



UNIVERSIDAD AUTÓNOMA DE ICA

UNIVERSIDAD AUTÓNOMA DE ICA

**FACULTAD DE INGENIERIA, CIENCIAS Y ADMINISTRACIÓN
PROGRAMA ACADÉMICO DE INGENIERÍA DE SISTEMAS**

TESIS

**Predicción de la demanda de medicamentos esenciales
mediante modelos de aprendizaje automático en farmacia
hospitalaria Puno 2025**

LINEA DE INVESTIGACION

Ciencia de datos, inteligencia artificial y analítica predictiva

PRESENTADO POR

Mamani Aparicio, Maritza

**TESIS DESARROLLADA PARA OPTAR EL TITULO
PROFESIONAL DE INGENIERA DE SISTEMAS**

DOCENTE ASESOR:

Dr. Velasquez Medina, Martin

<https://orcid.org/0000-0003-0742-911X>

Chincha, Perú, 2025



CONSTANCIA DE APROBACIÓN DE INVESTIGACIÓN

Chincha, 7 de diciembre del 2025

Dra. María Alejandra Campos Sobrino
Decana de la Facultad de Ingeniería, Ciencias y Administración
Universidad Autónoma de Ica.

Presente. -

De mi especial consideración:

Sirva la presente para saludarla e informar que, **Bach. Maritza Mamani Aparicio**, de la Facultad de Ingeniería, Ciencias y Administración, del programa Académico de INGENIERÍA DE SISTEMAS, ha cumplido con elaborar su:

PROYECTO DE TESIS

TESIS


TITULADO:

“Predicción de la demanda de medicamentos esenciales mediante modelos de aprendizaje automático en Farmacia Hospitalaria Puno 2025”

Por lo tanto, queda expedito para continuar con el procedimiento correspondiente para solicitar la emisión de la resolución para la designación de Jurado, fecha y hora de sustentación de la Tesis para la obtención del Título Profesional.

Agradezco por anticipado la atención a la presente, aprovecho la ocasión para expresar los sentimientos de mi especial consideración y deferencia personal. Cordialmente,

DR. MARTIN ISIDRO VELASQUEZ MEDINA
CODIGO ORCID: 0000-0003-0742-911X
DNI: 42940129

| | | |
|--|---|-------------------------|
|  UNIVERSIDAD AUTÓNOMA DE ICA | UNIVERSIDAD AUTÓNOMA DE ICA | Código: DIPI-IEI-DIR-04 |
| | DIRECTIVA DE LINEAMIENTOS Y GUIA METODOLÓGICA PARA LA ELABORACIÓN DE TRABAJOS DE INVESTIGACIÓN Y TESIS | Versión: 04 |
| | | Página 73 de 80 |

ANEXO 9

DECLARATORIA DE AUTENTICIDAD DE LA INVESTIGACIÓN

Yo, **Maritza Mamani Aparicio**, identificada con DNI N° 43147702, en mi condición de estudiante del programa de estudios de **INGENIERÍA DE SISTEMAS**, de la Facultad de Ingeniería, Ciencias y Administración, en la Universidad Autónoma de Ica y que habiendo desarrollado la Tesis titulada: "PREDICCIÓN DE LA DEMANDA DE MEDICAMENTOS ESENCIALES MEDIANTE MODELOS DE APRENDIZAJE AUTOMÁTICO EN FARMACIA HOSPITALARIA PUNO 2025", declaro bajo juramento que:

- La investigación realizada es de mi autoría.
- La tesis no ha cometido falta alguna a las conductas responsables de investigación, por lo que, no se ha cometido plagio, ni autoplagio en su elaboración.
- La información presentada en la tesis se ha elaborado respetando las normas de redacción para la citación y referenciación de las fuentes de información consultadas. Asimismo, el estudio no ha sido publicado anteriormente, ni parcial ni totalmente con fines de obtención de algún grado académico o título profesional.
- Los resultados presentados en el estudio, producto de la recopilación de datos son reales, por lo que, el(la) investigador(a) no ha incurrido ni en falsedad, duplicidad, copia o adulteración de estos, ni parcial, ni totalmente.
- La investigación cumple con el porcentaje de similitud establecido según la normatividad.

3%

Autorizo a la Universidad Autónoma de Ica, de identificar plagio, autoplagio, falsedad de información o adulteración de estos, se proceda según lo indicado en la normatividad vigente de la universidad, asumiendo las consecuencias o sanciones que se deriven de alguna de estas malas conductas.

Chincha Alta, 20 de diciembre del 2025



MARITZA MAMANI APARICIO
DNI: 43147702

La firma corresponde al o los responsables de la investigación



0121673460



NOTARIA ZEGARRA CABRERA JESSIE TARCILA SERVICIO DE AUTENTICACIÓN E IDENTIFICACIÓN BIOMÉTRICA



INFORMACIÓN PERSONAL

| | |
|------------------|----------|
| DNI | 43147702 |
| Primer Apellido | MAMANI |
| Segundo Apellido | APARICIO |
| Nombres | MARITZA |

CORRESPONDE

La primera impresión dactilar capturada corresponde al DNI consultado. La segunda impresión dactilar capturada corresponde al DNI consultado.



MAMANI APARICIO, MARITZA
DNI 43147702

INFORMACIÓN DE CONSULTA DACTILAR

Operador: 42095592 - Esther Quispe Apaza

Fecha de Transacción: 20-12-2025 09:28:14

Entidad: 10304227457 - ZEGARRA CABRERA JESSIE TARCILA

VERIFICACIÓN DE CONSULTA

Puede verificar la información en línea en: <https://serviciosbiometricos.reniec.gob.pe/identifica3/verification.do>

Número de Consulta: 0121673460



CERTIFICADO: La autenticación de la firma de Maritza Mamani Aparicio

con DNI. N° 43147702

quien ha firmado y estampado su huella dactilar en mi presencia. La Notaria que autoriza no asume responsabilidad sobre el contenido del documento.

(Art. 108 D. Leg. 1049) de que doy fe.

Puno, 20 Dic. 2025



NOTARÍA ZEGARRA CABRERA
SE SOLICITÓ ADHESIÓN DE IMPRESIÓN BIOMÉTRICA.-



Jessie Tarcila Zegarra Cabrera
ABOGADA - NOTARIA



DEDICATORIA

A Dios, fuente de sabiduría y fortaleza, por brindarme salud y perseverancia en cada etapa de este camino.

A mis padres, por su amor incondicional, ejemplo de esfuerzo y sacrificio, quienes sembraron en mí los valores que hoy guían mi vida profesional.

A mis docentes y amigos, por su compañía, estímulo y orientación constante.

AGRADECIMIENTO

Expreso mi más sincero agradecimiento a todas las personas que han hecho posible el desarrollo de este proyecto de investigación:

A mi asesor de tesis, Dr. Martín Velásquez Medina, por su valiosa orientación académica, sus aportes técnicos y su acompañamiento durante cada etapa de la investigación.

