



Desarrollo de aplicativo web para el pronóstico de ventas futuras utilizando
modelos predictivos.

Joann Esteban Bedoya Lopez

Universidad del Valle
Facultad de ingeniería
Escuela de ingeniería de sistemas y computación
Tuluá
2025

**Desarrollo de aplicativo web para el pronóstico de ventas futuras utilizando
modelos predictivos.**

Modalidad: proyecto de ingeniería

Joann Esteban Bedoya Lopez
Código 202059906
joann.bedoya@correounivalle.edu.co

Director
Héctor Fabio Ocampo Arbeláez
Profesor de la Escuela de Ingeniería de Sistemas y Computación
hector.ocampo@correounivalle.edu.co

Universidad del Valle
Facultad de ingeniería
Escuela de ingeniería de sistemas y computación
Tuluá
2025

Tabla de Contenido

Resumen	3
1. Introducción	5
2. Formulación del problema	6
2.1. Descripción del problema	6
2.2. Definición del problema	7
3. Marco Referencial	8
3.1. Marco teórico	8
3.2. Estado del arte	8
3.2.1. Análisis predictivo	8
3.2.2. Regresión lineal	9
3.2.3. Algoritmos especializados	9
3.3. Antecedentes	9
3.3.1. Herramienta de pronósticos para integrar en lenguajes de programación.	9
3.3.2. Trabajos previos.	10
3.3.3. Software y servicios.	10
3.4. Marco conceptual	11
3.4.1. Software	11
3.4.2. Machine learning	12
3.4.3. Pronóstico de ventas	13
3.4.4. Métricas de evaluación	13
3.4.5. Modelos predictivos	14
4. Alcance del proyecto	17
4.1. Declaración del alcance	17
4.1.1. Alcance Específico	17
4.2. Objetivos	18
4.2.1. Objetivo general	18
4.2.2. Objetivos Específicos	18
4.3. Resultados esperados	19
5. Metodología	20
5.1. Metodología de investigación	20
5.2. Metodología de desarrollo de software	21
5.2.1. Metodología elegida: Agile.	21
5.2.2. Modificaciones aplicadas a Scrum	21
5.2.3. Gestión del Proyecto	22

6. Datasets	23
6.1. Fuente de los datos	23
6.1.1. Kaggle	23
6.1.2. Data.gov	24
6.2. Tabla de datasets evaluados	24
6.2.1. Criterios de selección:	24
6.3. Análisis de la selección de datasets	25
6.4. Análisis de los Datasets	25
6.4.1. Retail Sales Dataset	25
6.4.2. Dataset Pizzería	27
7. Modelos Entrenados	28
7.1. Modelo 1: LSTM	28
7.2. Modelo 2: SVM	29
7.3. Modelo 3: SARIMA	31
7.4. Resumen de Resultados	32
8. Prototipo	33
8.1. Implementación	33
8.2. Creación de la base de datos	33
8.3. Creación del backend	33
8.4. Creación del frontend	36
8.4.1. Inicio de sesión	36
8.4.2. Panel principal	37
8.4.3. Carga de archivos	37
8.4.4. Visualización de resultados	38
8.5. Flujo del prototipo para hacer una predicción	39
9. Pruebas	40
9.1. Prueba de modelos de predicción con datos de negocio local de ensalada.	40
9.1.1. Entrenamientos adicionales con opciones avanzadas.	43
9.1.2. Evaluación con datos reales recopilados abril y mayo de 2025.	44
9.2. Prueba de modelos de predicción con datos de negocio local de comida rápida.	46
9.2.1. Entrenamientos adicionales con opciones avanzadas.	49
9.2.2. Evaluación con datos reales recopilados abril y mayo de 2025.	49
9.3. Encuesta realizada a usuarios dueños de negocio.	52
10. Conclusiones	57
11. Trabajos futuros	58
12. Referencias	59
13. Anexos	62

Resumen

En este trabajo de grado se aborda el desarrollo de un aplicativo de software para el pronóstico de ventas utilizando algoritmos de inteligencia artificial. El problema evidenciado es la carencia de una herramienta que permita a las pequeñas y medianas empresas (PYMES) tomar mejores decisiones de manera informada para la gestión de su inventario ya que para una empresa tener una buena gestión del inventario es crucial ya que las ventas pueden fluctuar de una manera poco predecible lo que puede llevar a pérdidas económicas por tener poco o demasiado inventario. Las pequeñas empresas, pueden enfrentar más dificultades en la gestión del inventario, lo cual puede resultar poco eficiente debido a que suelen realizar esta gestión de manera manual y sin un estudio previo.

El objetivo principal de este trabajo de grado es crear una herramienta capaz de generar pronósticos de ventas precisos a partir de datos históricos, integrando modelos predictivos entrenados con técnicas de machine learning. Para ello, se implementaron y evaluaron tres modelos distintos: una red neuronal LSTM (Long Short-Term Memory), un modelo de soporte vectorial para regresión (SVM) con distintos kernels, y un modelo estadístico SARIMA (Seasonal Autoregressive Integrated Moving Average). Las métricas utilizadas para evaluar el desempeño de los modelos fueron el Error Cuadrático Medio (MSE), el Error Absoluto Medio (MAE) y el Error Porcentual Absoluto Medio (MAPE).

El proyecto fue desarrollado bajo la metodología ágil Scrum, adaptada para un entorno de trabajo individual. Se construyó un backend en Python utilizando Django, con procesamiento de datos mediante bibliotecas como Pandas, NumPy, TensorFlow/Keras, Scikit-learn y Statsmodels. La base de datos utilizada fue MongoDB, y la interfaz web se diseñó con tecnologías modernas como React, permitiendo al usuario cargar datasets en formato CSV/Excel, configurar parámetros de entrenamiento y visualizar los resultados en gráficos interactivos.

Durante la fase experimental, se trabajó con datasets obtenidos de plataformas como Kaggle y Data.gov, incluyendo datos de ventas de una pizzería y comercios minoristas. El modelo SVM con kernel sigmoide presentó el mejor desempeño en términos absolutos, con un $MSE = 5.564$, $MAE = 1.937$ y $MAPE = 37.20\%$. Por otro lado, el modelo LSTM destacó en precisión relativa, con un $MAPE = 35.05\%$, aunque tuvo un mayor MAE y MSE. El modelo SARIMA mostró un rendimiento intermedio, con buen ajuste en patrones estacionales pero menor precisión porcentual ($MAPE = 40.04\%$).

Además de los experimentos con datos sintéticos, se realizaron pruebas reales con información de negocios locales de ensaladas y comida rápida. Los resultados arrojaron un MAPE por debajo del 20 % en escenarios controlados y hasta 37 % en contextos más ruidosos. También se aplicó una encuesta a usuarios finales, quienes destacaron la facilidad de uso, la utilidad de las visualizaciones y el potencial de la herramienta para apoyar la toma de decisiones.

Abstract

This paper is about the development of a software application for sales forecasting using artificial intelligence algorithms. The identified problem is the lack of a tool that enables small and medium-sized enterprises to make better-informed decisions regarding inventory management. For a business, having good inventory control is crucial, as sales can fluctuate unpredictably, potentially leading to financial losses from either overstocking or understocking. Small businesses, in particular, may face greater challenges in inventory management, often relying on manual processes without prior data analysis.

The main objective of this thesis is to create a tool capable of generating accurate sales forecasts based on historical data by integrating predictive models trained using machine learning techniques. To this end, three different models were implemented and evaluated: a Long Short-Term Memory (LSTM) neural network, a Support Vector Regression (SVR) model with various kernels, and a statistical Seasonal Autoregressive Integrated Moving Average (SARIMA) model. The performance of the models was assessed using the Mean Squared Error (MSE), Mean Absolute Error (MAE), and Mean Absolute Percentage Error (MAPE) metrics.

The project was developed using the agile Scrum methodology, adapted for a solo work environment. The backend was built in Python using Django, with data processing handled by libraries such as Pandas, NumPy, TensorFlow/Keras, Scikit-learn, and Statsmodels. MongoDB was used as the database, and the web interface was designed with modern technologies like React, allowing users to upload datasets in CSV/Excel format, configure training parameters, and visualize results through interactive graphs.

During the experimental phase, datasets were used, obtained from platforms such as Kaggle and Data.gov, including sales data from a pizzeria and retail stores. The SVR model with a sigmoid kernel showed the best absolute performance, with an MSE of 5.564, MAE of 1.937, and MAPE of 37.20 %. On the other hand, the LSTM model stood out in terms of relative precision, with a MAPE of 35.05 %, although it had higher MAE and MSE values. The SARIMA model demonstrated intermediate performance, with a good fit for seasonal patterns but lower percentage accuracy (MAPE = 40.04 %).

In addition to synthetic data experiments, real-world tests were conducted using data from local salad and fast-food businesses. Results showed a MAPE below 20 % in controlled scenarios and up to 37 % in noisier contexts. A user survey was also conducted, where participants highlighted the ease of use, the usefulness of the visualizations, and the tool's potential to support decision-making.

Key words

[1] Machine learning, regression analysis, economic forecasting, demand forecasting, artificial Intelligence.

1. Introducción

En la actualidad, la información es uno de los activos más valiosos, y su correcta gestión puede determinar el éxito o el fracaso de diversas iniciativas. En el ámbito del comercio de productos y servicios, la gestión eficiente del inventario es fundamental para satisfacer la demanda de los clientes. Tradicionalmente, muchas empresas han recurrido a métodos manuales para estimar sus necesidades de inventario, lo cual puede acarrear múltiples inconvenientes, tales como la sobrecompra o la escasez de productos, lo que a su vez puede llevar a pérdidas económicas. Predecir la demanda de forma manual es una tarea ardua y compleja, ya que las ventas fluctúan de manera impredecible de un período a otro.

En este contexto, el pronóstico de ventas se presenta como una solución para mejorar la gestión del inventario. Los grandes avances en la inteligencia artificial y en modelos predictivos de machine learning ofrecen soluciones para analizar datos históricos y hacer predicciones más precisas sobre tendencias futuras.

La importancia de este proyecto radica en la posibilidad de ofrecer una herramienta que no solo facilite la toma de decisiones, sino que permita al usuario planificar estrategias y mejorar la eficiencia en la gestión del inventario. Para ello, se tiene como objetivo el desarrollo de un aplicativo de software que utilice modelos de machine learning para el pronóstico de ventas futuras. Este objetivo se busca alcanzar mediante el uso de la metodología ágil con el marco de trabajo Scrum, el cual proporciona un flujo de trabajo ordenado y flexible para lograr los objetivos establecidos.

2. Formulación del problema

2.1. Descripción del problema

La gestión del inventario es crucial para las PYMES (pequeñas y medianas empresas) tales como droguerías, tiendas de ropa, restaurantes, ya que permiten optimizar y reducir los riesgos de tener mucho o poco inventario. El manejo del inventario es esencial para este tipo de negocios ya que debe existir un equilibrio entre la oferta y la demanda de los productos, así que un buen manejo del inventario, este se caracteriza en encontrar un equilibrio entre los costos asociados con mantener un inventario y las ganancias obtenidas[2]. Uno de los principales problemas que enfrentan muchas pequeñas empresas es la gestión manual del inventario, un método que resulta altamente ineficiente y propenso a errores [3] y sin tener ningún tipo de análisis de los datos. Además, los pequeños negocios pueden enfrentar restricciones financieras que les impiden operar como las grandes empresas, por ejemplo, contratando profesionales para realizar estudios de mercado y predecir que productos que más se demandan [4].

Con el avance y la globalización de la tecnología se han desarrollado aplicaciones para mejorar este proceso, como software de gestión del inventario, el cual permite tener un registro minucioso de todos los productos disponibles y las ventas registradas, esto permite que el usuario tenga un mejor control del inventario y pueda tomar mejores decisiones [5], además gracias al rápido crecimiento del aprendizaje automático en la computación se pueden entrenar modelos predictivos como la regresión lineal que es muy usada, estos modelos predictivos que pueden analizar datos de ventas pasadas y dar como resultado la predicción o estimación de un valor [6].

La falta de una gestión adecuada del inventario puede desencadenar una serie de problemas, entre los cuales destacan dos escenarios principales: la escasez de productos, que conlleva a la pérdida de ventas y la insatisfacción del cliente, y el exceso de inventario, que genera costos operativos elevados y puede resultar en el desperdicio de productos perecederos lo que puede llevar a la desconfianza de los clientes y la pérdida de futuras posibles ventas, por otro lado, tener mucho inventario requiere costos operativos como el almacenamiento, los impuestos [2] y también se debe tener la naturaleza de ciertos productos ya que estos presentan un deterioro a lo largo del tiempo, como los productos perecederos en los que la calidad del producto es lo más importante, para ello la rapidez y la eficiencia son factores claves [7], y no vender estos productos antes de las fechas de vencimiento conlleva a la pérdida de todos los productos y en consecuencia pérdidas económicas. En la figura 1: "árbol de problema", se puede ilustrar de mejor manera este problema.

Implementar un software especializado en la planificación del inventario, basado en modelos predictivos, permitirá a los negocios tomar decisiones informadas y precisas, minimizando tanto los riesgos de escasez como los costos asociados con el exceso de inventario. Esta solución tecnológica no solo optimiza el proceso de gestión, sino que también mejora la competitividad de las pequeñas y medianas empresas en un mercado cada vez más

dinámico [5].

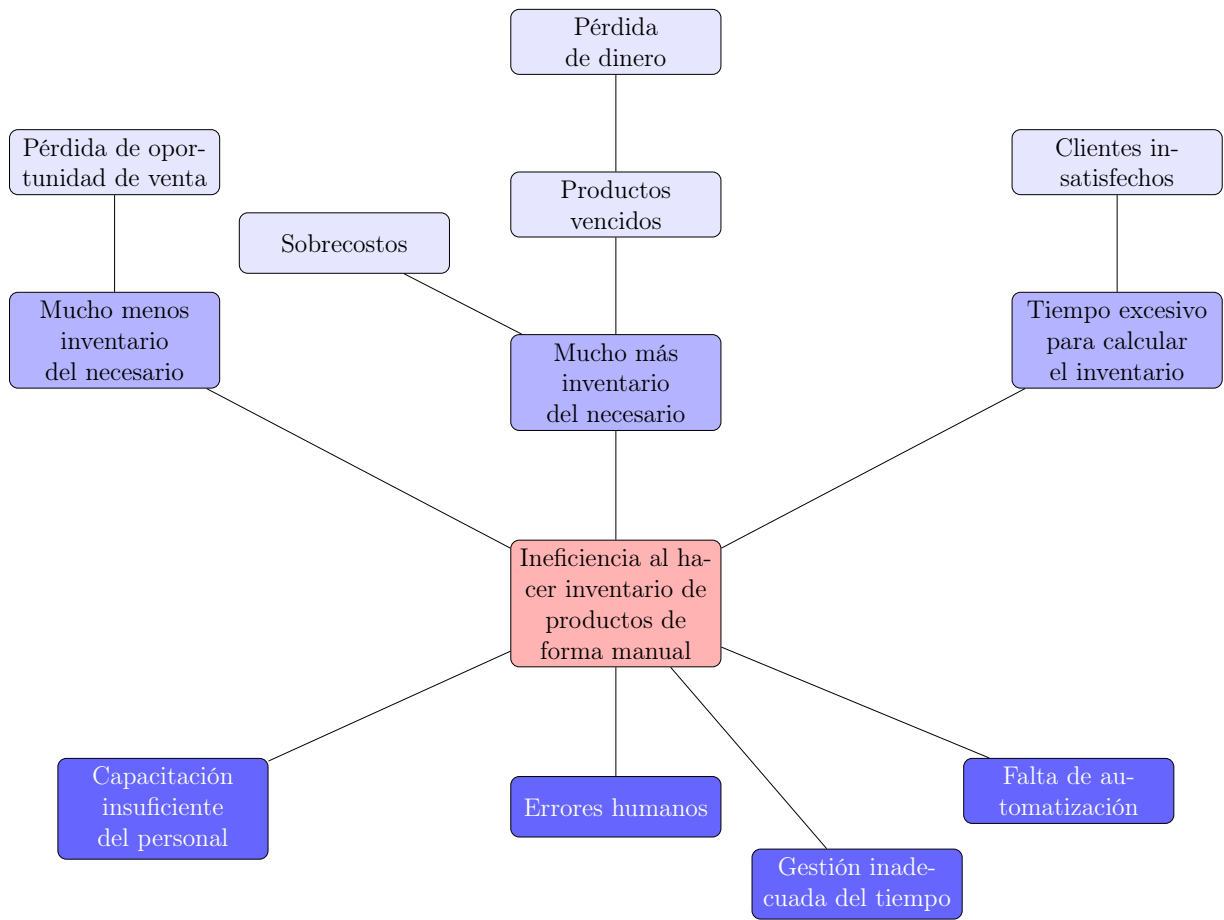


Figura 1: Árbol de problema.

2.2. Definición del problema

¿Cómo implementar un aplicativo web utilizando modelos predictivos para optimizar el pronóstico de ventas futuras en pequeños y medianos negocios?

3. Marco Referencial

3.1. Marco teórico

La teoría general de sistemas es un enfoque que busca modelar y comprender la realidad de manera sistemática y científica [8]. Esta teoría se caracteriza por las relaciones entre las distintas partes de un sistema, las cuales trabajan de manera coordinada y ordenada, dando como resultado conjuntos que emergen de esas interacciones. Según [9], la sinergia es el resultado de las interacciones entre las partes del sistema, que tienen como finalidad alcanzar un objetivo en común, dar una salida, por esto los sistemas tienen la capacidad de convertir entradas en salidas, estas salidas pueden ser medidas y se pueden ajustar las entradas para generar distintos tipos de salidas. Visto en un contexto de la informática se define como un sistema que se compone de entradas, procesos y salidas, las entradas son datos, estos sufren un procesamiento dentro del sistema y al final son transformados a datos de salida [10].

Aplicando estos principios al ámbito del Machine Learning, los modelos predictivos que utilizan redes neuronales se pueden ver como un sistema que reciben entradas, tiene un procesamiento y responde con unas salidas [11], enfocándose en el Machine Learning (ML) estas características son aprovechadas para el entrenamiento automático, ajustándose a diversos parámetros y dando respuestas, en este contexto el ML puede ser muy beneficioso al permitir a las máquinas superar a los humanos en tareas específicas a los humanos, el ML permite el desarrollo de modelos algorítmicos que optimizan la predicción de resultados, enfocándonos en el pronóstico de ventas, el ML ha demostrado ser una herramienta muy poderosa para hacer predicciones con mayor precisión [12].

3.2. Estado del arte

3.2.1. Análisis predictivo

El análisis predictivo se basa en estudiar datos históricos y actuales, el cual gracias a aplicar técnicas de machine learning avanzadas se pueden hacer predicciones a futuro, alrededor del mundo se ha convertido en una parte indispensable para muchas empresas las cuales lo usan para guiar sus decisiones futuras [13]. Estas empresas están en una amplia gama de sectores en la industria como los bancos que lo aplican para hacer predicciones de los prestamos tomados por sus clientes, también en la salud para hacer predicciones de que pacientes tienen un alto porcentaje de sobrevivir a ciertas infecciones, por otra parte, en el marketing son usados para predecir qué porcentaje de los clientes pueden quedar insatisfechos con algún producto, lo cual permite tomar acciones para retenerlos [14].

