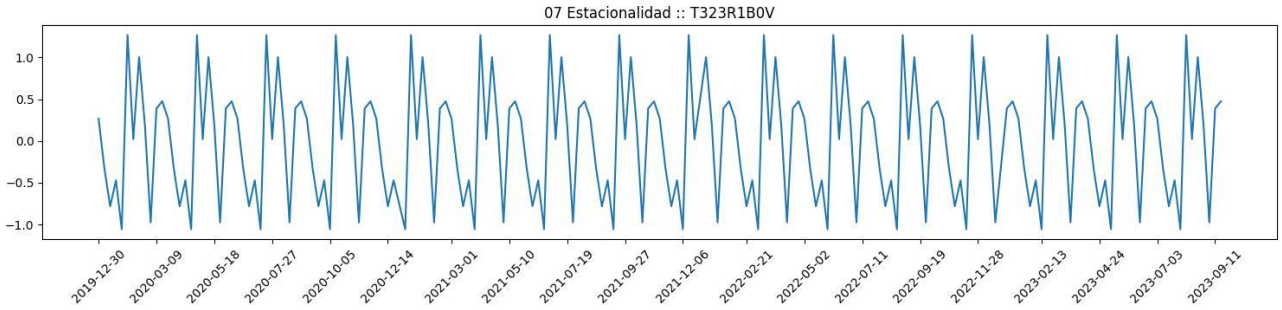


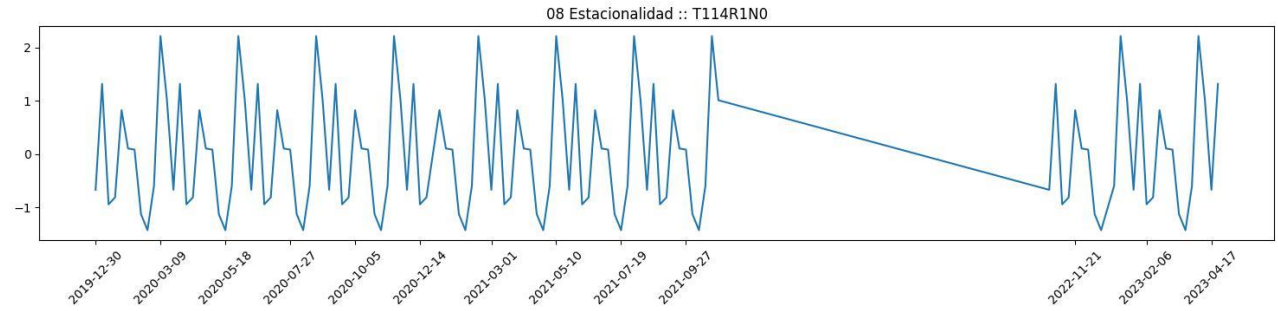


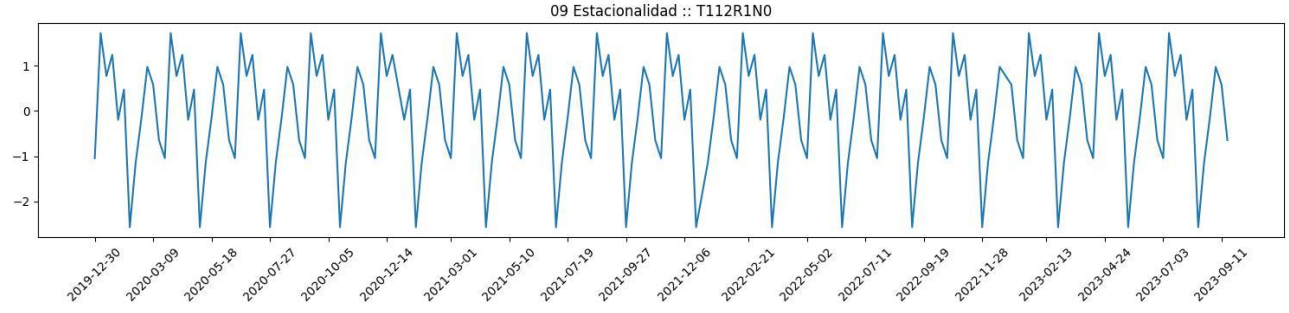


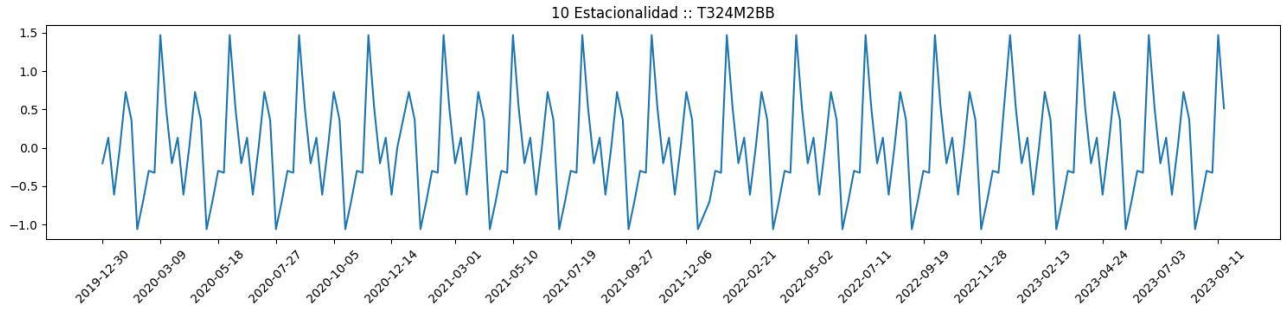

05 Estacionalidad :: T214R1B0

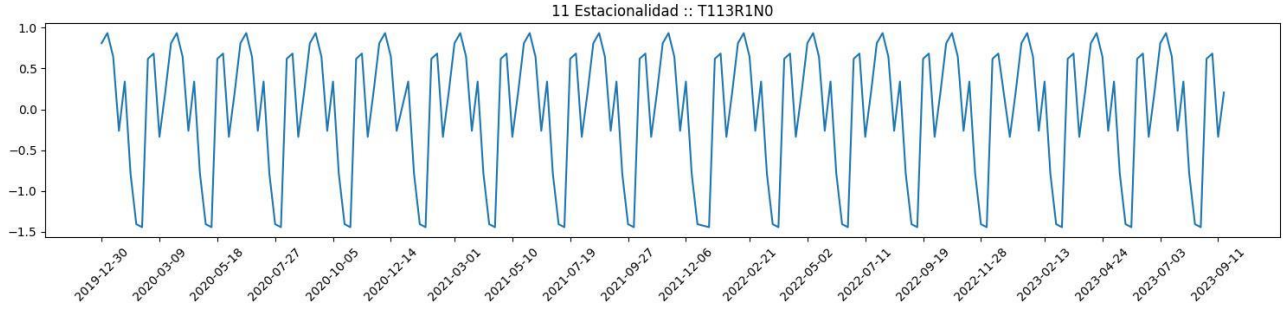

06 Estacionalidad :: T323R1BA

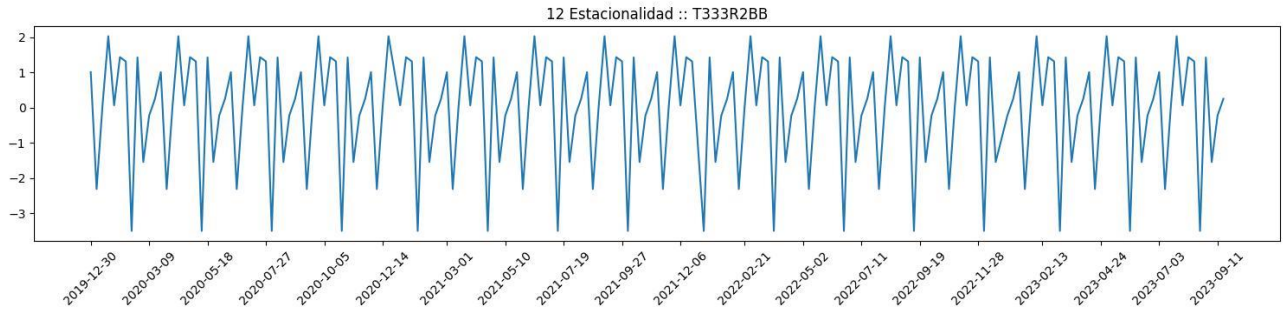

07 Estacionalidad :: T323R1B0V

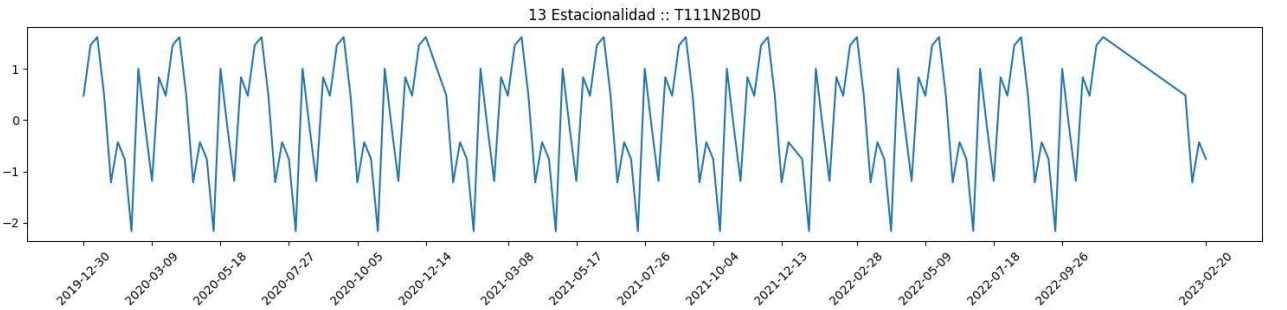

08 Estacionalidad :: T114R1N0

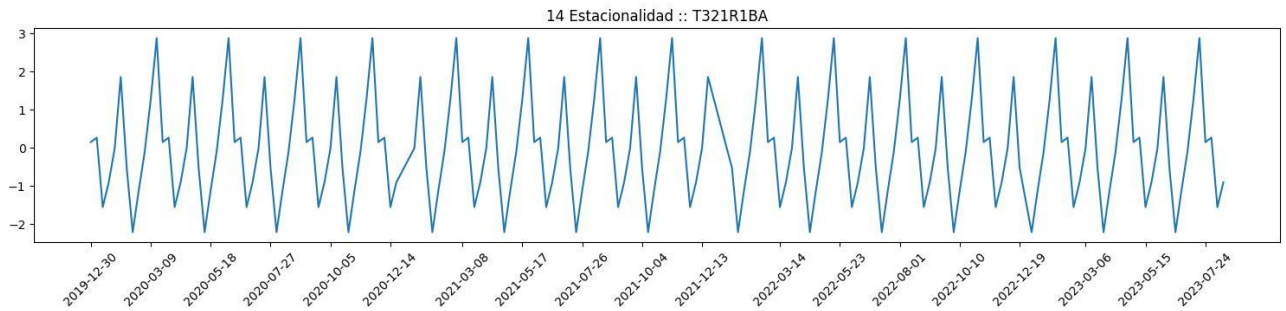

09 Estacionalidad :: T112R1N0

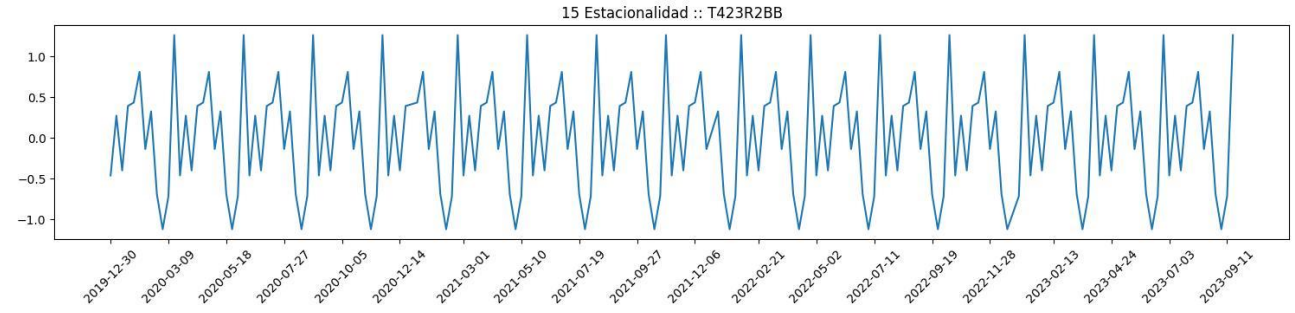

10 Estacionalidad :: T324M2BB

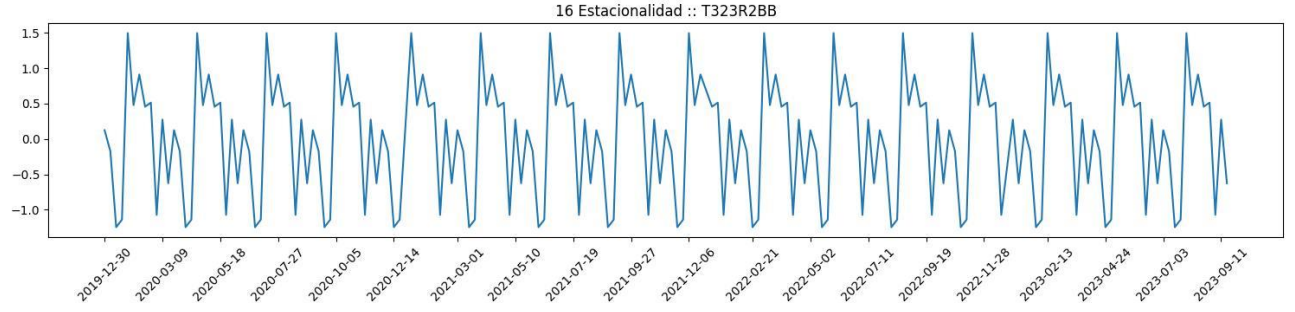

11 Estacionalidad :: T113R1N0

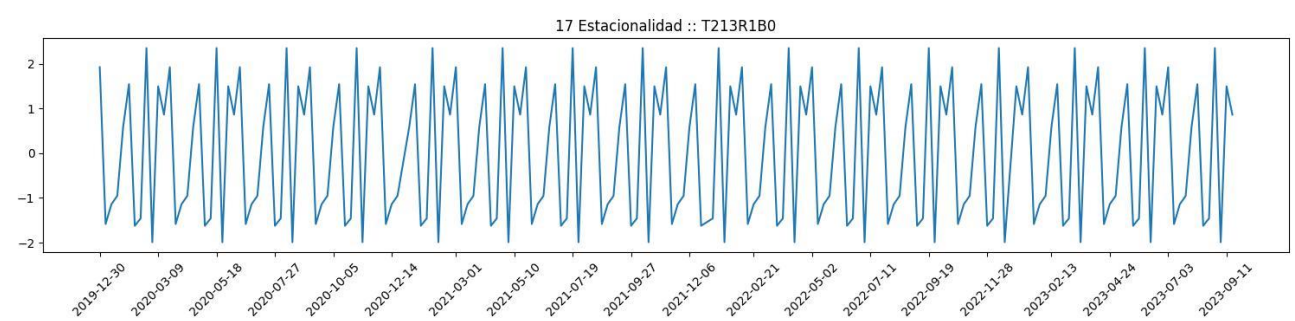

12 Estacionalidad :: T333R2BB

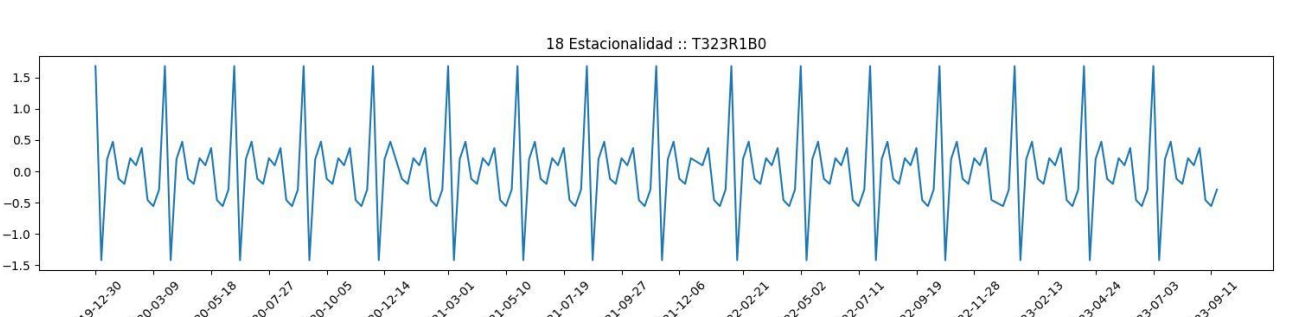

13 Estacionalidad :: T111N2B0D

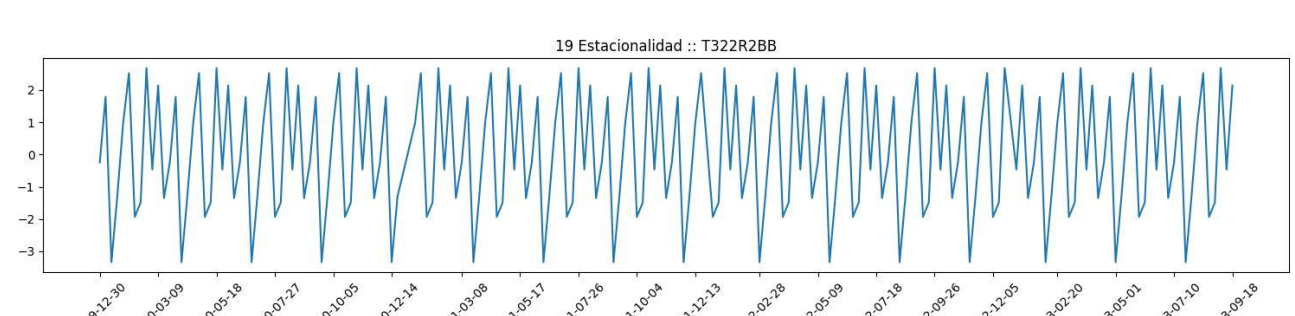

14 Estacionalidad :: T321R1BA

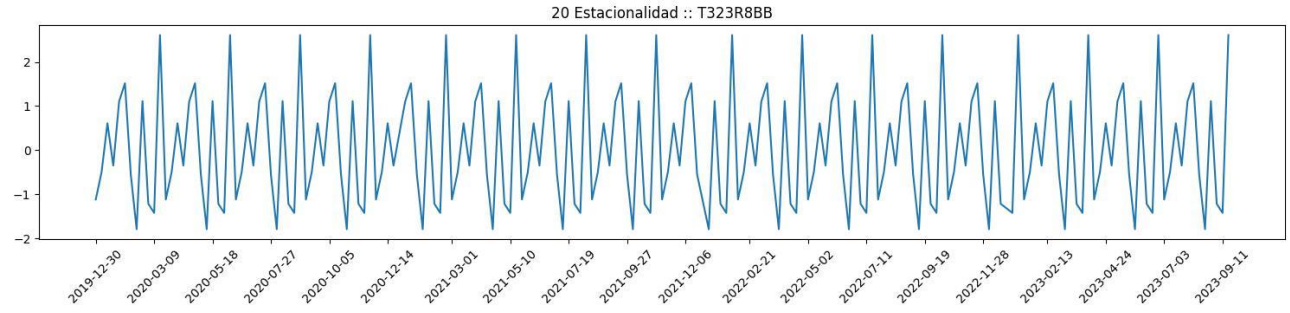

15 Estacionalidad :: T423R2BB
1.0
0.5
0.0
−0.5
−1.0
2019-12-30
2020-03-09
2020-05-18
2020-07-27
2020-10-05
2020-12-14
2021-03-01
2021-05-10
2021-07-19
2021-09-27
2021-12-06
2022-02-21
2022-05-02
2022-07-11
2022-09-19
2022-11-28
2023-02-13
2023-04-24
2023-07-03
2023-09-11

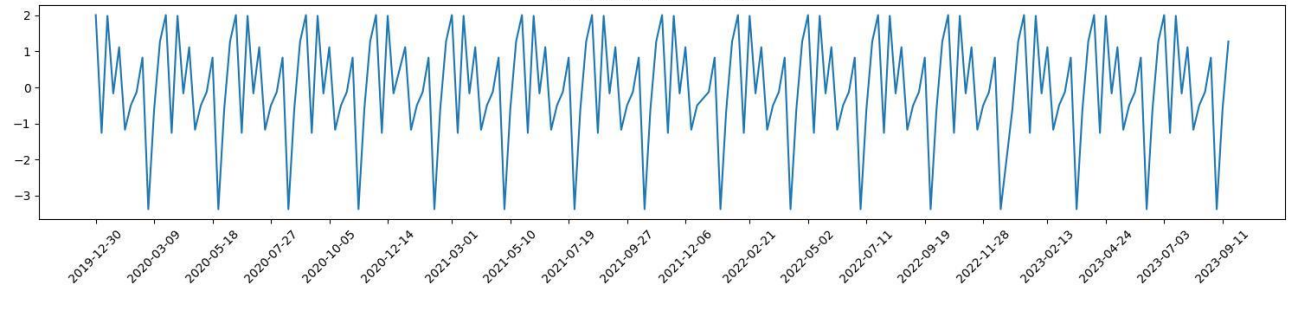

16 Estacionalidad :: T323R2BB
1.5
1.0
0.5
0.0
−0.5
−1.0
2019-12-30
2020-03-09
2020-05-18
2020-07-27
2020-10-05
2020-12-14
2021-03-01
2021-05-10
2021-07-19
2021-09-27
2021-12-06
2022-02-21
2022-05-02
2022-07-11
2022-09-19
2022-11-28
2023-02-13
2023-04-24
2023-07-03
2023-09-11

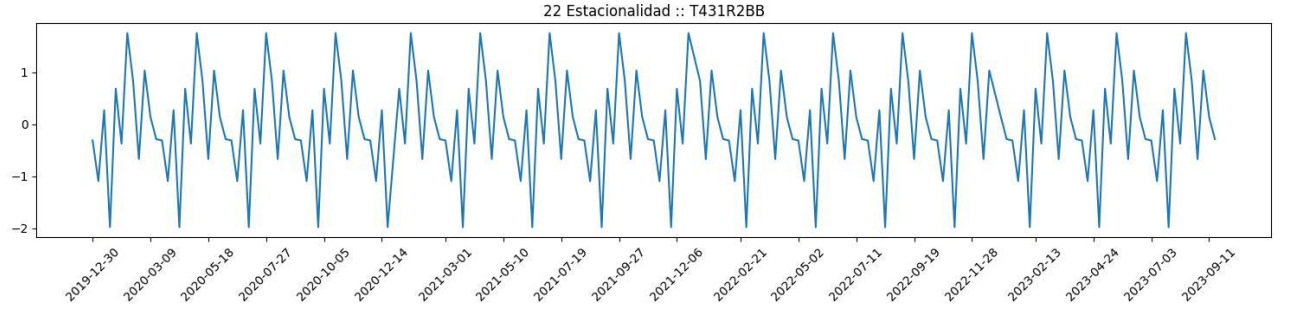

17 Estacionalidad :: T213R1B0
2
1
0
−1
−2
2019-12-30
2020-03-09
2020-05-18
2020-07-27
2020-10-05
2020-12-14
2021-03-01
2021-05-10
2021-07-19
2021-09-27
2021-12-06
2022-02-21
2022-05-02
2022-07-11
2022-09-19
2022-11-28
2023-02-13
2023-04-24
2023-07-03
2023-09-11

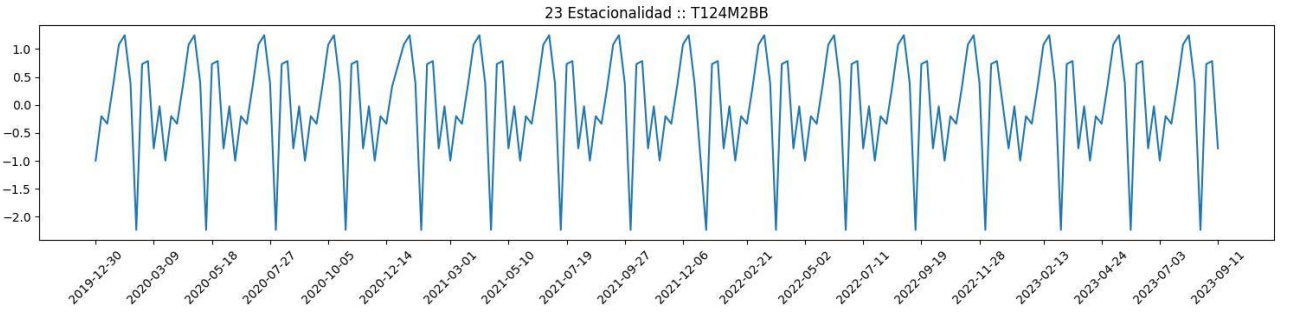

18 Estacionalidad :: T323R1B0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
2019-12-30
2020-03-09
2020-05-18
2020-07-27
2020-10-05
2020-12-14
2021-03-01
2021-05-10
2021-07-19
2021-09-27
2021-12-06
2022-02-21
2022-05-02
2022-07-11
2022-09-19
2022-11-28
2023-02-13
2023-04-24
2023-07-03
2023-09-11

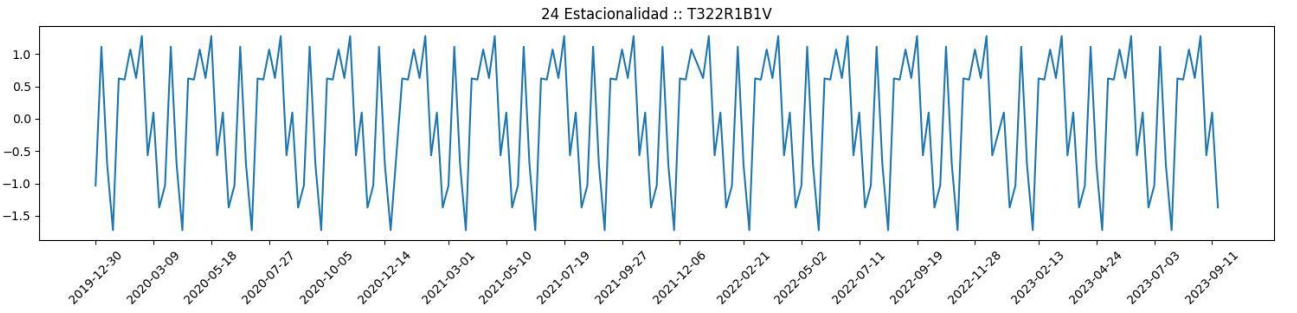

19 Estacionalidad :: T322R2BB
2
1
0
−1
−2
−3
2019-12-30
2020-03-09
2020-05-18
2020-07-27
2020-10-05
2020-12-14
2021-03-08
2021-05-17
2021-07-26
2021-10-04
2021-12-13
2022-02-28
2022-05-09
2022-07-18
2022-09-26
2022-12-05
2023-02-20
2023-05-01
2023-07-10
2023-09-18