A los profesionales del área de Farmacia del Hospital III de Puno – ESSALUD, por su colaboración en el acceso a los datos históricos y por compartir su experiencia en la gestión de medicamentos esenciales.

A mis compañeros y compañeras del programa de Ingeniería de Sistemas, por su apoyo, intercambio de ideas y por motivarme a continuar aprendiendo.

A mi familia, por creer en mí, por brindarme apoyo emocional incondicional, y por ser mi mayor fuente de inspiración.

RESUMEN

El estudio se desarrolló bajo un enfoque cuantitativo, aplicativo y comparativo, utilizando datos históricos estructurados de consumo diario de 30 medicamentos esenciales. El proceso metodológico se basó en la metodología CRISP-DM, abarcando las fases de comprensión de los datos, preprocesamiento, modelamiento, validación y evaluación estadística. Se implementaron cuatro modelos: Random Forest (RF), XGBoost, LSTM (Long Short-Term Memory) y GRU (Gated Recurrent Unit). El desempeño de cada modelo se evaluó utilizando tres métricas: MAE (Mean Absolute Error), RMSE (Root Mean Squared Error) y sMAPE (Symmetric Mean Absolute Percentage Error), complementadas con pruebas inferenciales: ANOVA de medidas repetidas, prueba de Friedman y contrastes post-hoc. Los resultados del ANOVA mostraron diferencias altamente significativas entre modelos ($p < 0.001$), con un tamaño del efecto elevado (η^2 parcial entre 0.55 y 0.62). El modelo GRU obtuvo los menores errores promedio (MAE ≈ 18.5 , RMSE ≈ 23.9 , sMAPE $\approx 4.2\%$), evidenciando mayor precisión y estabilidad temporal, seguido de XGBoost. En cambio, los modelos LSTM y Random Forest mostraron un rendimiento inferior con mayor dispersión de errores. Las pruebas de Friedman y post-hoc confirmaron la superioridad estadística del modelo GRU ($p < 0.001$ frente a LSTM y RF) y una tendencia favorable frente a XGBoost ($p \approx 0.06$). En conclusión, el modelo GRU se consolida como la alternativa óptima para la predicción de la demanda farmacéutica, al combinar precisión, eficiencia y estabilidad temporal.

Palabras clave: aprendizaje automático, farmacia hospitalaria, predicción de demanda, medicamentos esenciales, ciencia de datos.

ABSTRACT

The study was conducted under a quantitative, applied, and comparative approach, using structured historical data on the daily consumption of 30 essential medicines. The methodological process was based on the CRISP-DM methodology, covering the phases of data understanding, preprocessing, modeling, validation, and statistical evaluation. Four models were implemented: Random Forest (RF), XGBoost, LSTM (Long Short-Term Memory), and GRU (Gated Recurrent Unit). The performance of each model was evaluated using three metrics: MAE (Mean Absolute Error), RMSE (Root Mean Squared Error), and sMAPE (Symmetric Mean Absolute Percentage Error), complemented by inferential tests including repeated-measures ANOVA, the Friedman test, and post-hoc comparisons. The ANOVA results revealed highly significant differences among models ($p < 0.001$), with a large effect size (partial η^2 ranging from 0.55 to 0.62). The GRU model achieved the lowest average errors (MAE ≈ 18.5 , RMSE ≈ 23.9 , sMAPE $\approx 4.2\%$), demonstrating superior accuracy and temporal stability, followed by XGBoost. In contrast, the LSTM and Random Forest models exhibited inferior performance with greater error dispersion. The Friedman and post-hoc tests confirmed the statistical superiority of the GRU model ($p < 0.001$ compared to LSTM and RF) and a favorable trend when compared to XGBoost ($p \approx 0.06$).

In conclusion, the GRU model is consolidated as the optimal alternative for pharmaceutical demand forecasting, combining accuracy, efficiency, and temporal stability.

Keywords: machine learning, hospital pharmacy, demand forecasting, essential medicines, data science.

INDICE

| | |
|---|----|
| INDICE DE TABLAS | 11 |
| I. INTRODUCCION..... | 14 |
| II. PLANTEAMIENTO DEL PROBLEMA DE INVESTIGACION..... | 16 |
| 2.1 Descripción de la realidad problemática..... | 16 |
| 2.2 Pregunta de investigación general | 17 |
| 2.3 Preguntas de investigación específicas..... | 17 |
| 2.4 Objetivo general..... | 18 |
| 2.5 Objetivos específicos | 18 |
| 2.6 Justificación e importancia..... | 18 |
| III. MARCO TEORICO..... | 21 |
| 3.1 Antecedentes | 21 |
| 3.2 Bases teóricas de la variable..... | 25 |
| 3.3 Marco conceptual | 41 |
| IV. ESTRATEGIA METODOLOGICA | 44 |
| 4.1 Tipo y nivel de investigación..... | 44 |
| 4.2 Diseño de investigación | 44 |
| 4.3 Hipótesis General..... | 45 |
| 4.4 Hipótesis Específicas:..... | 45 |
| 4.5 Operacionalización de las variables..... | 46 |
| 4.6 Población y muestra: | 47 |
| 4.7 Técnicas e instrumentos de recolección de datos:..... | 49 |
| 4.8 Técnicas de análisis e interpretación de datos. | 49 |
| V. RESULTADOS | 65 |
| 5.1 Presentacion de resultados | 65 |
| 5.2 Interpretación de los resultados..... | 92 |

| | | |
|-------|---|-----|
| VI. | ANALISIS DE LOS RESULTADOS | 101 |
| 6.1 | Analisis inferencial..... | 101 |
| VII. | DISCUSION DE RESULTADOS | 109 |
| 7.1 | Comparacion de resultados..... | 109 |
| | CONCLUSIONES Y RECOMENDACIONES..... | 113 |
| VIII. | REFERENCIAS BIBLIOGRAFICAS..... | 115 |
| | ANEXOS | 120 |
| | Anexo 1: Matriz de consistencia | 121 |
| | Anexo 2: Instrumento de recolección de datos | 122 |
| | Anexo 3: Fragmento de la base de Datos..... | 124 |
| | Anexo 4: Informe de Turnitin al 28 % de similitud | 125 |
| | Anexo 5: Evidencia Fotográfica | 126 |

INDICE DE TABLAS

| | |
|---|-----|
| Tabla 1. Cuadro comparativo de justificación de modelos..... | 29 |
| Tabla 2. Operacionalización de variables | 46 |
| Tabla 3 Definición de variables de la clase ProductoDatos | 52 |
| Tabla 4 Tabla de frecuencias dispensadas por transacción para metamizol sodico1g/2ml..... | 55 |
| Tabla 5. Cuadro de Criterios generales y específicos | 66 |
| Tabla 6. Cuadro agrupación de criterios generales y específicos..... | 67 |
| Tabla 7 Evaluación del desempeño de los modelos predictivos GRU, LSTM, Random Forest y XGBoost..... | 91 |
| Tabla 8. Criterios técnicos y estadísticos para la selección de modelos. 92 | |
| Tabla 9. Modelos implementados y parámetros optimizados | 93 |
| Tabla 10 Evaluación del desempeño de los modelos predictivos GRU, LSTM, Random Forest y XGBoost..... | 95 |
| Tabla 11. Resultados del ANOVA de medidas repetidas entre modelos.. | 98 |
| Tabla 12 Resultados globales de la prueba de Friedman entre modelos | 104 |
| Tabla 13 Rango promedio de desempeño de modelos | 104 |
| Tabla 14. Comparación post-hoc entre modelos (p-ajustada) | 105 |