Uno de los servicios que hace uso del análisis predictivo es SageMaker Canvas, ofrecida por el globalmente conocido, Amazon Web Services, este servicio tiene modelos pre-integrados los cuales están listos para ser usados, aunque si lo desea puede crear un modelo de machine learning personalizado, además de la capacidad de importar datos de distintas fuentes, ahorrando mucho tiempo a la hora de procesar datos [15].

3.2.2. Regresión lineal

En estadística la regresión lineal se basa en medir ciertas semejanzas de un conjunto de datos en forma de una función, esta función nos ofrece información de la relación de estos datos tanto así que una regresión simple ofrece información de varias magnitudes como pueden serlo el crecimiento marginal, el valor de una si la otra es cero y que tan fuerte es la relación [16].

En un caso de ejemplo en el que se aplican los modelos de regresión lineal está un estudio en la agricultura de la producción de microalgas, teniendo varias variables como la temperatura, luz, pH y oxígeno se pueden hacer predicciones de la tendencia que sufre el crecimiento de la microalga [17].

3.2.3. Algoritmos especializados

Existen algoritmos listos para ser implementados en distintos lenguajes de programación como lo es Python, un ejemplo de este tipo de algoritmos es Prophet desarrollado por la reconocida Meta, la empresa detrás de Facebook, este algoritmo es un modelo de regresión lineal, Facebook lo utiliza para hacer predicciones de los recursos que usan y así mejorar su capacidad y eficiencia [18].

Prophet esta diseñada para tareas con enfoque empresarial, tiene una muy buena capacidad de adaptación en diferentes observaciones horarias como días, semanales o mensuales, también controla bien valores atípicos, eventos irregulares como el super bowl en Estados Unidos, se puede ajustar de una manera sencilla sin necesidad de tener conocimientos avanzados en series temporales. Todo esto lo hace muy eficaz para cubrir un amplio abanico de casos de negocio [18].

3.3. Antecedentes

En la actualidad existen múltiples maneras en las que se realizan predicciones de ventas, la estadística ha permitido el análisis de datos y ver sus tendencias a futuro, luego con la introducción de los sistemas informáticos ha permitido la automatización y procesamiento de grandes volúmenes de datos aplicando modelos predictivos en áreas como el machine learning y software enfocado específicamente a las predicciones de tendencias futuras han salido al mercado, a continuación, se hace una revisión más detallada.

3.3.1. Herramienta de pronósticos para integrar en lenguajes de programación.

La empresa meta ha desarrollado una herramienta llamada prophet, la cual puede ser integrada al código Python y R, está basada en series de tiempo y puede hacer pronósticos con bastante precisión, es usado en aplicaciones de Facebook para hacer planificación y ponerse objetivos [18] .

3.3.2. Trabajos previos.

En el trabajo de grado titulado “Uso de redes neuronales para predecir mercados financieros”, se desarrollaron dos modelos predictivos fundamentados en redes neuronales recurrentes (RNN), uno de ellos con arquitectura LSTM, para predecir el comportamiento de precios bursátiles, utilizando datos históricos de acciones como Amazon y 3D Systems Corporation. Este estudio se enmarca en la creciente tendencia del uso de inteligencia artificial en el análisis financiero, destacando la importancia de democratizar estas tecnologías para que inversionistas individuales (“retail”) puedan acceder a herramientas de pronóstico comparables a las utilizadas por grandes corporaciones. El trabajo no solo demostró que es posible obtener resultados precisos con recursos computacionales limitados, sino que también puso en evidencia la capacidad de las redes neuronales para aprender patrones complejos en series temporales. Esta investigación constituye una base sólida sobre la cual se puede continuar explorando y mejorando la aplicación de técnicas de aprendizaje especialmente usando redes neuronales LSTM[19].

Por otro lado en el trabajo titulado “Estimación de modelos para el pronóstico de precios en el sector agropecuario en el distrito especial de Buenaventura, una aproximación econométrica usando el lenguaje de programación R.”, desarrollaron un sistema de pronóstico de precios para tres productos agrícolas (Piña Gold, Yuca Chirrosa y Banano Criollo) en Buenaventura, aplicando modelos de series de tiempo en R. Utilizaron la metodología CRISP-DM, técnicas de imputación con el paquete `imputeTS`, y modelos como MA, SES, Holt-Winters, ARIMA, regresión y ensambles. Evaluaron el desempeño y visualizaron los resultados en un *dashboard* interactivo usando *Flexdashboard* del lenguaje R, que automatiza la recolección de datos y genera pronósticos semanales. Este trabajo fue un buen acercamiento hacia una implementación de software con múltiples modelos para la predicción de precios de distintos artículos especialmente usando ARIMA[20].

3.3.3. Software y servicios.

En el mercado existen software y servicios en la nube que ofrecen dentro de sus servicios herramientas para hacer pronósticos, a continuación, se revisaran algunos de ellos:

1. **Liveflow:**

Se integra a hojas de cálculo como Excel o sheets de Google volviéndolos en un panel que ofrece reportes de los datos, además ofrece la posibilidad de hacer un pronóstico a futuro, entregando una tasa de crecimiento que se espera [21].

2. **SalesForce:**

Es una plataforma de CRM fundada en 1999, ofrece una infraestructura dedicada a ayudar a las empresas en distintas industrias tales como marketing, ventas y servicio al consumidor, uno de sus servicios llamado Manufacturing Cloud ofrece la posibilidad de hacer pronósticos para identificar futuras oportunidades [22].

3. **Netsuite:**

Su propietario es Oracle, netsuite ofrece un software empresarial basado en la nube

con servicio como manejo de inventario, una de sus características es el pronóstico de ventas, este ofrece herramientas como, Sales Rep Forecast Editor permite a los representantes de ventas ajustar pronósticos basados en su experiencia, métodos integrales que dan un alto nivel de exactitud y herramientas que compilar los resultados para brindarle al usuario la información que necesita [23].

4. Zoho CRM:

Es un software desarrollado por la empresa zoho corporation, fue fundado en 1996 y se ha dedicado a dar soluciones en la gestión del ciclo de vida del cliente, tiene un panel centralizado donde se pueden encontrar todas las herramientas [24], entre sus herramientas tiene un sistema de proyecciones que se basa en datos históricos de ventas para dar una proyección a futuro.

A continuación, se muestra la tabla 1: Comparativa software predicción de ventas en el mercado, a modo de comparación y resumen de los softwares antes mencionados.

Productos	Características	Plataforma	Licencia
LiveFlow	Panel con reporte de datos, integración con hojas de cálculo Excel y Google sheets.	Web e implementación hoja de cálculo	De pago.
SalesForce	Infraestructura dedicada a la industria de ventas y servicio al consumidor.	Web	De pago.
Netsuite	Propiedad de Oracle, ofrece pronósticos de ventas precisos mediante herramientas avanzadas como el Sales Rep Forecast Editor.	Web	De pago.
Zoho CRM	Visión completa del ciclo de vida del cliente, panel de herramientas con sistema de proyección basado en datos histórico.	Web, programa de escritorio	De pago.

Tabla 1: Comparativa software predicción de ventas en el mercado

3.4. Marco conceptual

El desarrollo de un aplicativo de software que haga predicciones de ventas basado en modelos de aprendizaje automático y su posterior entrenamiento como nos permitiría hacer mejores predicciones del inventario futuro para comercios.

3.4.1. Software

Definición: Según la real academia de la lengua española su definición es “Conjunto de programas, instrucciones y reglas informáticas para ejecutar ciertas tareas en una compu-

tadora”. [25].

Características: Las características del software son [26]:

1. **Funcionalidad:** Son la serie de capacidades que tiene el software para que el usuario pueda lograr lo que quiere.
2. **Usabilidad:** Es la forma en la que el usuario interactúa con el software de una manera intuitiva y que el usuario se sienta satisfecho con esta.
3. **Eficiencia:** Es la forma en la que el software usa los recursos de la máquina de la mejor manera posible.
4. **Flexibilidad:** Es la manera en la que el software puede hacer cambios a los requerimientos del usuario.
5. **Fiabilidad:** Es como el software puede operar en diversas situaciones diferentes y tener un resultado acertado.
6. **Sostenibilidad:** Es la facilidad con la que el software se puede modificar, depurar y mejorar a lo largo de su ciclo de vida.
7. **Portabilidad:** Es la habilidad del software de funcionar en diversas plataformas.
8. **Integridad:** Es un aspecto que implica mantener la precisión y coherencia de los datos a lo largo de su ciclo de vida.

3.4.2. Machine learning

Definición: El aprendizaje automático (AM) es una rama de la inteligencia artificial (IA) y la informática que se centra en el uso de datos y algoritmos para que la IA imite la forma en que aprenden los humanos, mejorando gradualmente su precisión [27].

Métodos: según IBM [27] los métodos son:

1. **Aprendizaje supervisado:** Se utiliza conjuntos de datos etiquetados para entrenar algoritmos que clasifiquen datos o predigan resultados con precisión. El modelo ajusta los pesos a medida que más datos se van introduciendo; con esto se evitan sobreajustes o subajustes. Este tipo de aprendizaje es muy útil en la resolución de problemas del mundo real, tales como correos electrónicos no deseados, en redes neuronales y regresión lineal.
2. **Aprendizaje no supervisado:** Se emplean algoritmos para analizar y agrupar conjuntos de datos no etiquetados, es decir, datos no clasificados previamente, descubriendo patrones ocultos sin intervención humana. Es ideal para el análisis exploratorio de datos y estrategias de venta cruzada.

3. **Aprendizaje semi-supervisado:** Es la combinación de los dos aprendizajes: supervisado y no supervisado. Utiliza una pequeña porción de datos etiquetados como guía para la clasificación de un mayor volumen de datos no etiquetados. Este tipo de aprendizaje es muy útil cuando no hay suficientes datos etiquetados, permitiendo mejorar la precisión del modelo.

3.4.3. Pronóstico de ventas

Definición: el pronóstico de ventas se define como una estimación de las ventas para un período de tiempo determinado. Se puede describir como la predicción del número de ventas de un producto o servicio en el futuro, a menudo basada en el rendimiento de ventas pasadas o el producto o servicio en sí [28].

3.4.4. Métricas de evaluación

Las métricas de evaluación son herramientas utilizadas para medir el rendimiento de los modelos evaluados. En el caso de la predicción de ventas, estas métricas son especialmente útiles, ya que permiten evaluar modelos de regresión y de series temporales.

Error Absoluto Medio (MAE) El *Mean Absolute Error* (MAE) es una forma de medir qué tan lejos están, en promedio, las predicciones de un modelo respecto a los valores reales. Para calcularlo, se toma la diferencia entre el valor real y_i y el valor predicho $\hat{y}_i$ para cada dato, se convierte en valor absoluto, y luego se calcula el promedio de todas esas diferencias:

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

Cuanto más bajo sea el MAE, más preciso será el modelo. Una de sus ventajas es que sus resultados se expresan en las mismas unidades que los datos originales, lo que facilita su interpretación.

Error Porcentual Absoluto Medio (MAPE) El *Mean Absolute Percentage Error* (MAPE) permite conocer qué tan grande fue el error de las predicciones del modelo, pero expresado como un porcentaje del valor real:

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

Por ejemplo, si el MAPE es de 10 %, significa que, en promedio, el modelo se equivocó en un 10 % respecto a los valores reales. Es una métrica fácil de interpretar, pero tiene un inconveniente importante: si uno de los valores reales y_i es cero o muy cercano a cero, el cálculo se vuelve problemático o puede distorsionar mucho el resultado.

Error Cuadrático Medio (MSE) El *Mean Squared Error* (MSE) mide qué tan lejos están las predicciones del modelo de los valores reales, pero en lugar de tomar el valor absoluto del error, se eleva al cuadrado:

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

Esto hace que los errores grandes tengan un mayor impacto en el resultado final. El MSE es muy útil cuando se desea penalizar errores grandes. Sin embargo, como el error está elevado al cuadrado, el resultado no está en las mismas unidades que los datos originales, lo que puede dificultar su interpretación directa.

3.4.5. Modelos predictivos

El modelo de red neuronal **LSTM** (Long Short-Term Memory) es un modelo de red neuronal recurrente (RNN) que introduce una mejora que permite recordar información durante períodos largos de tiempo, superando una limitación de las RNN tradicionales, las cuales solo pueden recordar información a corto plazo. A este problema se le conoce como el desvanecimiento del gradiente. Para lograr tener una memoria a largo plazo, se implementan celdas de memoria y puertas que regulan el flujo de información. Estas puertas son redes neuronales que cumplen las siguientes funciones:

- **Forget gate:** Su función es eliminar elementos de la memoria.
- **Update gate:** Su función es ingresar nuevos elementos a la memoria.
- **Output gate:** Su función es crear el estado actualizado.

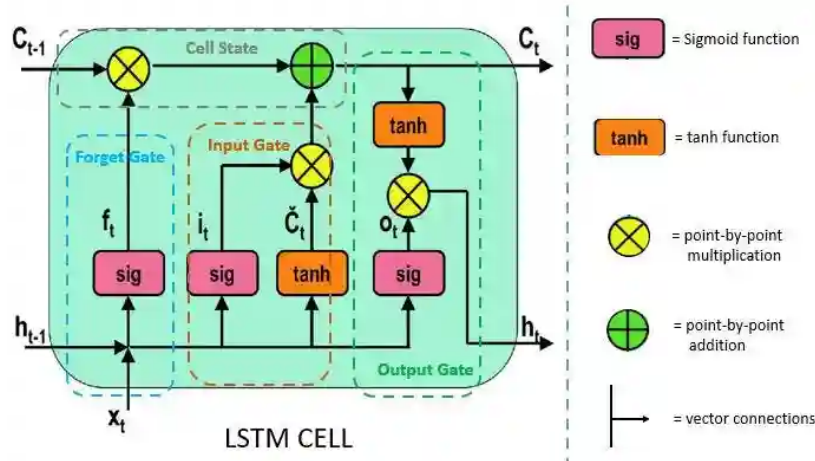


Figura 2: Estructura de una celda LSTM [29].

Support Vector Regression (SVR) es una técnica de aprendizaje automático basada en el modelo de máquinas de soporte vectorial (SVM), enfocada en resolver problemas

de regresión. A diferencia de las técnicas clásicas de regresión que buscan minimizar el error global, SVR busca encontrar una función que se desvíe lo menos posible de los valores reales dentro de un margen de tolerancia definido por un parámetro llamado epsilon (ϵ). En otras palabras, SVR intenta ajustar una línea o curva que pase lo más cerca posible de los datos, permitiendo cierto margen de error sin penalización, para los puntos dentro de dicho margen.

Además, existen distintos *kernels* para modelar relaciones lineales y no lineales. Algunos de los más comunes son: lineal, Radial Basis Function (RBF) y polinomial [30].

SVR tiene los siguientes parámetros:

- **C:** Controla el compromiso entre el error del entrenamiento y la rigidez del margen. Valores más altos penalizan más los errores.
- **gamma:** Parámetro del *kernel* que define cuánto influye un solo punto de entrenamiento.
- **epsilon:** Define un margen de tolerancia donde no se penaliza el error en la función de pérdida (*insensitive loss*).

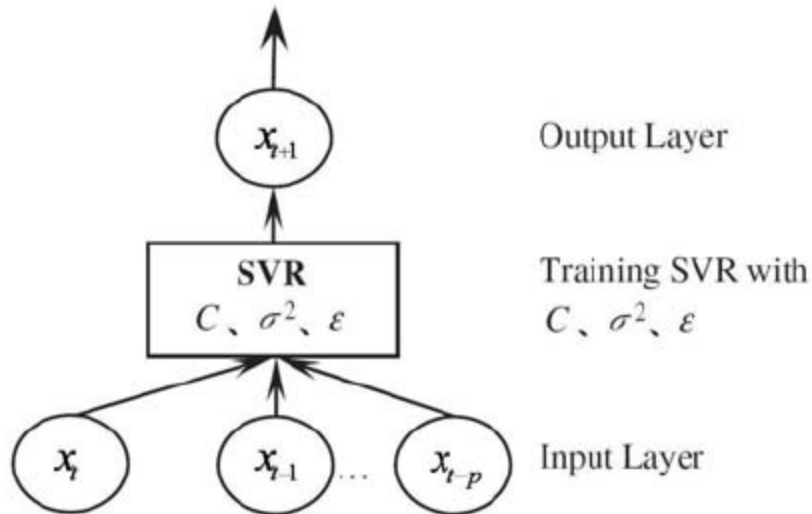


Figura 3: Esquema de SVR [31].

El modelo **SARIMA** (Seasonal Autoregressive Integrated Moving Average) es una extensión del modelo ARIMA que incorpora la capacidad de predecir series temporales con patrones estacionales que se repiten en un intervalo de tiempo determinado, como días, semanas o meses.

Este modelo cuenta con los siguientes componentes ajustables:

- **p:** Orden del componente autorregresivo (AR), que captura la relación entre una observación y sus valores pasados.

- **d**: Número de diferenciaciones necesarias para lograr la estacionariedad de la serie temporal.
- **q**: Orden del componente de media móvil (MA), que modela la relación entre una observación y los errores residuales de modelos anteriores.
- **P**: Orden del componente autorregresivo estacional.
- **D**: Número de diferenciaciones estacionales necesarias.
- **Q**: Orden del componente de media móvil estacional.
- **s**: Longitud del ciclo estacional (por ejemplo, $s = 12$ para datos mensuales con estacionalidad anual).

$$SARIMA \underbrace{(p, d, q)}_{non-seasonal} \underbrace{(P, D, Q)}_{seasonal}{}_m$$

Figura 4: Modelo SARIMA [32].

4. Alcance del proyecto

4.1. Declaración del alcance

Al finalizar el proyecto, se entregará una primera versión funcional del aplicativo de software en forma de un MVP (Minimum Viable Product), diseñado para realizar pronósticos de ventas basados en datos históricos. Un MVP es un prototipo de la aplicación que incluye las funcionalidades esenciales necesarias para validar su viabilidad, pero no todas las características finales o optimizaciones de rendimiento. Este prototipo se enfocará en demostrar el concepto y realizar pruebas iniciales, en lugar de ser una aplicación completa lista para la producción.

Los datos de entrada consistirán en archivos de hojas de cálculo que contengan registros de ventas pasadas, los cuales deberán estar en un formato específico, como archivos CSV o Excel, para ser procesados por el sistema.

El aplicativo contará con una interfaz gráfica web, desarrollada utilizando tecnologías modernas. Esta interfaz permitirá al usuario cargar los archivos de datos, configurar parámetros básicos del modelo predictivo, y visualizar los resultados del pronóstico a través de gráficos y tablas de datos.