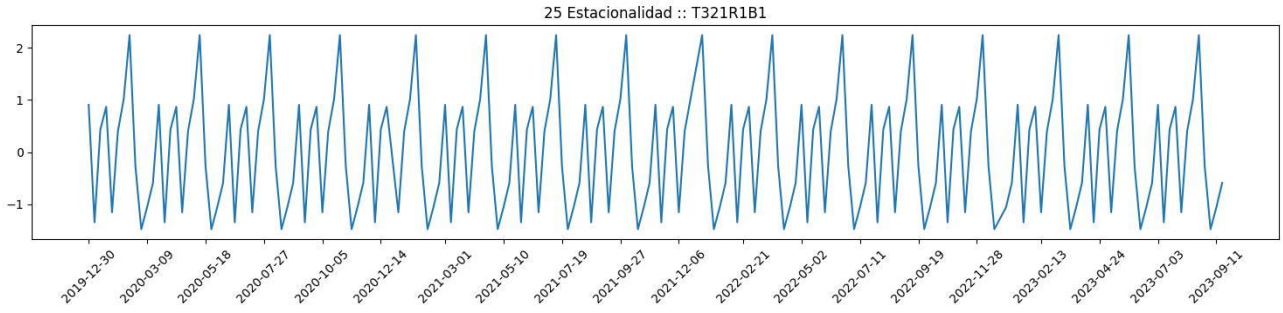

20 Estacionalidad :: T323R8BB

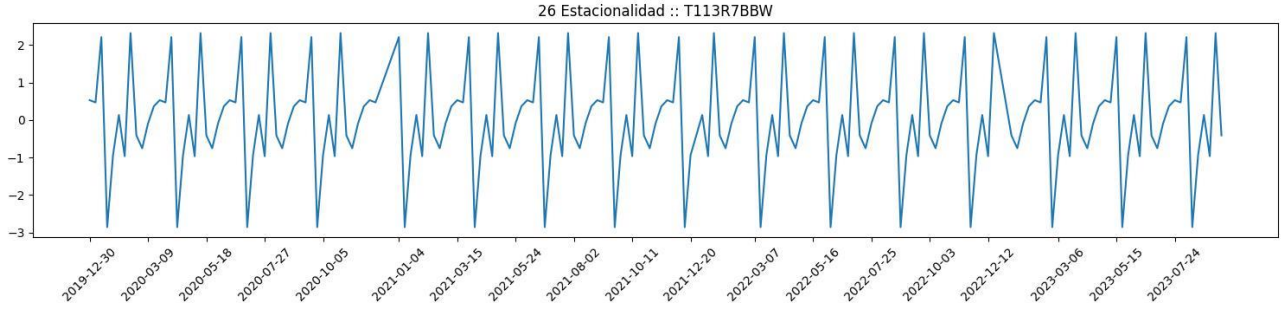

21 Estacionalidad :: T110N1B0

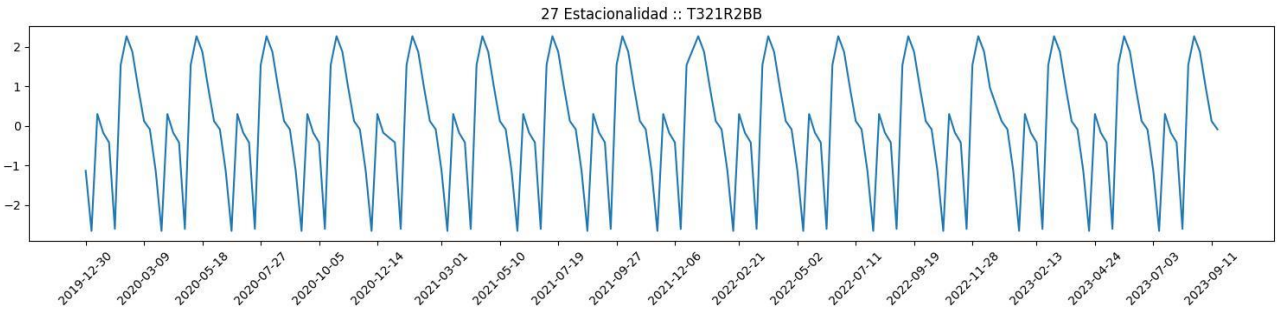

22 Estacionalidad :: T431R2BB

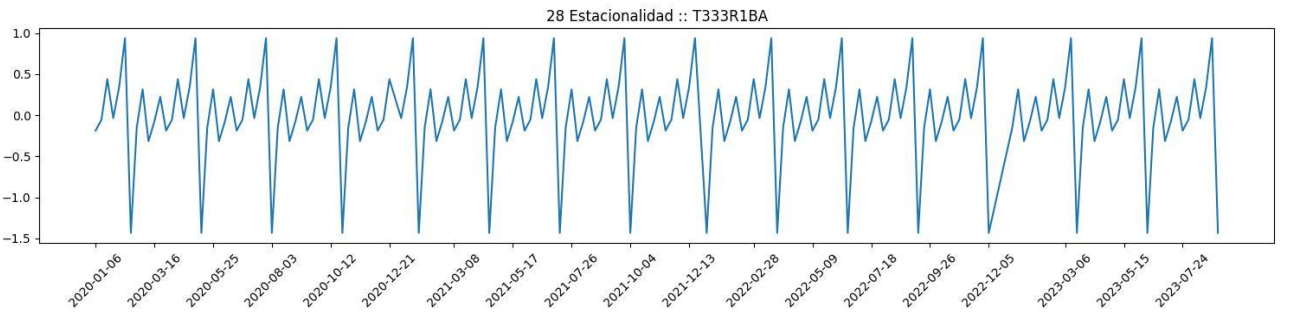

23 Estacionalidad :: T124M2BB

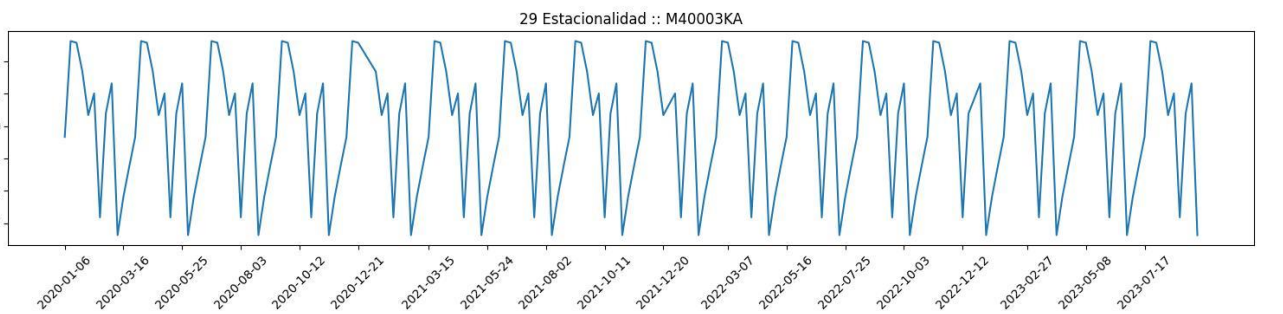

24 Estacionalidad :: T322R1B1V

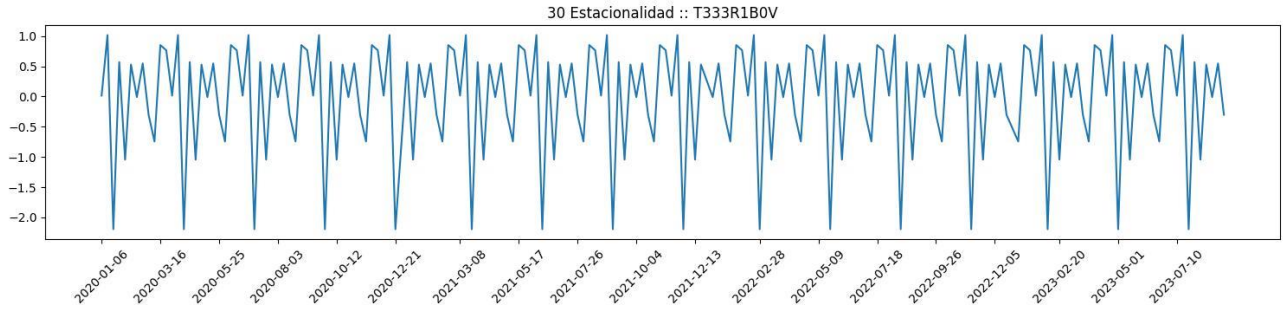

25 Estacionalidad :: T321R1B1

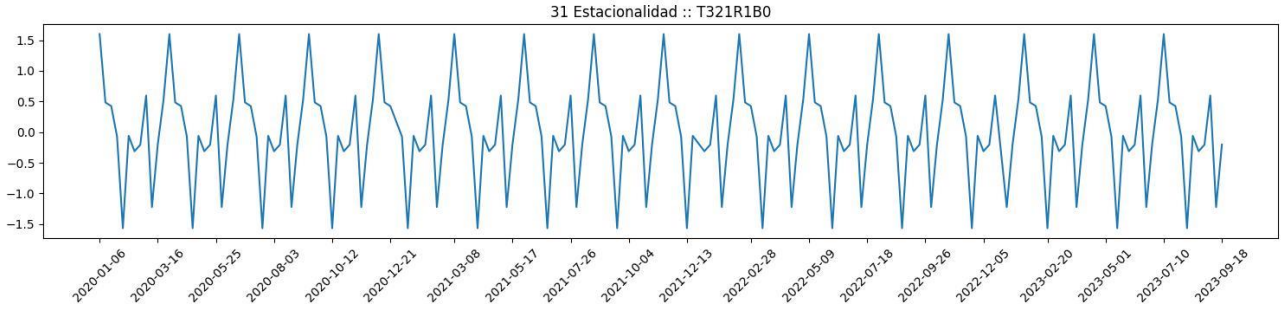

26 Estacionalidad :: T113R7BBW

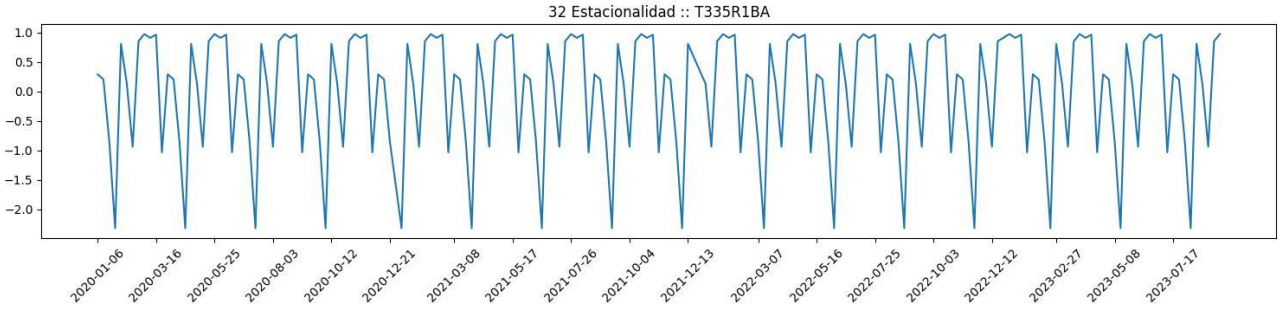

27 Estacionalidad :: T321R2BB

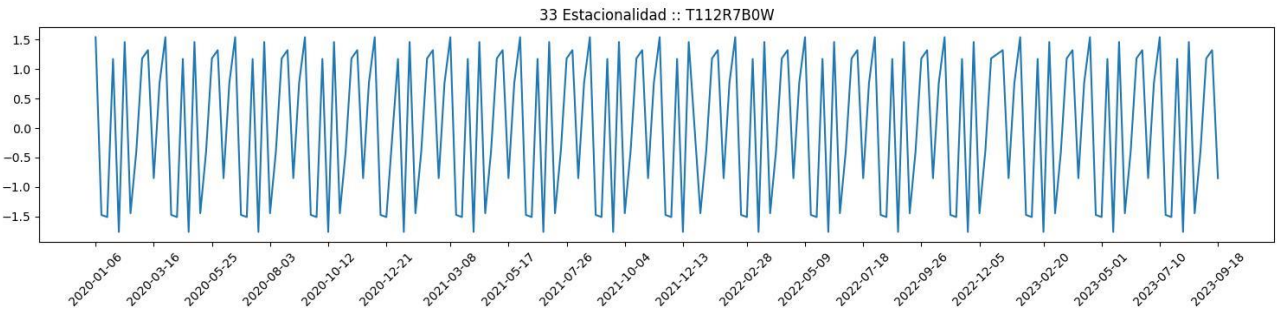

28 Estacionalidad :: T333R1BA

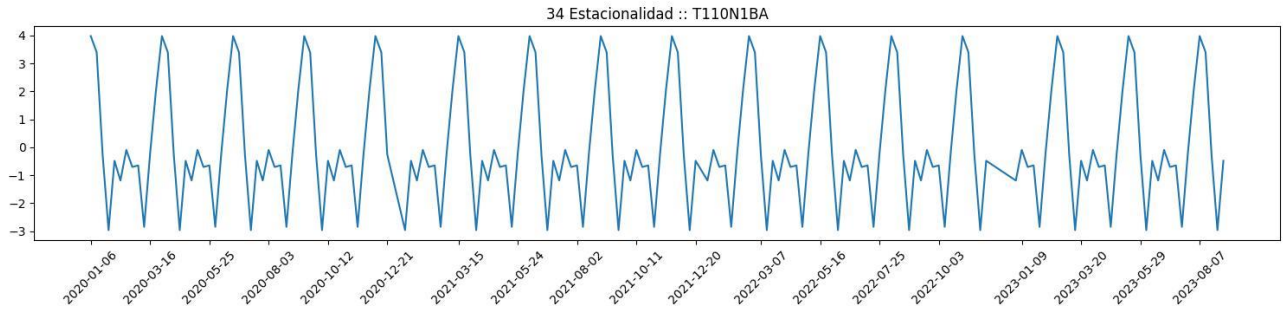

29 Estacionalidad :: M40003KA

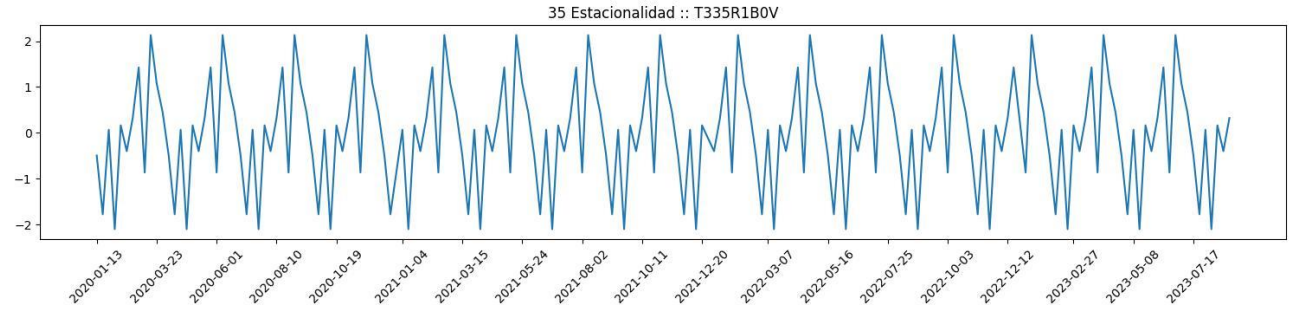

30 Estacionalidad :: T333R1B0V

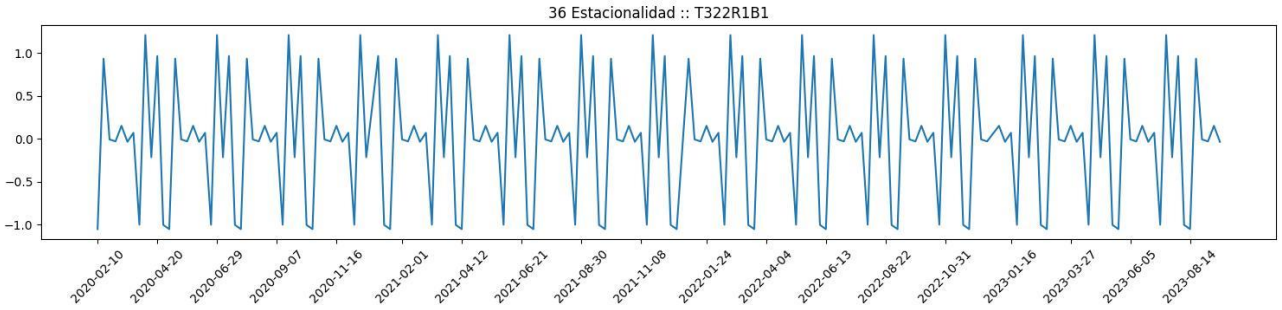

31 Estacionalidad :: T321R1B0

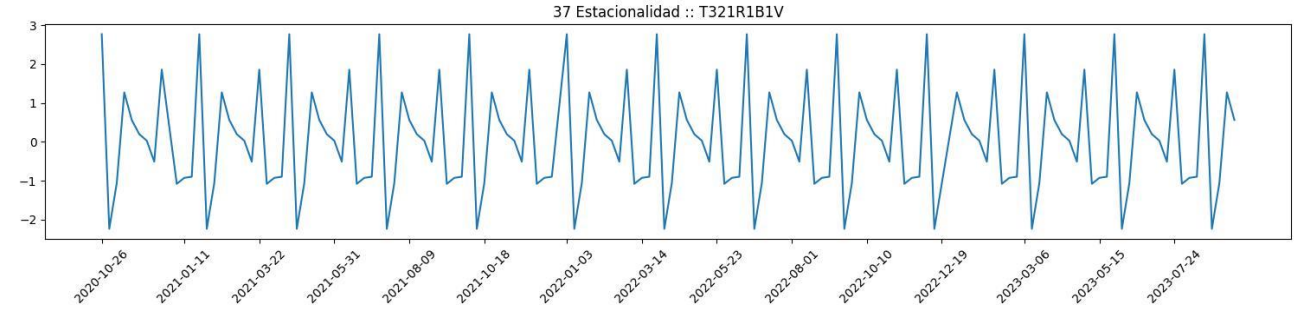

32 Estacionalidad :: T335R1BA


33 Estacionalidad :: T112R7B0W

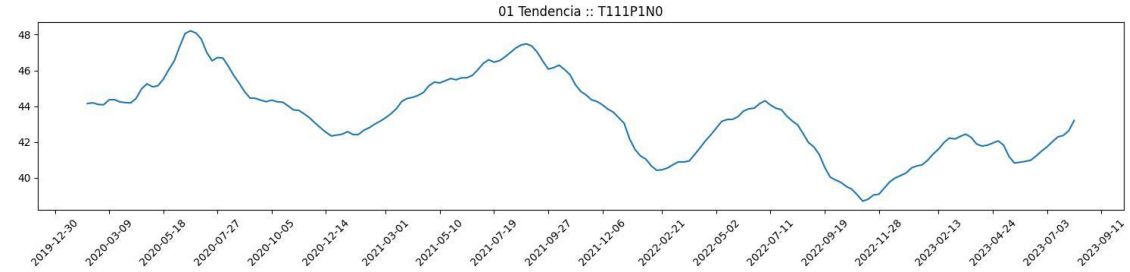

34 Estacionalidad :: T110N1BA

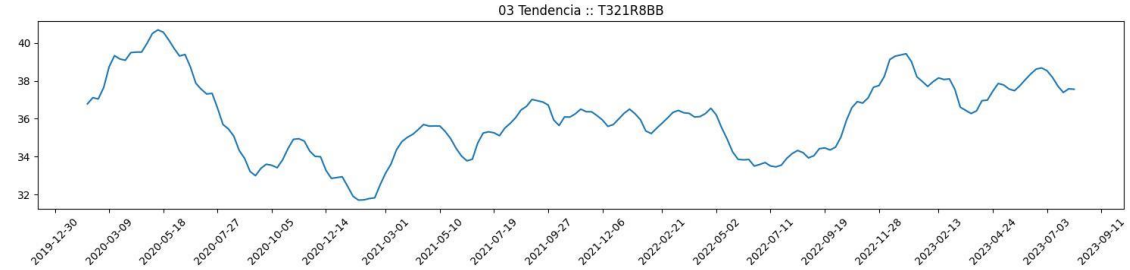

35 Estacionalidad :: T335R1B0V
2
1
0
−1
−2
2020-01-13
2020-03-23
2020-06-01
2020-08-10
2020-10-19
2021-01-04
2021-03-15
2021-05-24
2021-08-02
2021-10-11
2021-12-20
2022-03-07
2022-05-16
2022-07-25
2022-10-03
2022-12-12
2023-02-27
2023-05-08
2023-07-17


36 Estacionalidad :: T322R1B1
1.0
0.5
0.0
−0.5
−1.0
2020-02-10
2020-04-20
2020-06-29
2020-09-07
2020-11-16
2021-02-01
2021-04-12
2021-06-21
2021-08-30
2021-11-08
2022-01-24
2022-04-04
2022-06-13
2022-08-22
2022-10-31
2023-01-16
2023-03-27
2023-06-05
2023-08-14


37 Estacionalidad :: T321R1B1V
3
2
1
0
−1
−2
2020-10-26
2021-01-11
2021-03-22
2021-05-31
2021-08-09
2021-10-18
2022-01-03
2022-03-14
2022-05-23
2022-08-01
2022-10-10
2022-12-19
2023-03-06
2023-05-15
2023-07-24

Apéndice 2 Gráficos de Tendencias






06 Tendencia :: T323R1BA
46
44
42
40
38
2019-12-30
2020-03-09
2020-05-18
2020-07-27
2020-10-05
2020-12-14
2021-03-01
2021-05-10
2021-07-19
2021-09-27
2021-12-06
2022-02-28
2022-05-09
2022-07-18
2022-09-26
2023-02-13
2023-04-24
2023-07-03
2023-09-11


07 Tendencia :: T323R1B0V
44
42
40
38
2019-12-30
2020-03-09
2020-05-18
2020-07-27
2020-10-05
2020-12-14
2021-03-01
2021-05-10
2021-07-19
2021-09-27
2021-12-06
2022-02-21
2022-05-02
2022-07-11
2022-09-19
2022-11-28
2023-02-13
2023-04-24
2023-07-03
2023-09-11


08 Tendencia :: T114R1N0
47
46
45
44
43
42
2019-12-30
2020-03-09
2020-05-18
2020-07-27
2020-10-05
2020-12-14
2021-03-01
2021-05-10
2021-07-19
2021-09-27
2022-11-21
2023-02-06
2023-04-17


09 Tendencia :: T112R1N0
47
46
45
44
43
42
41
2019-12-30
2020-03-09
2020-05-18
2020-07-27
2020-10-05
2020-12-14
2021-03-01
2021-05-10
2021-07-19
2021-09-27
2021-12-06
2022-02-21
2022-05-02
2022-07-11
2022-09-19
2022-11-28
2023-02-13
2023-04-24
2023-07-03
2023-09-11


10 Tendencia :: T324M2BB
28
27
26
25
24
23
22
2019-12-30
2020-03-09
2020-05-18
2020-07-27
2020-10-05
2020-12-14
2021-03-01
2021-05-10
2021-07-19
2021-09-27
2021-12-06
2022-02-21
2022-05-02
2022-07-11
2022-09-19
2022-11-28
2023-02-13
2023-04-24
2023-07-03
2023-09-11


11 Tendencia :: T113R1N0
48
47
46
45
44
43
42
2019-12-30
2020-03-09
2020-05-18
2020-07-27
2020-10-05
2020-12-14
2021-03-01
2021-05-10
2021-07-19
2021-09-27
2021-12-06
2022-02-21
2022-05-02
2022-07-11
2022-09-19
2022-11-28
2023-02-13
2023-04-24
2023-07-03
2023-09-11


12 Tendencia :: T333R2BB
38
36
34
32
30
28
2019-12-30
2020-03-09
2020-05-18
2020-07-27
2020-10-05
2020-12-14
2021-03-01
2021-05-10
2021-07-19
2021-09-27
2021-12-06
2022-02-21
2022-05-02
2022-07-11
2022-09-19
2022-11-28
2023-02-13
2023-04-24
2023-07-03
2023-09-11


13 Tendencia :: T111N2B0D
48
46
44
42
40
38
36
2019-12-30
2020-03-09
2020-05-18
2020-07-27
2020-10-05
2020-12-14
2021-03-08
2021-05-17
2021-07-26
2021-10-04
2021-12-13
2022-02-28
2022-05-09
2022-07-18
2022-09-26
2023-02-20


14 Tendencia :: T321R1BA
55
50
45
40
35
2019-12-30
2020-03-09
2020-05-18
2020-07-27
2020-10-05
2020-12-14
2021-03-08
2021-05-17
2021-07-26
2021-10-04
2021-12-13
2022-03-14
2022-05-23
2022-08-01
2022-10-10
2022-12-19
2023-03-06
2023-05-15
2023-07-24


15 Tendencia :: T423R2BB
32
30
28
26
24
2019-12-30
2020-03-09
2020-05-18
2020-07-27
2020-10-05
2020-12-14
2021-03-01
2021-05-10
2021-07-19
2021-09-27
2021-12-06
2022-02-21
2022-05-02
2022-07-11
2022-09-19
2022-11-28
2023-02-13
2023-04-24
2023-07-03
2023-09-11


16 Tendencia :: T323R2BB


17 Tendencia :: T213R1B0


18 Tendencia :: T323R1B0


19 Tendencia :: T322R2BB


20 Tendencia :: T323R8BB


21 Tendencia :: T110N1B0
46
44
42
40
38
36
2019-12-30
2020-03-09
2020-05-18
2020-07-27
2020-10-05
2020-12-14
2021-03-01
2021-05-10
2021-07-19
2021-09-27
2021-12-06
2022-02-21
2022-05-02
2022-07-11
2022-09-19
2022-11-28
2023-02-13
2023-04-24
2023-07-03
2023-09-11