INDICE DE FIGURAS

| | |
|---|----|
| Figura 1. Ejemplo de datos de consumo de medicamentos esenciales .. | 53 |
| Figura 2. Función cargar datos | 53 |
| Figura 3. Ruta de los datos a trabajar | 54 |
| Figura 4. Código para generar la tabla de frecuencias | 55 |
| Figura 5. Cantidad de Ampollas de metamizol entregados a pacientes vs el % de veces de entrega | 56 |
| Figura 6. Código para eliminar registros con cantidad cero. | 57 |
| Figura 7. Código para graficar y poder visualizar la distribución de datos | 59 |
| Figura 8. Caja y bigotes (con outliers) Metamizol sódico 1 g/2ml | 59 |
| Figura 9. Código para graficar caja y bigotes (sin outliers) Metamizol sódico 1 g/2ml | 60 |
| Figura 10. Caja y bigotes (sin outliers) Metamizol sódico 1 g/2ml..... | 61 |
| Figura 11. Entregas diarias de recetas (dispensación) de Metamizol sodico 1g/2ml. | 62 |
| Figura 12. Fragmento de código procesamiento de datos para modelo Randon Forest. | 70 |
| Figura 13. Fragmento de código ingeniería de características en Randon Forest..... | 71 |
| Figura 14. Fragmento de código función de métricas en Randon Forest | 71 |
| Figura 15. Fragmento de código entrenamiento del modelo Randon Forest | 72 |
| Figura 16. Fragmento de código predicción de un día, alineando columnas del modelo Randon Forest..... | 73 |
| Figura 17. Ejemplo de código de la tabla de entrada (campos esenciales y consistencia temporal) del modelo XGBoost | 74 |
| Figura 18. Reglas codificadas como features binarias modelo XGBoost | 74 |
| Figura 19. Reglas codificadas Winsorizacion del modelo XGBoost | 75 |
| Figura 20. Matriz final de entrada (por fila/día)..... | 75 |
| Figura 21. Código escalamiento robusto modelo XGBoost..... | 76 |
| Figura 22. Esquema de pipeline preprocesamiento XGBoost, optimización métricas | 76 |
| Figura 23. Métricas definiciones compactas XGBoost | 77 |

| | |
|--|----|
| Figura 24. Consola de entrenamiento Bach (métricas por código y rutas) para el modelo XGBoost..... | 77 |
| Figura 25. Consola de entrenamiento para la carga de la base de datos para el modelamiento LSTM..... | 78 |
| Figura 26. Generación continua para el modelamiento LSTM | 79 |
| Figura 27. Creación de la variable calendario y operativas para el modelamiento LSTM..... | 79 |
| Figura 28. Tratamiento de ceros e imputaciones para el modelamiento LSTM | 80 |
| Figura 29. Codificación para el ventaneo del modelamiento LSTM | 82 |
| Figura 30. Codificación de escalamiento del modelamiento LSTM..... | 82 |
| Figura 31. Codificación línea base para el modelamiento LSTM | 83 |
| Figura 12. Codificación para el modelamiento LSTM..... | 84 |
| Figura 33. Codificación de evaluación del LSTM | 84 |
| Figura 34. Codificación para la predicción usando el modelo LSTM..... | 85 |
| Figura 35. Preprocesamiento diario con reglas operativas del GRU..... | 86 |
| Figura 36. Escalado robusto y generación de tensores (X, Y) del GRU . | 87 |
| Figura 37. Definición y entrenamiento optimizado del modelo GRU | 88 |
| Figura 38. Métricas y evaluación del modelo GRU | 88 |
| Figura 39. Proyección con modelo GRU, entrenado..... | 89 |
| Figura 40. Ejemplo de ajuste del modelo GRU a la demanda real del medicamento metamizol 1ml. | 94 |
| Figura 41. Comparativo de métricas de error promedio por modelo MAE y RMSE entre modelos predictivos..... | 95 |
| Figura 42. Comparativo de sMAPE promedio (%) entre modelos predictivos | 96 |

I. INTRODUCCION

La disponibilidad continua de medicamentos esenciales constituye un pilar fundamental para garantizar una atención médica segura, equitativa y oportuna. Sin embargo, según reportes de la Organización Mundial de la Salud OMS, (2025), más del 30% de los hospitales públicos en América latina enfrentan periódicamente episodios de desabastecimiento de medicamentos críticos, afectando la calidad del servicio y la seguridad del paciente. En el contexto del ámbito peruano, estudios del ministerio de salud MINSA, (2023), evidencian que las deficiencias en la planificación y gestión del stock hospitalario generan pérdidas económicas y comprometen la capacidad de respuesta ante emergencias.

La farmacia del Hospital III Base Puno, constituye un caso representativo de esta problemática: el volumen elevando de dispensaciones, la variabilidad diaria de la demanda y la ausencia de mecanismos analíticos de pronóstico dificultan la toma de decisiones precisas sobre compras, almacenamiento y programación de insumos. En escenarios de creciente presión sobre los recursos sanitarios, se vuelven indispensables implementar métodos que permitan reducir la incertidumbre y mejorar la planificación operativa.

En este marco, la ciencia de datos y los modelos de aprendizaje automáticamente ofrecen herramientas robustas para analizar series temporales de consumo y generar predicciones más precisas que los métodos tradicionales en particular técnicas como Randon Forest y XGBoost permiten modelar relaciones o lineales y patrones estacionales, mientras que las arquitecturas de aprendizaje profundo basadas en redes neuronales recurrentes, como LSTM y GRU, destacan por su capacidad para capturar dependencias temporales complejas y fluctuaciones irregulares en los datos. La aplicación de estos modelos en el ámbito hospitalario representa una alternativa innovadora, reproducible y de bajo costo, sobre todo cuando se dispone de registros históricos estructurados y software especializado.

El estudio tiene como objetivo predecir la demanda diaria de medicamentos esenciales en la farmacia del hospital base III de Puno, mediante modelos de aprendizaje automático, con el fin de mejorar la precisión de las estimaciones, optimizar la gestión de medicamentos y reducir los riesgos de quiebre de stock o sobreabastecimiento. Para ello, se emplean cuatro modelos representativos: Random Forest; XGBoost, LSTM y GRU, evaluados bajo metodología CRISP-DM y comparados mediante métricas estándar de la literatura científica (MAE, RMSE y sMAPE). Esta comparación permite no solo identificar el modelo más adecuado para el contexto hospitalario, sino también evidencia científica sobre el desempeño de diferentes enfoques predictivos en la gestión farmacéutica del sector público.