4.1.1. Alcance Específico

1. Funcionalidades Principales

- **Cargar y Validar Datos:** La aplicación permitirá al usuario cargar archivos de hojas de cálculo. Se implementará un sistema de validación para asegurar que los datos estén en el formato correcto antes de ser procesados.
- **Modelos Predictivos:** Se implementará un modelo predictivo básico para generar pronósticos de ventas, utilizando técnicas de Machine Learning como la regresión lineal.
- **Visualización de Resultados:** Los resultados del pronóstico se presentarán a través de gráficos (por ejemplo, gráficos de línea para tendencias de ventas), y tablas que permitan una interpretación clara y rápida.

2. Pruebas con Usuarios

- **Pruebas Piloto:** La aplicación será probada por un usuario final, quien proporcionará retroalimentación sobre la funcionalidad, usabilidad y precisión de los pronósticos. Estas pruebas son esenciales para validar la utilidad del MVP y realizar ajustes antes de un posible desarrollo futuro más amplio.
- **Recopilación de Retroalimentación:** Se recopilarán y analizarán los comentarios de estos usuarios para identificar posibles mejoras en el modelo predictivo y la interfaz de usuario.

3. Limitaciones

- **Uso Restringido:** Esta versión de la aplicación no será productiva ni estará abierta al público en general. Su objetivo es servir como una herramienta de prueba y validación interna, enfocada en evaluar la viabilidad del concepto y la efectividad de los modelos predictivos implementados.
- **Precisión del Modelo:** Es importante destacar que los pronósticos generados son aproximaciones basadas en datos históricos y no garantizan precisión absoluta. La exactitud del pronóstico dependerá en gran medida de la calidad y relevancia de los datos de entrada.
- **Formato de Datos:** El sistema aceptará únicamente datos en formatos CSV o Excel, y no se procesarán archivos con formatos distintos o datos incompletos.
- **Capacidad del Sistema:** La primera versión del software estará diseñada para manejar volúmenes de datos pequeños a medianos. No se recomienda su uso para grandes volúmenes de datos hasta que se realicen optimizaciones futuras.

Este MVP está destinado a servir como una herramienta de apoyo y validación conceptual, proporcionando información preliminar que puede ayudar en la toma de decisiones estratégicas. No obstante, se recomienda a los usuarios combinar estos pronósticos con otros métodos de análisis para maximizar su efectividad en la planificación comercial.

4.2. Objetivos

4.2.1. Objetivo general

Desarrollar un aplicativo web utilizando modelos predictivos para optimizar el pronóstico de ventas futuras en pequeños y medianos negocios.

4.2.2. Objetivos Específicos

1. Analizar diferentes datos de ventas que serán requeridos para el entrenamiento del modelo.
2. Diseñar e implementar modelos predictivos utilizando técnicas de machine learning, como la regresión lineal y redes neuronales, optimizando su desempeño para generar pronósticos de ventas futuros con la mayor precisión posible.
3. Desarrollar un prototipo funcional de un aplicativo web que integre los modelos predictivos implementados, permitiendo a los usuarios cargar datos de ventas, ejecutar pronósticos, y visualizar los resultados mediante gráficos.
4. Realizar pruebas para evaluar la usabilidad, precisión de los pronósticos, y la efectividad de la interfaz gráfica, recopilando métricas y retroalimentación para futuras mejoras.

4.3. Resultados esperados

A Continuación se exponen los resultados esperados por cada uno de los objetivos específicos en la tabla 2 : Resultado esperado por objetivo específico.

Objetivo específico	Resultados esperados
Analizar diferentes datos de ventas que serán requeridos para el entrenamiento del modelo.	■ Contar con una amplia colección de datos que serán útiles para el desarrollo del proyecto. ■ Documentación detallada donde se explique los tipos de datos necesarios y su importancia para el desarrollo del proyecto.
Diseñar e implementar modelos predictivos utilizando técnicas de machine learning, como la regresión lineal y redes neuronales, optimizando su desempeño para generar pronósticos de ventas futuros con la mayor precisión posible.	■ Documento detallado que especifique el diseño de los algoritmos.
Desarrollar un prototipo funcional de un aplicativo web que integre los modelos predictivos implementados, permitiendo a los usuarios cargar datos de ventas, ejecutar pronósticos, y visualizar los resultados mediante gráficos.	■ Contar con el código fuente del software. ■ Contar con un prototipo del aplicativo de software con todas las características implementadas e interfaz gráfica para hacer pronósticos de ventas.
Realizar pruebas para evaluar la usabilidad, precisión de los pronósticos, y la efectividad de la interfaz gráfica, recopilando métricas y retroalimentación para futuras mejoras.	■ Documento detallado con gráficos del desempeño y certeza del aplicativo desarrollado.

Tabla 2: Resultado esperado por objetivo específico.

5. Metodología

5.1. Metodología de investigación

La metodología de investigación se muestra en la tabla 3 usando los niveles de madurez tecnológica, cabe resaltar que es una investigación aplicada de base tecnológica.

Actividad	TRL	Característica	Productos representativos
Investigación básica	Investigación aplicada	Entendimiento y análisis de los datos	Artículo de investigación
Investigación básica	Formulación de conceptos	Formulación del modelo	Artículo de investigación
Investigación aplicada	Prueba experimental del concepto	Modelado e implementación del sistema	Artículo de investigación

Tabla 3: Metodología de investigación

A continuación, se presenta la aplicación de la metodología basada en los niveles de madurez tecnológica (TRL), detallando las actividades desarrolladas durante este proyecto: Esta metodología permitió estructurar el proceso investigativo desde la comprensión teórica hasta la implementación práctica de la solución tecnológica:

- **Nivel 1: Investigación básica – Entendimiento y análisis de los datos**

En esta fase se realizó una búsqueda y revisión exhaustiva de literatura científica y técnica en bases de datos académicas como SpringerLink, Google Académico y repositorios universitarios. El objetivo fue identificar trabajos previos relacionados con predicción de series temporales, así como comprender a fondo los fundamentos de modelos como LSTM, SARIMA y SVM. Esta etapa permitió construir una base teórica sólida y entender las características de los datos necesarios para alimentar dichos modelos. En este caso, esta investigación inicial fue fundamental para orientar el enfoque del proyecto y seleccionar las técnicas más apropiadas para el desarrollo.

- **Nivel 2: Investigación básica – Formulación del modelo**

Una vez comprendidas las técnicas, se pasó a la etapa de formulación del modelo, en la cual se definieron los algoritmos que serían implementados en la plataforma web. En esta fase, se decidió cómo estructurar y entrenar modelos de predicción de series temporales, teniendo en cuenta la estructura de los datos, la validación cruzada, el ajuste de hiperparámetros, y las métricas de evaluación como MSE, MAE y MAPE. Esta formulación fue clave para garantizar que los modelos generaran resultados confiables en la interfaz desarrollada.

- **Nivel 3: Investigación aplicada – Prueba experimental del concepto**

Finalmente, esta etapa corresponde al desarrollo práctico y experimental del proyecto: la implementación de la página web. Aquí se integraron los modelos previamente formulados en un entorno funcional, donde los usuarios pueden cargar sus propios archivos y generar predicciones. Este nivel demuestra la aplicación tecnológica del conocimiento adquirido, ya que no solo se desarrollaron los modelos, sino que también se construyó una herramienta web que permite su uso real, cumpliendo con los principios de una investigación aplicada orientada a la solución de problemas concretos.

5.2. Metodología de desarrollo de software

5.2.1. Metodología elegida: Agile.

Esta metodología se basa en desarrollos cortos y continuos, pone importancia en la entrega rápida y se adapta a cambios, lo que quiere decir que permite ajustes frecuentes durante el proceso, lo que promueve la retroalimentación continua y mejora del producto final [33]. Para esta metodología se aplicará el marco de trabajo Scrum.

La elección de esta metodología con el marco de trabajo Scrum se realiza debido al flujo de trabajo ordenado y especialmente por la flexibilidad que ofrece con los cronogramas y las historias de usuario además al tener retroalimentación de los anteriores sprints es posible hacer modificaciones adecuadas para los siguientes sprints para así lograr los objetivos planteados. Cabe aclarar que, aunque el marco de trabajo scrum se oriente al trabajo en equipo, también puede ser usado por una persona con ciertas modificaciones.

5.2.2. Modificaciones aplicadas a Scrum

Debido a que el desarrollo del proyecto se realiza por una sola persona, se hicieron las siguientes adaptaciones al marco de trabajo Scrum para ajustarlo a un entorno individual:

- **Unificación de roles:** En Scrum existen tres roles definidos: *Product Owner*, *Scrum Master* y *Equipo de Desarrollo*. Al tratarse de un proyecto desarrollado por una sola persona, el mismo desarrollador asumió los tres roles, lo que implicó una mayor autogestión y flexibilidad. Esto permitió, por ejemplo, definir requerimientos (como Product Owner), planificar y realizar el seguimiento del trabajo (como Scrum Master), y ejecutar las tareas técnicas (como desarrollador).
- **Sprints flexibles:** Aunque se mantuvo la estructura de sprints de dos semanas, su duración se flexibilizó en función de la complejidad de las tareas asignadas y las limitaciones de tiempo del desarrollador. Esto permitió una mejor adaptación al ritmo real del trabajo.
- **Gestión de historias de usuario:** Las historias de usuario fueron redactadas con un enfoque más técnico de lo habitual, especificando claramente los requerimientos

de desarrollo tanto para el *frontend* como para el *backend*, en lugar de centrarse únicamente en la experiencia del usuario final.

- **Uso de tablero Kanban:** En lugar del tablero Scrum tradicional, que organiza tareas por sprint, se optó por un tablero *Kanban*, más adecuado para un flujo de trabajo individual y continuo.

5.2.3. Gestión del Proyecto

Para la planificación y seguimiento del desarrollo se empleó inicialmente la herramienta **Jira**, ampliamente utilizada en entornos profesionales por su versatilidad y compatibilidad con Scrum. No obstante, al agotarse el periodo de prueba gratuita y considerando las restricciones presupuestarias del proyecto, se migró a una alternativa gratuita: **GitHub Projects**.

GitHub Projects permite organizar tareas mediante tableros *Kanban*, integrados directamente al repositorio del código fuente, lo cual facilita una gestión unificada del código y las tareas. Este tablero se divide en columnas que representan el estado de las tareas:

- **To Do:** Tareas planificadas que aún no han sido iniciadas.
- **In Progress:** Tareas en proceso de desarrollo.
- **Done:** Tareas finalizadas y validadas.

Cada tarea está redactada como una historia de usuario, lo que permite definir con claridad qué se quiere lograr, para quién y con qué propósito. Por ejemplo: “Como desarrollador, quiero entrenar un modelo de predicción para obtener una métrica de precisión aceptable.”

Se definieron **sprints** de dos semanas, en los cuales se priorizaron tareas relacionadas con tres grandes componentes del proyecto:

1. Diseño y entrenamiento de modelos predictivos (ARIMA, LSTM, SVR).
2. Implementación del backend y la arquitectura de datos.
3. Desarrollo del frontend para visualización de resultados.

Durante cada sprint se realizó una revisión de las tareas completadas y una planificación del siguiente ciclo, ajustando prioridades y corrigiendo errores según fuera necesario. Esta dinámica facilitó mantener un ritmo constante de trabajo y permitió detectar y resolver cuellos de botella a tiempo.

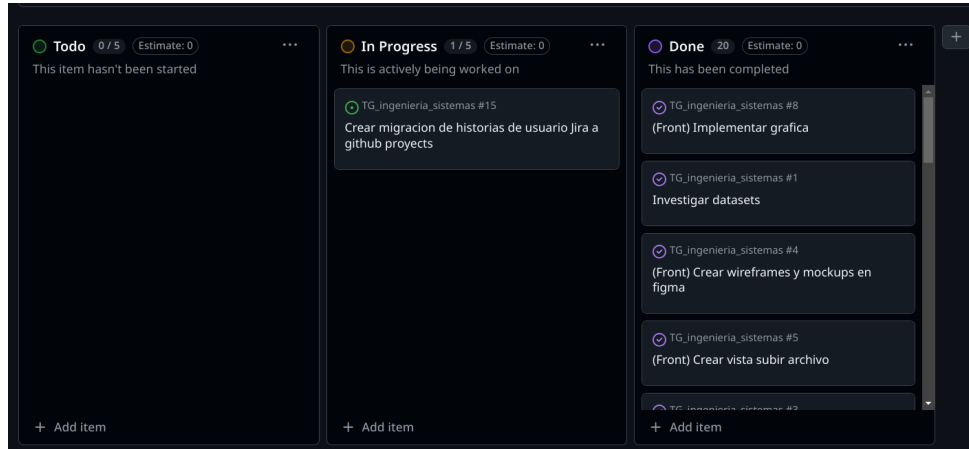


Figura 5: Tablero Kanban utilizado en GitHub Projects

Finalmente, se pone a disposición el enlace al tablero de tareas que documenta de forma transparente el avance del desarrollo como se muestra en el anexo A en la sección anexos 13.

6. Datasets

Para el desarrollo y entrenamiento de los modelos predictivos fue necesaria hacer una recopilación y análisis de diversos datasets que contengan los datos necesarios, esta parte es primordial para el proyecto ya que influye en la precisión y ajuste del modelo.

A continuación se analizaron distintas fuentes, los datasets recopilados como candidatos, sus características, qué criterios se usaron para el descarte y finalmente cuáles serán usados para el desarrollo del proyecto.

6.1. Fuente de los datos

Para la recopilación de los conjuntos de datos necesarios para el desarrollo del proyecto, se recurrió a fuentes ampliamente reconocidas por la comunidad científica y de libre acceso. Se priorizaron plataformas confiables que ofrecen datasets de calidad y diversidad temática, facilitando así el análisis y modelado de datos con fines de investigación y desarrollo.

6.1.1. Kaggle

Kaggle es una de las plataformas más reconocidas y utilizadas en el ámbito de la ciencia de datos y el aprendizaje automático. Esta comunidad en línea, propiedad de Google, proporciona acceso gratuito a miles de conjuntos de datos de distintas categorías y niveles de complejidad. Además, permite a los usuarios compartir sus propios datasets y participar en competencias de modelado predictivo, lo que contribuye a enriquecer constantemente la base de datos disponible [34].

6.1.2. Data.gov

Data.gov es el repositorio oficial de datos abiertos del gobierno de los Estados Unidos. Este portal ofrece una amplia variedad de recursos, incluyendo políticas públicas, herramientas analíticas, estudios de caso y conjuntos de datos provenientes de diferentes agencias gubernamentales. Su objetivo principal es promover la transparencia, la innovación y el aprovechamiento de los datos públicos para investigaciones, desarrollos tecnológicos y toma de decisiones basada en evidencia [35].

6.2. Tabla de datasets evaluados

	Nombre	Registros	Características	Fuente	Link	¿Usado?
1	Supermarket Sales	1000	Registra las ventas en tres sucursales de supermercados, incluyendo tipo de cliente, línea de productos, método de pago, hora de compra y calificación del servicio.	Kaggle	Enlace	No
2	Retail Sales Dataset	1000	Proporciona datos sobre ventas minoristas y características de clientes, permitiendo análisis de tendencias de compra, comportamiento del consumidor y segmentación de mercado.	Kaggle	Enlace	No
3	Customer Shopping Dataset	99457	Recopila información de compras en centros comerciales de Estambul entre 2021 y 2023. Incluye ID de cliente, edad, género, métodos de pago, categorías de productos, cantidad, precio y fechas de compra.	Kaggle	Enlace	Sí
4	Bakery	9684	Datos de transacciones en una panadería de Edimburgo. Contiene pedidos en línea, fecha y hora de compra, número de transacción y cantidad de productos vendidos entre 2016 y 2017.	Kaggle	Enlace	No
5	Pizza Sales Dataset	48600	Este conjunto de datos contiene información detallada sobre pedidos de pizza, incluyendo detalles sobre las variantes de pizza, cantidades, precios, fechas, horarios y categorización.	Kaggle	Enlace	Si
6	Warehouse and Retail Sales	307646	Este conjunto de datos contiene una lista de ventas y datos de movimiento por artículo y departamento, actualizados mensualmente.	data.gov	Enlace	No

Tabla 4: Lista de Datasets obtenidos después de la investigación

6.2.1. Criterios de selección:

Para seleccionar los datasets que serán utilizados para el entrenamiento de los modelos predictivos se consideraron los siguientes criterios:

- **Tema:** El dataset debe contener datos de ventas.

- **Variables:** Se requieren dos variables claves:
 - Fecha de la transacción.
 - Cantidad de artículos vendidos.
- Se aceptan categorías en lugar de artículos, ya que representan un conjunto más grande de productos.
- **Consistencia de los datos:** No debe haber brechas en las fechas donde los valores sean nulos.
- **Cantidad de registros:** El dataset debe contener al menos un año de datos para cubrir todos los meses y analizar su comportamiento.

6.3. Análisis de la selección de datasets

Después de analizar las columnas y datos de los datasets candidatos se decidió que se descartan los datasets 1 y 2, ya que no cuentan con fechas, también se descarta el database 4 ya que no contiene información continua de alguna de sus categoría por mínimo un año.

Se concluyó que se opta por utilizar los datasets 3 y 5 para el desarrollo del proyecto, esto debido a que pasan los criterios y por ende contienen las variables que se necesitan, ambos contienen las fechas de las transacciones, contienen artículos o categorías, cantidad vendida y contienen la cantidad de datos necesarios.

6.4. Análisis de los Datasets

6.4.1. Retail Sales Dataset

El dataset contiene un total de 1000 registros y abarca distintas categorías como ropa (*clothing*), libros (*books*), cosméticos (*cosmetics*), y alimentos y bebidas (*Food & Beverage*). Las columnas más relevantes para este análisis son: **cantidad vendida**, **fecha de la factura** y **categoría**.

En la Figura 6 se puede observar un histograma que muestra las categorías con la cantidad total vendida durante el periodo comprendido entre el 31 de diciembre de 2022 y el 31 de diciembre de 2023. Se destaca que la categoría más vendida fue *clothing*, seguida por *cosmetics* y *Food & Beverage*. La categoría menos vendida fue *books*.

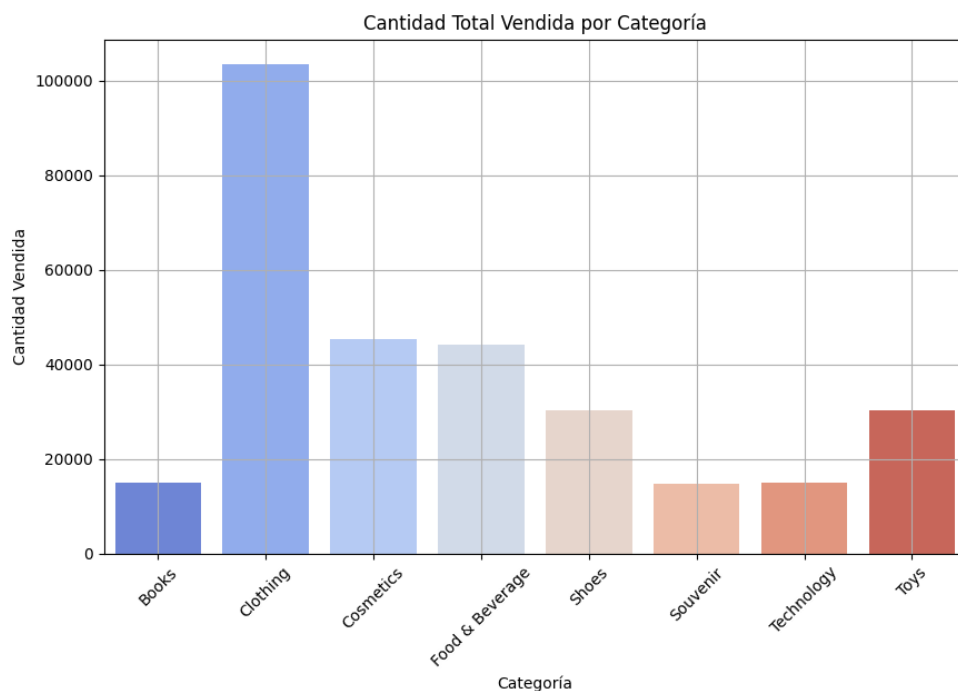


Figura 6: Histograma de ventas por categoría entre 2022-12-31 y 2023-12-31.