22 Tendencia :: T431R2BB
34
32
30
28
26
24
2019-12-30
2020-03-09
2020-05-18
2020-07-27
2020-10-05
2020-12-14
2021-03-01
2021-05-10
2021-07-19
2021-09-27
2021-12-06
2022-02-21
2022-05-02
2022-07-11
2022-09-19
2022-11-28
2023-02-13
2023-04-24
2023-07-03
2023-09-11


23 Tendencia :: T124M2BB
28
27
26
25
24
23
22
21
2019-12-30
2020-03-09
2020-05-18
2020-07-27
2020-10-05
2020-12-14
2021-03-01
2021-05-10
2021-07-19
2021-09-27
2021-12-06
2022-02-21
2022-05-02
2022-07-11
2022-09-19
2022-11-28
2023-02-13
2023-04-24
2023-07-03
2023-09-11


24 Tendencia :: T322R1B1V
44
42
40
38
36
2019-12-30
2020-03-09
2020-05-18
2020-07-27
2020-10-05
2020-12-14
2021-03-01
2021-05-10
2021-07-19
2021-09-27
2021-12-06
2022-02-21
2022-05-02
2022-07-11
2022-09-19
2022-11-28
2023-02-13
2023-04-24
2023-07-03
2023-09-11


25 Tendencia :: T321R1B1
45
40
35
30
25
2019-12-30
2020-03-09
2020-05-18
2020-07-27
2020-10-05
2020-12-14
2021-03-01
2021-05-10
2021-07-19
2021-09-27
2021-12-06
2022-02-21
2022-05-02
2022-07-11
2022-09-19
2022-11-28
2023-02-13
2023-04-24
2023-07-03
2023-09-11


26 Tendencia :: T113R7BBW
50.0
47.5
45.0
42.5
40.0
37.5
35.0
2019-12-30
2020-03-09
2020-05-18
2020-07-27
2020-10-05
2021-01-04
2021-03-15
2021-05-24
2021-08-02
2021-10-11
2021-12-20
2022-03-07
2022-05-16
2022-07-25
2022-10-03
2022-12-12
2023-03-06
2023-05-15
2023-07-24


27 Tendencia :: T321R2BB
50
48
46
44
42
40
38
2019-12-30
2020-03-09
2020-05-18
2020-07-27
2020-10-05
2020-12-14
2021-03-01
2021-05-10
2021-07-19
2021-09-27
2021-12-06
2022-02-21
2022-05-02
2022-07-11
2022-09-19
2022-11-28
2023-02-13
2023-04-24
2023-07-03
2023-09-11


28 Tendencia :: T333R1BA
37
36
35
34
33
32
31
2020-01-06
2020-03-16
2020-05-25
2020-08-03
2020-10-12
2020-12-21
2021-03-08
2021-05-17
2021-07-26
2021-10-04
2021-12-13
2022-02-28
2022-05-09
2022-07-18
2022-09-26
2022-12-05
2023-03-06
2023-05-15
2023-07-24


29 Tendencia :: M40003KA
35.0
32.5
30.0
27.5
25.0
22.5
20.0
17.5
2020-01-06
2020-03-16
2020-05-25
2020-08-03
2020-10-12
2020-12-21
2021-03-15
2021-05-24
2021-08-02
2021-10-11
2021-12-20
2022-03-07
2022-05-16
2022-07-25
2022-10-03
2022-12-12
2023-02-27
2023-05-08
2023-07-17


30 Tendencia :: T333R1B0V
36
34
32
30
28
2020-01-06
2020-03-16
2020-05-25
2020-08-03
2020-10-12
2020-12-21
2021-03-08
2021-05-17
2021-07-26
2021-10-04
2021-12-13
2022-02-28
2022-05-09
2022-07-18
2022-09-26
2022-12-05
2023-02-20
2023-05-01
2023-07-10


31 Tendencia :: T321R1B0
48
46
44
42
40
2020-01-06
2020-03-16
2020-05-25
2020-08-03
2020-10-12
2020-12-21
2021-03-08
2021-05-17
2021-07-26
2021-10-04
2021-12-13
2022-02-28
2022-05-09
2022-07-18
2022-09-26
2022-12-05
2023-02-20
2023-05-01
2023-07-10
2023-09-18


32 Tendencia :: T335R1BA
42
40
38
36
34
2020-01-06
2020-03-16
2020-05-25
2020-08-03
2020-10-12
2020-12-21
2021-03-08
2021-05-17
2021-07-26
2021-10-04
2021-12-13
2022-03-07
2022-05-16
2022-07-25
2022-10-03
2022-12-12
2023-02-27
2023-05-08
2023-07-17


33 Tendencia :: T112R7B0W
52.5
50.0
47.5
45.0
42.5
40.0
37.5
2020-01-06
2020-03-16
2020-05-25
2020-08-03
2020-10-12
2020-12-21
2021-03-08
2021-05-17
2021-07-26
2021-10-04
2021-12-13
2022-02-28
2022-05-09
2022-07-18
2022-09-26
2022-12-05
2023-02-20
2023-05-01
2023-07-10
2023-09-18

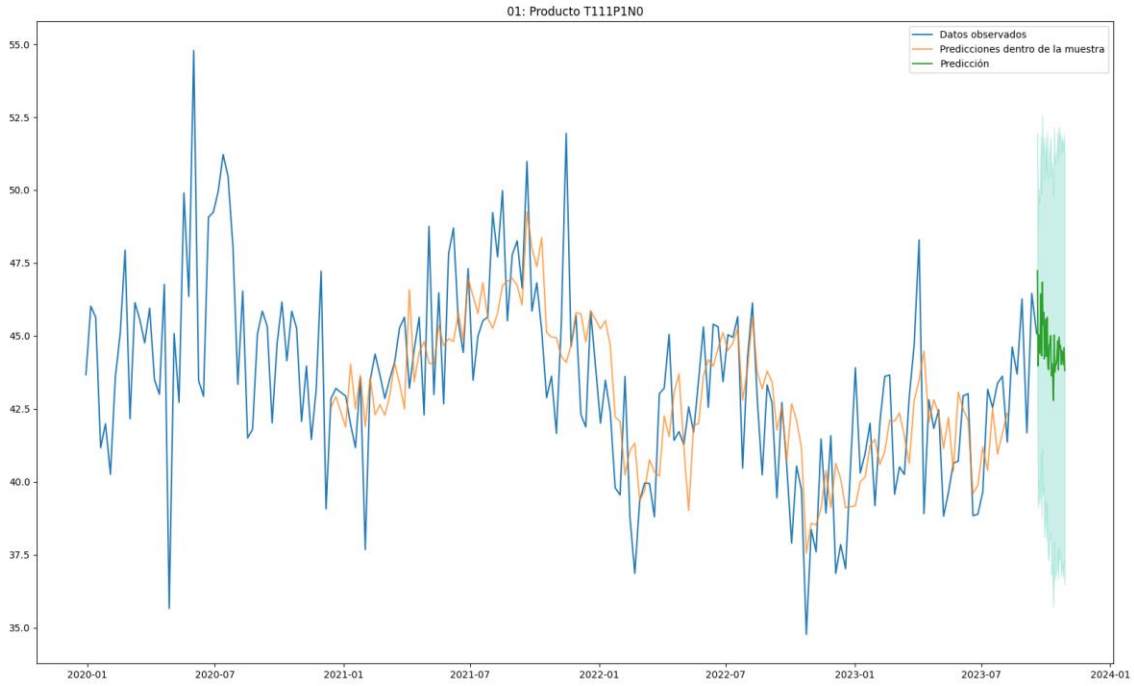

34 Tendencia :: T110N1BA
50.0
47.5
45.0
42.5
40.0
37.5
35.0
2020-01-06
2020-03-16
2020-05-25
2020-08-03
2020-10-12
2020-12-21
2021-03-15
2021-05-24
2021-08-02
2021-10-11
2021-12-20
2022-03-07
2022-05-16
2022-07-25
2022-10-03
2023-01-09
2023-03-20
2023-05-29
2023-08-07

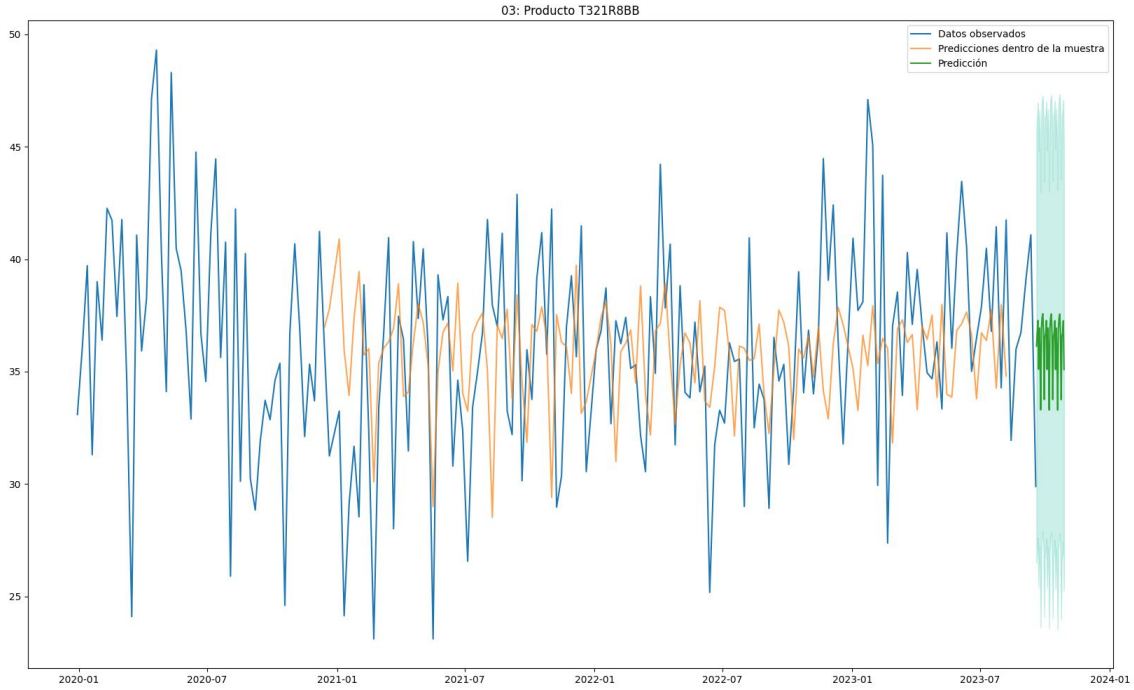

35 Tendencia :: T335R1B0V
42
40
38
36
34
32
30
28
2020-01-13
2020-03-23
2020-06-01
2020-08-10
2020-10-19
2021-01-04
2021-03-15
2021-05-24
2021-08-02
2021-10-11
2021-12-20
2022-03-07
2022-05-16
2022-07-25
2022-10-03
2022-12-12
2023-02-27
2023-05-08
2023-07-17

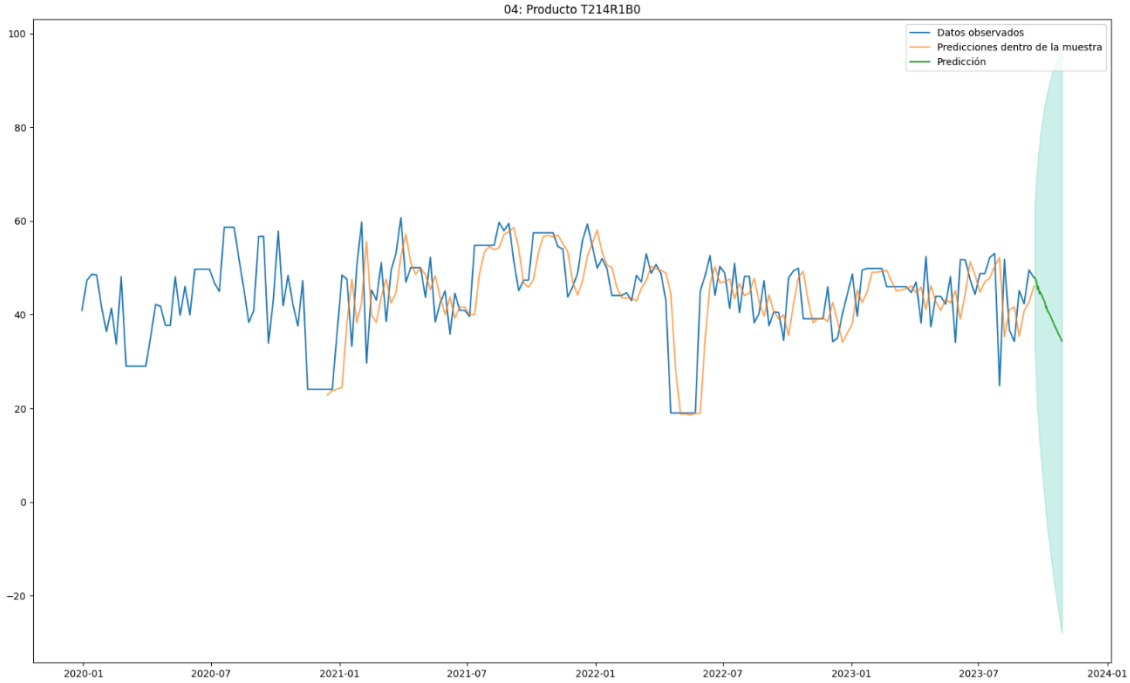

36 Tendencia :: T322R1B1
44
42
40
38
36
34
32
2020-02-10
2020-04-20
2020-06-29
2020-09-07
2020-11-16
2021-02-01
2021-04-12
2021-06-21
2021-08-30
2021-11-08
2022-01-24
2022-04-04
2022-06-13
2022-08-22
2022-10-31
2023-01-16
2023-03-27
2023-06-05
2023-08-14

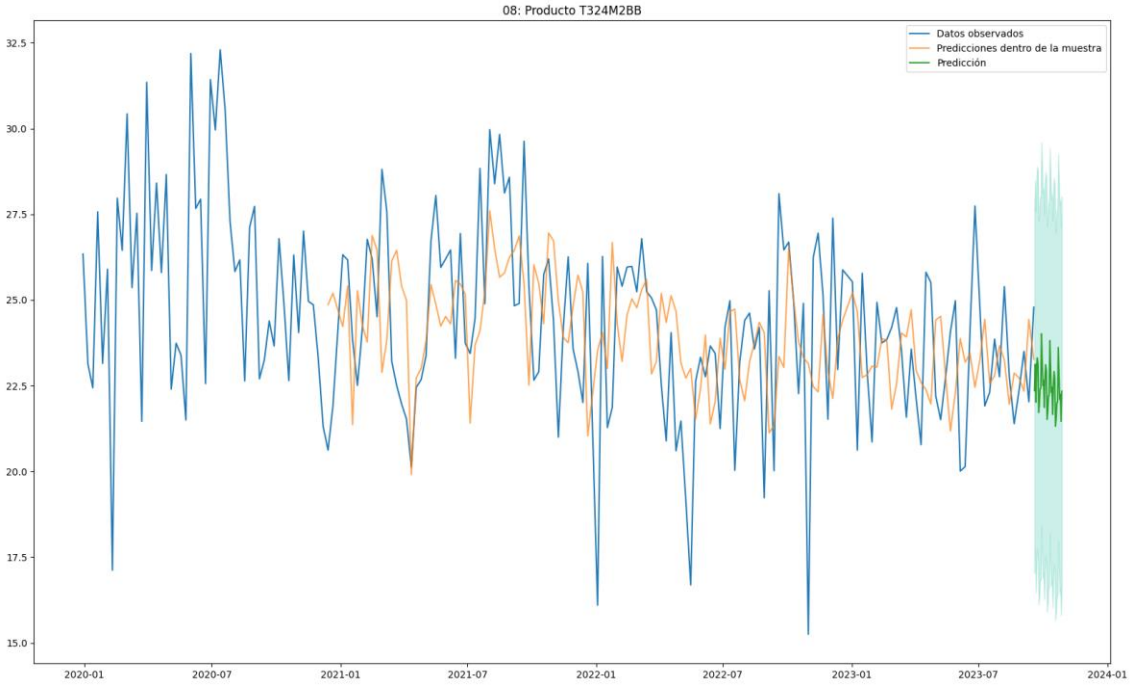

37 Tendencia :: T321R1B1V
48
46
44
42
40
2020-10-26
2021-01-11
2021-03-22
2021-05-31
2021-08-09
2021-10-18
2022-01-03
2022-03-14
2022-05-23
2022-08-01
2022-10-10
2022-12-19
2023-03-06
2023-05-15
2023-07-24

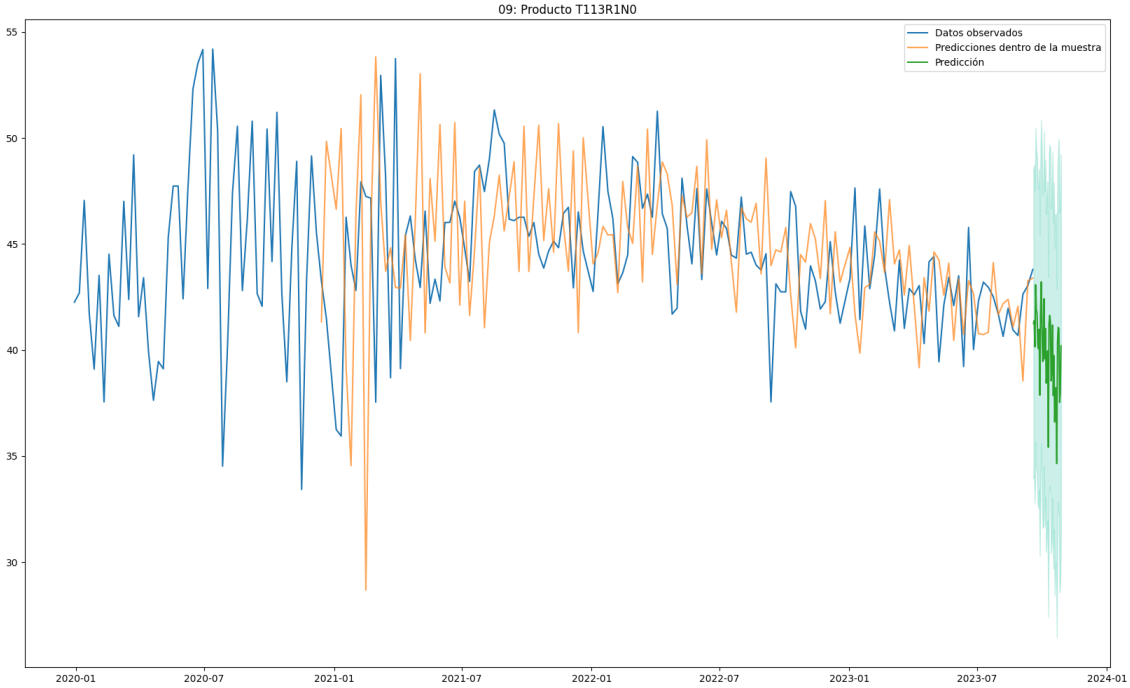
Apéndice 3 Gráficos de Modelo Ajustado SARIMA

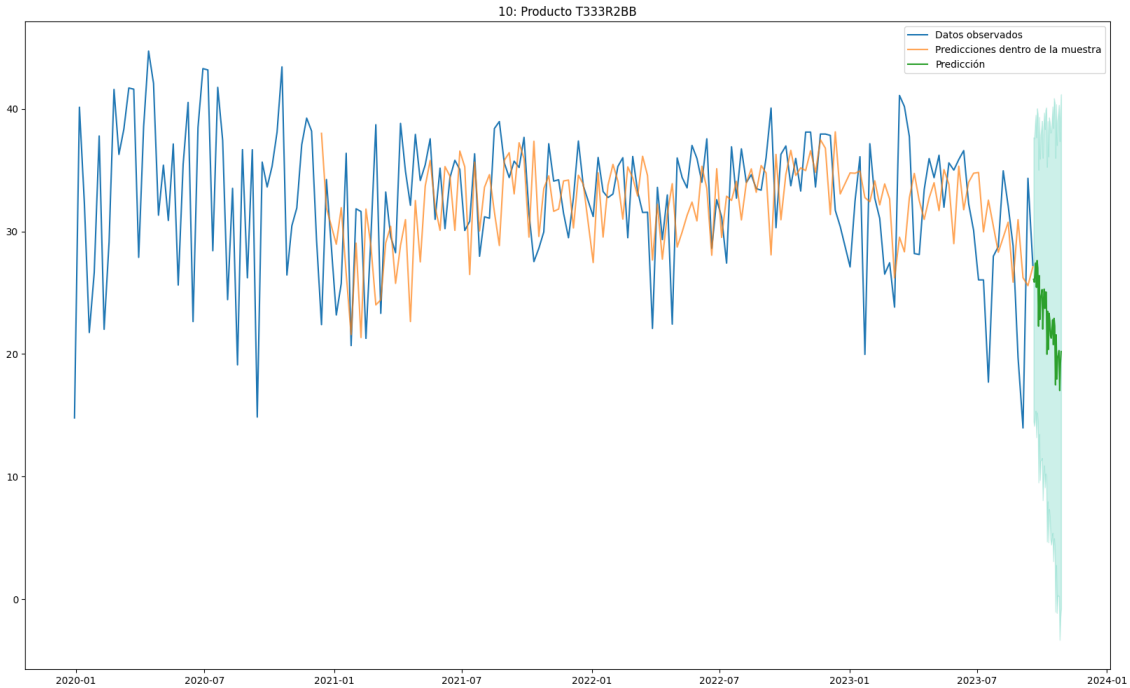


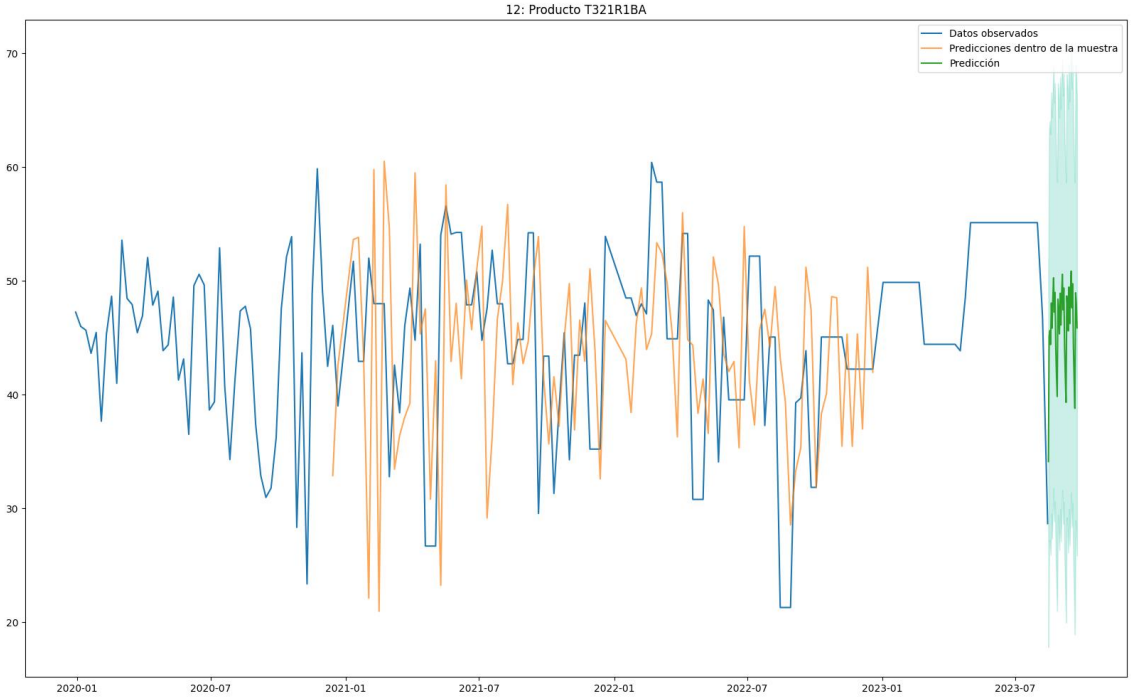


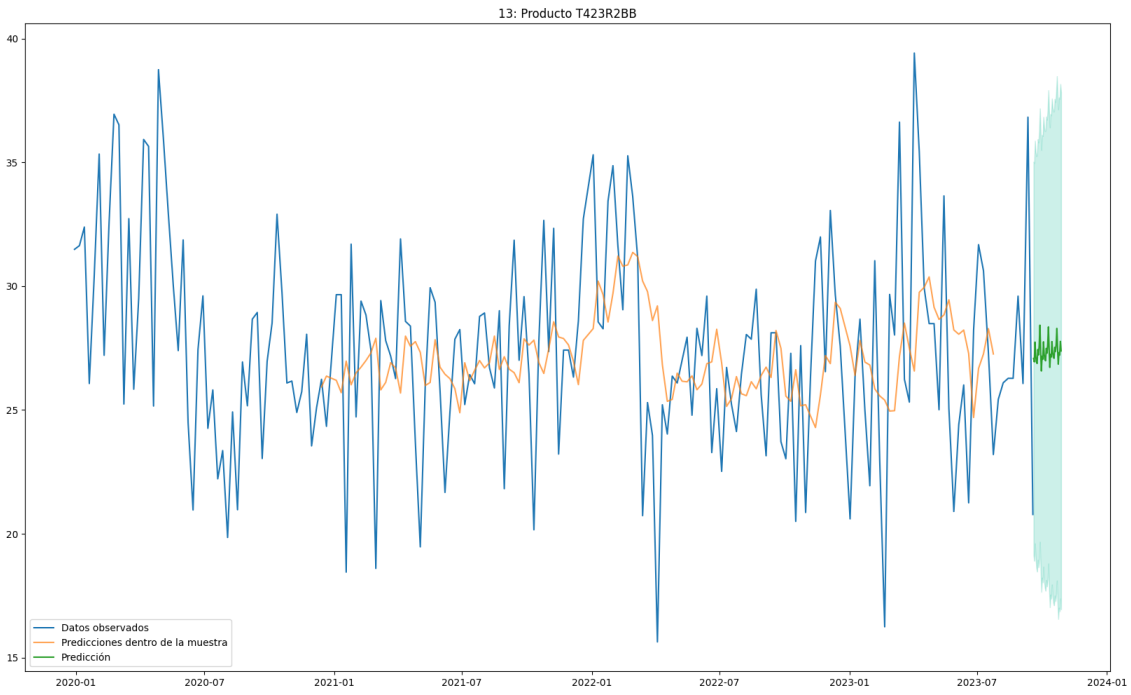

04: Producto T214R1B0
Datos observados
Predicciones dentro de la muestra
Predicción
100
80
60
40
20
0
−20
2020-01
2020-07
2021-01
2021-07
2022-01
2022-07
2023-01
2023-07
2024-01