El trabajo se estructura de la siguiente manera:

En el capítulo I, se contextualizó el problema desde la perspectiva internacional, nacional y local, se formularon las preguntas de investigación, los objetivos y la justificación académica, metodológica y social del estudio.

En el capítulo II, se desarrolló el marco teórico, que incluye antecedentes relevantes, fundamentos del aprendizaje automático, predicción de demanda, series temporales y conceptos operativos esenciales para el estudio.

En el capítulo III, se presentó la estrategia metodológica, detallando el tipo y nivel de investigación, el diseño, la población y muestra, las técnicas de recolección y procesamiento de datos, así como el procesamiento de construcción, entrenamiento y evaluación de los modelos predictivos.

El capítulo IV expone los resultados obtenidos, el análisis comparativo entre modelos, las pruebas estadísticas inferenciales y la discusión en relación con estudios nacionales e internacionales. Finalmente se presentan las conclusiones, recomendaciones y las referencias utilizadas.

Con el desarrollo de este estudio se busca demostrar que el uso de las técnicas de ciencia de datos en la farmacia hospitalaria constituye una

herramienta efectiva, escalable y replicable, para mejorar la planificación del abastecimiento, optimizar recursos y fortalecer la toma de decisiones.

II. PLANTEAMIENTO DEL PROBLEMA DE INVESTIGACION

2.1 Descripción de la realidad problemática

A nivel mundial, los procesos de gestión de medicamentos representan uno de los mayores retos para los sistemas de salud, el cual garantiza una atención de calidad para los pacientes, tanto en países desarrollados como en lugares con recursos limitados (Bilal et al., 2024).

La demanda de medicamentos puede ser a veces incierta y también cambiante, específicamente en Hospitales Públicos, motivos que pueden provocar el desabastecimiento o sobrestock de medicamentos esenciales, afectando directamente la calidad de atención al paciente o generando pérdidas económicas para las instituciones de Salud (Hu et al., 2024)(Joshi et al., 2025).

Frente a esta problemática, existen varios estudios en el ámbito internacional que desarrollaron el uso de modelos predictivos que se basan en inteligencia artificial y aprendizaje automatizado, para predecir la demanda y optimizar los procesos logísticos hospitalarios (Kumar et al., 2023)(Fourkiotis & Tsadiras, 2024).

A nivel de América Latina, se desarrollaron trabajos de investigación donde demostraron que incluso modelos simples, como regresión lineal, KNN o arboles de decisión, pueden brindar predicciones precisas y útiles en la gestión de farmacias hospitalarias (Rivero Albarran et al., 2022);(Correa, 2020). Estas técnicas admiten tomar decisiones informadas sobre la reposición de stock, informar la falta de medicamentos y realizar las programaciones de medicamentos, incluso en contexto con datos limitados (Ndakondja & Dongo, 2024).

En el Perú, aun es escasa la aplicación metódica de herramientas tecnológicas de predicción en el ámbito Hospitalario. La correcta y adecuada gestión de la demanda de medicamentos es vital para garantizar una atención médica de calidad. El Hospital Base III Puno, que es un establecimiento de salud en el que la farmacia se ve obligada a captar datos solo de consumo en promedio para realizar sus programaciones, no midiendo exactamente el nivel de variabilidad en la demanda de medicamentos esenciales, que se generan por las diferentes especialidades médicas.

Esta situación de atención de grandes cantidades de medicamentos produce desafíos en planificación del abastecimiento, el cual en caso de no realizar una programación exacta provocaría ya sea desabastecimiento o sobrestock, problemas que a futuro conllevarían a pérdidas económicas. Ante lo cual, disponer de técnicas modernas como la aplicación de técnicas de aprendizaje automático para la construcción de modelos predictivos basados en datos históricos que permita anticipar la demanda futura, aunque no ha sido aplicada ni validada en contextos de hospitalarios regionales del país, representa una oportunidad innovadora y de bajo costo para la mejora de la gestión farmacéutica, en beneficio de la población asegurada con atención eficiente y oportuna.

2.2 Pregunta de investigación general

¿De qué manera los modelos de aprendizaje automático permiten predecir la demanda de medicamentos esenciales en la farmacia hospitalaria?

2.3 Preguntas de investigación específicas

¿Cuáles son los criterios relevantes que permiten seleccionar los modelos de aprendizaje automático más adecuados para la predicción de la demanda de medicamentos esenciales?

¿Cómo se pueden implementar los modelos de aprendizaje automático para predecir la demanda de medicamentos esenciales a partir de los datos procesados?

¿Qué nivel de desempeño presentan los modelos predictivos según las métricas estadística?

2.4 Objetivo general

Predecir la demanda de medicamentos esenciales en farmacia hospitalaria, utilizando modelos de aprendizaje automático.

2.5 Objetivos específicos

Identificar los criterios relevantes que permitan seleccionar los modelos de aprendizaje automático.

Implementar modelos de aprendizaje automático para predecir la demanda de medicamentos esenciales sobre los datos preprocesados

Evaluar el desempeño de los modelos predictivos utilizando métricas estadísticas y pruebas paramétricas para elegir el modelo con mayor precisión y estabilidad temporal.

2.6 Justificación e importancia

Actualmente en el servicio de farmacia la programación para el abastecimiento de medicamentos suele ser de forma empírica, basándose a menudo en estimaciones o normativas generales en lugar de tener datos exactos, por tanto la información no es exacta y puede dar lugar a errores de cálculo que afecta la disponibilidad y stock de medicamentos, motivo por el cual se plantea este estudio propone la implementación de modelos de aprendizaje automático como herramienta innovadora para predecir la demanda de medicamentos, esta solución permitirá fortalecer y optimizar el proceso de planificación de compras de medicamentos, reducir

perdidas por vencimiento, minimizar quiebres de stock y garantizar una atención oportuna y satisfacción del asegurado, esta investigación se sustenta en la creciente evidencia internacional que demuestran la eficacia de técnicas como egresión lineal, árboles de decisión o KNN para anticipar patrones de consumo y optimizar la gestión de la demanda de medicamentos en hospitales (Joshi et al., 2025)(Hu et al., 2024).

Además, cabe mencionar que la justificación práctica de esta investigación no requiere de inversión tecnológica significativa ya que puede implementarse con herramientas de software libre y datos históricos que se tiene disponibles. Por tanto, su aplicabilidad en farmacias hospitalarias con limitaciones presupuestales hace que se convierta en una alternativa viable y sostenible.

Este trabajo de investigación se fundamenta en teorías de aprendizaje automático, predicción de series temporales y la inteligencia artificial aplicada a la salud, se revisa antecedentes donde existen trabajos sólidos que demuestran la efectividad de modelos como el uso de regresión lineal, árboles de decisión y algoritmos KNN, para anticipar la demanda de productos en diferentes sectores. Estudios internacionales como la de (Joshi et al., 2025), (Hu et al., 2024), quienes validan la pertinencia de estas metodologías en contextos hospitalarios, lo que representa una oportunidad ideal para incorporar soluciones tecnológicas que están accesibles, reproducibles y adaptadas a la realidad de los diferentes Hospitales regionales, donde aún el uso de estas tecnologías es limitados. La aplicación de modelos supervisados entrenados con datos históricos, no requiere de mucho presupuesto ni grandes inversiones menos infraestructura especial, motivo por el cual contribuirá al marco teórico sobre inteligencia artificial en salud, proporcionando evidencia contextualizada y replicable.