A continuación, se realiza un análisis detallado de la categoría *clothing*. En la Figura 7 se muestra un gráfico de series temporales que ilustra el comportamiento de las ventas diarias. Aunque el comportamiento es variable, se observa una media de aproximadamente 140 ventas por día.

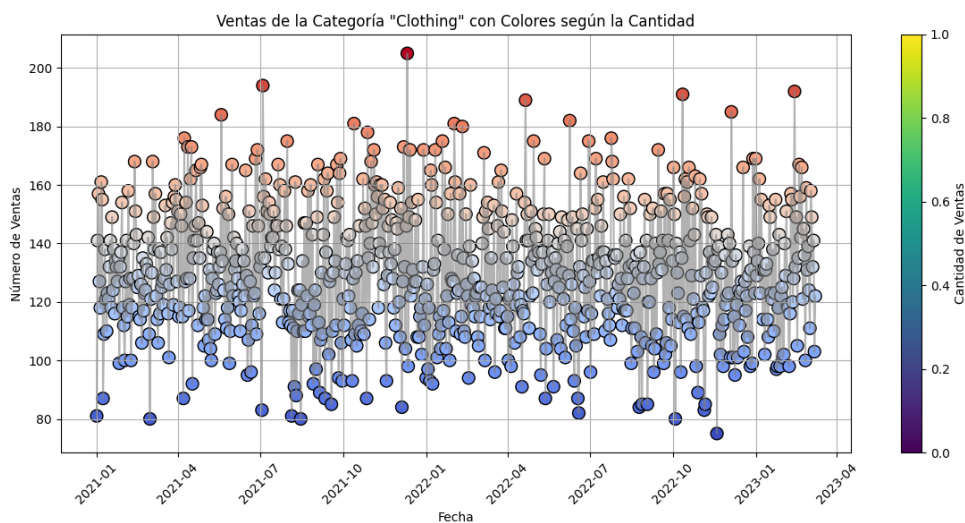


Figura 7: Serie temporal de ventas diarias para la categoría *clothing*.

6.4.2. Dataset Pizzería

Este dataset contiene 48,621 registros que abarcan desde el 31 de diciembre de 2014 hasta el 11 de diciembre de 2015. Incluye distintos tipos de pizzas, siendo las columnas clave: **nombre de la pizza** y **fecha de venta**.

Primero se realizó una normalización en la columna de fechas, ya que se presentaban dos formatos distintos.

En la Figura 9 se presenta la cantidad total de pizzas vendidas. Se observa que la pizza *The Classic Deluxe Pizza* fue la más vendida en todo el periodo.

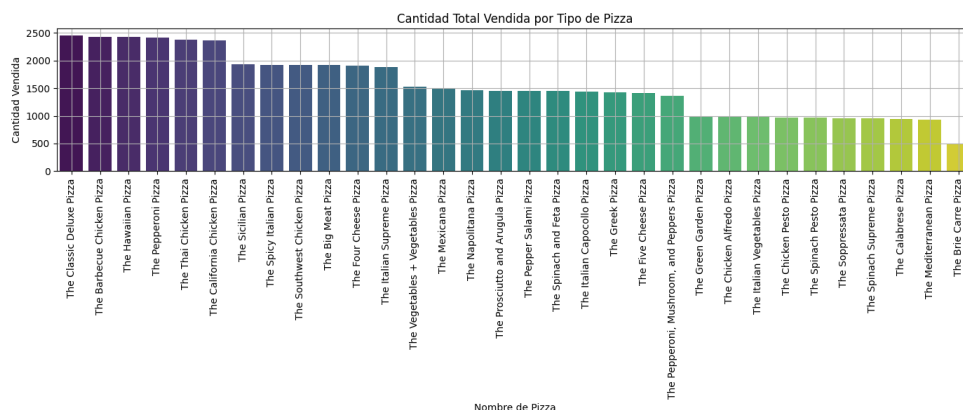


Figura 8: Total de pizzas vendidas por tipo.

A continuación, se realiza un análisis específico de la pizza *The Classic Deluxe Pizza*, que fue la de mayor volumen de ventas durante el periodo estudiado.

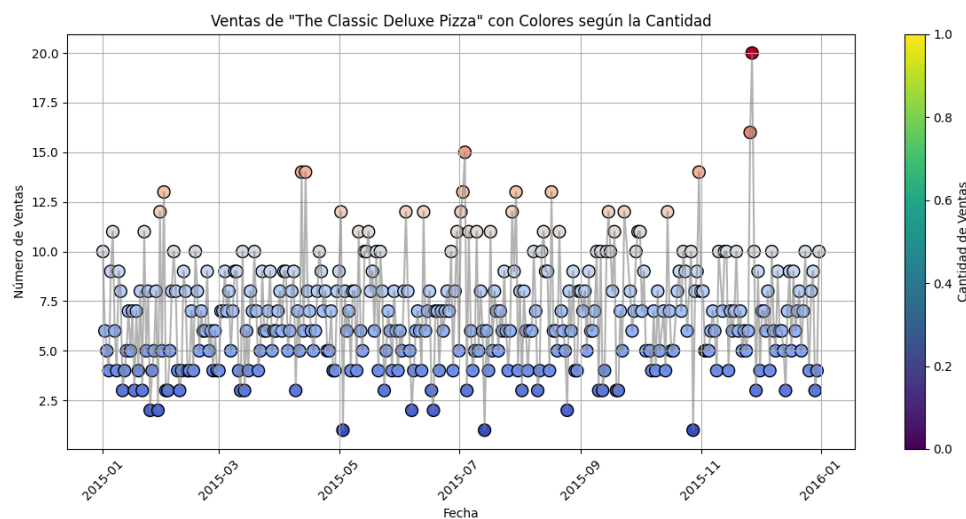


Figura 9: Total de pizzas vendidas por tipo.

7. Modelos Entrenados

Se presentan los tres modelos que se utilizarán para la predicción de las ventas: **LSTM**, **SVM** y **SARIMA**. Cada modelo tiene características que los hacen adecuados para el análisis de datos de series temporales. Se utiliza el dataset *pizza* con los datos de "The Classic Deluxe Pizza", como referencia para comparar los modelos. Las métricas utilizadas son:

- **MSE (Mean Squared Error)**: mide el promedio de los errores al cuadrado.
- **MAE (Mean Absolute Error)**: calcula el valor absoluto de los errores.
- **MAPE (Mean Absolute Percentage Error)**: mide el error en porcentaje relativo.

A continuación, se revisan cada uno de los modelos, su justificación, implementación y resultados.

7.1. Modelo 1: LSTM

El primer modelo es una red neuronal recurrente especializada llamada **Long Short-Term Memory (LSTM)**. Esta puede identificar dependencias temporales de largo plazo en datos secuenciales, como las ventas históricas. A diferencia de modelos tradicionales de series temporales, LSTM mitiga el problema del desvanecimiento del gradiente, lo cual le permite aprender patrones complejos y no lineales presentes en los datos recolectados. Además, se pueden ajustar los pesos para mejorar su desempeño, en la figura 10 se muestra un entrenamiento hecho con lstm en el que la línea naranja representa el entrenamiento y la línea azul los datos reales, en el que se logra evidenciar un mejor ajuste de la predicción a los datos reales con respecto al modelo svm.

Implementación:

- **Primera capa LSTM**: 100 unidades, `return_sequences=True`, función de activación `tanh`, `stateful=False`.
- **Primera capa Dropout**: tasa de 20 %, para evitar sobreajuste.
- **Segunda capa LSTM**: 100 unidades, `return_sequences=False`, función de activación `tanh`, `stateful=False`.
- **Segunda capa Dropout**: tasa de 20 %.
- **Capa de salida**: capa densa con una unidad, activación lineal.
- **Compilación**: optimizador Adam con `learning rate = 0.001`, función de pérdida `mean_squared_error`.
- **Entrenamiento**: 200 épocas, batch size de 2, 80 % entrenamiento, 20 % prueba.

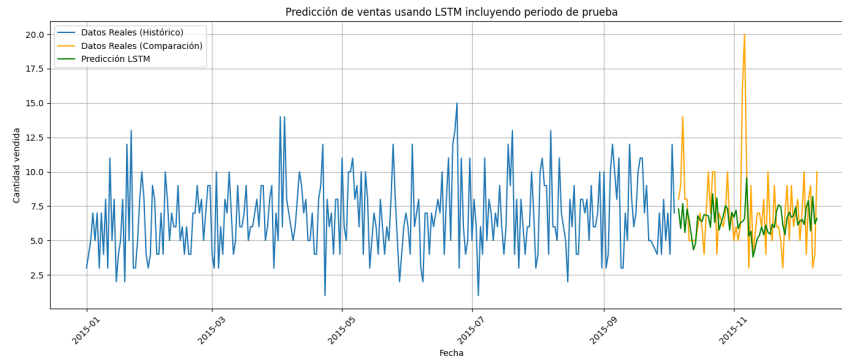


Figura 10: Comparación con datos reales del dataset ventas de pizza contra predicción hecha por el modelo LSTM.

7.2. Modelo 2: SVM

Las **Máquinas de Soporte Vectorial (SVM)** no están diseñadas específicamente para series temporales, pero pueden adaptarse mediante la creación de ventanas temporales. En este caso, se utilizó una ventana `lag = 7` (7 días).

Implementación: Se hace uso de la librería Keras para facilitar la creación y entrenamiento del modelo, a continuación se presenta la arquitectura del modelo y se usa 80

Se hace uso de un Grid Search para buscar los mejores hiperparámetros `c`, `gamma` y `epsilon`. Se definen los parámetros del grid search de la siguiente manera para hacer la búsqueda.

- Se usó `GridSearchCV` para encontrar los mejores hiperparámetros: `C`, `gamma` y `epsilon`.
- **Parámetros explorados:**

```
param_grid = {
    'C': [10, 100, 500],
    'gamma': [0.001, 0.01, 0.1],
    'epsilon': [0.01, 0.1, 1]
}
```

A continuación, se realizan pruebas utilizando tres tipos de kernels en el modelo SVM: RBF (Radial Basis Function), lineal y sigmoide. Esto con el fin de evaluar y comparar el rendimiento de cada kernel para determinar cuál se adapta mejor a los datos y ofrece mejores resultados en términos de precisión y capacidad predictiva.

RBF (Radial Basis Function): en la figura 11 se muestra los resultados del entrenamiento usando el kernel RBF, en la gráfica está en el eje x la fecha y en el eje y la cantidad de pizzas vendidas. La línea de color azul en la gráfica representa los datos reales, el color naranja representa los datos reales y el color verde representa los datos del entrenamiento, los datos del entrenamiento comienzan a partir del mes d de octubre, se puede evidencia una tendencia de la gráfica hacia la media tratando de ajustarse a los picos..

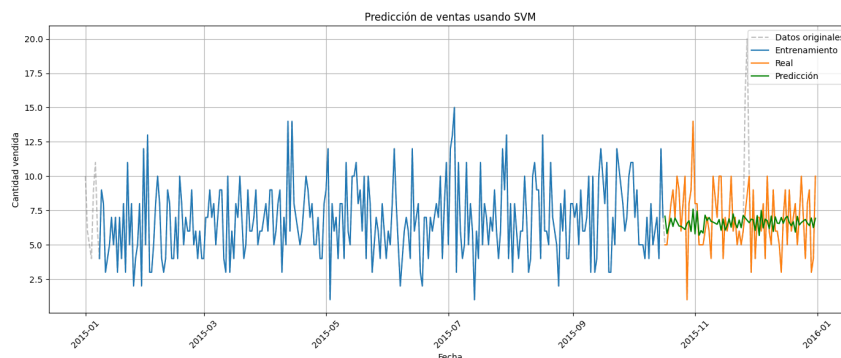


Figura 11: Comparación con datos reales del dataset ventas de pizza contra predicción hecha por el modelo SVM con el kernel RBF.

Lineal: en la figura 12 se muestra el resultado obtenido del entrenamiento con el kernel lineal, se puede apreciar un comportamiento similar al visto en la gráfica del kernel rbf, la gráfica tiende hacia la media y trata de ajustarse a los picos pero de una manera suavizada y más cercana a la media.icos pero de una manera suaviazada y mas cercana a la media.

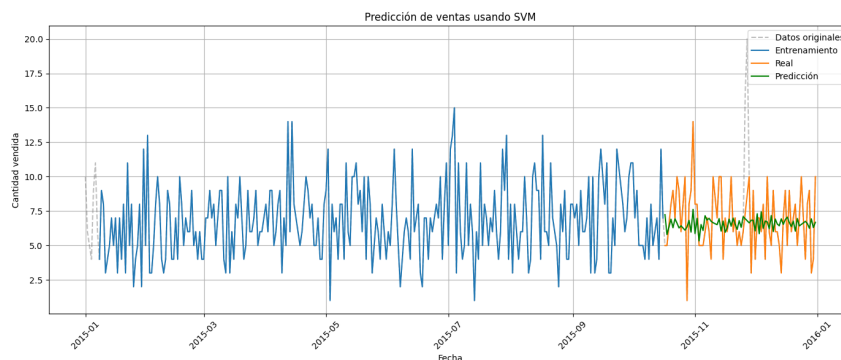


Figura 12: Comparación con datos reales del dataset ventas de pizza contra predicción hecha por el modelo SVM con el kernel lineal.

Sigmoide: finalmente se realiza el entrenamiento con el kernel sigmoide, en la figura 13 se muestran los resultados obtenidos, nuevamente se evidencia un comportamiento de la grafica tendiendo a la media.

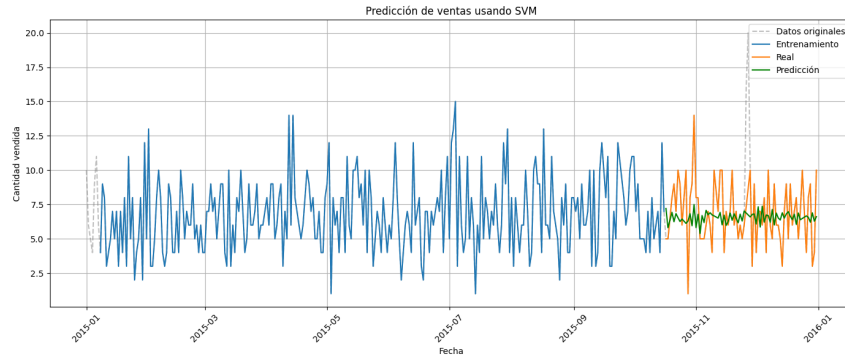


Figura 13: Comparación con datos reales del dataset ventas de pizza contra predicción hecha por el modelo SVM con el kernel sigmoide.

Resultados de comparación entre kernels:

Kernel	MSE	MAE	MAPE	Tiempo de Ejecución
RBF	5.570	1.932	37.687 %	3.5 seg
Lineal	5.536	1.931	37.325 %	208 seg
Sigmoide	5.564	1.937	37.208 %	2.6 seg

Tabla 5: Comparación de kernels en el modelo SVM.

Conclusión: Se selecciona el **kernel sigmoide** debido a su menor MAPE y mejor tiempo de ejecución.

7.3. Modelo 3: SARIMA

El modelo **SARIMA (Seasonal Autoregressive Integrated Moving Average)** es una extensión de ARIMA que incorpora estacionalidad. Es ideal para analizar datos que presentan patrones repetitivos a lo largo del tiempo (diarios, mensuales, anuales).

Para SARIMA se hace un grid search para encontrar el valor óptimo se usan los siguientes rangos:

$$\begin{array}{ll}
 p, d, q \in \{1, 2\} & \text{(ARIMA: orden no estacional)} \\
 P, D, Q \in \{1, 2\} & \text{(ARIMA: orden estacional)}
 \end{array}$$

Justificación: En los datos se observan variaciones dependiendo del día de la semana, semana del mes y época del año, por lo cual SARIMA es apropiado para capturar dichas variaciones. A continuación, en la figura 14 se muestra un entrenamiento de prueba utilizando este modelo, en el que se logra evidenciar un mejor ajuste de la predicción a los datos reales con respecto al modelo svm.

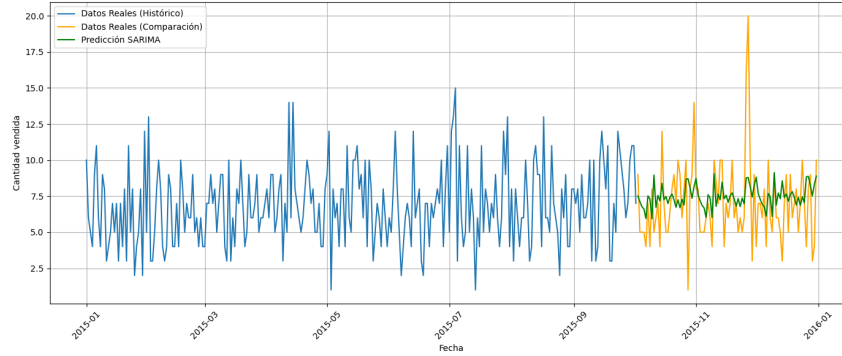


Figura 14: Comparación con datos reales del dataset ventas de pizza contra predicción hecha por el modelo SARIMA.

7.4. Resumen de Resultados

A continuación en la tabla 6, se muestra un resumen con los resultados obtenidos por los tres modelos:

Modelo	MSE	MAE	MAPE
LSTM	10.423	2.274	35.053 %
SVM (Sigmoide)	5.564	1.937	37.208 %
SARIMA	7.062	1.937	40.04 %

Tabla 6: Comparación final de métricas entre modelos.

Podemos concluir que:

- El modelo **SVM (Sigmoide)** presenta el menor error cuadrático medio ($MSE = 5.564$) y el menor error absoluto medio ($MAE = 1.937$), lo cual indica una mayor precisión en términos absolutos.
- El modelo **LSTM**, aunque tiene el mayor MSE (10.423) y MAE (2.274), logra el **menor error porcentual absoluto medio** ($MAPE = 35.053 \%$), lo cual sugiere un mejor rendimiento relativo respecto al tamaño de las predicciones.
- El modelo **SARIMA** se ubica en un punto intermedio con un MSE de 7.062 y el mismo MAE que SVM (1.937), pero registra el **MAPE más alto** (40.04 %), lo cual indica menor precisión relativa.