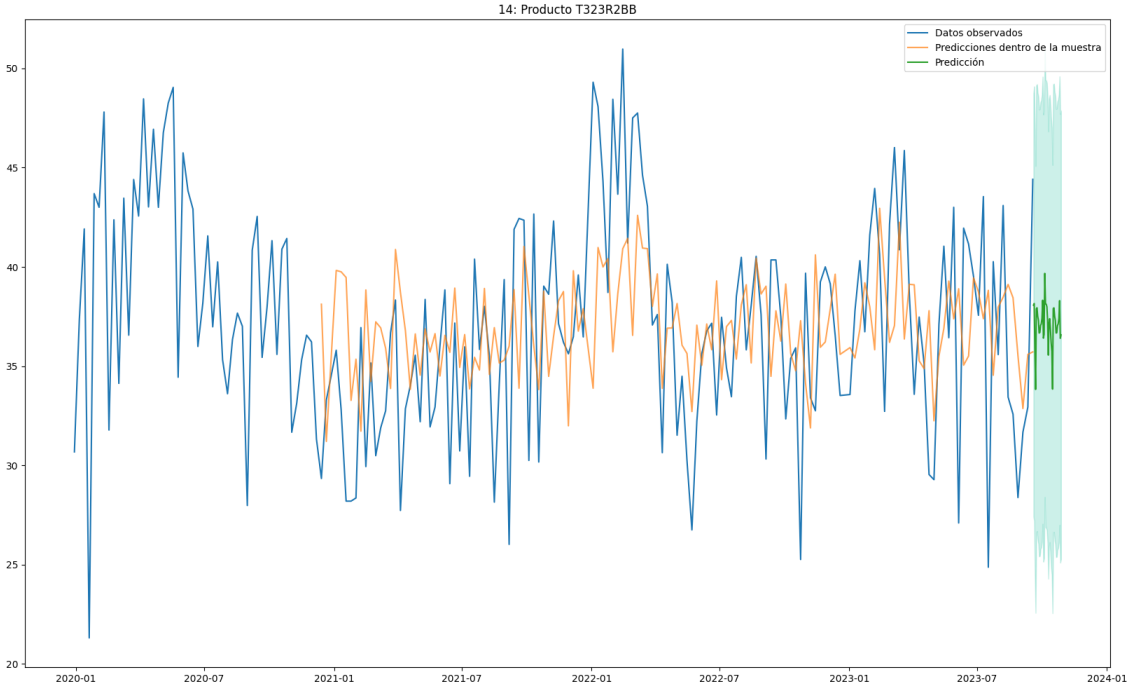

08: Producto T324M2BB
Datos observados
Predicciones dentro de la muestra
Predicción
32.5
30.0
27.5
25.0
22.5
20.0
17.5
15.0
2020-01
2020-07
2021-01
2021-07
2022-01
2022-07
2023-01
2023-07
2024-01

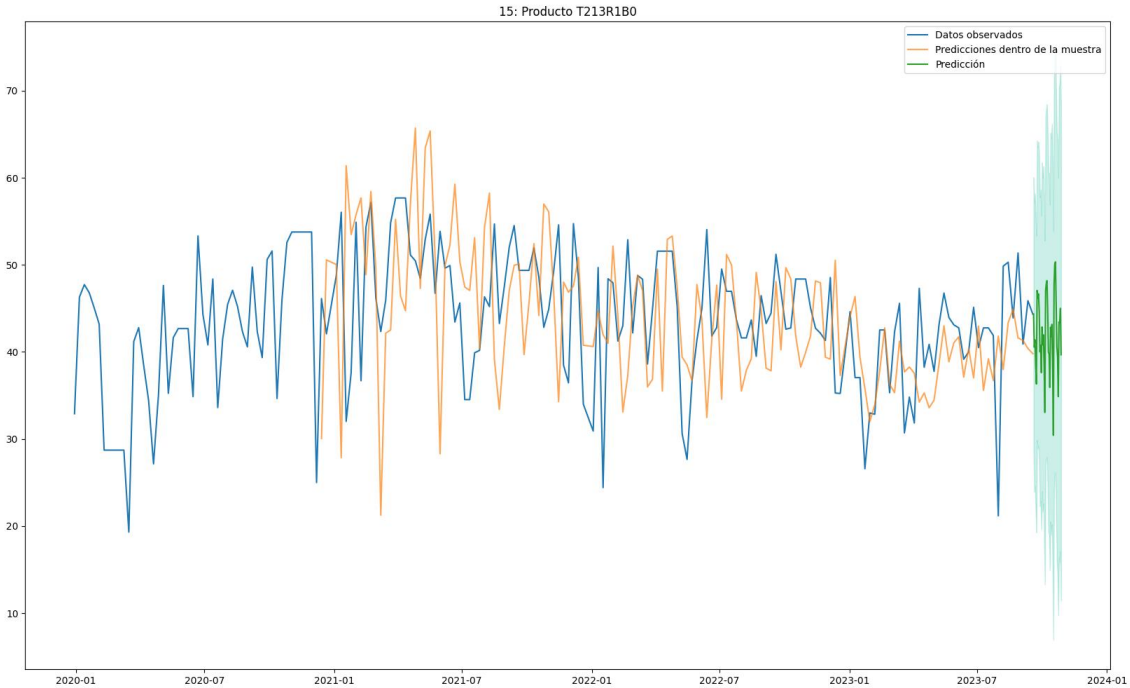

09: Producto T113R1N0
Datos observados
Predicciones dentro de la muestra
Predicción
55
50
45
40
35
30
2020-01
2020-07
2021-01
2021-07
2022-01
2022-07
2023-01
2023-07
2024-01

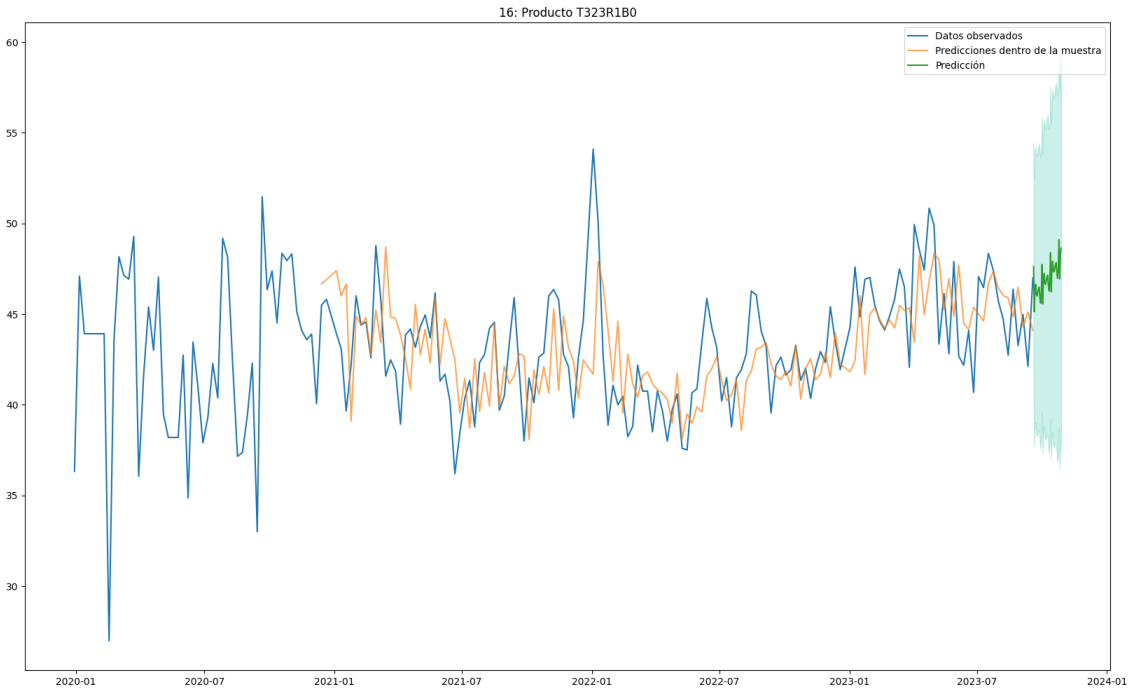

10: Producto T333R2BB
Datos observados
Predicciones dentro de la muestra
Predicción
40
30
20
10
0
2020-01
2020-07
2021-01
2021-07
2022-01
2022-07
2023-01
2023-07
2024-01

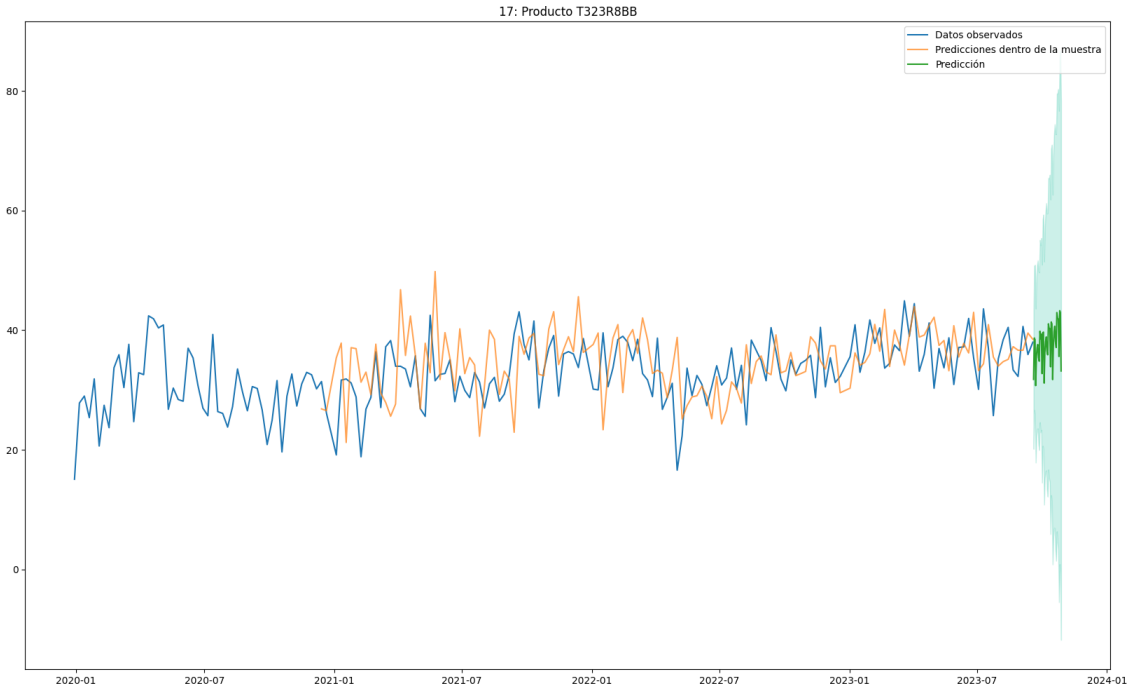

12: Producto T321R1BA
Datos observados
Predicciones dentro de la muestra
Predicción
70
60
50
40
30
20
2020-01
2020-07
2021-01
2021-07
2022-01
2022-07
2023-01
2023-07

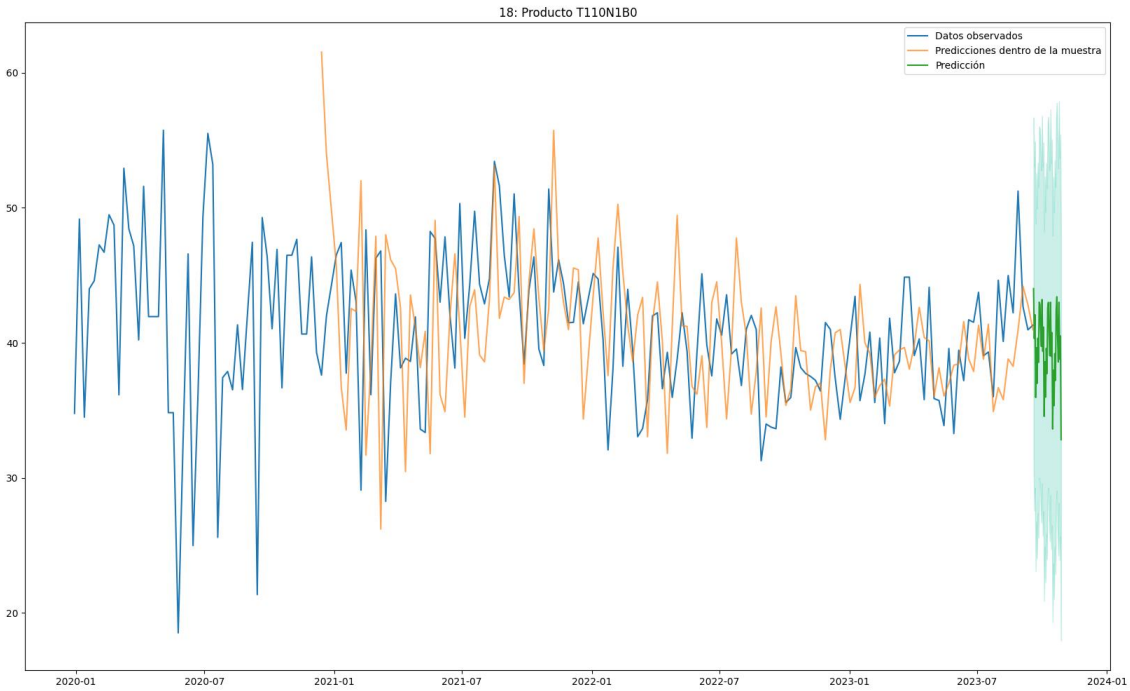

13: Producto T423R2BB
Datos observados
Predicciones dentro de la muestra
Predicción
40
35
30
25
20
15
2020-01
2020-07
2021-01
2021-07
2022-01
2022-07
2023-01
2023-07
2024-01

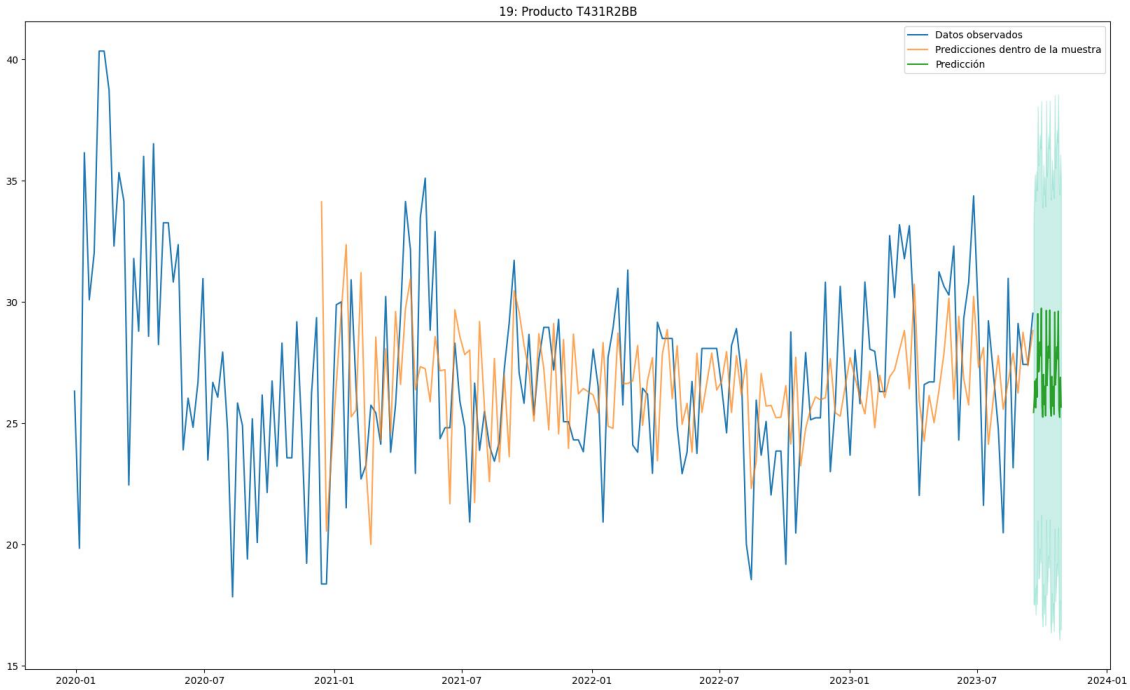

14: Producto T323R2BB
Datos observados
Predicciones dentro de la muestra
Predicción
50
45
40
35
30
25
20
2020-01
2020-07
2021-01
2021-07
2022-01
2022-07
2023-01
2023-07
2024-01

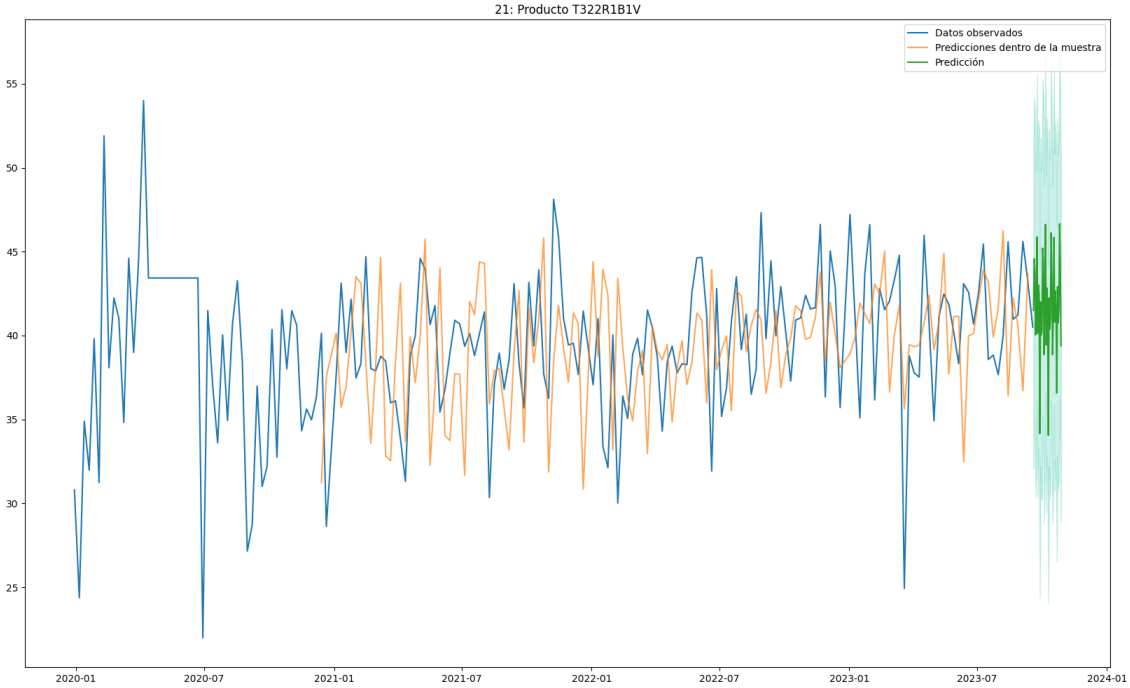

15: Producto T213R1B0
Datos observados
Predicciones dentro de la muestra
Predicción
70
60
50
40
30
20
10
2020-01
2020-07
2021-01
2021-07
2022-01
2022-07
2023-01
2023-07
2024-01

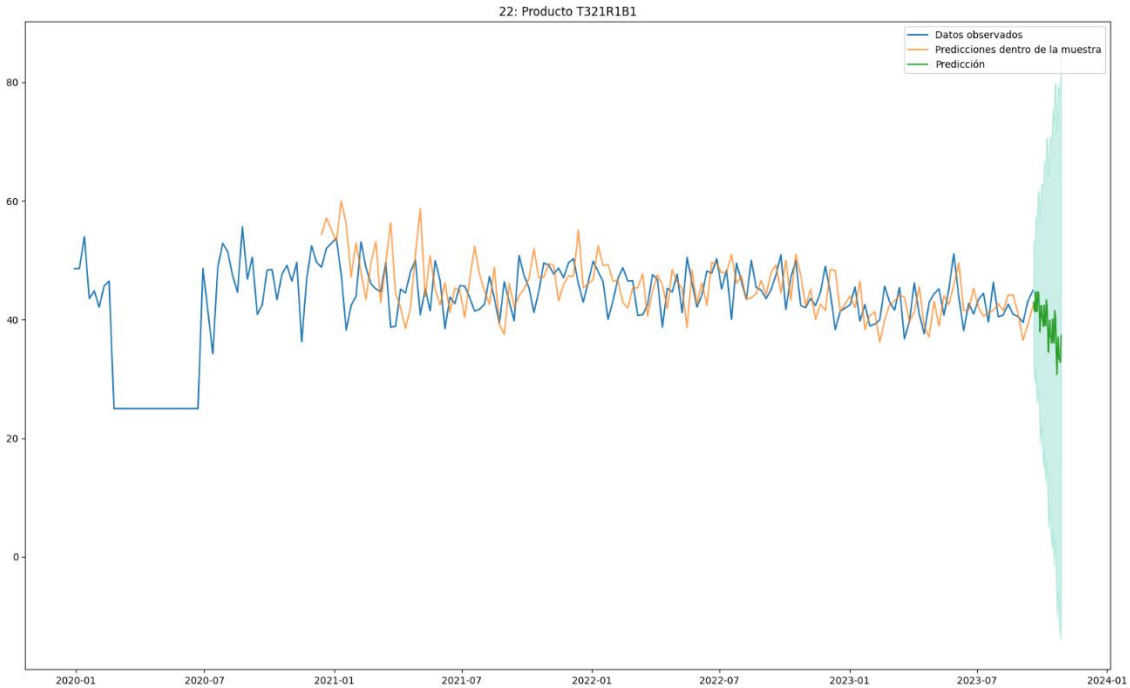

16: Producto T323R1B0
Datos observados
Predicciones dentro de la muestra
Predicción
60
55
50
45
40
35
30
2020-01
2020-07
2021-01
2021-07
2022-01
2022-07
2023-01
2023-07
2024-01

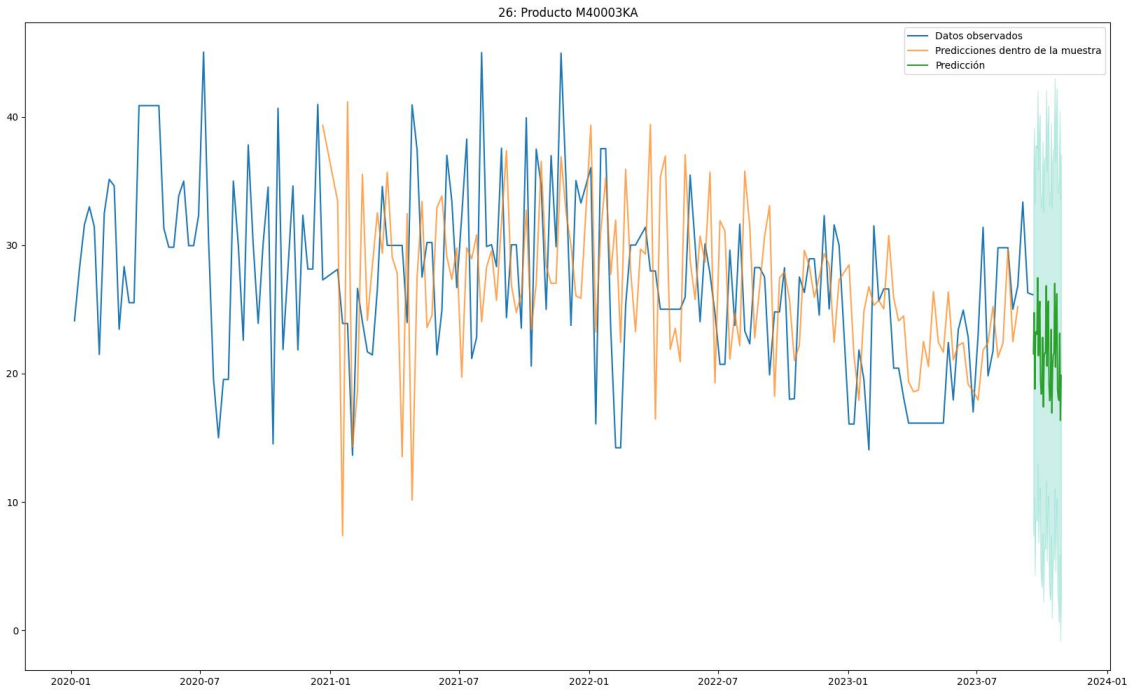

17: Producto T323R8BB
Datos observados
Predicciones dentro de la muestra
Predicción
80
60
40
20
0
2020-01
2020-07
2021-01
2021-07
2022-01
2022-07
2023-01
2023-07
2024-01

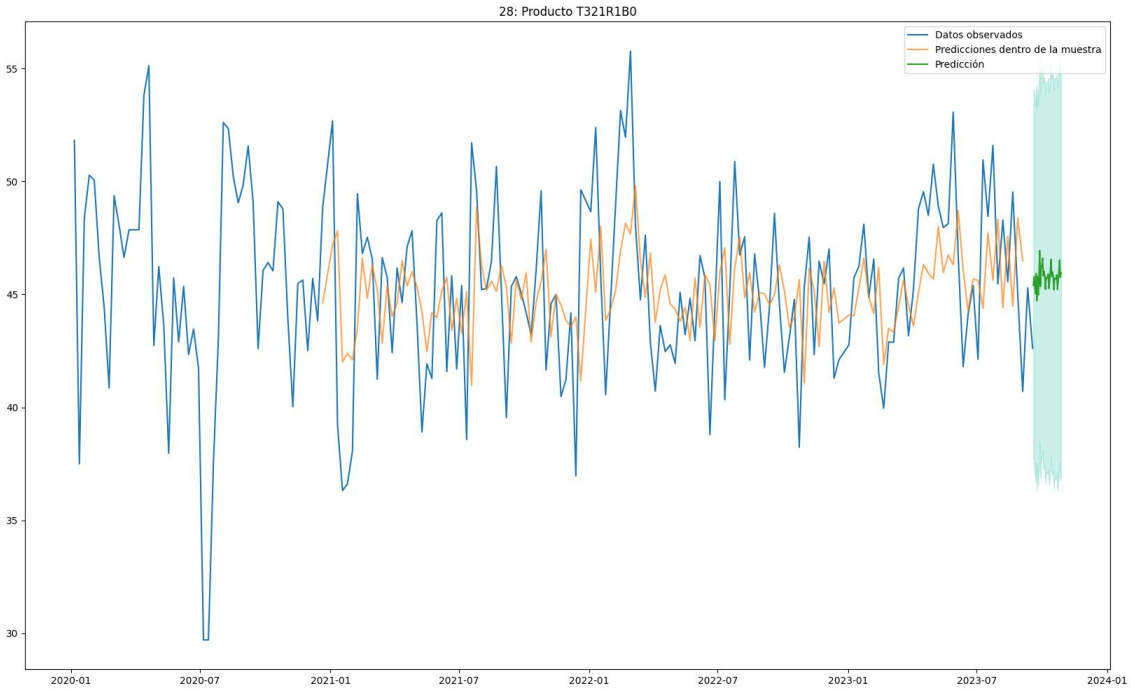

18: Producto T110N1B0
Datos observados
Predicciones dentro de la muestra
Predicción
60
50
40
30
20
2020-01
2020-07
2021-01
2021-07
2022-01
2022-07
2023-01
2023-07
2024-01