Así mismo desde el punto de vista metodológico, esta investigación tiene valor por la comparación sistémica de diferentes modelos predictivos mediante métricas cuantitativas, que permitirá identificar la técnica más precisa y robusta en el contexto local, el proceso de entrenamiento, evaluación y selección del modelo será reproducible, documentado y escalable a otros servicios hospitalarios. Motivo que fortalece el rigor metodológico del estudio y su potencial de generalización.

Además, desde el punto de vista académico este trabajo de investigación resulta ideal para el desarrollo de competencias en ciencia de datos, programación aplicada, análisis de series temporales y evaluación de modelos predictivos. Procesos que fortalecerán el perfil de profesional, motivo por el cual esta investigación se justifica en el aspecto social, que contribuirá a la mejorará de la atención al paciente en la farmacia del hospital, en el aspecto tecnológico, por hacer uso de métodos de aprendizaje automático accesible en la gestión hospitalaria, mientras que en el aspecto académico fomenta la práctica del conocimiento en ingeniería de sistemas al servicio de la salud pública.

III. MARCO TEORICO

3.1 Antecedentes

Se han identificado diferentes estudios, nacionales e internacionales, donde la gestión de medicamentos es una línea de investigación que va en crecimiento en el ámbito de la salud pública y la ingeniería, estos trabajos de investigación han aplicado técnicas de aprendizaje automático, para llegar a su objetivo de predecir la demanda de medicamentos en hospitales, habiendo en algunos casos modelos simples como regresión lineal que pueden alcanzar una precisión exacta.

A nivel internacional tenemos los trabajos de:

Tenemos el trabajo de Joshi et al., (2025), en la revista Research Journal of Pharmacy and Technology en la India, cuyo objetivo fue de predecir medicamentos faltantes y optimizar la cadena de abastecimiento hospitalario. Ellos usaron los modelos LSTM (Long Short-Term Memory), Random Forest y Gradient Boosting, la población fue los datos de consumo de medicamentos en una farmacia hospitalaria durante 5 años. logrando mejorar la planificación y reducción de desabastecimientos críticos, concluyendo que el aprendizaje automático mejora la gestión logística en lugares críticos reduciendo pérdidas por falta de medicamentos.

Hu et al., (2024) efectuaron un estudio exhaustivo sobre la predicción incierta de demanda de medicamentos en hospitales, en Israel, que establecieron como objetivo evaluar la precisión de modelos simples y avanzados en la predicción de demanda hospitalaria, usando el método de comparación lineal, LSTM y XGBoost usando series temporales hospitalarias, con respecto a la población fue la atención de pacientes al Ramban Medical Center, los resultados que obtuvieron indicaron que, en ciertas condiciones, modelos simples pueden ser igual o más efectivos que modelos

complejos, por tanto concluyendo que los modelos más sencillos pueden ser adecuados para hospitales con capacidades técnicas limitadas.

A nivel internacional, hay estudios que han explorado rumbos más complejos como los realizados por Fourkiotis & Tsadiras, (2024) la Universidad Aristotélica de Grecia, cuyo objetivo fue mejorar la precisión de las predicciones de ventas farmacéuticas, mediante un enfoque híbrido que combinó modelos estadísticos tradicionales y técnicas de aprendizaje automático, especialmente frente a patrones estacionales y alta variabilidad en consumo, el método usado es el análisis de univariante de series de tiempo, con segmentación por categoría ATC para obtener estacionalidad relevante. La población con la que se trabajó fue de 600000 registros de ventas de una única farmacia, llegando a conclusión que los modelos híbridos en combinación con series temporales y técnicas de machine learning mejoran significativamente la exactitud de las predicciones de ventas farmacéuticas frente a modelos puramente estadísticos.

Kumar et al., (2023), en la revista IJISRT, en la India, donde plantearon el objetivo de comparar la eficacia de distintos modelos avanzados de predicción en el control de stock. El método implementado fue RNN, LSTM, GRU y modelos ensamblados. La población fue los datos de stock de una cadena farmacéutica, resultando que algunos modelos simples presentaron desempeño competitivo frente a complejos, llegando a la conclusión que algunos modelos simples también llegan a presentar un rendimiento competitivo y son útiles en contextos de alta y baja capacidad tecnológica. Predicción de la demanda de medicamentos para una gestión óptima del inventario médico.

Rivero Albarran et al., (2022), estudio que se realizó en la ciudad de Ibarra en una Farmacia del Instituto del Seguro Social del Ecuador,

plantearon un modelo para predecir la demanda en farmacia, empleando herramientas de inteligencia artificial, modelos basados en series temporales tradicionales, modelo usando redes neuronales, la población fue los datos de dispensación real de medicamentos de la farmacia. El trabajo de investigación evidenció que el uso de modelos computacionales basados en inteligencia artificial mejora sustancialmente la precisión de las estimaciones, lo cual resulta en una mejor gestión del stock y reducción de costos por exceso o falta de medicamentos.

Correa, (2020), implemento un modelo que permitió estimar la demanda para una empresa farmacéutica, en Chile, teniendo como objetivo general, desarrollar un modelo que permita estimar la demanda mensual de medicamentos, utilizando metodología CRISP-DM para estructurar el ciclo del modelado, aplicando ARIMA, promedio simple, series temporales, entre otros. La población con la que trabajo fueron los datos reales de ventas mensuales de medicamentos de la empresa farmacéutica técnicas estadísticas convencionales. Su investigación concluye en la importancia del modelo estadístico convencional basado en ARIMA y series temporales mejora la estimación de demanda frente a métodos muy simples destaca la importancia de la planificación de stock de medicamentos para impedir pérdidas económicas y mejorar la atención al cliente, valiendo como referencia básica para investigaciones más avanzadas en contextos hospitalarios.

A nivel nacional tenemos los trabajos de:

Manrique, (2025) realizo el estudio Machine learning para la gestión de inventarios de medicamentos, en Perú, cuyo objetivo fue desarrollar y validar un modelo de aprendizaje automático para estimar niveles óptimos en stock de medicamentos, con la finalidad de reducir el exceso y desabastecimiento. La metodología aplicada fue el uso de modelos supervisados (regresión, arboles de decisión)

a la vez evaluación mediante MAE, la población fueron los datos históricos de consumo y stock de medicamentos. Llegando a reducir de aproximadamente en 10% en exceso de stock, demostrando la viabilidad practica de Machine Learning en entornos hospitalarios peruanos.