8. Prototipo

8.1. Implementación

La implementación de los algoritmos, así como la del prototipo, se realizó bajo la metodología **Scrum**, se hizo uso de la herramienta. Esta implementación se divide en las siguientes partes:

- Creación de base de datos.
- Creación del backend.
- Interfaz gráfica.

8.2. Creación de la base de datos

En esta etapa se eligió un motor de base de datos. La elección final fue **MongoDB**, dado que al ser un motor NoSQL permite trabajar con documentos en formato JSON, lo cual facilita el almacenamiento de información proveniente de los datasets y de los resultados obtenidos por los diferentes modelos predictivos.

La base de datos se organiza en las siguientes colecciones:

- **users**: contiene la información de los usuarios, como nombre, correo, contraseña e identificador.
- **files**: almacena los datos de las ventas antes de ser predichas.
- **sarima_trains**: contiene los resultados de las predicciones realizadas con el modelo SARIMA.
- **lstm_trains**: contiene los resultados de las predicciones realizadas con el modelo LSTM.
- **svm_trains**: contiene los resultados de las predicciones realizadas con el modelo SVM.
- **datasets**: relaciona la colección **files** con las diferentes predicciones hechas por los modelos SARIMA, LSTM y SVM, así como con el usuario correspondiente. Además, incluye el nombre, descripción y fecha de creación del dataset.

8.3. Creación del backend

En esta parte se desarrolló el backend del prototipo, encargado de gestionar toda la lógica y las APIs consumidas por el frontend.

Las APIs implementadas son:

- `/upload-file`: valida y guarda en la base de datos los archivos con datos de ventas.

- `/get-databases`: obtiene una lista de los datasets almacenados en la base de datos, permitiendo aplicar filtros por fecha, usuario y orden.
- `/get-database-by-id`: permite obtener un dataset específico de la base de datos a través de su identificador.
- `/get-file`: esta API contiene varios flujos lógicos. Si la variable `file_train_id` es diferente de `null`, se buscará en la base de datos una predicción ya realizada anteriormente utilizando el modelo especificado en la variable `model_type`. En caso contrario, el sistema ejecutará un nuevo entrenamiento y predicción usando el modelo correspondiente indicado en `model_type`.

El backend está hecho en python usando el framework django y se usaron las siguientes librerías como se muestra en la tabla 7.

Para el control de versiones y la gestión del desarrollo del proyecto, se utilizó la plataforma *GitHub*. El repositorio con el código fuente y la documentación desarrollados por el autor se encuentra disponible en el anexo B en la sección anexos 13.

Librería / Módulo	Descripción y Uso
<code>numpy</code>	Permite realizar operaciones matemáticas y manipulación de arreglos numéricos.
<code>pandas</code>	Facilita el manejo de estructuras de datos tipo DataFrame.
<code>tensorflow.keras.models.Sequential</code>	Permite construir modelos de redes neuronales de forma secuencial, como LSTM.
<code>tensorflow.keras.layers.LSTM</code>	Capa de red neuronal recurrente ideal para procesar series temporales.
<code>tensorflow.keras.layers.Dense</code>	Capa completamente conectada usada al final del modelo para la predicción.
<code>tensorflow.keras.layers.Dropout</code>	Capa que reduce el sobreajuste al apagar aleatoriamente neuronas durante el entrenamiento.
<code>tensorflow.keras.optimizers.Adam</code>	Optimizador eficiente y adaptativo comúnmente usado en redes neuronales profundas.
<code>tensorflow.keras.optimizers.SGD</code>	Optimizador clásico basado en descenso del gradiente estocástico.
<code>tensorflow.keras.optimizers.RMSprop</code>	Optimizador adaptativo útil en tareas con datos secuenciales como LSTM.
<code>tensorflow.keras.optimizers.Adagrad</code>	Optimizador que adapta la tasa de aprendizaje a cada parámetro.
<code>sklearn.preprocessing.MinMaxScaler</code>	Escala características numéricas entre un rango definido, generalmente [0, 1], para mejorar el rendimiento de los modelos.
<code>sklearn.metrics.mean_squared_error</code>	Mide el error cuadrático medio, útil para evaluar modelos regresivos. Penaliza errores grandes.
<code>sklearn.metrics.mean_absolute_error</code>	Mide el error absoluto medio, útil para evaluar la precisión general del modelo.
<code>sklearn.svm.SVR</code>	Implementación de máquinas de soporte vectorial para tareas de regresión. Útil cuando los datos no son lineales.
<code>sklearn.model_selection.GridSearchCV</code>	Busca de forma automatizada los mejores hiperparámetros para un modelo, con validación cruzada.
<code>datetime, timedelta</code>	Manejo de fechas y tiempos, útil para manipular y dividir series temporales.
<code>statsmodels.api</code>	Biblioteca de modelado estadístico que proporciona herramientas como SARIMAX.
<code>statsmodels.tsa.statespace.SARIMAX</code>	Modelo estadístico de series temporales con soporte para componentes estacionales y variables exógenas.
<code>itertools.product</code>	Genera combinaciones cartesianas de parámetros, útil para búsquedas de hiperparámetros.

Tabla 7: Resumen de librerías y módulos utilizados en el backend del proyecto

8.4. Creación del frontend

La interfaz gráfica de usuario fue implementada como una aplicación web que interactúa directamente con las API del sistema backend. Para su diseño inicial, se elaboraron *mockups* que sirvieron como guía visual y funcional durante el proceso de desarrollo.

Los mockups se realizaron con la herramienta figma, el enlace puede ser consultado en el anexo C en la sección anexos 13.

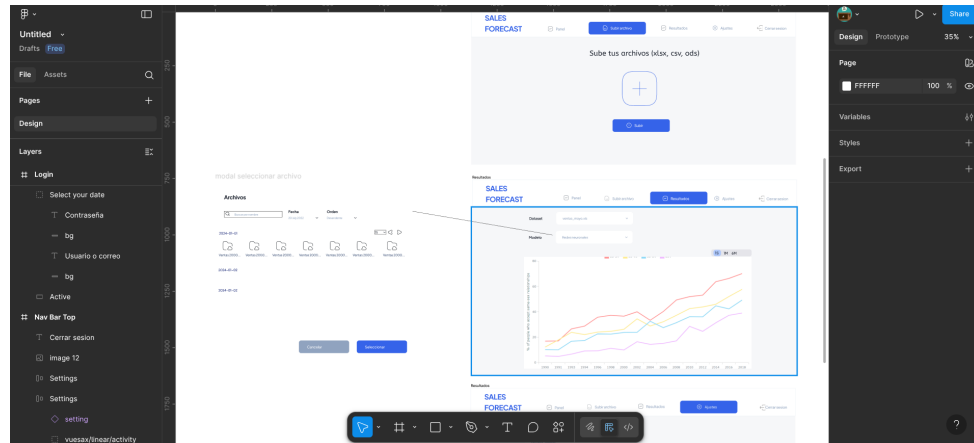


Figura 15: Tablero herramienta figma

A continuación se hace un repaso de las diferentes vistas:

8.4.1. Inicio de sesión

El sistema cuenta con una pantalla de autenticación donde los usuarios deben ingresar sus credenciales (nombre de usuario y contraseña) para acceder a las funcionalidades disponibles. Esta etapa garantiza el acceso seguro y personalizado a la plataforma.

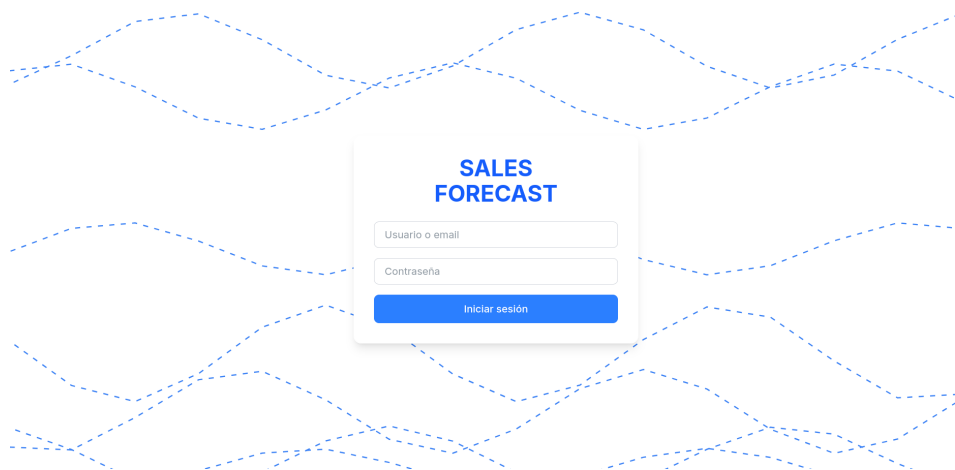


Figura 16: Pantalla de inicio de sesión del usuario

8.4.2. Panel principal

Una vez autenticado, el usuario accede al panel principal, donde se presenta un resumen de las actividades realizadas. En esta vista se puede consultar información sobre los entrenamientos previamente ejecutados, así como los conjuntos de datos cargados por el usuario.



Figura 17: Vista general del panel principal

8.4.3. Carga de archivos

La sección destinada a la carga de archivos permite al usuario subir nuevos datasets al sistema. Estos archivos serán utilizados posteriormente para entrenar los modelos predictivos. La interfaz de carga fue diseñada para facilitar el proceso y garantizar una experiencia intuitiva.

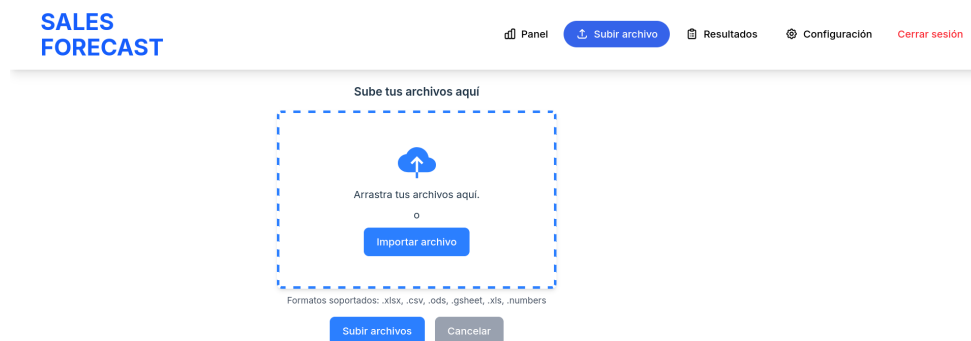


Figura 18: Interfaz para la carga de archivos

8.4.4. Visualización de resultados

En esta sección, el usuario tiene la posibilidad de ejecutar entrenamientos de modelos de predicción. Se puede seleccionar uno de los datasets previamente cargados y elegir el modelo con el cual se desea trabajar. Además, la interfaz ofrece una serie de parámetros avanzados para personalizar el proceso de entrenamiento ajustando los hiperparámetros del modelo seleccionado.

Una vez finalizado el entrenamiento, los resultados se presentan de forma visual mediante una gráfica comparativa: la línea azul representa los datos reales y la línea naranja, los datos estimados por el modelo. En la parte superior se muestran las métricas de evaluación que permiten medir el desempeño del modelo.

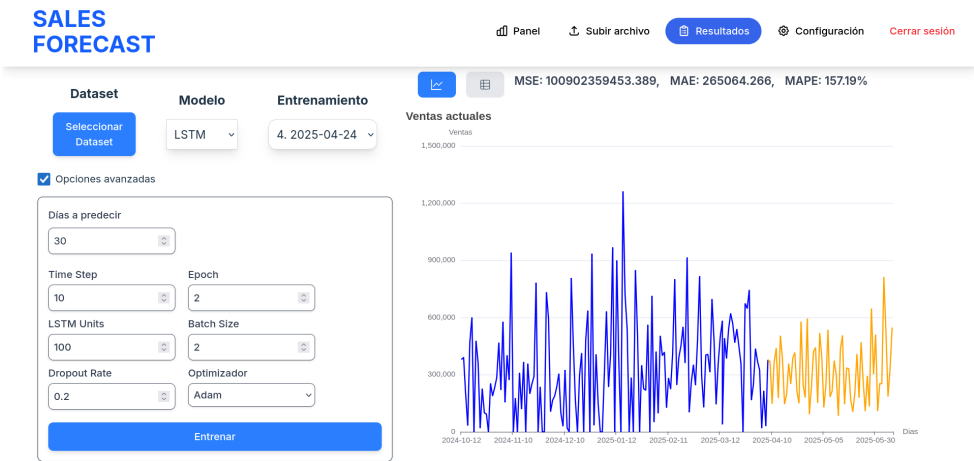


Figura 19: Gráfica de comparación entre valores reales y predichos

Adicionalmente, el sistema ofrece una opción para visualizar los resultados en formato tabular, permitiendo una inspección más detallada de los valores predichos frente a los datos reales.

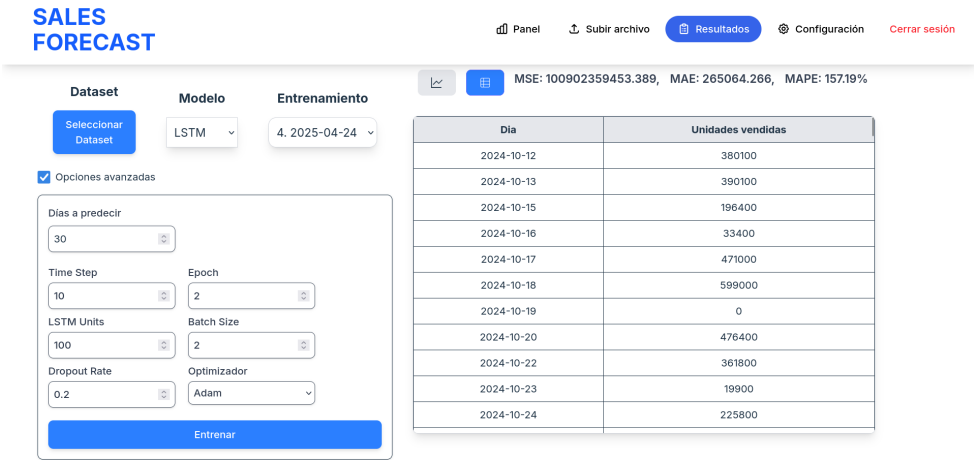


Figura 20: Vista tabular de los resultados del modelo

8.5. Flujo del prototipo para hacer una predicción

En resumen, prototipo sigue un proceso sencillo dividido en cinco pasos. Primero, el usuario carga un archivo con la información que desea analizar. Luego, el sistema lee ese archivo y guarda los datos de forma ordenada. Después, se elige el modelo que se usará para hacer las predicciones. En el siguiente paso, el sistema aprende a partir de esos datos y genera un resultado. Finalmente, ese resultado se presenta de manera visual, ya sea en una tabla o en una gráfica, para que sea fácil de entender. A continuación en la figura 21 se muestra un diagrama del flujo.

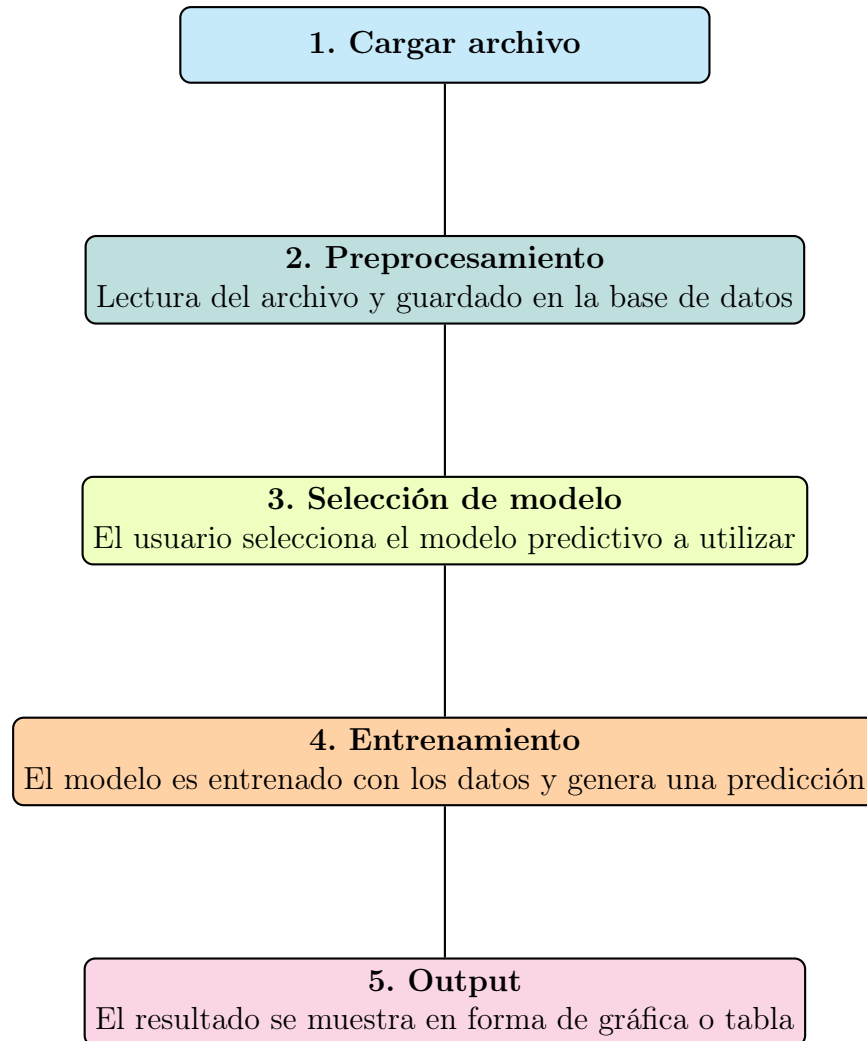


Figura 21: Flujo del prototipo del sistema de predicción

9. Pruebas

9.1. Prueba de modelos de predicción con datos de negocio local de ensalada.

Se llevaron a cabo pruebas con datos reales de un comercio local dedicado a la venta de ensaladas. La prueba se enfocó en un tipo de producto en específico llamado *ensalada dulce*. Se recopilaron los datos de las ventas diarias de este producto durante el periodo de tiempo comprendido entre el 7 de octubre del 2024 y el 30 de marzo del 2025.

Se generó una gráfica con los datos recolectados durante este periodo. En ella se observa una variación en las ventas, con patrones que sugieren fluctuaciones semanales y una disminución progresiva a partir de febrero de 2025. Una característica importante que se debe resaltar es que los días domingos no hay servicio, por lo cual las ventas son cero en esos días. En la siguiente figura se muestra una gráfica con los datos recopilados.