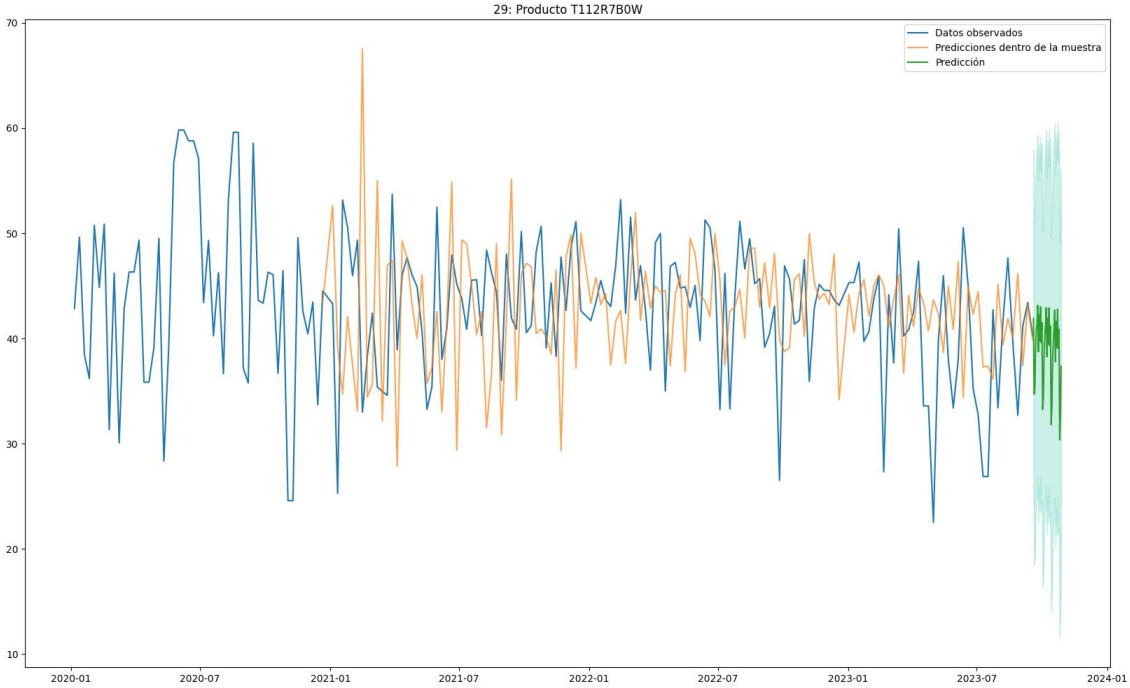

19: Producto T431R2BB
Datos observados
Predicciones dentro de la muestra
Predicción
40
35
30
25
20
15
2020-01
2020-07
2021-01
2021-07
2022-01
2022-07
2023-01
2023-07
2024-01

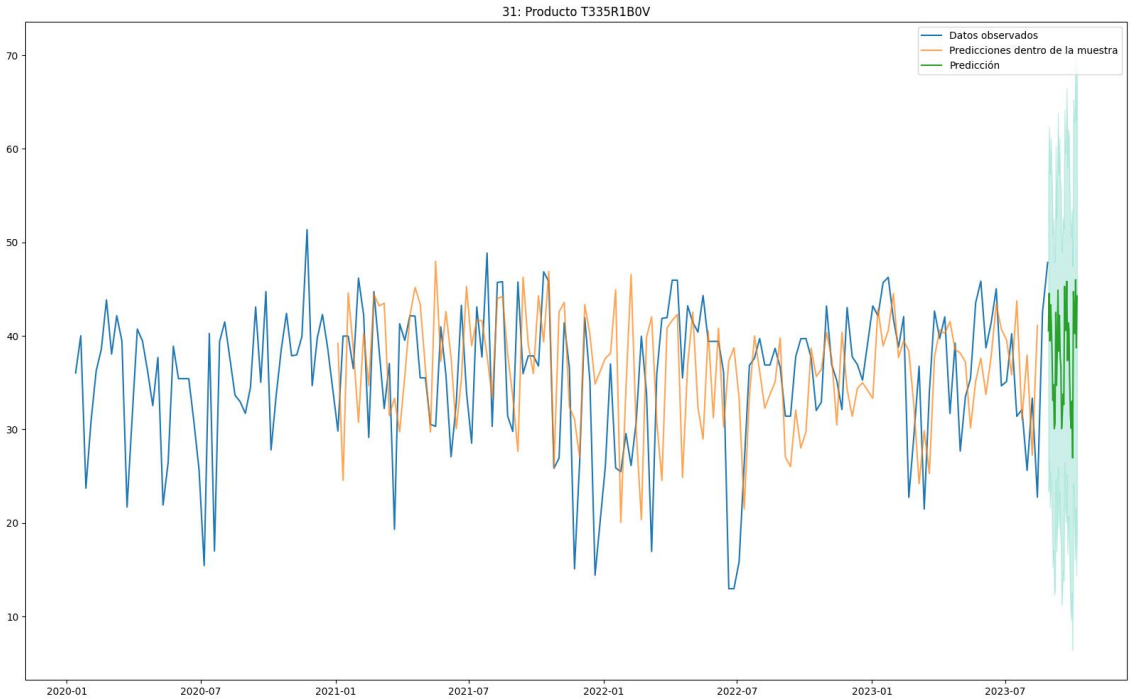

21: Producto T322R1B1V
Datos observados
Predicciones dentro de la muestra
Predicción
55
50
45
40
35
30
25
2020-01
2020-07
2021-01
2021-07
2022-01
2022-07
2023-01
2023-07
2024-01


22: Producto T321R1B1
Datos observados
Predicciones dentro de la muestra
Predicción
80
60
40
20
0
2020-01
2020-07
2021-01
2021-07
2022-01
2022-07
2023-01
2023-07
2024-01

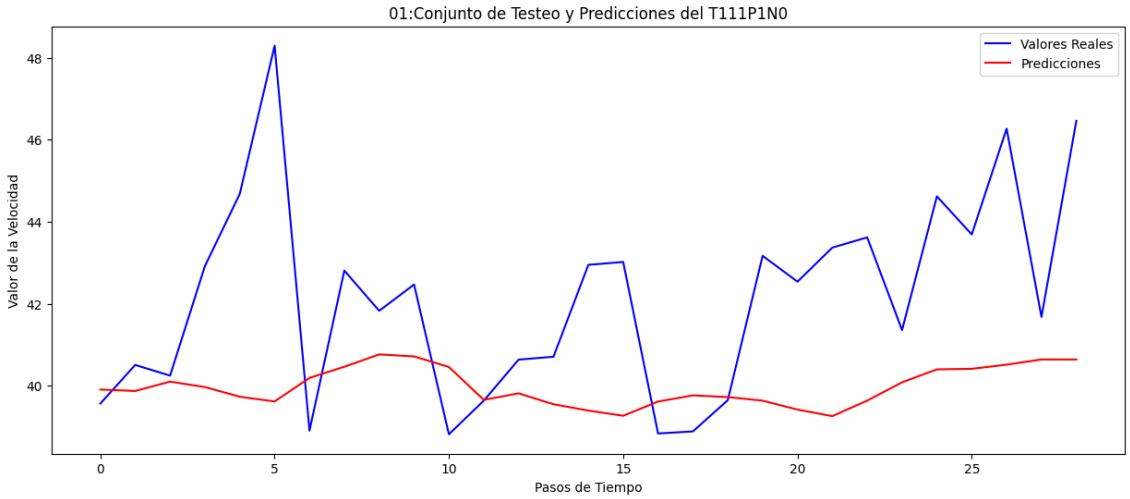

26: Producto M40003KA
Datos observados
Predicciones dentro de la muestra
Predicción
40
30
20
10
0
2020-01
2020-07
2021-01
2021-07
2022-01
2022-07
2023-01
2023-07
2024-01

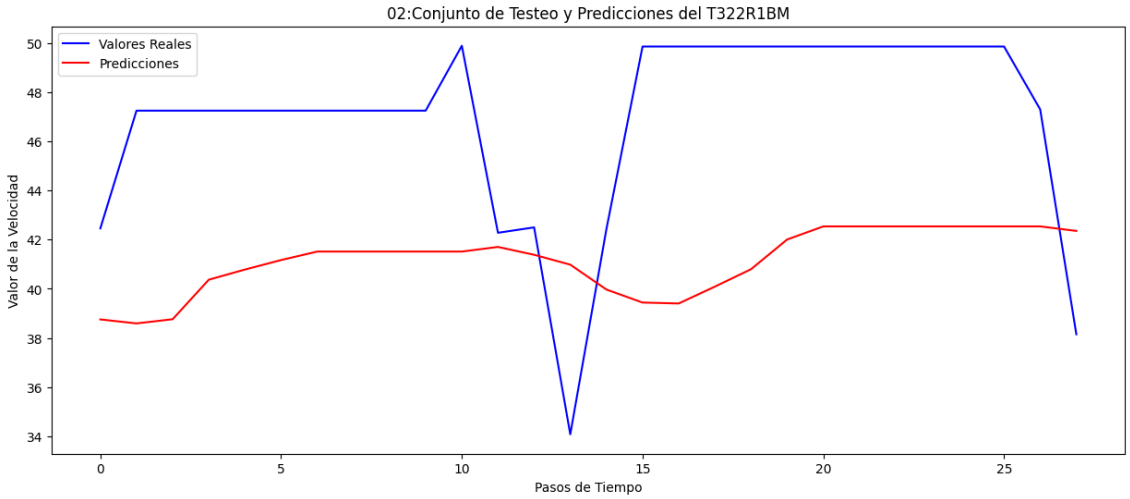

28: Producto T321R1B0
Datos observados
Predicciones dentro de la muestra
Predicción
55
50
45
40
35
30
2020-01
2020-07
2021-01
2021-07
2022-01
2022-07
2023-01
2023-07
2024-01

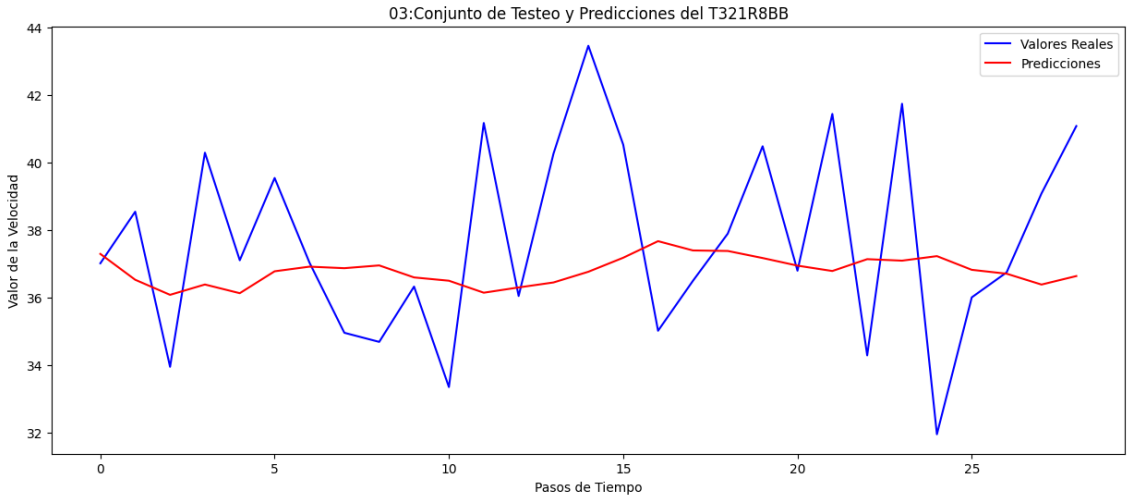

29: Producto T112R7B0W
Datos observados
Predicciones dentro de la muestra
Predicción
70
60
50
40
30
20
10
2020-01
2020-07
2021-01
2021-07
2022-01
2022-07
2023-01
2023-07
2024-01

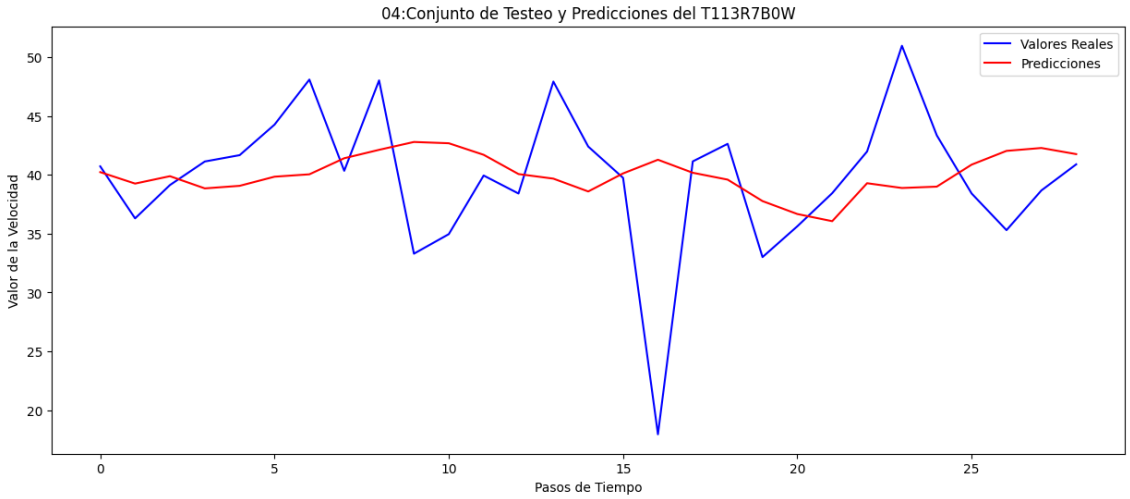

31: Producto T335R1B0V
Datos observados
Predicciones dentro de la muestra
Predicción
70
60
50
40
30
20
10
2020-01
2020-07
2021-01
2021-07
2022-01
2022-07
2023-01
2023-07

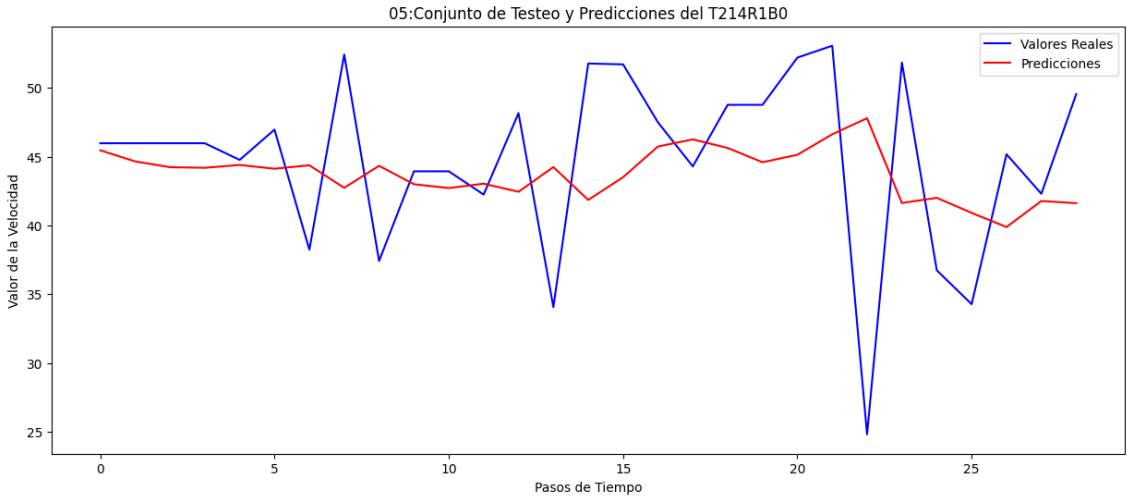
Apéndice 4 Gráficos Modelado LSTM

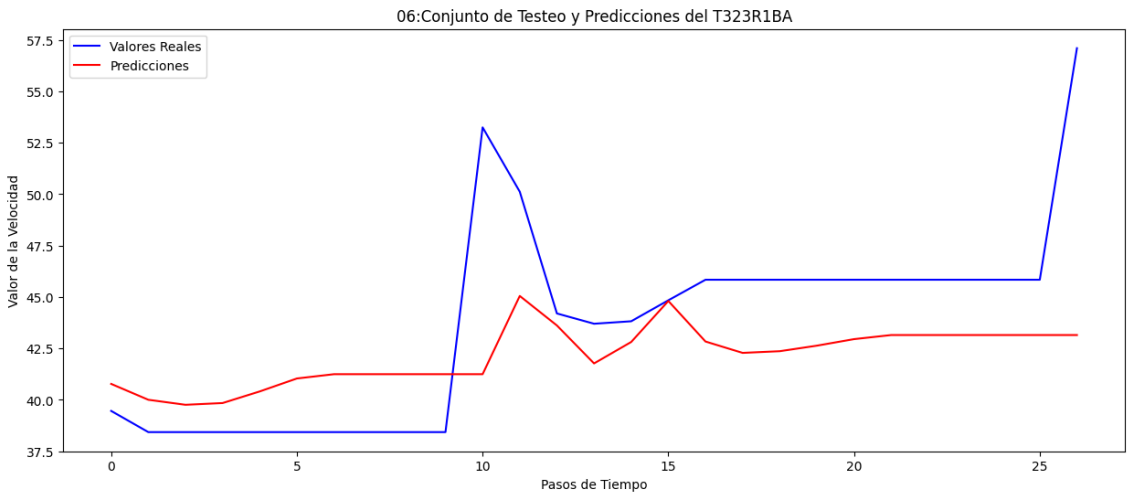


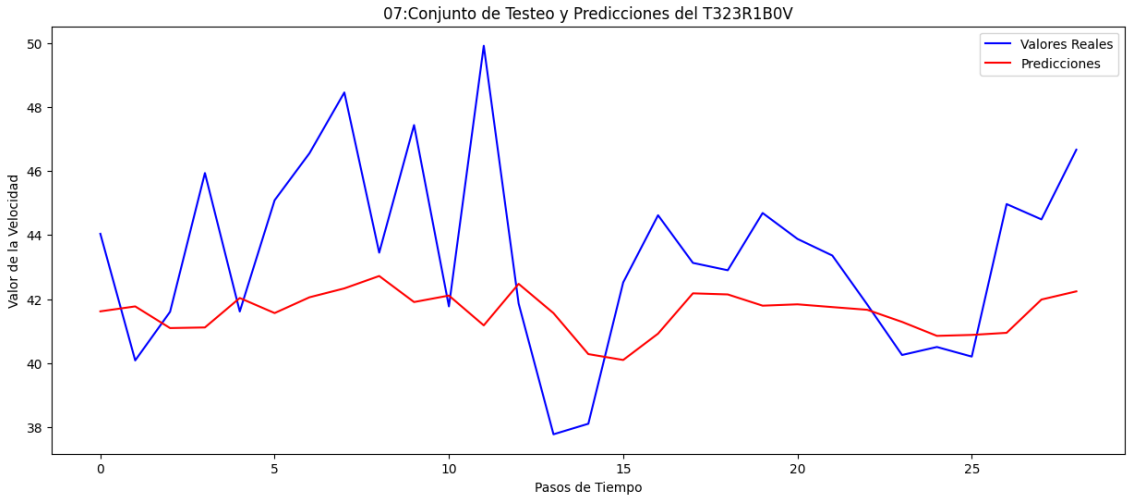


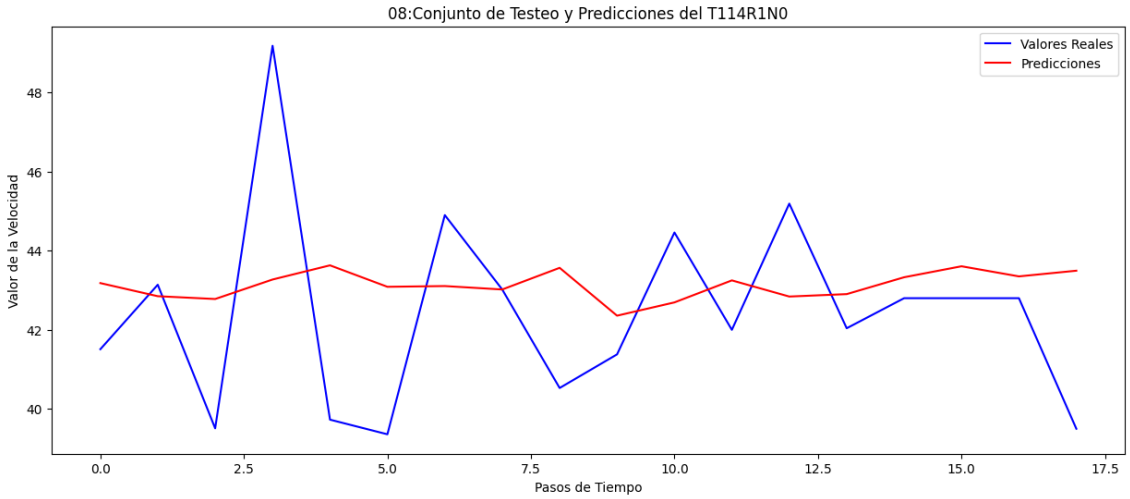


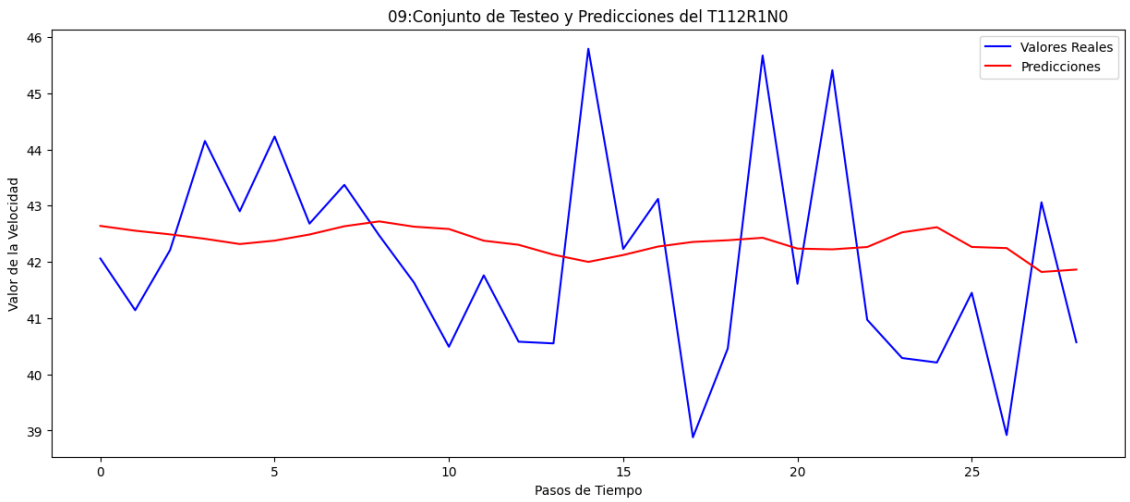

04:Conjunto de Testeo y Predicciones del T113R7B0W
Valores Reales
Predicciones
Valor de la Velocidad
Pasos de Tiempo

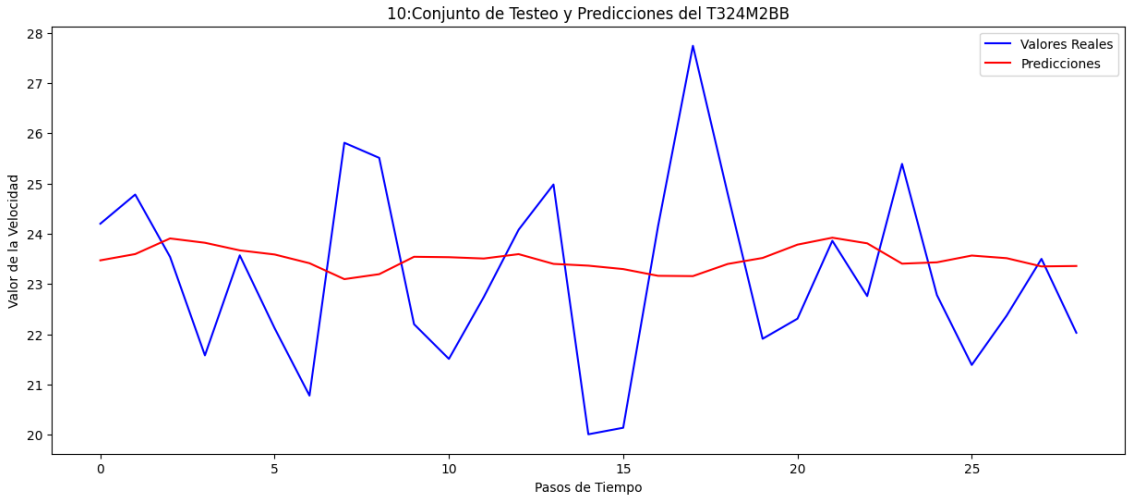

05:Conjunto de Testeo y Predicciones del T214R1B0
Valores Reales
Predicciones
Valor de la Velocidad
Pasos de Tiempo