Flores & Montalvo, (2024), realizaron el estudio sobre Machine learning para la predicción en la gestión de inventario dirigida a PYMES de venta de productos tecnológicos, en Perú, cuyo objetivo que se plantearon fue analizar la influencia de un sistema basado en aprendizaje automático en la predicción de gestión de inventario para un PYME, la metodología aplicada fue la comparativa de 3 algoritmos de (regresión lineal, árbol aleatorio , mejora gradual) y la aplicación de la metodología CRISP-DM para el desarrollo del sistema, la población es 64 productos que tenia inventariada la empresa del caso de estudio. Los resultados mostraron que el algoritmo de mejora gradual obtuvo mejores resultados (R^2 mas alto) y que la implementación del sistema permitió mejorar la rotación y disminuir la duración del inventario.

Díaz, (2023) realizo el estudio sobre la predicción de la demanda de medicamentos en un clínica privada de Chiclayo, donde su objetivo fue de desarrollar un modelo basado en minería de datos para estimar la demanda diaria/mensual de medicamentos, usando como técnicas la minería de datos y aprendizaje automático, cuya muestra fue la cantidad de registros reales de dispensación de medicamentos en los periodos de 2021 al 2022, resultando que el modelo logro buenas estimaciones del consumo, comprobando que el aprendizaje automático permite anticipar demanda y reducir errores en la programación de abastecimiento.

Huanco, (2023) realiza el trabajo de investigación sobre modelo de identificación de Covid 19 usando técnicas de Deep Learning a partir de imágenes de rayos x de tórax de los pulmones de los pacientes,

como objetivo se planteó determinar el modelo de proceso diagnóstico del COVID-19 mediante la aplicación de técnicas de Deep Learning a partir de imágenes de rayos X del tórax de los pulmones de los pacientes. Con un total 21165 radiografías de tórax de imágenes de pacientes y se seleccionan las infectadas con COVID-19; posteriormente, el preprocesamiento y procesamiento, fue utilizando modelos de redes neuronales convolucionales como VGG19, DenseNet169, ResNet101 y EfficientNetB0. Asimismo, se evaluó y comparo el rendimiento de los algoritmos mediante matrices de confusión. Se han codificado utilizando el lenguaje de programación Python. El resultado de desempeño más efectivo entre los cuatro modelos evaluados fue EfficientNetB0, con un 99.130% de exactitud y un 99% de precisión en la implementación del algoritmo.

Martinez, (2023) trabajo en la aplicación web basada en machine learning para predecir la demanda de medicamentos en una clínica de Chiclayo, cuya finalidad de estudio fue implementar un prototipo de Python que permitiera al personal de farmacia predecir la demanda futura de medicamentos. Donde la población fueron los datos históricos de dispensación registrados en el sistema hospitalario de la mencionada clínica, llegando a la conclusión que la aplicación presento una precisión aceptable y permitió al personal visualizar estimaciones mensuales de consumo, facilitando la gestión de stock.

3.2 Bases teóricas de la variable

3.2.1 Predicción de la demanda

La predicción de la demanda es una técnica esencial en la gestión de medicamentos en una farmacia, que permite anticipar la cantidad de productos que se requerirán en un periodo futuro. En el ámbito de la salud esta estimación es (Kumar et al., 2023) importante y que garantiza el abastecimiento continuo de medicamentos y evita el desabastecimiento como el sobrestock

que afectaría directamente la atención al paciente y generar pérdidas económicas.

De acuerdo con Díaz, (2023) la predicción de la demanda se consolida como un problema clásico de series temporales, donde el objetivo es de anticipar el comportamiento futuro de una variable dependiente del tiempo. Esta tarea adquiere mayor complejidad cuando se enfrenta a demandas intermitentes. Caracterizadas por largos periodos sin consumo seguidos de picos repentinos de ventas o dispensaciones. En contextos hospitalarios, este comportamiento puede observarse en medicamentos de baja rotación o de uso estacional, lo cual demanda modelos capaces de capturar irregularidades y la no linealidad de los datos.

3.2.2 Aprendizaje automático para demanda

Es una rama de la inteligencia artificial que permite a los modelos aprender patrones a partir de datos etiquetados, es decir, conjunto en los que las variables de entrada y salida son conocidas. En el contexto de predicción de demanda, estos modelos permiten estimar valores futuros (como la cantidad de medicamentos a programar para su compra cada año), basándose en el comportamiento histórico de datos.

Entre los algoritmos más apropiados para este escenario destacan:

Regresión lineal simple: es útil cuando la relación entre variable que no cuentan con infraestructura avanzada (Mousa & Al-Khateeb, 2023).

3.2.2.1 Principios de aprendizaje automático:

El aprendizaje automático (ML) es una rama de la inteligencia artificial que aprende patrones a partir de datos etiquetados (supervisados), donde las variables de entrada (por ejemplo) históricos de ventas, estacionalidad, factores externos) se relacionan con la variable objetivo (demanda futura).

Este paradigma permite capturar estructuras más complejas y no lineales que los métodos clásicos, dando un volumen de datos suficiente. En su revisión, *Machine Learning y algorithms for demand Forecasting* explora varios algoritmos de ML, sus ventajas y limitaciones (Andrewson & Al-Mubarak, 2024).

Tipos de aprendizaje relevantes:

- **Supervisado:** donde se conoce la variable objetivo (demanda futura).
- **No supervisado / clustering:** puede utilizarse para segmentar productos o perfiles de demanda antes de modelar.
- **Aprendizaje reforzado / otros enfoques:** menos comunes en forecasting de demanda, pero potencialmente útiles en escenarios de decisión de inventario adaptativo.

Ciclo metodológico: CRISP-DM adaptado para forecasting.

- **Comprensión del negocio,** definir la necesidad de pronóstico y las implicancias operativas.
- **Comprensión de los datos,** exploración, limpieza, detección de estacionalidad, valores faltantes.
- **Preparación de los datos,** normalización, creación de ventanas de tiempo (*lags*), extracción de características (*features*).
- **Construcción del modelo,** seleccionar algoritmos relevantes, ajustar *hiperparámetros*, validación cruzada temporal.
- **Evaluación,** uso de métricas adecuadas y técnicas de validación robustas.
- **Despliegue (opcional),** integración del modelo en el sistema de gestión hospitalaria.

Ventajas del ML en forecasting.

- Puede modelar relaciones no lineales entre variables.
- Permite incorporar múltiples variables exógenas (clima, campañas médicas, etc.).
- Escalable a grandes volúmenes de datos.
- Puede actualizarse o reentrenarse periódicamente.

Limitaciones y desafíos.

- Requiere volumen suficiente de datos históricos para generalizar bien.
- Riesgo de sobreajuste si no se regulariza correctamente.
- La naturaleza de series de tiempo (autocorrelaciones, no estacionariedad) puede debilitar modelos típicos si no se manejan bien.
- Evaluación adecuada y selección de métricas es clave para evitar conclusiones erróneas (por ejemplo, no comparar modelos con distintos sesgos). En ese sentido, el trabajo de Hewamalage et al., (2023). sobre “Forecast Evaluation for Data Scientists” es una guía muy útil sobre las trampas comunes y mejores prácticas en evaluación de modelos de forecasting con ML

3.2.2.2 Modelos de aprendizaje automático aplicados a predicción de demanda.

Esta sección describe cada algoritmo que se usara, su fundamento teórico, ventajas e inconvenientes y ejemplos de aplicación.