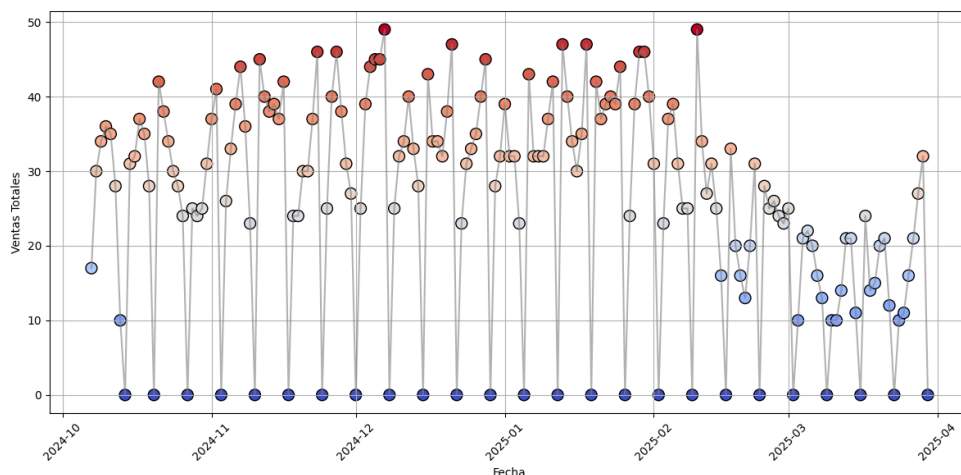


Figura 22: Histograma ventas de ensalada dulce, octubre 2024 a Marzo 2025

Los datos están almacenados dentro de una hoja de cálculo en formato `xlsx`, con dos columnas: `fecha` y `ventas totales` como se muestra en la siguiente tabla 8.

fecha	ventas totales
07/10/2024	17
08/10/2024	30
09/10/2024	34
10/10/2024	36
11/10/2024	35
12/10/2024	28
13/10/2024	10
14/10/2024	0
15/10/2024	31
16/10/2024	32
17/10/2024	37
18/10/2024	35
19/10/2024	28
20/10/2024	0
21/10/2024	42
22/10/2024	38
23/10/2024	34
24/10/2024	30
25/10/2024	28
26/10/2024	24
27/10/2024	0
28/10/2024	25
29/10/2024	24
30/10/2024	25

Tabla 8: Muestra de ejemplo cantidad vendida de ensaladas dulces diarias desde octubre 7 a octubre 30 del 2024

Resultados Iniciales

Se procedió a cargar el archivo en el prototipo, cabe recalcar que las métricas obtenidas por los modelos son calculadas en base a, el 80 % de los datos son usados para entrenamiento y el 20 % restantes para pruebas de este 20 % es donde se obtienen como una metrica preliminar, la linea azul representa los datos reales y la linea naranja los datos pronosticados, se obtuvieron los siguientes resultados usando los valores por defecto de cada modelo:

SVM: Se generan los datos usando el modelo SVM en su configuración por defecto y se obtienen los siguientes resultados como se muestra en la figura 23.

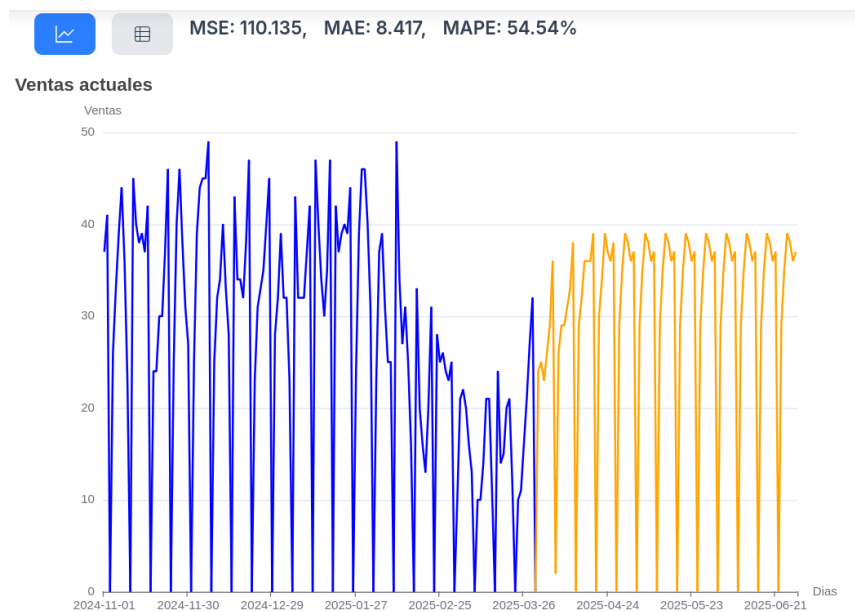


Figura 23: Predicción SVM ventas de ensalada dulce abril a mayo 2025

LSTM: También se evaluó el modelo LSTM, obteniendo los resultados que se muestran a continuación en la figura 24

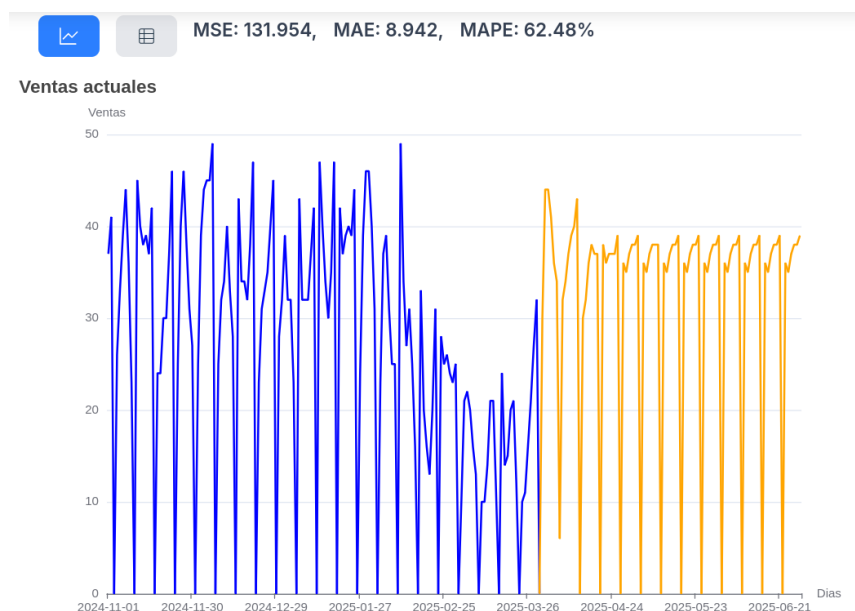


Figura 24: Predicción LSTM ventas de ensalada dulce abril a mayo 2025

SARIMA Se graficaron los resultados iniciales del modelo SARIMA en la figura 25, el cual desde el primer momento mostró un mejor ajuste comparado con los otros modelos.

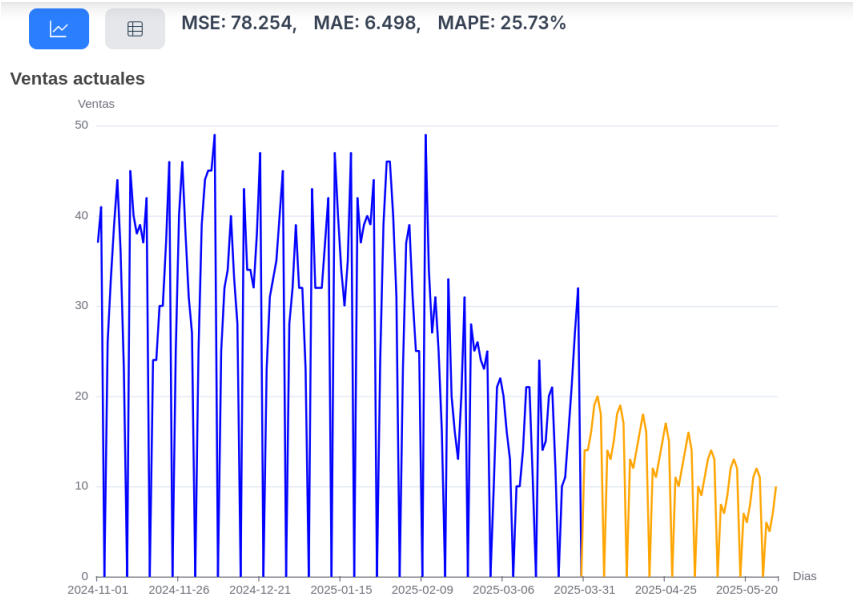


Figura 25: Predicción SARIMA ventas de ensalada dulce abril a mayo 2025

De este primer acercamiento, se concluye que el modelo SARIMA mostró un mejor comportamiento general en comparación con SVM y LSTM. En la siguiente tabla se muestran los resultados obtenidos por parte de los tres modelos.

Modelo	MSE	MAE	MAPE
SVM	110.135	8.417	54.54 %
LSTM	131.954	8.942	62.48 %
SARIMA	78.254	6.498	25.73 %

Tabla 9: Comparativo de Modelos de Predicción de Ventas de ensalada dulce

9.1.1. Entrenamientos adicionales con opciones avanzadas.

Se exploraron otras configuraciones personalizadas en cada uno de los modelos y se hicieron pruebas en la tabla 10 se muestran los datos de entrada y resultados obtenidos de estos entrenamientos.

Modelo	Configuración	MSE	MAE	MAPE (%)
SVM	$C=1$, $\gamma=0.1$, $\epsilon=5$, kernel=RBF	334.337	16.144	94.96 %
SVM	$C=1$, $\gamma=0.1$, $\epsilon=5$, kernel=RBF	338.5	16.265	95.72 %
LSTM	epochs=300, units=100, batch size=2, optimizador=Adam	147.447	9.846	70.47 %
LSTM	epochs=300, units=100, batch size=10, optimizador=Adam	51.624	5.537	45.43 %
LSTM	epochs=200, units=100, batch size=2, optimizador=SGD	238.915	13.093	76.11 %
SARIMA	(1,1,1)(1,1,0)[7]	98.106	7.236	28.41 %
SARIMA	(2,1,2)(1,2,0)[7]	94.509	6.69	26.05 %
SARIMA	(2,2,2)(1,2,1)[7]	214.245	8.188	30.19 %

Tabla 10: Comparación de modelos y métricas de predicción

9.1.2. Evaluación con datos reales recopilados abril y mayo de 2025.

Se recopilaron datos del número de ventas de ensalada dulce durante los meses de abril y mayo del año 2025, como se muestra en la figura 26. El propósito de esta recopilación fue evaluar las predicciones generadas por los modelos propuestos. Con base en las métricas obtenidas, se determinó que el modelo con mejor desempeño fue el modelo SARIMA utilizando su configuración por defecto haciendo uso de un grid search. Este proceso identificó que los mejores hiperparámetros son: $p = 1$, $d = 1$, $q = 2$, $P = 1$, $D = 1$, $Q = 1$, y $s = 7$.

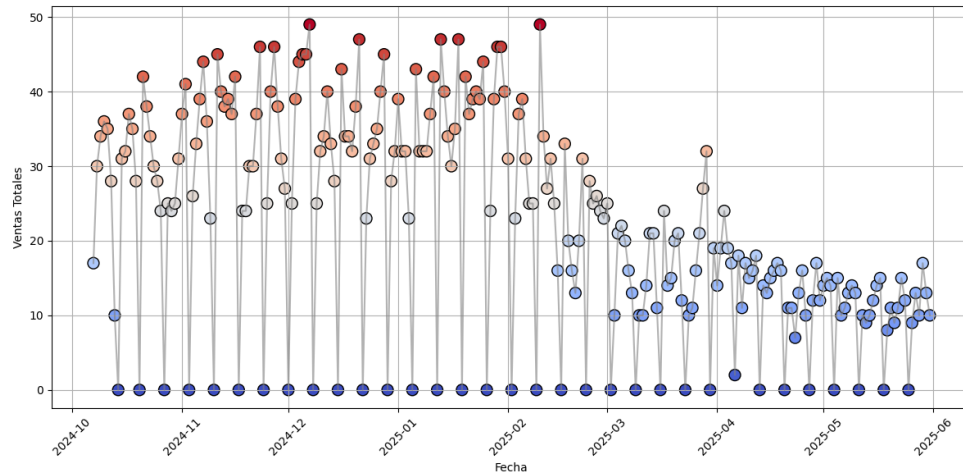


Figura 26: Histograma ventas de ensalada dulce octubre 2024 a Mayo 2025

A continuación se seleccionará uno de los resultados de cada modelo y se comparará con los datos reales, la selección se basará en las métricas y ya que los modelos han tenido

dificultades para predecir usando estos datos se descartaran los que sus gráfica tiendan a irse hacia la media y a aplanarse.

SVM: Se evalúa el modelo entrenado con los parámetros por defecto, este modelo fue el que tuvo más problemas para seguir la tendencia del precio, obtuvo resultados más altos que los reales como se muestra en la figura 27.

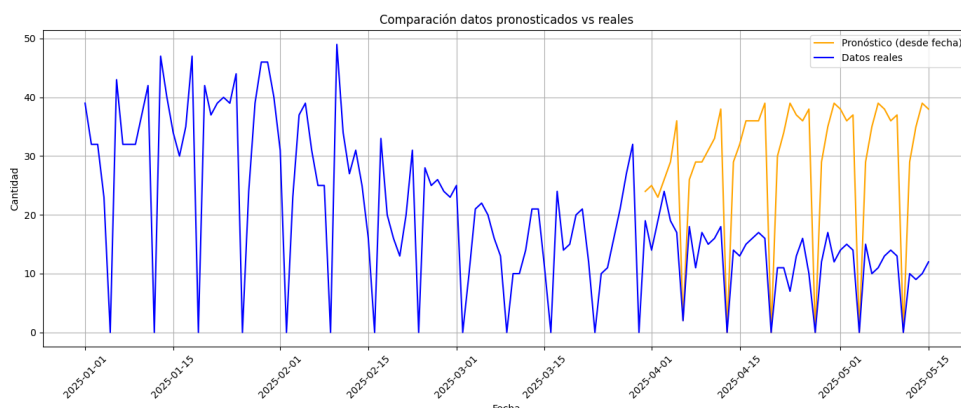


Figura 27: Histograma comparación predicción SVM contra datos reales de venta de ensalada dulce abril 2025 a Mayo 2025

SVM: Se evalúa el modelo entrenado con parámetros personalizados los cuales con los siguientes epochs=300, units=100, batch size=10, optimizador=Adam, según lo evidenciado en la figura 28, el modelo tuvo resultados un más altos que los valores reales.

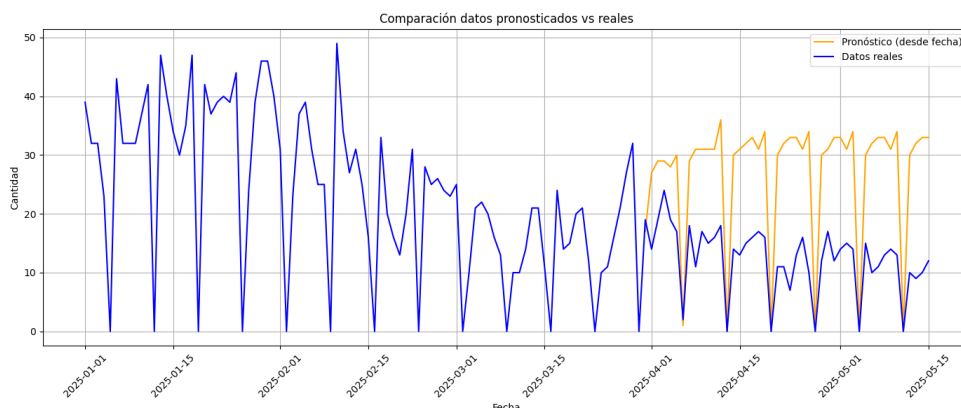


Figura 28: Histograma comparación predicción LSTM contra datos reales de venta de ensalada dulce abril 2025 a Mayo 2025

SARIMA: El modelo SARIMA utilizando su configuración por defecto con autogrid. Este proceso identificó que los mejores hiperparámetros son: $p = 1$, $d = 1$, $q = 2$, $P = 1$, $D = 1$, $Q = 1$, y $s = 7$, se obtuvo los siguientes resultados en la figura 29

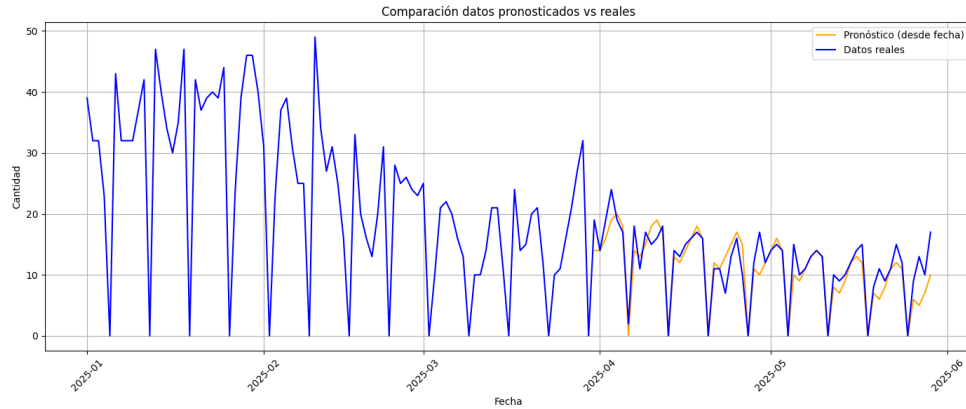


Figura 29: Histograma comparación predicción SARIMA contra datos reales de venta de ensalada dulce abril 2025 a Mayo 2025

Al evaluar las metricas con los datos reales se obtuvieron las metricas como se ven en la tabla 11.

Modelo	MAE	MSE	RMSE	MAPE (%)
SVM	5.73	124.64	11.16	19.14
LSTM	5.07	93.54	9.67	19.29
SARIMA	0.72	2.94	1.71	8.08

Tabla 11: Resultados de las métricas de error comparada con datos reales abril y mayo 2025 cantidad de ensaladas dulces vendidas

Se puede concluir que el modelo SARIMA obtuvo las mejores puntuaciones según las métricas para estos datos de venta, obteniendo los siguientes valores:

- **MAE (Error Absoluto Medio):** Aproximadamente **0.72**. Esto significa que el modelo se equivoca, en promedio, por menos de una ensalada. Las predicciones son bastante cercanas a la realidad.
- **MSE (Error Cuadrático Medio):** Valor de **2.97**. Indica que los errores grandes no son muy frecuentes. Aunque el MSE penaliza más los errores grandes, el valor bajo muestra que el modelo se mantiene estable.
- **MAPE (Error Porcentual Absoluto Medio):** **8.08 %**. En promedio, el modelo se equivoca solo un 8 % respecto a los valores reales. En predicción de ventas, un error menor al 10 % se considera bastante aceptable, por lo que el modelo está funcionando bien.

9.2. Prueba de modelos de predicción con datos de negocio local de comida rapida.

Adicionalmente se llevaron a cabo pruebas con datos reales de un comercio local dedicado a la venta de comidas rapidas. La prueba hace uso de las ventas diarias totales.

Se recopilaron los datos de las ventas en el periodo desde 11 de octubre del 2024 al 30 de marzo del 2025. A continuación en la figura 30 se presenta un histograma de las ventas.