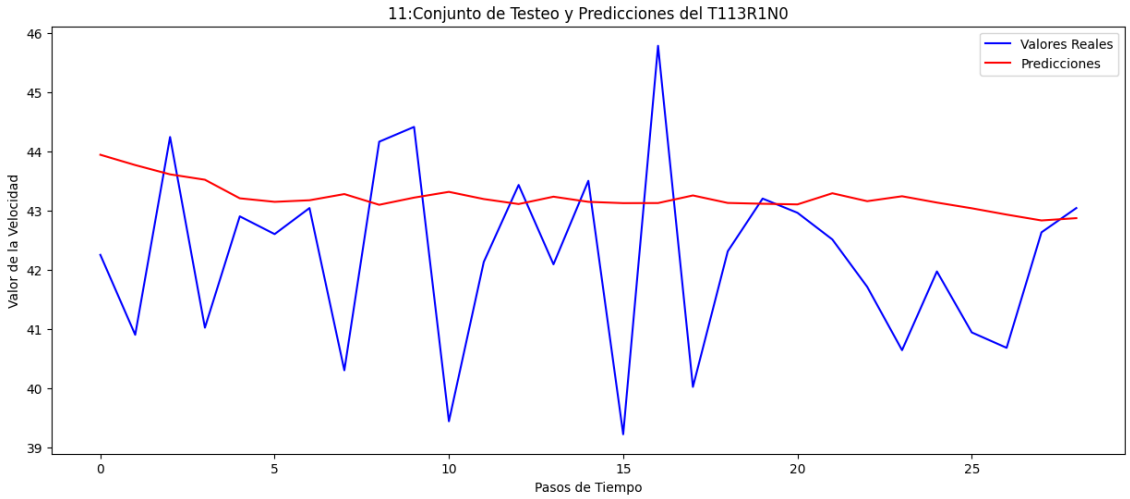

06:Conjunto de Testeo y Predicciones del T323R1BA
Valores Reales
Predicciones
Valor de la Velocidad
Pasos de Tiempo

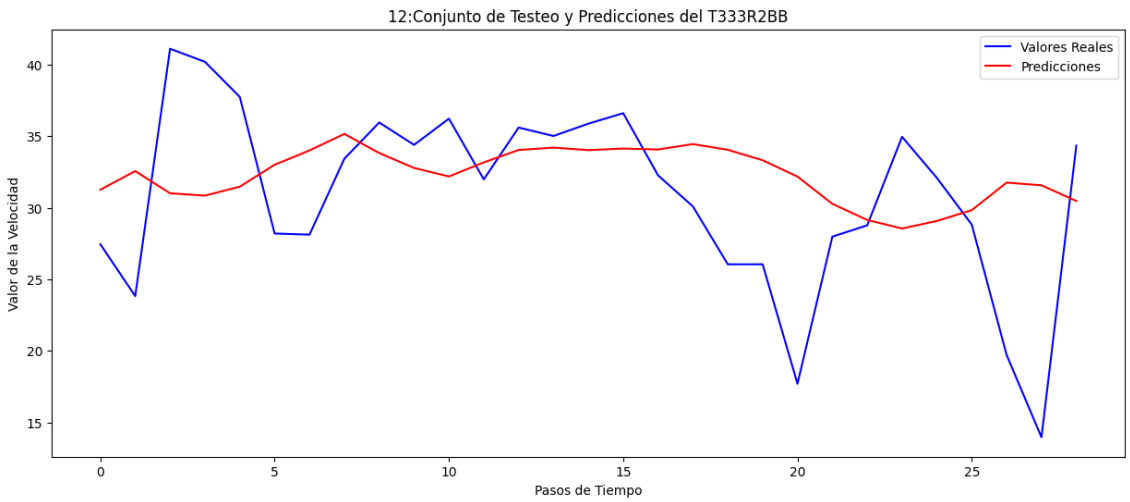

07:Conjunto de Testeo y Predicciones del T323R1B0V
Valores Reales
Predicciones
Valor de la Velocidad
Pasos de Tiempo

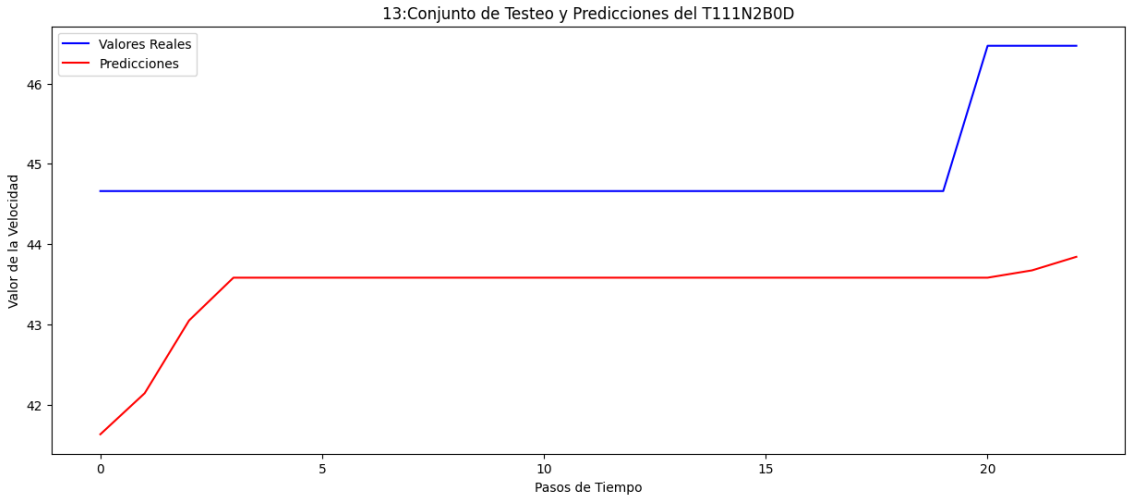

08:Conjunto de Testeo y Predicciones del T114R1N0
Valores Reales
Predicciones
Valor de la Velocidad
Pasos de Tiempo

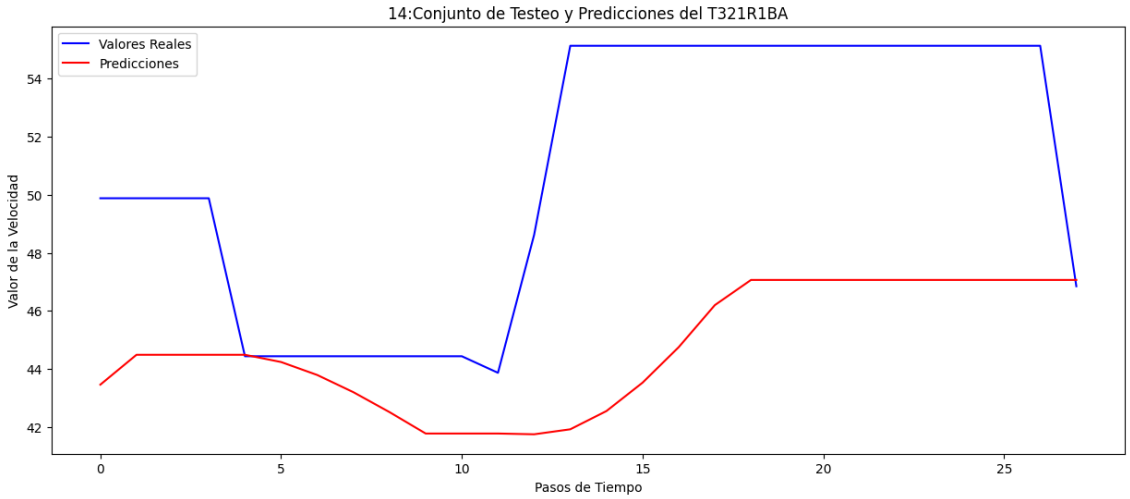

09:Conjunto de Testeo y Predicciones del T112R1N0
Valores Reales
Predicciones
Valor de la Velocidad
Pasos de Tiempo

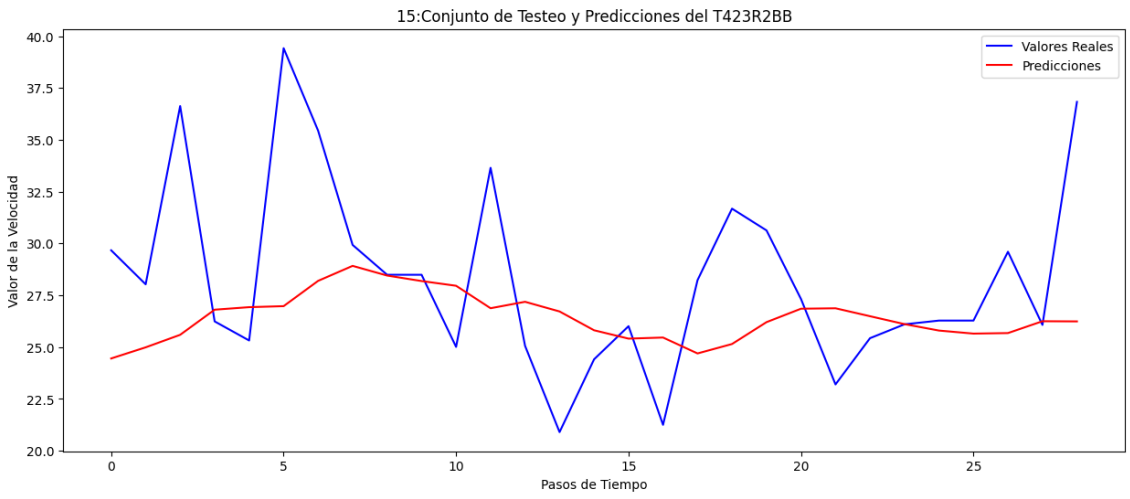

10:Conjunto de Testeo y Predicciones del T324M2BB
Valores Reales
Predicciones
Valor de la Velocidad
Pasos de Tiempo

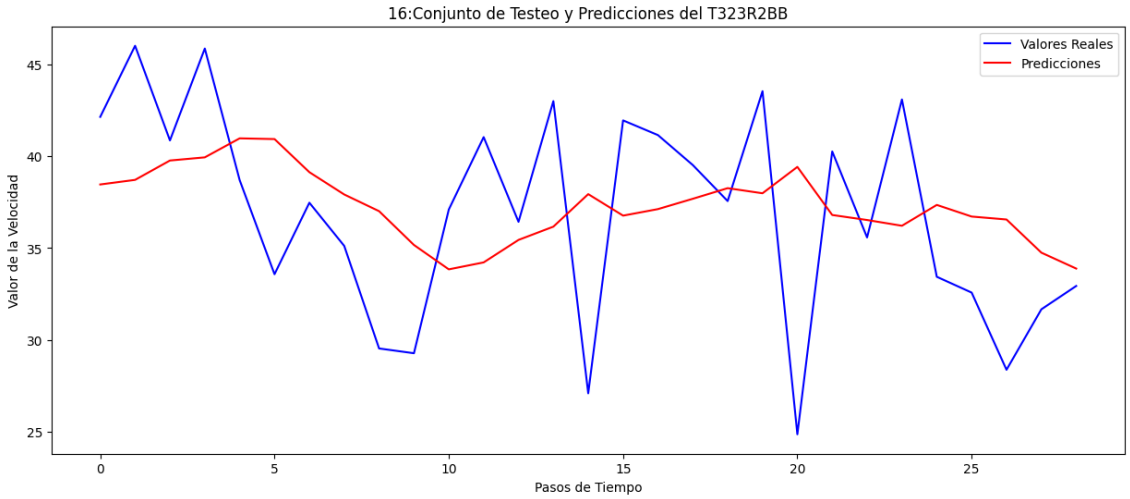

11:Conjunto de Testeo y Predicciones del T113R1N0
Valores Reales
Predicciones
Valor de la Velocidad
Pasos de Tiempo

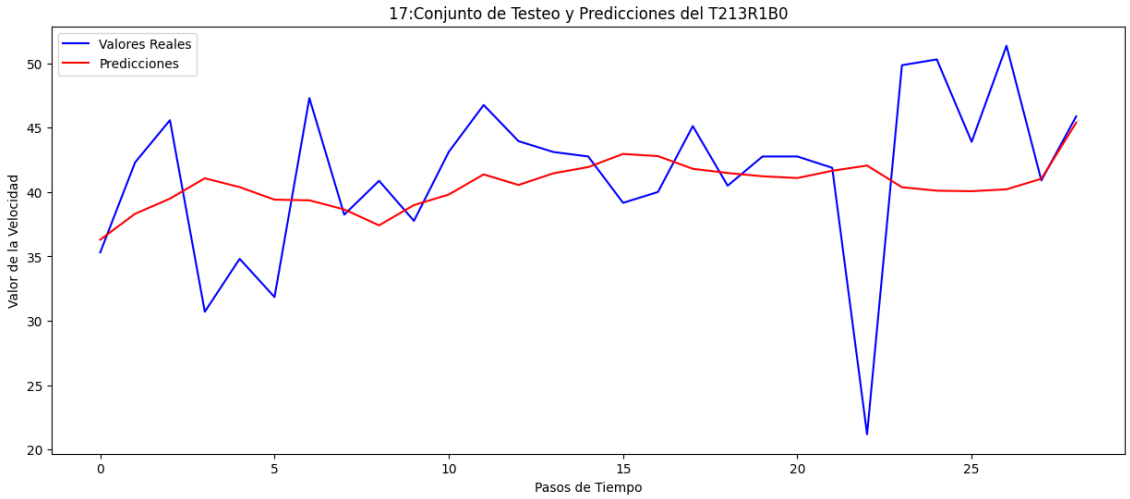

12:Conjunto de Testeo y Predicciones del T333R2BB
Valores Reales
Predicciones
Valor de la Velocidad
Pasos de Tiempo

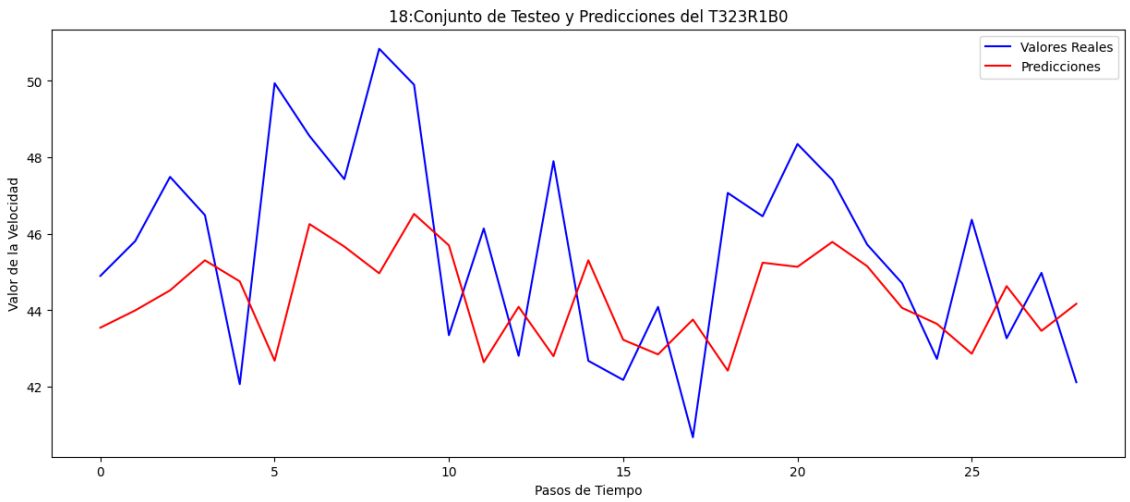

13:Conjunto de Testeo y Predicciones del T111N2B0D
Valores Reales
Predicciones
Valor de la Velocidad
Pasos de Tiempo

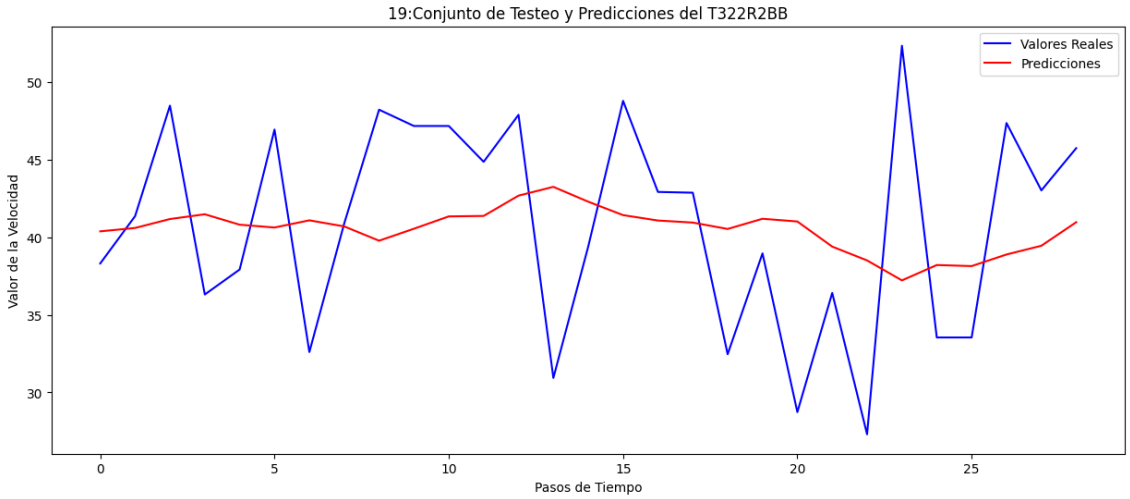

14:Conjunto de Testeo y Predicciones del T321R1BA
Valores Reales
Predicciones
Valor de la Velocidad
Pasos de Tiempo

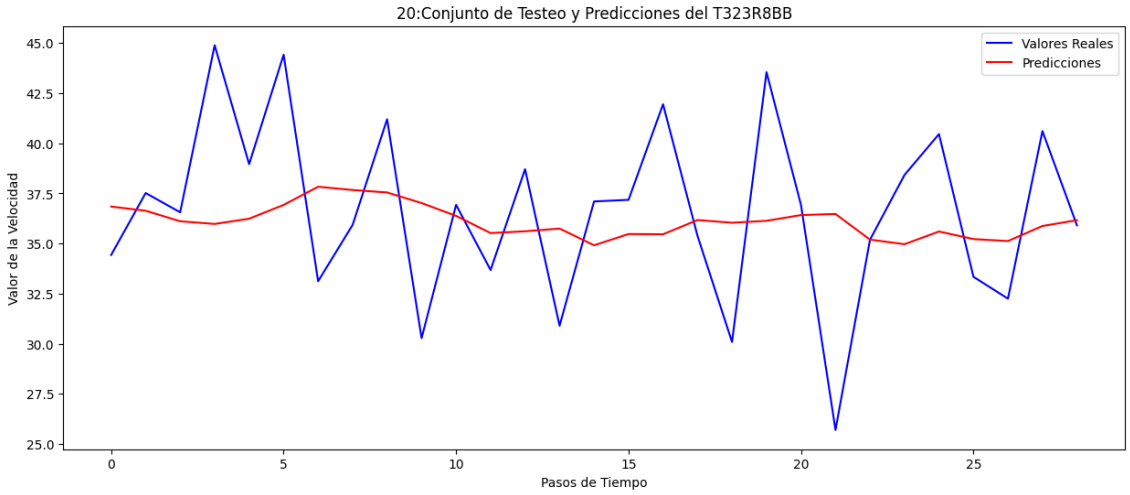

15:Conjunto de Testeo y Predicciones del T423R2BB
Valores Reales
Predicciones
Valor de la Velocidad
Pasos de Tiempo

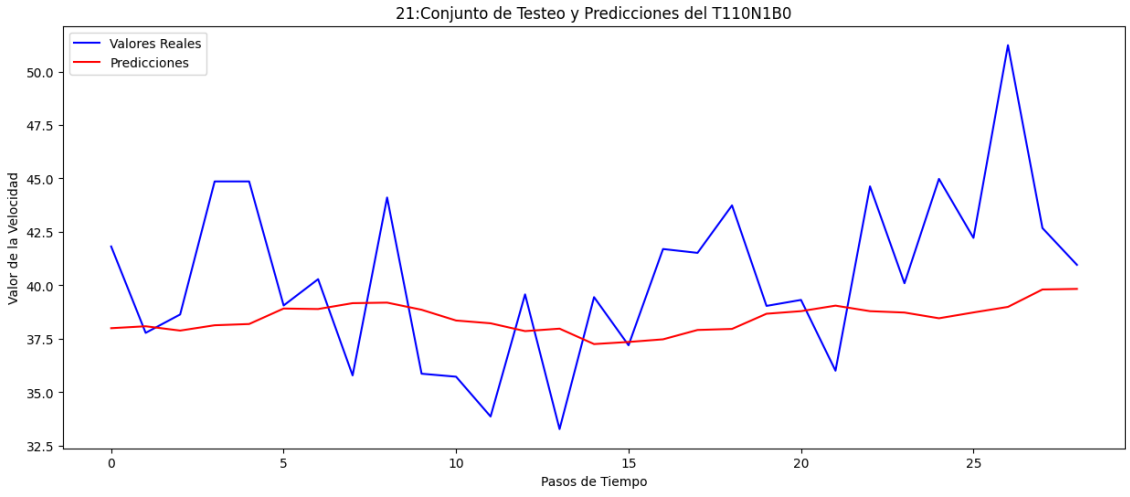

16:Conjunto de Testeo y Predicciones del T323R2BB
Valores Reales
Predicciones
Valor de la Velocidad
Pasos de Tiempo

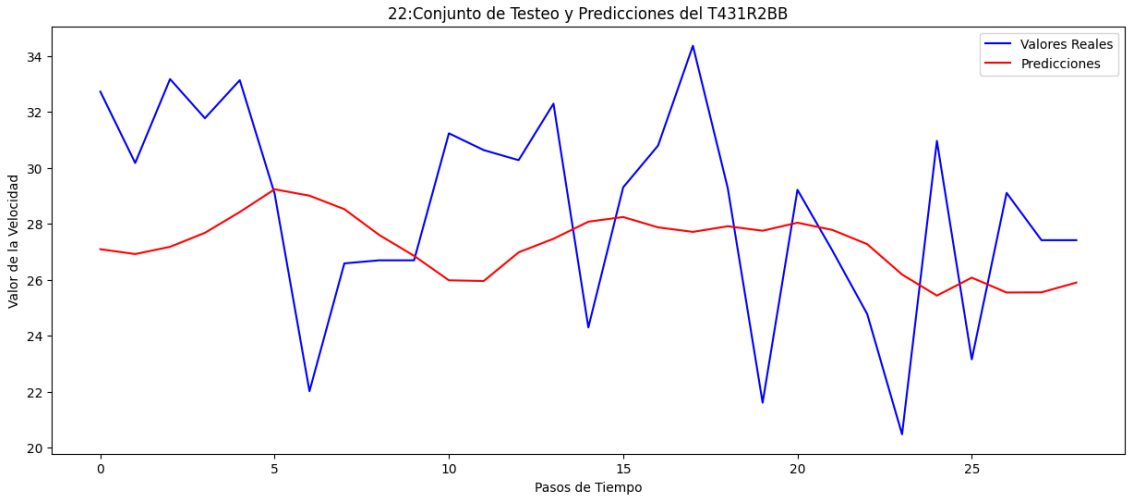

17:Conjunto de Testeo y Predicciones del T213R1B0
Valores Reales
Predicciones
Valor de la Velocidad
Pasos de Tiempo

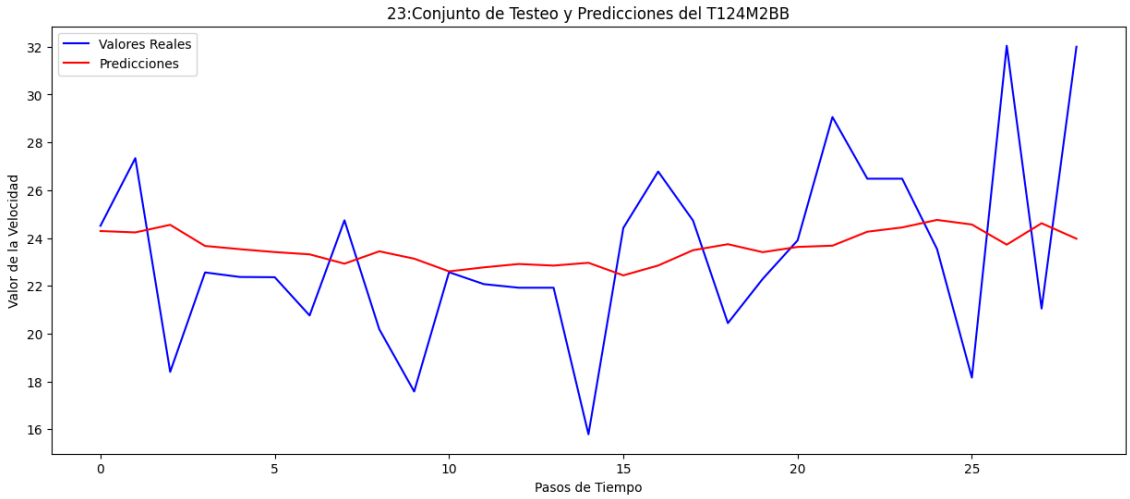

18:Conjunto de Testeo y Predicciones del T323R1B0
Valores Reales
Predicciones
Valor de la Velocidad
Pasos de Tiempo

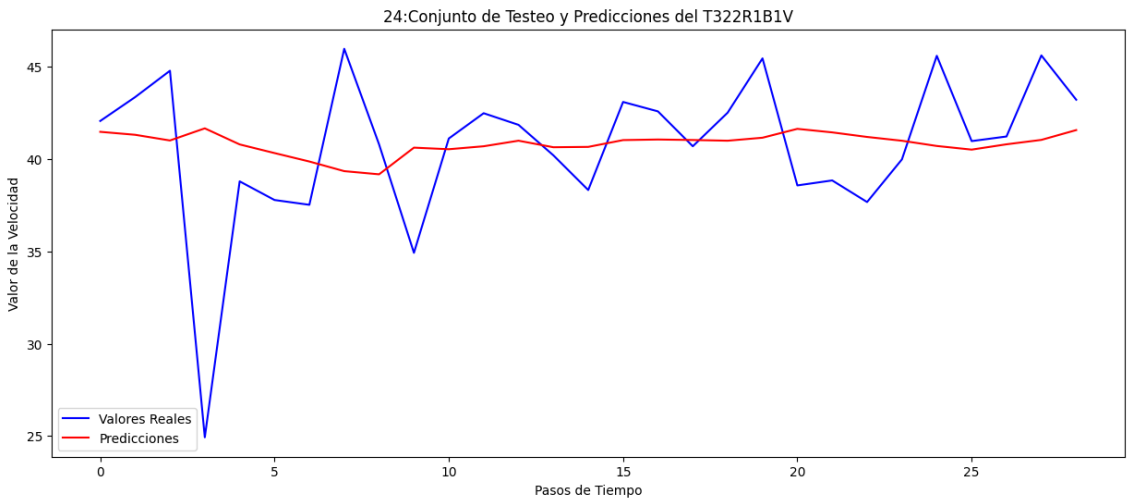

19:Conjunto de Testeo y Predicciones del T322R2BB
Valores Reales
Predicciones
Valor de la Velocidad
Pasos de Tiempo