Comparación y criterios de selección

En el “Cuadro de Justificación de Modelos” deberás contrastar criterios como precisión (C2), robustez (C3), interpretabilidad (C4), capacidad de captura de patrones temporales (C1), eficiencia computacional (C5) y generalización (C6).

Por ejemplo, Random Forest puede ser preferido por interpretabilidad y robustez, mientras que XGBoost sobresale en precisión optimizada de errores residuales.

En estudios aplicados al sector farmacéutico, Random Forest y métricas de árbol han superado modelos tradicionales en exactitud (Merkuryeva et al., 2019).

Tabla 1. Cuadro comparativo de justificación de modelos

| Tip o | Modelo | Criterios relevante s | Ventajas principales | Referencia |
|------------------|---------------------------------|--------------------------------------|--|---------------------------|
| ML | Random Forest (RF) | C3, C4, C5, C6 | Modelo no lineal, robusto al ruido, maneja interacciones complejas e interpretabilidad alta | (Nikmah et al., 2023) |
| ML | XGBoost | C2, C3, C4, C5 | Alta precisión predictiva, optimiza errores residuales, incluye regularización y control de complejidad | (Kong et al., 2025) |
| ML | SVR (Support Vector Regression) | C3, C4, C6 | Generaliza bien en conjuntos medianos, control de margen, evita sobreajuste mediante kernel | (Smola & Olkopf, 1996) |
| DL | LSTM (Long Short-Term Memory) | C1, C2, C3, C4 | Captura dependencia de largo plazo, evita el desvanecimiento o del gradiente, adecuada para secuencias largas. | (Bayani et al., 2024) |
| DL | GRU (Gated Recurrent Unit) | C1, C2, C3, C5 | Menor complejidad que LSTM, eficiente en | (Hewamalage et al., 2021) |

| | | | | |
|----|--------------------|--------|--|----------------------|
| | | | datos pequeños y secuencias medianas. | |
| DL | CNN-1D | C3, C5 | Detecta patrones locales, útil en secuencias con ruido o alta frecuencia, capta dependencias cortas. | (Bai et al., 2018) |
| DL | CNN-LSTM (hibrido) | C1–C5 | Combina detección de patrones locales (CNN) y dependencias temporales (LSTM), adecuado para picos y tendencias | (Lim & Zohren, 2021) |

3.2.3 Definición y naturaleza de series temporales

Una serie temporal es una secuencia de observaciones de una variable registradas en intervalos regulares de tiempo. Matemáticamente, puede representarse como $X(t) = \{X_1, X_2, \dots, X_n\}$, donde t denota el tiempo y X_t es el valor observado en el instante t . Estas series permiten analizar el comportamiento de un fenómeno dinámico a lo largo del tiempo, identificar patrones y generar predicciones sobre su evolución futura.

Las series temporales pueden clasificarse en univariantes, cuando se registran una sola variable dependiendo del tiempo (por ejemplo, la demanda diaria de un medicamento), o multivariantes, cuando varias variables interdependientes evolucionan simultáneamente (por ejemplo, demanda, temperatura y estacionalidad). También se distinguen en discretas y continuas según la naturaleza de la medición temporal (Raval, 2021).

Componentes fundamentales: el análisis de series temporales descompone los datos en cuatro componentes esenciales, cuyo

estudio permite comprender la estructura interna de los datos a formular modelos de predicción más precisos.

- a. **Tendencia (Trend):** representa la dirección general del comportamiento de la serie a largo plazo, ya sea creciente o decreciente.
- b. **Estacionalidad (Seasonality):** describe fluctuaciones periódicas que se repiten regularmente en intervalos fijos, como días, meses o estaciones del año.
- c. **Ciclicidad (Cycle):** recoge las oscilaciones de mediano o largo plazo que no representan una periodicidad constante, generalmente asociadas a factores económicos o estructurales.
- d. **Ruido (noise):** componente aleatorio que agrupa las variaciones impredecibles o no explicadas por los demás factores.

Estos componentes pueden combinarse según dos esquemas clásicos:

Modelo aditivo: $y_t = T_t + S_t + C_t + R_t$

Modelo multiplicativo: $y_t = T_t \times S_t \times C_t \times R_t$

El modelo aditivo supone independencia entre los componentes, mientras que el multiplicativo considera que los efectos son proporcionales entre sí (Aguirre, 2022).

3.2.4 Modelos estadísticos tradicionales

Las series de temporales se han modelado tradicionalmente mediante métodos estadísticos basados en la autocorrelación y la dependencia temporal, entre los que destacan:

- **Modelos ARIMA (Autoregressive integrate Moving Average),** que combinan procesos autorregresivos y de promedio móvil aplicando diferenciación para garantizar la estacionalidad.
- **Modelo de suavizamiento exponencial:** útiles para capturar tendencias y estacionalidades con mayor simplicidad computacional

- **Modelo de Croston:** diseñado específicamente para demandas intermitentes, donde se estiman de forma separada la magnitud y la frecuencia de los eventos de consumo.(Raval, 2021).

3.2.5 Series temporales y aprendizaje profundo (Deep Learning) para series temporales

con el auge del aprendizaje profundo (**deep learning**), las redes neuronales se han convertido en una alternativa poderosa para el modelado de series temporales complejas. Estas técnicas aprenden representaciones no lineales de los datos y capturan dependencias de largo plazo imposibles de modelar con métodos tradicionales.

Aquí abordas los modelos más avanzados para forecasting, su arquitectura, ventajas y retos.

- LSTM (Long Short-Term Memory)

Redes recurrentes diseñadas para capturar dependencias de largo plazo en secuencias. Tiene puertas de entrada, olvido y salida que regulan el flujo de información.

Los LSTM fueron propuestas por Hochreiter y Schmidhuber (1997). Su principal innovación es la introducción de un mecanismo de memoria interna controlado por tres puertas que determinan que información recordar, cual olvidar y cual usar para la predicción.

El bloque LSTM tiene los siguientes componentes:

a. Puertas de olvido (forget Gate)

Decide que parte de la información anterior debe conservarse o eliminarse.

$$f_t = \sigma(W_f[h_{t-1}, x_t] + b_f)$$

b. Puerta de entrada (Input Gate)

Determina que nueva información será añadida a la memoria

$$i_t = \sigma(W_i[h_{t-1}, x_t] + b_i)$$

c. Candidata a Memoria (Cell candidate)

Genera una versión actualizada del contenido que podría almacenarse

$$\tilde{c}_t = \tanh(W_c[h_{t-1}, x_t] + b_c)$$

d. Actualización del estado de celda

Combina lo que se conserva del pasado con lo nuevo que se agrega:

$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t$$

donde c_t es la **memoria a largo plazo**, y \odot representa una multiplicación elemento a elemento

e. Puerta de salida (Output Gate)

Decide que parte de la memoria interna se usara como salida del modelo

$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o)$$

$$h_t = o_t \odot \tanh(c_t)$$

Aquí h_t es el estado oculto final, que resume toda información aprendida hasta el tiempo t .