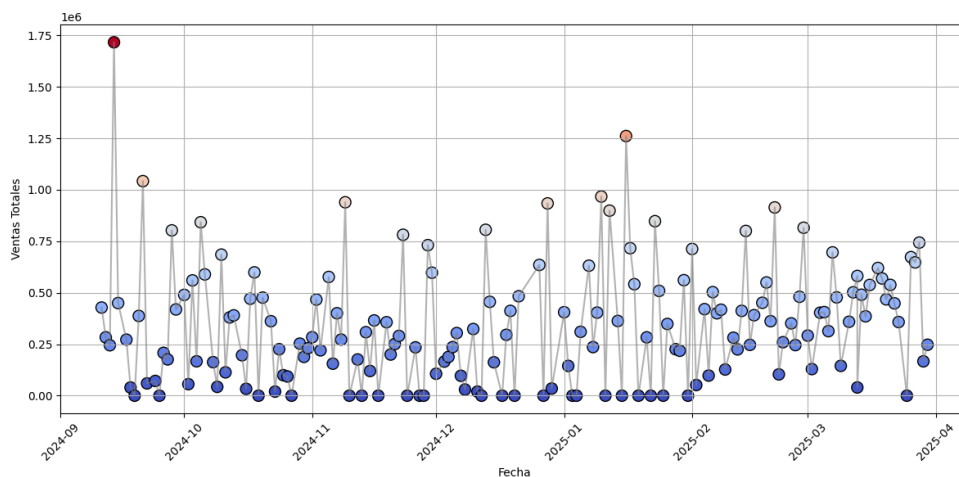


Figura 30: Histograma ventas totales diarias de negocio de comidas rapidas de 11 de octubre de 2024 al 30 de marzo 2025 en millones de pesos colombianos

Se proceden a realizar pruebas con los tres modelos en su configuración por defecto.

SVM: Resultados generados por SVM.

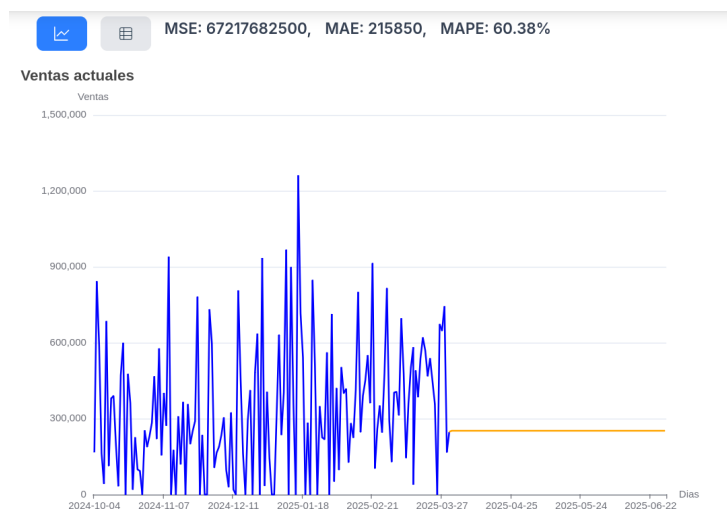


Figura 31: Predicción SVM ventas totales diarias de negocio de comidas rapidas abril 2025 a Mayo 2025 en millones de pesos colombianos

LSTM: También se evaluó el modelo LSTM, mostrando su comportamiento ante la serie temporal proporcionada.

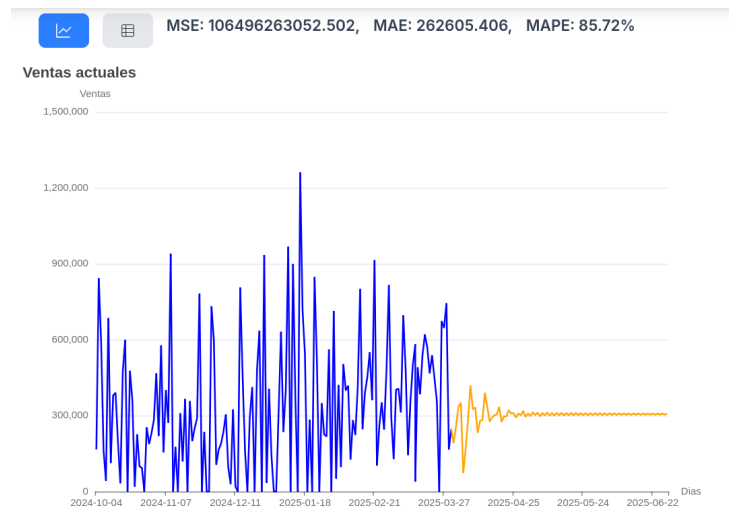


Figura 32: Predicción LSTM ventas totales diarias de negocio de comidas rapidas abril 2025 a Mayo 2025 en millones de pesos colombianos

SARIMA: Se graficaron los resultados iniciales del modelo SARIMA, el cual desde el primer momento mostró una predicción al alza.

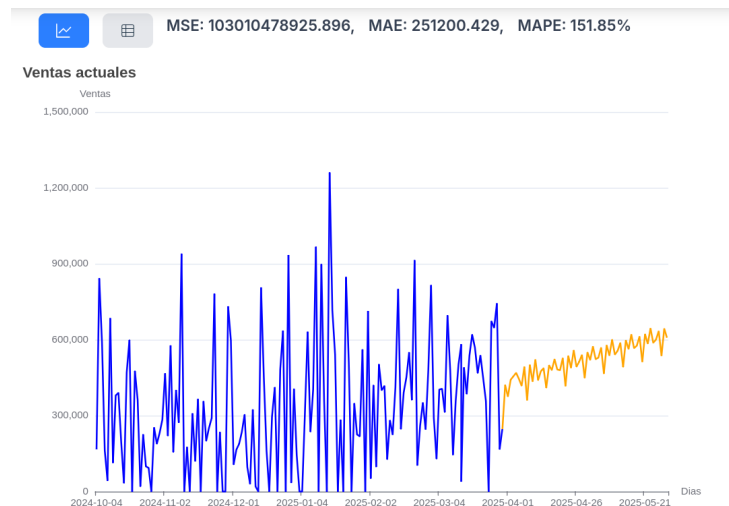


Figura 33: Predicción SARIMA ventas totales diarias de negocio de comidas rapidas abril 2025 a Mayo 2025 en millones de pesos colombianos

En este primer entrenamiento podemos ver que los modelos no han tenido resultados muy satisfactorios, aunque SVM tuvo el mejor resultado en las métricas, el modelo está tendiendo a irse a una media. Se presentan los resultados obtenidos por las métricas en la tabla 12.

Modelo	MSE	MAE	MAPE (%)
SVM	67,217,682,500	215,850.000	60.38
LSTM	115,714,682,004.15	277,812.994	113.97
SARIMA	103,010,478,925.90	251,200.429	151.85

Tabla 12: Comparación de desempeño entre modelos de predicción predicción ventas de comida rápida periodo abril a mayo del 2025

9.2.1. Entrenamientos adicionales con opciones avanzadas.

Se exploraron otras configuraciones personalizadas en cada uno de los modelos, a continuación en la tabla 13 se muestran los datos de entrada y resultados obtenidos de estos entrenamientos.

Model	Parameters	MSE	MAE	MAPE (%)
SVM	C=0.1, $\gamma=0.001$, $\epsilon=0.1$, kernel=RBF	67,217,682,500	215,850	60.38
SVM	C=1, $\gamma=1$, $\epsilon=5$, kernel=sigmoid	67,217,682,500	215,850	60.38
LSTM	epochs=300, units=100, batch size=2, optimizer=Adam	87,186,484,259.232	244,805.21	161.93
LSTM	epochs=250, units=100, batch size=2, optimizer=Adam	115,336,240,111.216	275,569.319	98.31
SARIMA	(2,1,2)(1,1,0)[7]	136,623,979,640.923	282,205.81	151.64
SARIMA	(2,0,2)(2,1,0)[7]	124,723,420,681.502	263,613.457	125.86

Tabla 13: Comparación modelos entrenamiento con datos adicionales ventas de comida rapida

9.2.2. Evaluación con datos reales recopilados abril y mayo de 2025.

Se recopilaron datos del número de ventas de totales por dia de comida rapida durante todo el mes de abril hasta el 14 de mayo del año 2025, como se muestra en la figura 34. El propósito de esta recopilación fue evaluar las predicciones generadas por los modelos propuestos.

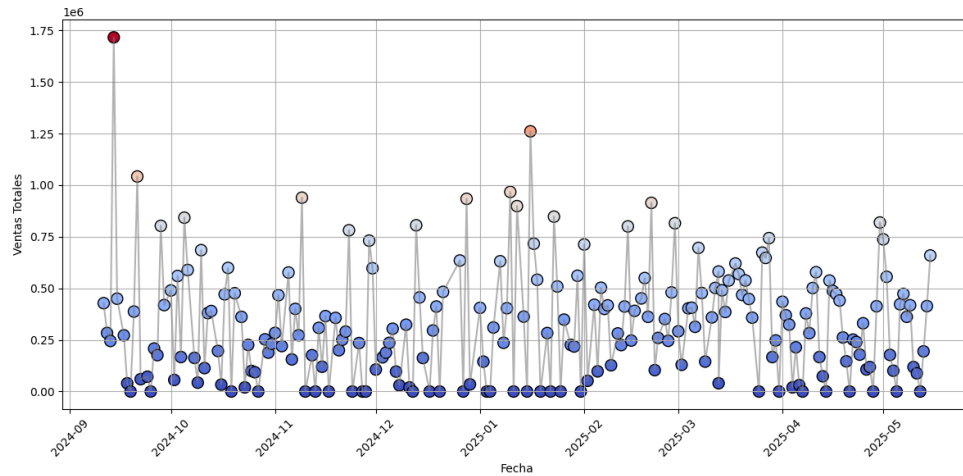


Figura 34: Histograma ventas totales diarias de comida rapida en el periodo octubre 2024 a Mayo 2025 en millones pesos colombianos

A continuación se seleccionará uno de los resultados de cada modelo y se comparará con los datos reales, la selección se basará en las métricas y ya que los modelos han tenido dificultades para predecir usando estos datos se descartaran los que sus gráfica tiendan a irse hacia la media y a aplanarse.

SVM Se evalúa el modelo entrenado con los parámetros por defecto, este modelo fue el que tuvo más problemas para seguir la tendencia del precio, se vio un comportamiento de que la gráfica se va a la media y se aplanar como se muestra en la figura 35.

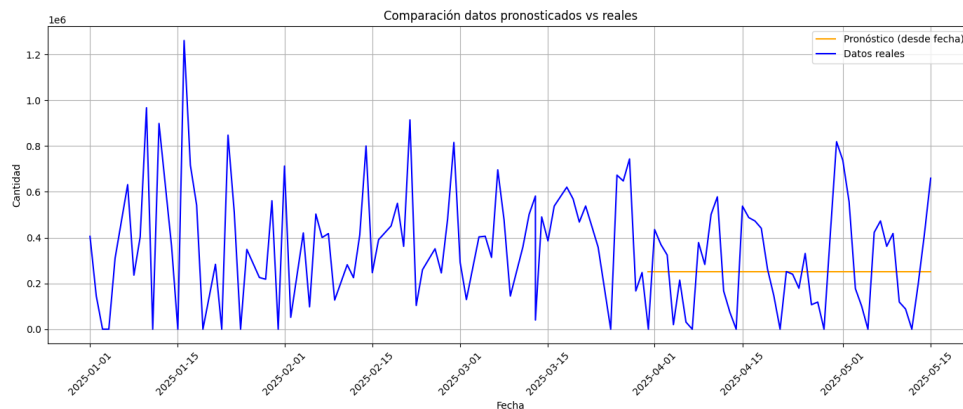


Figura 35: Histograma comparación predicción SVM venta de ensalada dulce abril a Mayo 2025 en millones de pesos colombianos

LSTM Se evalúa el modelo entrenado con los parámetros epoch= 250, lstm units=100, batch size = 2, optimizador= Adam, la gráfica se mantiene en el rango de los datos reales aunque parece que tiene problemas para predecir con exactitud, a continuación se muestra en la figura 36 los resultados.

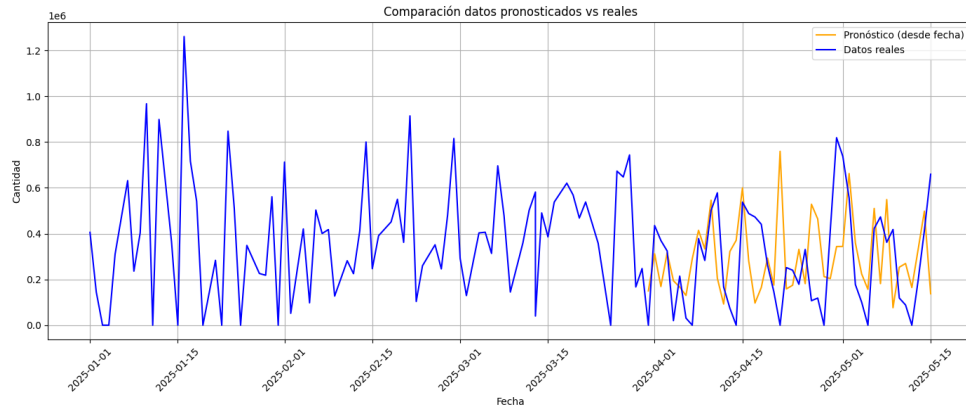


Figura 36: Histograma comparación predicción LSTM venta de ensalada dulce abril a Mayo 2025 en millones de pesos colombianos

SARIMA Se evalúa el modelo entrenado con los parámetros $p=2$, $d=0$, $q=2$, $P=2$, $D=1$, $Q=0$, $s=7$, la gráfica se mantiene en el rango de los datos reales aunque parece que tiende a hacer un patrón repetitivo alrededor de la media, a continuación se muestran los resultandos en la figura 37.

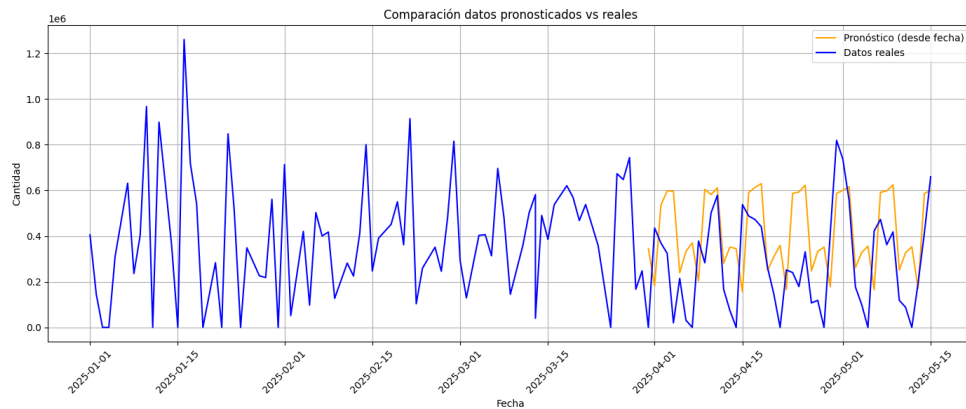


Figura 37: Histograma comparación predicción SARIMA venta de ensalada dulce abril 2025 a Mayo 2025 en millones de pesos colombianos

A continuación se dejan los resultados obtenidos en la tabla 14, se puede evidenciar de que el modelo SARIMA tuvo los mejores resultados de los modelos con un porcentaje en MAPE de 24.79 %.

Modelo	MAE	MSE	RMSE	MAPE (%)
SVM	68,133.87	17,507,961,250.00	132,317.65	29.16
LSTM	74,563.10	23,599,789,378.85	153,622.23	36.24
SARIMA	80,502.85	23,120,355,841.06	152,053.79	24.79

Tabla 14: Comparación de desempeño entre modelos de predicción de ventas de comida rápida contra datos reales en el periodo abril a mayo del 2025

Se puede concluir que todos los modelos fueron capaces de generar predicciones razonables, sin embargo, su desempeño vario de una manera significativa según el tipo de datos utilizados. En especial se observo que los modelos presentaron una mejor precisión con los datos de ventas de ensaladas, donde el modelo LSTM obtuvo un porcentaje en MAPE inferior al 20 %, en contraste las ventas de comida rápida presentaron una mayor aleatoriedad, lo cual impacto en la precisión de los resultados en todos los modelos, elevando el porcentaje MAPE alrededor de 37 %.

El modelo SVM se destacó en precisión absoluta obteniendo el menor MAE y MSE en la mayoría de los casos con las ventas de ensaladas, sin embargo, en las pruebas con los datos de comida rápida sus predicciones tendieron a ir hacia la media, por otro lado, el modelo LSTM presento un mejor desempeño relativo y el modelo SARIMA mostro un rendimiento muy bueno y estable con datos con estacionalidad clara.

Estas diferencias en el desempeño de los distintos modelos indican que no existe un modelo universal superior, sino que la elección del modelo optimo depende de los datos de entrada con el patrón de las ventas. además, el ajuste de los hiperparámetros es clave para encontrar una buena configuración que de un mejor desempeño en los modelos.

9.3. Encuesta realizada a usuarios dueños de negocio.

Se realizó una pruebas con los dueños del negocio que proporcionaron los datos de ventas, esto con el fin de probar el funcionamiento y la interfaz gráfica del prototipo con usuarios reales al final de la prueba se les realizó una encuesta hecha con la herramienta de google forms. El enlace a los resultados de la encuesta se encuentra disponible en el anexo D en la sección anexos 13.

Se obtuvieron dos respuestas al formulario. A continuación se presenta un análisis de los resultados:

- En cuanto al diseño de la interfaz gráfica se muestra el resultado en la figura 38, ambos participantes evaluaron la apariencia y organización visual como aceptable y excelente, lo cual indica una buena recepción del componente estético y funcional. Este resultado es relevante, ya que en el desarrollo del sistema se priorizó la simplicidad visual para facilitar su adopción por parte de usuarios sin conocimientos técnicos.

¿Cómo calificarías el diseño de la interfaz web?

2 respuestas

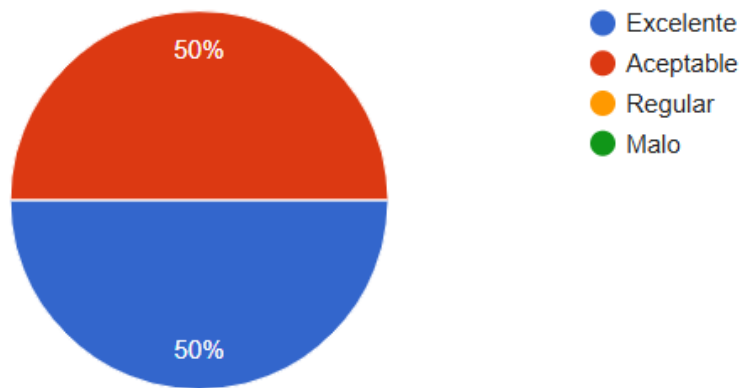


Figura 38: Resultados pregunta diseño grafico

- Respecto a la facilidad de navegación y comprensión general del funcionamiento mostrado el resultado en la figura 39, los participantes calificaron la experiencia como “algo fácil”, lo que sugiere que la curva de aprendizaje del sistema no representa una barrera significativa.