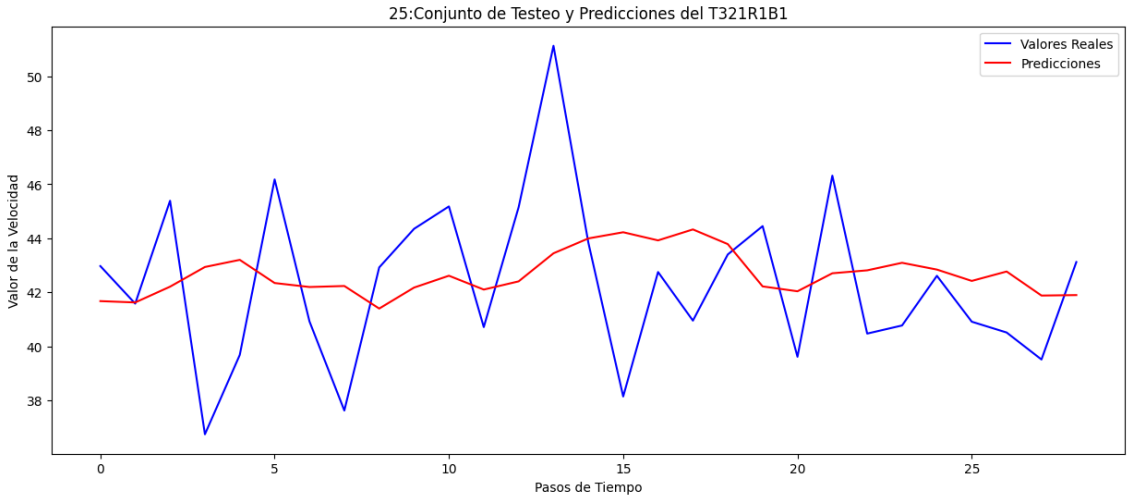

20:Conjunto de Testeo y Predicciones del T323R8BB
Valores Reales
Predicciones
Valor de la Velocidad
Pasos de Tiempo

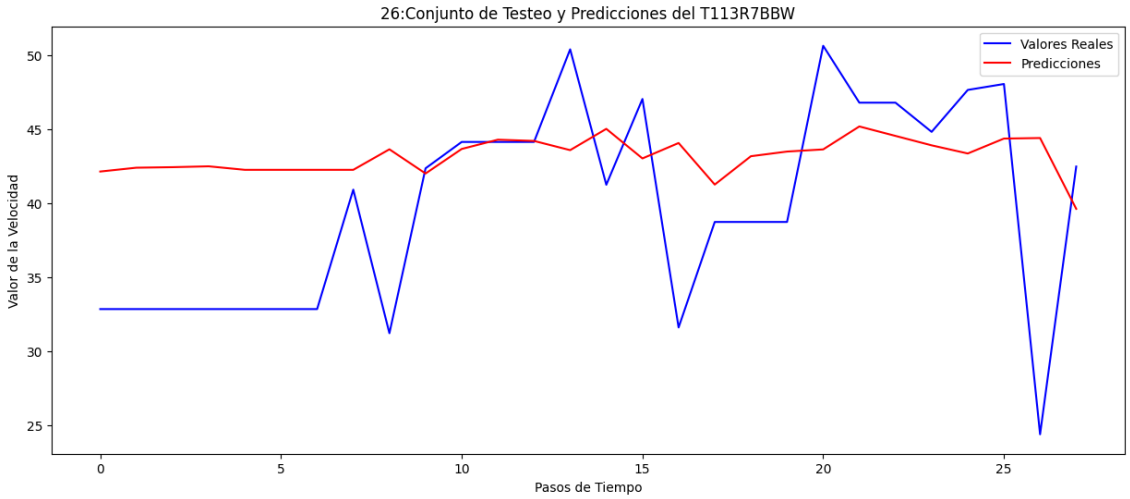

21:Conjunto de Testeo y Predicciones del T110N1B0
Valores Reales
Predicciones
Valor de la Velocidad
Pasos de Tiempo

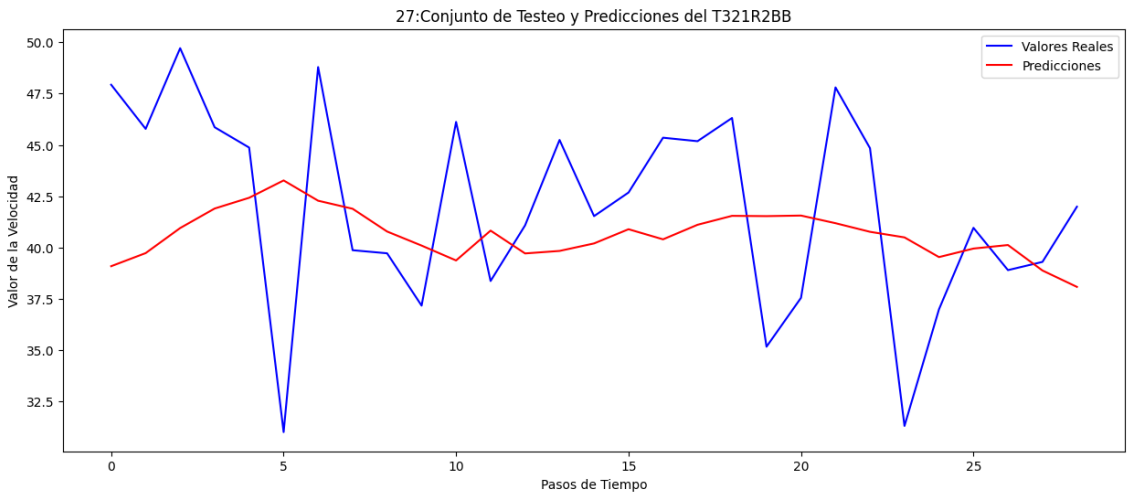

22:Conjunto de Testeo y Predicciones del T431R2BB
Valores Reales
Predicciones
Valor de la Velocidad
Pasos de Tiempo

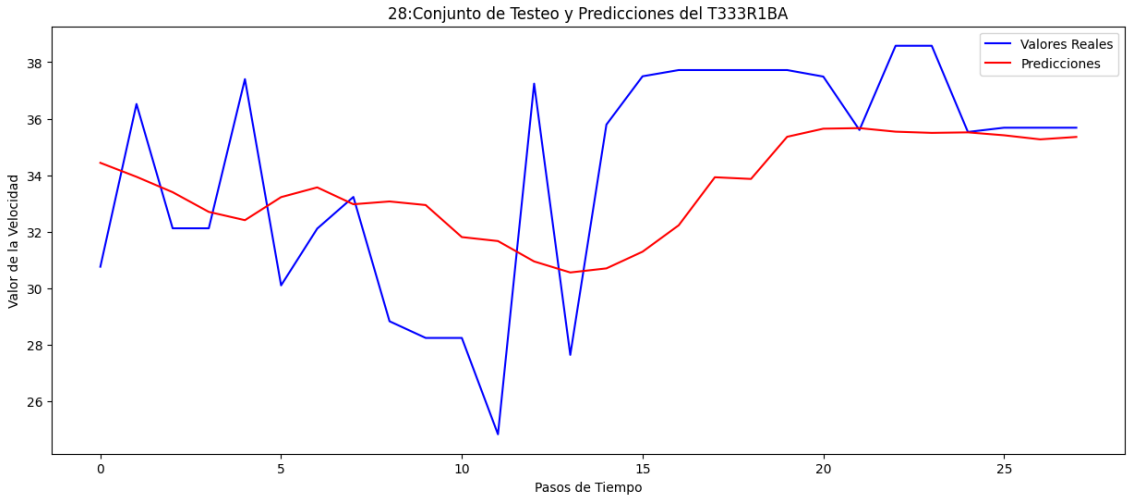

23:Conjunto de Testeo y Predicciones del T124M2BB
Valores Reales
Predicciones
Valor de la Velocidad
Pasos de Tiempo

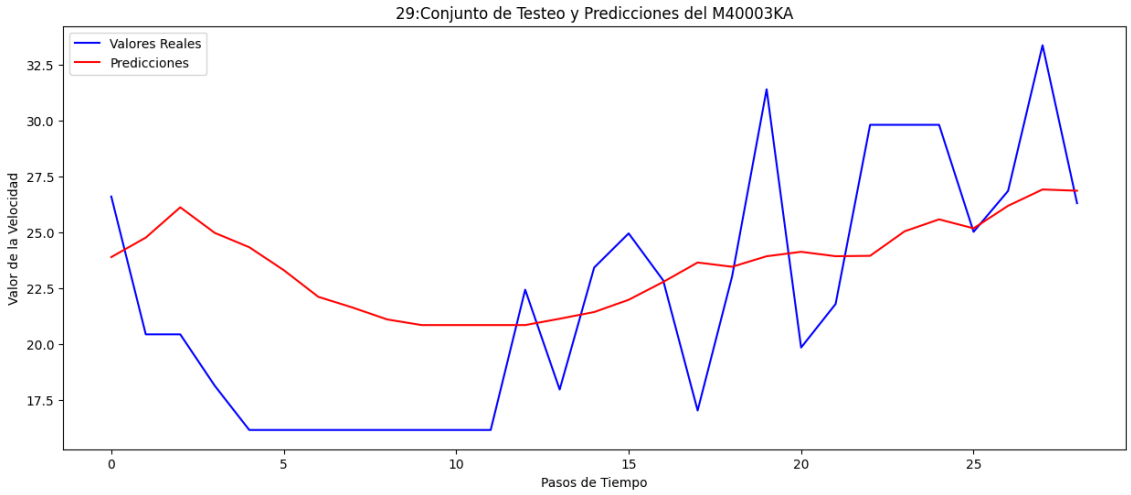

24:Conjunto de Testeo y Predicciones del T322R1B1V
Valores Reales
Predicciones
Valor de la Velocidad
Pasos de Tiempo

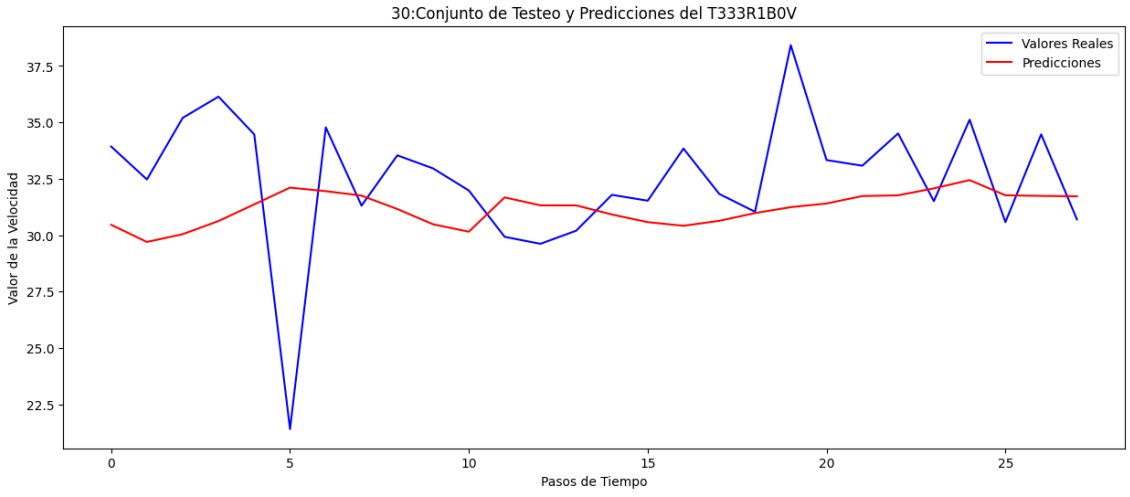

25:Conjunto de Testeo y Predicciones del T321R1B1
Valores Reales
Predicciones
Valor de la Velocidad
Pasos de Tiempo

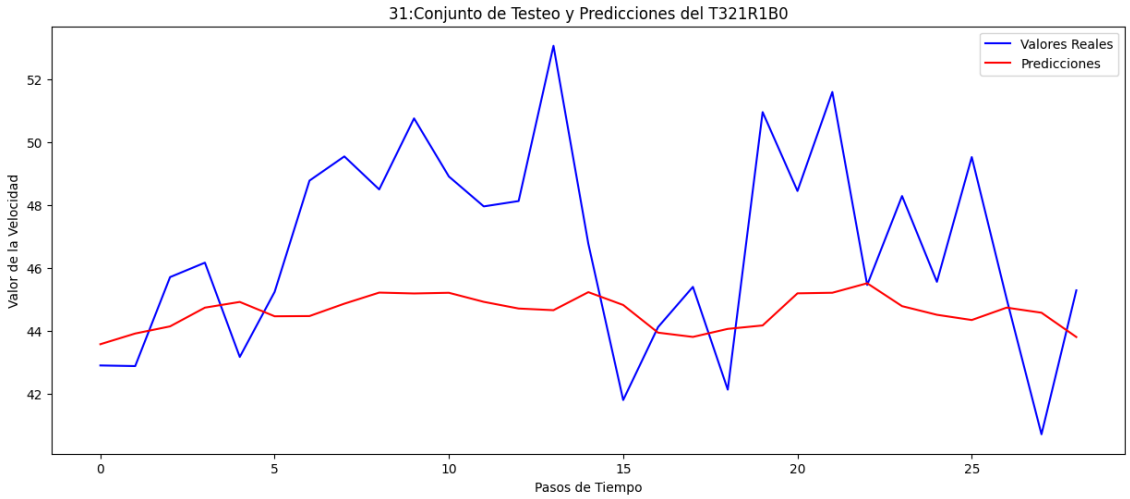

26:Conjunto de Testeo y Predicciones del T113R7BBW
Valores Reales
Predicciones
Valor de la Velocidad
Pasos de Tiempo

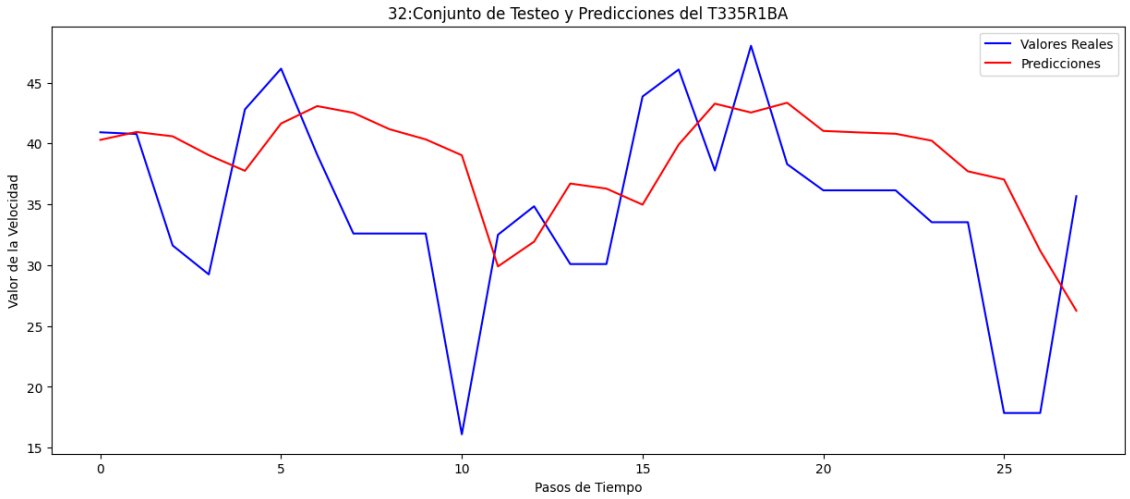

27:Conjunto de Testeo y Predicciones del T321R2BB
Valores Reales
Predicciones
Valor de la Velocidad
Pasos de Tiempo

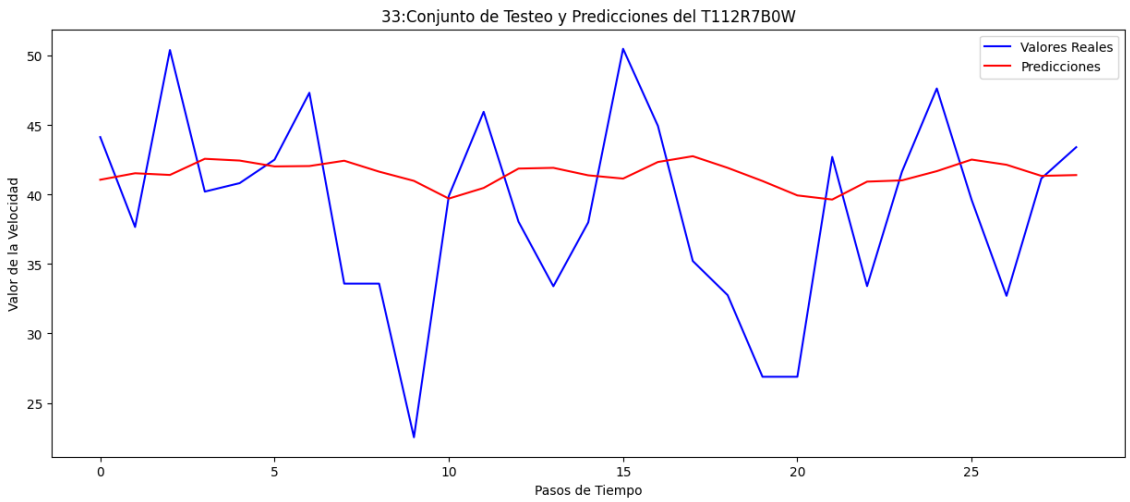

28:Conjunto de Testeo y Predicciones del T333R1BA
Valores Reales
Predicciones
Valor de la Velocidad
Pasos de Tiempo

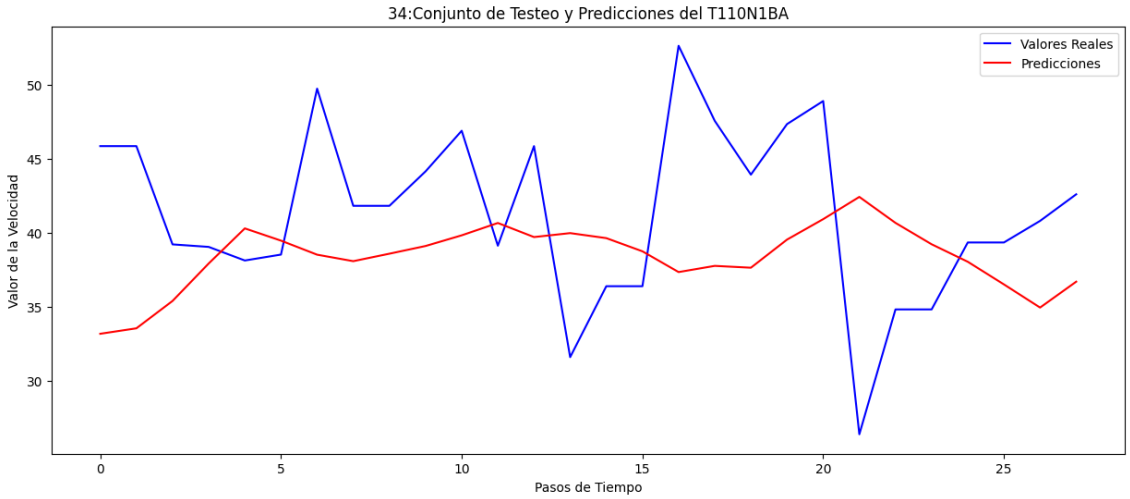

29:Conjunto de Testeo y Predicciones del M40003KA
Valores Reales
Predicciones
Valor de la Velocidad
Pasos de Tiempo

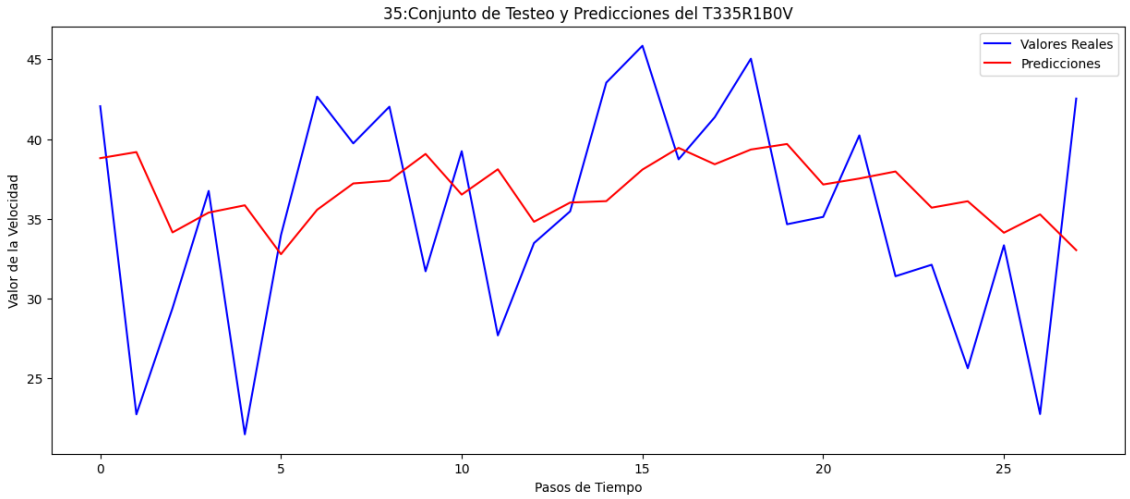

30:Conjunto de Testeo y Predicciones del T333R1B0V
Valores Reales
Predicciones
Valor de la Velocidad
Pasos de Tiempo

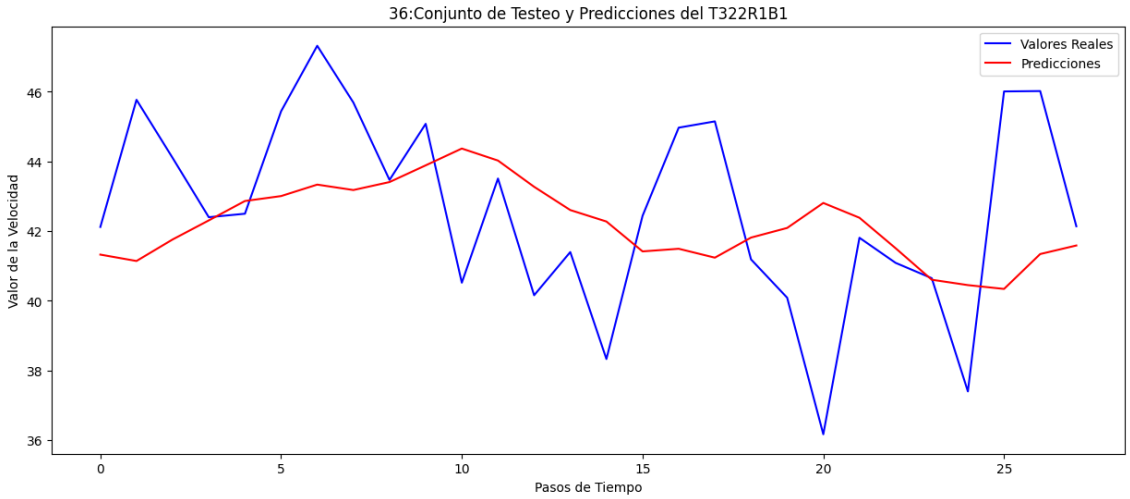

31:Conjunto de Testeo y Predicciones del T321R1B0
Valores Reales
Predicciones
Valor de la Velocidad
Pasos de Tiempo

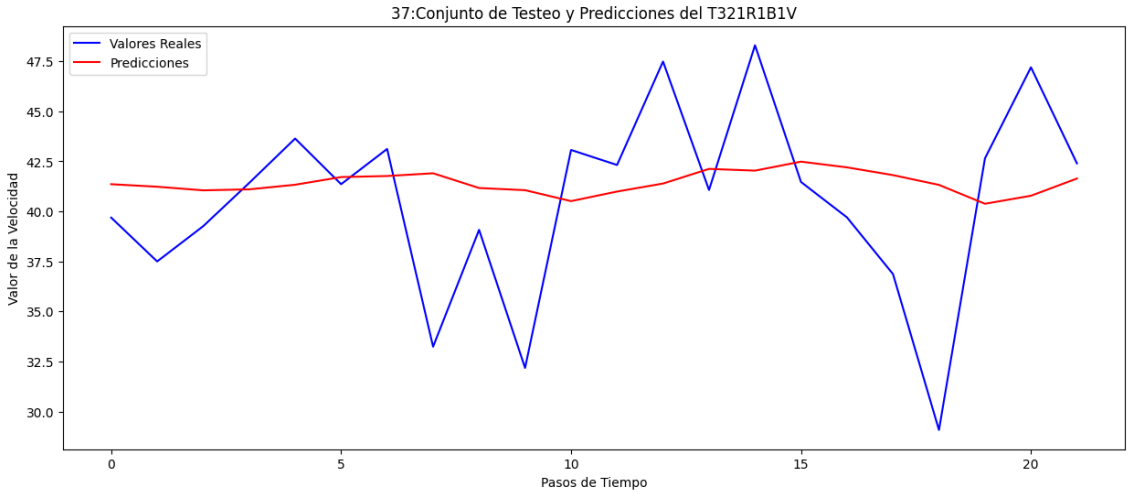

32:Conjunto de Testeo y Predicciones del T335R1BA
Valores Reales
Predicciones
Valor de la Velocidad
Pasos de Tiempo


33:Conjunto de Testeo y Predicciones del T112R7B0W
Valores Reales
Predicciones
Valor de la Velocidad
Pasos de Tiempo


34:Conjunto de Testeo y Predicciones del T110N1BA
Valores Reales
Predicciones
Valor de la Velocidad
Pasos de Tiempo
50
45
40
35
30
0
5
10
15
20
25


35:Conjunto de Testeo y Predicciones del T335R1B0V
Valores Reales
Predicciones
Valor de la Velocidad
Pasos de Tiempo
45
40
35
30
25
0
5
10
15
20
25