¿Fue fácil navegar por el sistema y entender cómo funciona?

2 respuestas

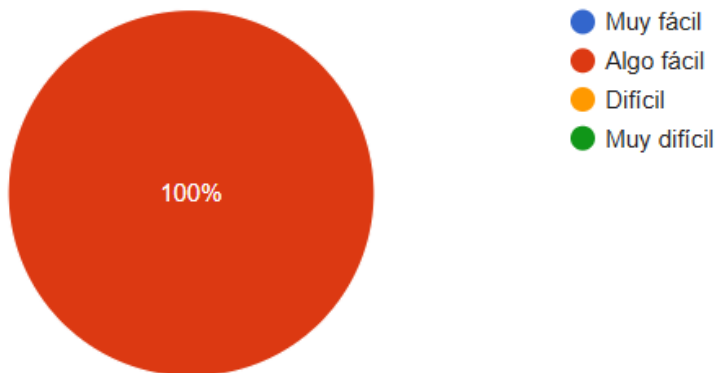


Figura 39: Resultados pregunta navegación y comprensión del prototipo.

- A su vez, la carga de archivos, una funcionalidad crítica del sistema, fue evaluada como “muy fácil”, como se muestra en la figura 40, lo que valida la decisión de permitir la importación de datos mediante archivos Excel comunes en entornos empresariales pequeños.

¿Fue fácil la carga de archivo?

2 respuestas

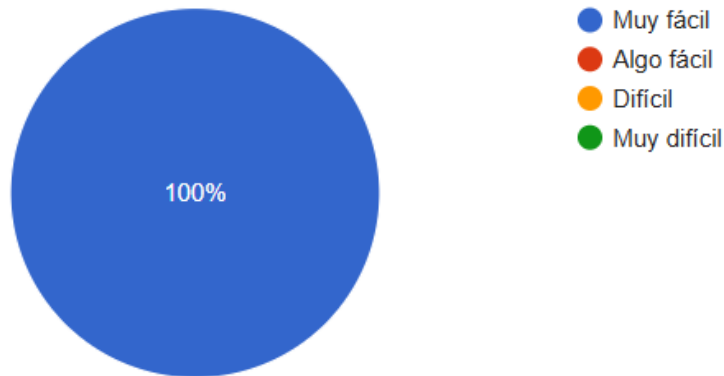


Figura 40: Resultados pregunta facilidad para cargar archivos.

- El proceso de entrenamiento de los modelos fue calificado entre “muy fácil” y “algo fácil” como se ilustra en la figura 41, lo que evidencia que el sistema logra abstraer la complejidad técnica del machine learning a través de una interacción accesible. Este punto es clave, ya que permite que usuarios sin formación técnica puedan aprovechar herramientas de predicción avanzadas sin necesidad de intervención experta.

¿Fue fácil realizar el entrenamiento de los modelos?

2 respuestas

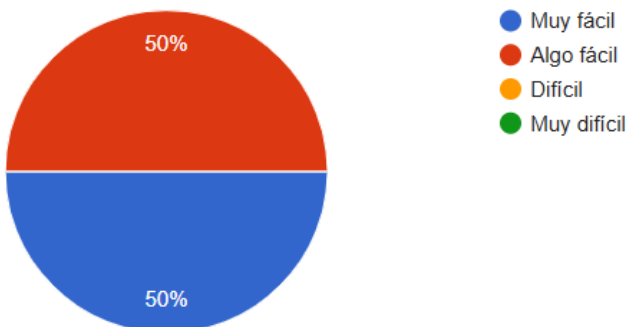


Figura 41: Resultados pregunta facilidad entrenamiento modelos.

- En cuanto a la interpretación de los resultados generados por el sistema, ambos usuarios respondieron afirmativamente como se muestra en la figura 42, indicando que comprendieron adecuadamente las predicciones visualizadas. Este resultado refuerza la decisión de presentar los pronósticos de forma gráfica y con información clara en pantalla lo cual fue útil para hacer más fácil a los usuarios la interpretación de los resultados.

¿Las predicciones generadas por el sistema fueron comprensibles?

2 respuestas

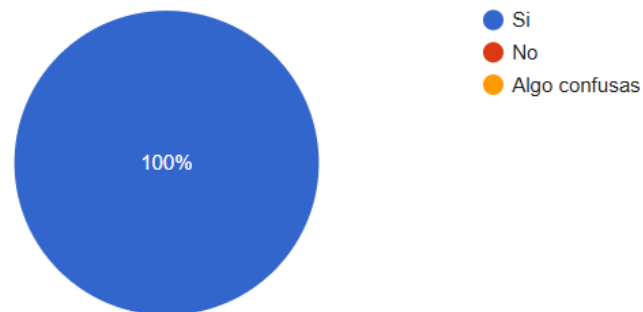


Figura 42: Resultados pregunta comprensión de los resultados.

- A la pregunta sobre la utilidad de dichas predicciones para la toma de decisiones en sus negocios, se obtuvo una respuesta de “sí, totalmente” como se visualiza en la figura 43, lo que valida el enfoque práctico del sistema desarrollado.

¿Consideras útiles las predicciones para tomar decisiones en un negocio real?

2 respuestas

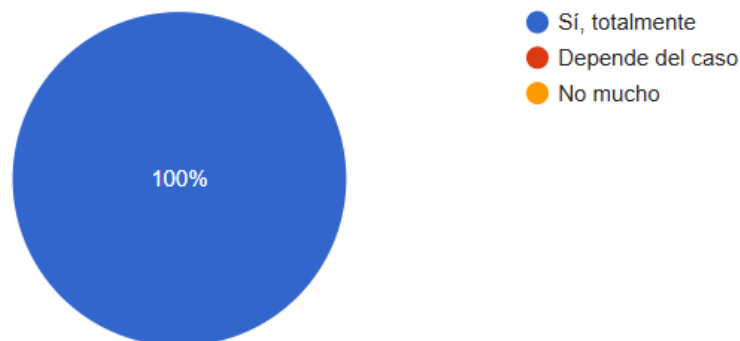


Figura 43: Resultados pregunta si es util para tomar desiciones en un negocio.

- Finalmente, en la figura 44 se muestra la precisión percibida de las predicciones fue calificada como “aceptable”, lo cual es coherente con los valores reales de error (MAPE) obtenidos durante las pruebas técnicas del sistema. Este resultado indica una percepción realista por parte de los usuarios y sugiere que, aunque existe margen de mejora, el nivel de precisión es suficiente para generar valor en la toma de decisiones empresariales.

¿Qué tan precisas te parecieron las predicciones?

2 respuestas

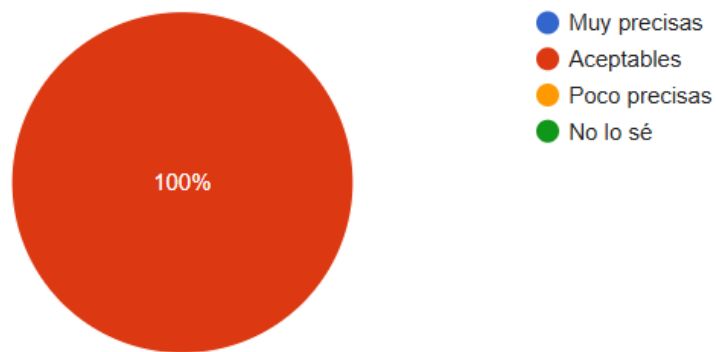


Figura 44: Resultados pregunta precisión de los resultados.

En conjunto, los resultados de la encuesta dan una primera validación positiva del sistema desde la perspectiva del usuario. La interfaz gráfica fue bien recibida, las funcionalidades resultaron accesibles, y las predicciones generadas fueron comprendidas y consideradas útiles para ser tomadas en cuenta en la toma de decisiones en el negocio. Esta retroalimentación da un respaldo a la viabilidad del prototipo en situaciones reales y sugiere que, con mayor difusión y ajustes, podría ser adoptado por más negocios.

No obstante, basándonos en los comentarios obtenidos, es posible identificar oportunidades de mejora en aspectos como la apariencia visual de la interfaz, con el fin de lograr un diseño más estético y profesional. Asimismo, se recomienda optimizar la experiencia de usuario dentro de la plataforma web, especialmente en lo relacionado con la navegación y el proceso de entrenamiento de los modelos, con el objetivo de hacerlo aún más intuitivo y accesible para usuarios sin formación técnica.

10. Conclusiones

Conclusiones

1. El desarrollo de este trabajo de grado permitió demostrar que es posible integrar modelos predictivos a entornos reales en PYMES, incluso con datos que presentan un nivel de ruido alto, y obtener resultados aceptables y útiles que puede usar una empresa para tomar decisiones que puedan mejorar sus ventas. A lo largo del desarrollo, se presentaron diversos retos como la calidad de los datos, la selección de modelos adecuados para hacer las predicciones y la creación de una interfaz intuitiva. Desde un enfoque de la ingeniería, fue necesario aplicar criterios técnicos y metodológicos para superar los retos y lograr completar el proyecto.
2. Uno de los principales retos y el pilar de este proyecto consistió en trabajar con diversos datasets, en el que se tuvo que aplicar criterios de selección, técnicas de preprocesamiento y normalización de los datos para obtener datos consistentes y útiles. Gracias a ello, se logró estructurar una base robusta que permitió entrenar los modelos con resultados satisfactorios. La elección de tres modelos predictivos (SARIMA, SVM y LSTM) no solo buscaba comparar rendimientos, sino también ofrecer una solución versátil al usuario, capaz de adaptarse a distintas condiciones y comportamientos de los datos.
3. Los resultados obtenidos en términos de la métrica MAPE confirman que el sistema puede ofrecer pronósticos útiles: en escenarios más controlados (ensalada dulce) se obtuvo un error menor al 20 %, mientras que con datos con más ruido (comida rápida) se mantuvo por debajo del 37 %, cifras aceptables que pueden ser útiles para la toma de decisiones informadas para una empresa. Estos resultados no solo son evidencia del funcionamiento técnico, sino también, desde un punto de vista de ingeniería, de la capacidad para aplicar herramientas matemáticas, estadísticas y computacionales de manera integrada.
4. Otro aspecto que exigió criterio de ingeniería fue la gestión del proyecto. Se optó por una metodología ágil que permitió organizar las tareas en sprints y gestionar el tiempo. Implementar una metodología facilitó el cumplimiento de los objetivos, la planificación eficiente y se solventaron ajustes imprevistos sin retrasos significativos.
5. La implementación de una interfaz gráfica funcional representó otro reto técnico importante, ya que implicó la creación de una experiencia de usuario amigable y fácil de usar por un usuario sin conocimientos avanzados en informática. Esta solución técnica buscó cerrar la brecha entre la tecnología y su adopción real en entornos donde no se cuenta con personal especializado.
6. En síntesis, el proyecto no solo cumplió los objetivos propuestos, sino que permitió poner en práctica competencias clave de un ingeniero: análisis de problemas reales, diseño de soluciones empleando tecnologías, investigación de información para la resolución de las necesidades y la toma de decisiones técnicas informadas.

11. Trabajos futuros

- **Integrar nuevos modelos:** Añadir más modelos de predicción a la aplicación permitirá ofrecer al usuario un conjunto más amplio de opciones, lo cual puede derivar en un mejor ajuste a diferentes tipos de datos y contextos, superando incluso el rendimiento de los modelos implementados en este prototipo.
- **Integrar reentrenamiento de modelos:** Incorporar una funcionalidad que permita cargar datos actualizados y reentrenar los modelos de forma periódica permitiría ajustar las predicciones a los cambios reales en las ventas, mejorando así la precisión con el tiempo y adaptándose a nuevas condiciones del mercado.
- **Desplegar la aplicación en la nube:** Publicar la aplicación en un servicio en la nube facilitaría el acceso remoto desde cualquier lugar con conexión a Internet, permitiendo a más usuarios aprovechar sus funcionalidades sin necesidad de instalar software localmente.
- **Pruebas de uso con usuarios reales:** Realizar pruebas más amplias con distintos tipos de negocios y usuarios reales permitirá evaluar la efectividad práctica del sistema. También facilitará la recolección de retroalimentación para futuras mejoras, especialmente en la toma de decisiones basadas en las predicciones generadas por el software.

12. Referencias

- [1] Institute of Electrical and Electronics Engineers (IEEE), “January 2024 iee thesaurus version 1.03 created by the institute of electrical and electronics engineers (ieee),” 2024. n.d.
- [2] F. D. Hedrick, D. C. Frank, C. Barnes, E. W. Davis, M. Krieger, and J. Budding, “Inventory management,” 2008.
- [3] M. T. Tanaman, J. L. A. Baylosis, B. J. A. Abiles, M. L. P. Catungal, and P. C. Encarnacion, “Web-based inventory management system,” *International Journal of Science and Applied Information Technology*, 2023.
- [4] B. J. Grablowsky, “Finantial management of inventory,”
- [5] e. a. GASTERMANN, Bernd, “Implementation of a software prototype with conwip characteristics for production planning and stock management.” cite-seerx.ist.psu.edu/document?repid=rep1type=pdfdoi=50d4e7cc540baa457291c0e0532803406a74fdac 2014.
- [6] S. Nowak, “Modelos predictivos ¿qué son y para qué se usan?.” nucio.school/blog/modelos-predictivos-que-son-y-usos/Que-son-los-modelos-predictivos, 2022.
- [7] J. W. Escobar, R. Linfati, and W. A. Jaimes, “Inventory management for distributors of perishable products,” *Ingeniería y Desarrollo*, 2017.
- [8] M. A. y Francisco Osorio, “Introducción a los conceptos básicos de la teoría general de sistemas.” www.redalyc.org/pdf/101/10100306.pdf, 1998.
- [9] V. A. D. y Miguel Angel Lopez, “Teoría general de sistemas, un enfoque práctico.” salazarvirtual.sistemaeducativosalazar.mx/assets/biblioteca/2eed553326d9cd2d96e78a7bea248ef3-2
- [10] E. Monterrubio-Hernández, “Componentes de un sistema informático.” epository.uaeh.edu.mx/revistas/index.php/ixtlahuaco/article/view/10403/9949, 2023.
- [11] J. L. G. Antonino Parisi, Franco Parisi, “Modelos predictivos de redes neuronales en Índices bursátiles.” www.jstor.org/stable/20856798?seq=1, 2003.
- [12] S. S. S. V. P. D. N. S. Purvika Bajaj1, Renesa Ray, “Sales prediction using machine learning algorithms,” 2020.
- [13] I. Amazon.com, “¿qué es el análisis predictivo?.” aws.amazon.com/es/what-is/predictive-analytics/.
- [14] IBM, “What is predictive analytics?.” www.ibm.com/topics/predictive-analytics.
- [15] I. Amazon.com, “Amazon sagemaker canvas.” <https://aws.amazon.com/es/sagemaker/canvas/>.

- [16] R. M. Granados, “Modelos de regresión lineal múltiple.” www.ugr.es/~montero/matematicas/regresion_lineal.pdf, 2016.
- [17] K. N. O. G. J. V. M. G.-B. Arys Carrasquilla, Alfonso Chacón, “Modelos de regresión lineal múltiple,” 2016.
- [18] B. L. Sean J. Taylor, “Prophet: forecasting at scale.” research.facebook.com/blog/2017/2/prophet-forecasting-at-scale/, 2017.
- [19] A. F. M. Mahecha, “Uso de redes neuronales para predecir mercados financieros.” <https://repositorio.uniandes.edu.co/server/api/core/bitstreams/c51474d1-6999-4392-ab87-176360818fe1/content>, 2023.
- [20] J. L. Gamarra Palacios, “Estimación de modelos para el pronóstico de precios en el sector agropecuario en el distrito especial de buenaventura, una aproximación econométrica usando el lenguaje de programación r.” <http://hdl.handle.net/10906/89009>, 2021.
- [21] LiveFlow, “How to create a live pl forecast in google sheets with liveflow.” www.youtube.com/watch?v=JU2IrOl1T1Q, 2022.
- [22] K. Yasar, “What is salesforce? a beginner’s guide.” www.techtarget.com/searchcustomerexperience/definition/Salesforcecom, 2023.
- [23] N. Inc, “Sales forecasting.” www.inboundcycle.com/blog-de-inbound-marketing/zoho-crm-que-es.
- [24] S. Meijomil, “Zoho crm: qué es y cómo funciona.” www.inboundcycle.com/blog-de-inbound-marketing/zoho-crm-que-es, 2021.
- [25] Real Academia Española, “Real academia española: Diccionario de la lengua española,” 2023. Versión 23.6 En Línea.
- [26] E. Talent, “What are the characteristics of software in software development?,” 2023.
- [27] IBM, “What is machine learning?.” www.ibm.com/topics/machine-learning, 2005.
- [28] H. Timo, “Sales forecasting in small and medium-sized enterprises,” 2005.
- [29] A. Jaiswal, “Celda lstm.” <https://www.analyticsvidhya.com/blog/2022/01/tutorial-on-rnn-lstm-gru-with-implementation/>, 2024.
- [30] geeksforgeeks, “Support vector regression (svr) using linear and non-linear kernels in sci-kit learn.” <https://www.geeksforgeeks.org/support-vector-regression-svr-using-linear-and-non-linear-kernels-in-scikit-learn/>, 2025.
- [31] C. Huang, “Basic architecture of svr model.” https://www.researchgate.net/figure/Basic-architecture-of-SVR-model_fig3_228677761, June 2009. Recuperado de ResearchGate.

- [32] J. J. Kim, “Sarima..” <https://medium.com/@jihyun.kim423/exploring-the-seasonal-arima-sarima-model-for-forecasting-differences-from-arima-e30c> Apr. 2024.
- [33] Coursera, “What is agile? and when to use it?.” <https://www.coursera.org/articles/what-is-agile-a-beginners-guide>, 2024.
- [34] Coursera, “What is kaggle and what is it used for?,” July 2025.
- [35] T. Lowden, “Data.gov launches new metrics tools.” <https://digital.gov/2024/09/13/datagov-launches-metrics-tools>, Sept. 2024.

13. Anexos

A continuación se presentan los enlaces relevantes que complementan el desarrollo del proyecto:

Anexo	Título	Descripción	Enlace / Ubicación
A	Tablero del proyecto	Tablero Kanban con la planeación y gestión del proyecto usando GitHub Projects.	https://github.com/users/ekesteban/projects/3/views/1
B	Repositorio del proyecto	Contiene el código fuente completo del backend y frontend, así como la documentación técnica.	https://github.com/ekesteban/TG_ingenieria_sistemas
C	Mockups de interfaz	Diseños visuales del prototipo creados con la herramienta Figma.	https://www.figma.com/design/NuZeawx3bgQiisaXbcZs2y/Untitled?node-id=0-1&t=1TcVeCDS3B1itGe2-1
D	Resultados encuesta	Enlace al análisis de la encuesta realizada a usuarios sobre el uso del sistema.	https://shorturl.at/t2gkq

Tabla 15: Anexos del trabajo de grado