Apéndice 5 Código fuente de creación del Pipeline

```
import pandas as pd
import json
import boto3
import pathlib
import io
import sagemaker
from sagemaker.workflow.parameters import ParameterInteger, ParameterString
from sagemaker.workflow.steps import ProcessingStep, TrainingStep, CreateModelStep
from sagemaker.workflow.step_collections import RegisterModel
from sagemaker.workflow.pipeline import Pipeline
from sagemaker.sklearn.processing import SKLearnProcessor
from sagemaker.estimator import Estimator
from sagemaker.inputs import TrainingInput
from sagemaker.model import Model
```

```
from sagemaker import get_execution_role
from sagemaker.processing import ProcessingInput, ProcessingOutput
from sagemaker import image_uris
from sagemaker.processing import ScriptProcessor
from sagemaker.tensorflow import TensorFlow
from sagemaker.s3 import S3Uploader
from sagemaker.inputs import CreateModelInput
from sagemaker.workflow.properties import PropertyFile
from sagemaker.workflow.lambda_step import (
    LambdaStep,
    LambdaOutput,
    LambdaOutputTypeEnum,
)
from sagemaker.processing import ScriptProcessor
from sagemaker.workflow.steps import ProcessingStep

from sagemaker.workflow.lambda_step import Lambda
from sagemaker.processing import ScriptProcessor
from sagemaker.workflow.steps import ProcessingStep
from sagemaker.workflow.parameters import ParameterString, ParameterInteger
from sagemaker.workflow.pipeline import Pipeline
import sagemaker

sess = sagemaker.Session()
sagemaker_role = sagemaker.get_execution_role()
region = sess.boto_region_name

image_uri = image_uris.retrieve(
    framework="tensorflow",
    region="us-east-1",
    version="2.3.1",
    py_version="py37",
    instance_type="ml.m5.xlarge",
    image_scope="training"
)

input_data = ParameterString(name="InputData", default_value="s3://sagemaker-us-east-1-
221551376986/data/PARAS_DE_BACHEO_2020_2023.csv")
model_approval_status = ParameterString(name="ModelApprovalStatus", default_value="PendingManualApproval")

script_processor = ScriptProcessor(
    command=['python3'],
    image_uri='221551376986.dkr.ecr.us-east-1.amazonaws.com/pipeline1:latest',
    role=sagemaker_role,
    instance_count=1,
    instance_type='ml.m5.xlarge',
```

```
)

processing_step = ProcessingStep(
    name="PasoDeProcesamiento",
    processor=script_processor,
    inputs=[ProcessingInput(source=input_data, destination='/opt/ml/processing/input')],
    outputs=[ProcessingOutput(output_name='train', source='/opt/ml/processing/train',
                              destination='s3://sagemaker-us-east-1-221551376986/data_procesada/')],
    code='procesamiento.py'
)

estimator = Estimator(
    entry_point='entrenamiento.py',
    image_uri='221551376986.dkr.ecr.us-east-1.amazonaws.com/pipeline1-training-step-v3:latest',
    role=sagemaker_role,
    instance_count=1,
    instance_type='ml.m5.xlarge',
    output_path='s3://sagemaker-us-east-1-221551376986/modelosTF/'
)

training_input =
TrainingInput(s3_data=processing_step.properties.ProcessingOutputConfig.Outputs['train'].S3Output.S3Uri)

training_step = TrainingStep(
    name="PasoDeEntrenamiento",
    estimator=estimator,
    inputs={'train': training_input}

)

script_decompress = ScriptProcessor(
    command=['python3'],
    image_uri='763104351884.dkr.ecr.us-east-1.amazonaws.com/tensorflow-inference:2.3.0-cpu-py37-
ubuntu18.04',
    role=sagemaker_role,
    instance_count=1,
    instance_type='ml.m5.xlarge',
)

decompress_step = ProcessingStep(
    name="PasoDeDescompresion",
    processor=script_decompress,
    inputs=[
        ProcessingInput(
            source=training_step.properties.ModelArtifacts.S3ModelArtifacts,
            destination='/opt/ml/processing/model'
```

```
        )
    ],
    outputs=[
        ProcessingOutput(
            output_name='models',
            source='/opt/ml/processing/extracted_models',
            destination='s3://sagemaker-us-east-1-221551376986/extracted_modelsTF/'
        )
    ],
    code='descompresion.py'
)

sagemaker_role = sagemaker.get_execution_role()
region = sagemaker.Session().boto_region_name
# Definición de parámetros
bucket_name_param = ParameterString(name="BucketName", default_value="sagemaker-us-east-1-221551376986")
s3_model_directory_param = ParameterString(name="S3ModelDirectory", default_value="extracted_modelsTF/")
role_param = ParameterString(name="Role", default_value=sagemaker_role)
framework_version_param = ParameterString(name="FrameworkVersion", default_value="2.8.0")
instance_type_param = ParameterString(name="InstanceType", default_value="ml.m5.large")
initial_instance_count_param = ParameterInteger(name="InitialInstanceCount", default_value=1)
region_param = ParameterString(name="Region", default_value=region)

script_processor_deploy = ScriptProcessor(
    command=["python3"],
    image_uri='221551376986.dkr.ecr.us-east-1.amazonaws.com/deploy-models-v1:latest',
    role=sagemaker_role,
    instance_count=1,
    instance_type="ml.m5.large"
)

deploying_step = ProcessingStep(
    name="PasoDeDespiegue",
    processor=script_processor_deploy,
    inputs=[
        ProcessingInput(
            source=decompress_step.properties.ProcessingOutputConfig.Outputs['models'].S3Output.S3Uri,
            destination='/opt/ml/processing/model'
        )
    ],
    outputs=[],
    job_arguments=[
        "--bucket-name", bucket_name_param,
        "--s3-model-directory", s3_model_directory_param,
        "--role", role_param,
        "--framework-version", framework_version_param,
```

```
        "--region", region_param
    ],
    code="despliegue.py"
)


# Creacion del pipeline
pipeline = Pipeline(
    name="PipelineV2",
    parameters=[bucket_name_param, s3_model_directory_param, role_param, framework_version_param,
instance_type_param, initial_instance_count_param, region_param, input_data, model_approval_status],
    steps=[
        processing_step,
        training_step,
        decompress_step,
        deploying_step
    ]
)

# Ejecutar el pipeline
pipeline.upsert(role_arn=sagemaker_role)
execution = pipeline.start()
```

Apéndice 6 Código Fuente del script procesamiento.py

```
import glob
import numpy as np
import os
from sklearn.preprocessing import StandardScaler

import pandas as pd
from scipy import stats
from pandas.tseries.offsets import DateOffset
from scipy.stats import normaltest
from statsmodels.stats.diagnostic import acorr_ljungbox
import warnings

#Seccion de funciones
def valor_p_ruido_blanco(df):
    es_ruido_blanco = False
    resultado = acorr_ljungbox(df['Tm/h'], lags=[10])  # Aquí 10 es el número de retrasos
    p_valor = resultado['lb_pvalue'].iloc[0]  # Segunda posición contiene los p-valores
    return p_valor
```

```python
def es_ruido_blanco(df):
    es_ruido_blanco = False
    # Aplicar la prueba de Ljung-Box
    resultado = acorr_ljungbox(df['Tm/h'], lags=[10])  # Aquí 10 es el número de retrasos
    # Interpretar el resultado
    p_valor = resultado['lb_pvalue'].iloc[0]  # Segunda posición contiene los p-valores
    if p_valor > 0.05:
        es_ruido_blanco = True
    return es_ruido_blanco

def verificar_ruido_blanco(df, producto):
    df_filtrado = df.loc[df['PRODUCTO'] == producto,['Tm/h']]
    # Aplicar la prueba de Ljung-Box
    resultado = acorr_ljungbox(df_filtrado['Tm/h'], lags=[5])  # Aquí 10 es el número de retrasos

    # Interpretar el resultado
    p_valor = resultado['lb_pvalue'].iloc[0]  # Segunda posición contiene los p-valores
    if p_valor > 0.05:
        print("Probablemente es ruido blanco")
    else:
        print("Probablemente no es ruido blanco")
    plot_acf(df_filtrado['Tm/h'])
    plt.show()

def completar_datos_por_anio(anio, dict_prod_vel_semana):
    df_completo = pd.DataFrame()
    df_concatenado = pd.DataFrame()
    for clave in dict_prod_vel_semana:
        df = dict_prod_vel_semana[clave]
        df = df.loc[ df['ANIO_AJUSTADO'] == anio,: ]
        df_completo = llenar_semanas_faltantes(df)
        df_completo = imputar_valores_semana(df_completo)
        df_completo['PRODUCTO'] = clave
        df_concatenado = pd.concat([df_completo, df_concatenado], ignore_index=True)
    return df_concatenado

def imputar_valores_semana(df):
    df_ffill = df.copy()
    df_ffill['Tm/h'] = df_ffill['Tm/h'].ffill()
    df_ffill['ANIO_AJUSTADO'] = df_ffill['ANIO_AJUSTADO'].ffill()
    return df_ffill

def llenar_semanas_faltantes(df):
    # Elimina filas donde 'SEMANA' es NaN
    df_limpio = df.dropna(subset=['SEMANA'])
```

```
    # Continúa con tu lógica original
    min_semana = int(np.amin(df_limpio['SEMANA']))
    max_semana = int(np.amax(df_limpio['SEMANA']))
    print(f"min_semana: {min_semana}")
    print(f"max_semana: {max_semana}")
    semanas_completas = range(min_semana, max_semana + 1)
    df_limpio.sort_values(by='SEMANA', inplace=True)
    df_completo = df_limpio.set_index('SEMANA').reindex(semanas_completas).reset_index()
    return df_completo


def consolida_por_semana(df):
    df_agrupado = df.groupby(['ANIO_AJUSTADO', 'SEMANA'])['Tm/h'].mean().reset_index()
    df_agrupado = pd.DataFrame(df_agrupado)
    df_agrupado['Tm/h'] = round(df_agrupado['Tm/h'], 2)
    return df_agrupado

mes_a_numero = {'ENE': '01','FEB': '02','MAR': '03','ABR': '04','MAY': '05','JUN': '06','JUL': '07','AGO':
'08','SEP': '09','OCT': '10','NOV': '11','DIC': '12'}
columna_analizada = 'Tm/h'

if __name__=='__main__':

    input_files = glob.glob('{}/*.npy'.format('/opt/ml/processing/input'))

    df = pd.read_csv('/opt/ml/processing/input/PARAS_DE_BACHEO_2020_2023.csv', delimiter=";")
    df.dropna(inplace=True)

    df['MES'] = df['MES'].replace(mes_a_numero)
    df['MES'] = df['MES'].astype(int)
    df['ANIO'] = df['ANIO'].astype(int)
    df['COD_ALIM'] = df['COD_ALIM'].astype(str)

    df['Tm/h'] = df['Tm/h _REP'].astype(str).str.replace(',', '.').astype(float)

    df2 = pd.DataFrame({'Ho':[]})
    df2['Ho'] = df['Ho'].astype(str).str.split('.').str[0]
    df2 = pd.DataFrame({'Hf':[]})
    df2['Hf'] = df['Hf'].astype(str).str.split('.').str[0]

    df['Ho_'] = df['Ho'].astype(str).str.split('.').str[0]
    df['Hf_'] = df['Hf'].astype(str).str.split('.').str[0]
    df['Fecha'] = df['DIA'].astype(str) + '-' + df['MES'].astype(str) + '-' + df['ANIO'].astype(str)
    df['f_inicio'] = df['DIA'].astype(str) + '-' + df['MES'].astype(str) + '-' + df['ANIO'].astype(str) +'
'+ df['Ho_'].astype(str)
```

```
    df['f_fin'] = df['DIA'].astype(str) + '-' + df['MES'].astype(str) + '-' + df['ANIO'].astype(str) +' '+
df['Hf_'].astype(str)
    df['Fecha'] = pd.to_datetime(df['Fecha'], format='%d-%m-%Y', errors='coerce')
    df['f_inicio'] =  pd.to_datetime(df['f_inicio'], format='%d-%m-%Y %H:%M:%S', errors='coerce')
    df['f_fin'] =  pd.to_datetime(df['f_fin'], format='%d-%m-%Y %H:%M:%S', errors='coerce')
    df['Anio_Semana'] = df['Fecha'].dt.strftime('%G-%V')
    df['ANIO_AJUSTADO'] = df['Fecha'].dt.strftime('%G').astype(int)
    df['SEMANA'] = df['Fecha'].dt.strftime('%V').astype(int)
    df['Tm/h'] = round(df['Tm/h'], 2)
    df['COD_ALIM'] = df['COD_ALIM'].astype(str)

    #vuelvo a eliminar por si acaso se produjeron NaT en la transformación de las fechas
    df.dropna(inplace=True)

    df =
df.loc[:,['Fecha','ANIO','MES','DIA','LOTE','COD_ALIM','Tm/h','f_inicio','f_fin','Anio_Semana','ANIO_AJUST
ADO','SEMANA']]

    #Eliminación de datos atípicos
    df = df.loc[df[columna_analizada]!=0,:]
    df.loc[ :,'Fecha'] = pd.to_datetime(df['Fecha'], format='%Y-%m-%d')
    df.loc[ :,'f_inicio'] =  pd.to_datetime(df['f_inicio'], format='%Y-%m-%d %H:%M:%S')
    df.loc[ :,'f_fin'] =  pd.to_datetime(df['f_fin'], format='%Y-%m-%d %H:%M:%S')

    df_heatmap = df.loc[:, ['ANIO_AJUSTADO','SEMANA', 'COD_ALIM','Tm/h']]
    df_consolidado = df_heatmap.groupby(['ANIO_AJUSTADO', 'SEMANA',
'COD_ALIM']).size().reset_index(name='Conteo')
    df_consolidado['ANIO_SEMANA'] = df_consolidado['ANIO_AJUSTADO'].astype(str) + '-' +
df_consolidado['SEMANA'].astype(str).str.zfill(2)
    df_consolidado.sort_values(by='ANIO_SEMANA', ascending=True, inplace=True)
    df_consolidado['Conteo'] = df_consolidado['Conteo'].astype(int)

    # Calcular Q1, Q3 y IQR
    Q1 = df[columna_analizada].quantile(0.25)
    Q3 = df[columna_analizada].quantile(0.75)
    IQR = Q3 - Q1
    # Definir límites para los outliers
    limite_inferior = Q1 - 1.5 * IQR
    limite_superior = Q3 + 1.5 * IQR
    # Identificar outliers
    outliers = df[(df[columna_analizada] < limite_inferior) | (df[columna_analizada] > limite_superior)]
    df = df.drop(outliers.index)

    # Calcular la duración entre las dos fechas
    df['Duracion'] = df['f_fin'] - df['f_inicio']
    # Encontrar el punto medio de la duración
```

```
    df['Mitad_Duracion'] = df['Duracion'] / 2
    # Calcular el punto medio en términos de fecha y hora
    df['Fecha_Media'] = df['f_inicio'] + df['Mitad_Duracion']

    #Paso 03 Consolidación por semana

    df =
df.loc[:,['Fecha','ANIO','MES','DIA','LOTE','COD_ALIM','Tm/h','f_inicio','f_fin','Anio_Semana','ANIO_AJUST
ADO','SEMANA']]

    df.set_index('Fecha', inplace=True)
    df.sort_index(inplace=True)

    resultado = df.groupby('COD_ALIM').size().reset_index(name='Conteo')
    resultado_ordenado = resultado.sort_values(by='Conteo', ascending=False)

    # Calcular el total de la columna 'Conteo'
    total = resultado_ordenado['Conteo'].sum()
    # Ordenar el DataFrame por 'Conteo' en orden descendente (si aún no está ordenado)
    resultado_ordenado = resultado_ordenado.sort_values(by='Conteo', ascending=False)
    # Calcular la suma acumulada de la columna 'Conteo'
    resultado_ordenado['Suma_Acumulada'] = resultado_ordenado['Conteo'].cumsum()
    # Encuentre el punto donde la suma acumulada alcanza el 80% del total
    corte_80 = total * 0.8
    # Filtrar los registros que forman el 80% del total
    resultado_filtrado = resultado_ordenado[resultado_ordenado['Suma_Acumulada'] <= corte_80]
    # Opcionalmente, puede eliminar la columna 'Suma_Acumulada' si ya no la necesita
    del resultado_filtrado['Suma_Acumulada']

    df = df[df['COD_ALIM'].isin(resultado_filtrado['COD_ALIM'])]

    list_productos = df['COD_ALIM'].unique()

    dict_prod = {}
    for registro in list_productos:
        dict_prod[registro] = df.loc[df['COD_ALIM'] == registro, : ]


    dict_velocidades_semana = {}
    for registro in list_productos:
        dict_velocidades_semana[registro] = consolida_por_semana(dict_prod[registro])


    warnings.filterwarnings('ignore')
    df_final = completar_datos_por_anio(2020, dict_velocidades_semana)
```

```
    df_final = pd.concat([df_final, completar_datos_por_anio(2021, dict_velocidades_semana)],
ignore_index=True)
    df_final = pd.concat([df_final, completar_datos_por_anio(2022, dict_velocidades_semana)],
ignore_index=True)
    df_final = pd.concat([df_final, completar_datos_por_anio(2023, dict_velocidades_semana)],
ignore_index=True)

    #esta linea es necesaria porque el df final del anterior paso es df_final
    df = df_final
    df['ANIO_AJUSTADO'] = df['ANIO_AJUSTADO'].astype(int)

    #Calculo de la fecha del primer día de la semana
    df['FECHA'] = pd.to_datetime(df['ANIO_AJUSTADO'].astype(str) + '-W' + df['SEMANA'].astype(str) + '-1',
format='%Y-W%U-%w')

    mask_2020 = df['ANIO_AJUSTADO'] == 2020
    df.loc[mask_2020, 'FECHA'] = df.loc[mask_2020, 'FECHA'] - pd.Timedelta(days=7)

    list_productos = df['PRODUCTO'].unique()

    #obtengo la lista de productos de los cuales sus series de tiempo no son ruido blanco
    lista_prod_no_ruido_blanco = []
    for producto in list_productos:
        df_filtrado = df.loc[df['PRODUCTO'] == producto,['Tm/h']]
        if not es_ruido_blanco(df_filtrado):
            lista_prod_no_ruido_blanco.append(producto)

    df_filtrado = df[ df['PRODUCTO'].isin(lista_prod_no_ruido_blanco)]
    df_filtrado.to_csv('/opt/ml/processing/train/series_para_sarima.csv', index=False)
```

Apéndice 7 Código Fuente del script entrenamiento.py

```
import os
import pandas as pd
import numpy as np
from sklearn.preprocessing import MinMaxScaler
from keras.models import Sequential
from keras.models import load_model
from keras.layers import LSTM, Dense
import warnings
from keras.callbacks import EarlyStopping, ModelCheckpoint
from sagemaker_training import environment
import boto3

def print_files_in_directory(directory):
```

```
    for filename in os.listdir(directory):
        print(filename)

# Definición de funciones
def mean_absolute_percentage_error(y_true, y_pred):
    y_true, y_pred = np.array(y_true), np.array(y_pred)
    return np.mean(np.abs((y_true - y_pred) / y_true)) * 100

def crear_arreglo_entradas_y_salidas(df):
    df['Tm/h_scaled'] = scaler.fit_transform(df[['Tm/h']])
    # Crear secuencias de tiempo
    time_step = 5
    X, y = [], []
    for i in range(len(df) - time_step - 1):
        X.append(df['Tm/h_scaled'].iloc[i:i + time_step].values)
        y.append(df['Tm/h_scaled'].iloc[i + time_step])
    X, y = np.array(X), np.array(y)
    return X, y

def crear_conjunto_entre_valid_test(X, y):
    data_size = X.shape[0]
    train_size = int(data_size * 0.7)
    val_size = int(data_size * 0.15)
    test_size = data_size - train_size - val_size

    X_train, X_val , X_test = X[:train_size], X[train_size:train_size + val_size], X[train_size +
val_size:]
    y_train, y_val , y_test = y[:train_size], y[train_size:train_size + val_size], y[train_size +
val_size:]

    return X_train, y_train, X_val, y_val, X_test, y_test

def crear_modelo(X_train):
    model = Sequential()
    model.add(LSTM(units=50, return_sequences=True, input_shape=(X_train.shape[1], 1)))
    model.add(LSTM(units=50))
    model.add(Dense(units=1))
    model.compile(optimizer='adam', loss='mean_squared_error')
    return model

def entrenar_modelo(producto, model, X_train, y_train, X_val, y_val):
    early_stopping = EarlyStopping(monitor='val_loss', patience=100, mode='min',
restore_best_weights=True)
    model_dir = os.path.join(os.environ['SM_MODEL_DIR'], f'{producto}/00000001')
    print(f'Guardando el modelo en {model_dir}')
    os.makedirs(model_dir, exist_ok=True)
```

```
    model_checkpoint = ModelCheckpoint(model_dir, monitor='val_loss', save_best_only=True,
save_format='tf')
    X_train = np.reshape(X_train, (X_train.shape[0], X_train.shape[1], 1))
    model.fit(X_train, y_train, epochs=50000, batch_size=32, callbacks=[early_stopping, model_checkpoint],
validation_data=(X_val, y_val), verbose=0)

def predecir_para_test(X_test, model):
    X_test = np.reshape(X_test, (X_test.shape[0], X_test.shape[1], 1))
    predicted_values = model.predict(X_test)
    predicted_values = scaler.inverse_transform(np.array(predicted_values).reshape(-1, 1))
    return predicted_values

def entrenar_modelo_lstm(producto, df):
    df_temporal = df.copy()
    X, y = crear_arreglo_entradas_y_salidas(df_temporal)
    X_train, y_train, X_val, y_val, X_test, y_test = crear_conjunto_entre_valid_test(X, y)
    model = crear_modelo(X_train)
    entrenar_modelo(producto, model, X_train, y_train, X_val, y_val)

if __name__ == '__main__':
    print('OKOK')
    env = environment.Environment()
    print(f'directorio env={env.channel_input_dirs}')

    train_data_location = env.channel_input_dirs['train']
    print(f'Training data is saved at: {train_data_location}')

    df = pd.read_csv(f'{train_data_location}/series_para_sarima.csv')
    df['FECHA'] = df['FECHA'].apply(pd.to_datetime, format='%Y-%m-%d')
    df.set_index('FECHA', inplace=True)

    list_productos = df['PRODUCTO'].unique()

    scaler = MinMaxScaler(feature_range=(0, 1))

    for producto in list_productos:
        print(producto)
        df_filtrado = df.loc[df['PRODUCTO'] == producto, ['Tm/h']]
        entrenar_modelo_lstm(producto, df_filtrado)
    print('Fin del entrenamiento')
```

Apéndice 8 Código Fuente del script descompresion.py

```
import os
import tarfile
```

```
model_dir = '/opt/ml/processing/model'
output_dir = '/opt/ml/processing/extracted_models'

# Crear el directorio de salida si no existe
os.makedirs(output_dir, exist_ok=True)

# Descomprimir el archivo model.tar.gz
tar_path = os.path.join(model_dir, 'model.tar.gz')
with tarfile.open(tar_path, 'r:gz') as tar:
    tar.extractall(path=output_dir)

# Comprimir cada directorio en un archivo .tar.gz y luego eliminar el directorio original
for item in os.listdir(output_dir):
    item_path = os.path.join(output_dir, item)
    if os.path.isdir(item_path):
        tar_gz_path = os.path.join(output_dir, f"{item}.tar.gz")

        # Comprimir el directorio en .tar.gz
        with tarfile.open(tar_gz_path, 'w:gz') as tar_gz:
            tar_gz.add(item_path, arcname=os.path.basename(item_path))

        # Eliminar el directorio original
        for root, dirs, files in os.walk(item_path, topdown=False):
            for file in files:
                os.remove(os.path.join(root, file))
            for dir in dirs:
                os.rmdir(os.path.join(root, dir))
        os.rmdir(item_path)
```

Apéndice 9 Código Fuente del script despliegue.py

```
import boto3
import sagemaker
from sagemaker.tensorflow import TensorFlowModel
from sagemaker.serverless import ServerlessInferenceConfig
import os
import argparse

def deploy_models(bucket_name, s3_model_directory, role, framework_version, region):
    # Configurar la región de AWS
    boto3.setup_default_session(region_name=region)
    sagemaker_session = sagemaker.Session(boto3.Session(region_name=region))

    # Crear cliente de boto3 para S3
```

```
    s3_client = boto3.client('s3', region_name=region)

    # Listar todos los objetos en el directorio del bucket S3
    response = s3_client.list_objects_v2(Bucket=bucket_name, Prefix=s3_model_directory)

    # Filtrar los archivos de modelo (asumiendo que los modelos son archivos .tar.gz)
    model_files = [obj['Key'] for obj in response.get('Contents', []) if obj['Key'].endswith('.tar.gz')]

    # Iterar sobre cada archivo de modelo y desplegarlo
    for model_file in model_files:
        model_url = f's3://{bucket_name}/{model_file}'
        tensorflow_model = TensorFlowModel(model_data=model_url,
                                           role=role,
                                           framework_version=framework_version,
                                           sagemaker_session=sagemaker_session)

        # Configurar la inferencia serverless
        serverless_config = ServerlessInferenceConfig(memory_size_in_mb=1024, max_concurrency=1)

        # Desplegar el modelo como endpoint serverless
        predictor = tensorflow_model.deploy(serverless_inference_config=serverless_config)

        print(f"Modelo serverless desplegado desde {model_url}")

if __name__ == "__main__":
    parser = argparse.ArgumentParser()
    parser.add_argument('--bucket-name', type=str, required=True)
    parser.add_argument('--s3-model-directory', type=str, required=True)
    parser.add_argument('--role', type=str, required=True)
    parser.add_argument('--framework-version', type=str, default='2.11.0')
    parser.add_argument('--region', type=str, default='us-east-1')  # Agregar región como argumento

    args = parser.parse_args()

    deploy_models(args.bucket_name, args.s3_model_directory, args.role, args.framework_version,
args.region)
```

Apéndice 10 Dockerfile creación de imagen personalizada

```
# Start from the TensorFlow Docker image
FROM 683313688378.dkr.ecr.us-east-1.amazonaws.com/sagemaker-scikit-learn:0.23-1-cpu-py3

# Install Keras
RUN pip install --no-cache-dir pandas numpy scikit-learn keras tensorflow flask==1.1.2 werkzeug==1.0.1
```