Interpretación intuitiva:

- La LSTM funciona como una memoria selectiva
- Puede “recordar” patrones importantes de largo plazo (como tendencia estacionales o cambios de comportamiento) y “olvidar” ruido y variaciones irrelevantes.
- Este mecanismo la hace especialmente para predecir series temporales complejas como la demanda irregular de medicamentos

hospitalarios, donde los patrones no siguen una periodicidad fija.

Ventajas: maneja bien secuencias largas, retiene información importante pasada, es estándar en forecasting secuencial.

Limitaciones: entrenamiento más costoso, riesgo de sobreajuste si no hay suficientes datos.

- **GRU (Gated Recurrent Unit)**

Variante más simple de LSTM con menos parámetros (combinación de puertas).

La

Ventajas: menor complejidad, velocidad de entrenamiento más alta, rendimiento similar en muchos casos.

Limitaciones: en series muy complejas, puede perder capacidad frente a LSTM en dependencia muy larga.

- **Red GRU (Gated Recurrent Unit)**

La GRU, propuesta por Joshi et al., (2025) es una versión simplificada de la LSTM que busca mantener su capacidad de memoria pero con menor complejidad computacional. En lugar de tres puertas y dos estados (como LSTM), la GRU utiliza dos puertas y un único estado oculto.

a. **Puerta de actualización (Update Gate)**

Controla cuanto de la información anterior se mantiene.

$$z_t = \sigma(W_z[h_{t-1}, x_t] + b_z)$$

b. **Puerta de reinicio (Reset Gate)**

Regula cuanta información del pasado se olvida al generar una nueva predicción:

$$r_t = \sigma(W_r[h_{t-1}, x_t] + b_r)$$

c. Candidato de nuevo estado

Combina la información nueva con la memoria previa filtrada por la puerta de reinicio

$$\tilde{h}_t = \tanh(W_h[r_t \odot h_{t-1}, x_t] + b_h)$$

d. Actualización del estado oculto:

Mezcla la información vieja la nueva según el valor de z_t :

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h}_t$$

interpretación intuitiva:

Si $z_t \approx 1$, la red mantiene el estado anterior (memoria)

Si $z_t \approx 0$, la red se “reinicia” y genera un nuevo estado como la información mas reciente

Este comportamiento hace que la GRU aprenda mas rápido y seas más reciente.

Este comportamiento hace que la GRU aprenda mas rápido y sea mas eficiente en contextos donde las secuencias no son demasiado largas o los datos son limitados,

3.2.6 Series temporales y aprendizajes automático (Machine Learning) para series temporales: Random Forest y XGBoost

1. **Fundamento teórico:** En los últimos años, además de los enfoques clásicos (ARIMA modelos de suavizamiento, redes neuronales recurrentes), se han explorado modelos de aprendizaje automático supervisado (Machine Learning) para predecir series temporales. entre ellos Random Forest y XGBoost son populares por su capacidad de modelar relaciones no lineales, manejar interacciones entre variables y trabajar bien con variables exógenas (atributos adicionales). A continuación, se describe como se aplican estos modelos a

series temporales sus ventajas, limitaciones y ejemplos de aplicaciones (Bayani et al., 2024).

- **Random Forest para series temporales**

El modelo es un ensamble de múltiples árboles de decisión $T_b(x)$, cada uno entrenando sobre un subconjunto de datos D_b generado mediante muestreo Bootstrap.

El pronóstico final se obtiene mediante el promedio de las predicciones de los árboles:

$$\hat{y}_t = \frac{1}{B} \sum_{b=1}^B T_b(X_t)$$

Donde B : número total de árboles en el bosque,

$T_b(X_t)$: predicción del árbol b para la entrada X_t .

Cada árbol se ajusta minimizando un criterio de impureza, como el error cuadrático medio (MSE) en el nodo:

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

donde \bar{y} es el valor medio de las observaciones dentro del nodo.

El algoritmo introduce aleatoriedad al:

- a. Seleccionar aleatoriamente muestra de entrenamiento (baggin).
- b. Seleccionar aleatoriamente subconjunto de variables en cada división

Esto reduce la varianza y evita el sobreajuste.

2. **Aplicación a series temporales**

En un contexto temporal, los datos deben organizarse respetando la secuencia. Así, se construye una ventana de entrenamiento (Rolling window)

$$\begin{array}{l} \text{Input: } [y_{t-1}, y_{t-2}, \dots, y_{t-p}] \\ \text{Output: } y_t \end{array}$$

Y se entrena el modelo para aprender la función:

$$f_{RF}: \mathbb{R}^p \rightarrow \mathbb{R}, \hat{y}_t = f_{RF}(y_{t-1}, \dots, y_{t-p})$$

El pronóstico multi-paso se puede realizar:

Iterativamente: usar la predicción \hat{y}_{t+1} como entrada para estimar \hat{y}_{t+2} ;

Directamente: entrenar un modelo independiente para cada horizonte h .

3. Propiedades matemáticas

El RF aproxima la esperanza condicional del valor futuro:

$$E[Y_t | X_t] \approx \frac{1}{B} \sum_{b=1}^B T_b(X_t)$$

y su varianza esperada se reduce aproximadamente según:

$$Var(\hat{y}_t) = \rho\sigma^2 + \frac{1-\rho}{B}\sigma^2$$

donde ρ es la correlación promedio entre los árboles. A medida que $B \rightarrow \infty$, el segundo termino tiende a cero, mejorando la estabilidad del modelo

- Random Forest para series temporales

a. Principio del método

XGBoost (extreme gradient Boosting), es un algoritmo de boosting de gradiente basado en arboles de decisión, a diferencia del bagging de RF, XGBoost, entrena los árboles secuencialmente, donde cada árbol corrige los errores del anterior.

Sea el conjunto de datos $\{(x_i, y_i)\}_{i=1}^n$. El modelo consiste en la suma de K árboles de regresión:

$$\hat{y}_i = \sum_{k=1}^K f_k(x_i), f_k \in \mathcal{F}$$

donde \mathcal{F} es el espacio de funciones representadas por árboles.

La función objetivo general combina la pérdida y la regularización:

$$\mathcal{L}(\phi) = \sum_{i=1}^n l(y_i, \hat{y}_i) + \sum_{k=1}^K \Omega(f_k)$$

donde:

- $l(y_i, \hat{y}_i)$: función de pérdida (por ejemplo, cuadrática),
- $\Omega(f_k) = \gamma T + \frac{1}{2} \lambda \|w\|^2$: término de regularización que penaliza la complejidad del árbol (número de hojas T y pesos w).

b. Expansión de boosting de gradiente

El algoritmo agrega nuevos árboles ajustando los residuos del modelo anterior. En la interacción t , se construye un árbol que minimiza la expansión de segundo orden de la pérdida:

$$\mathcal{L}^{(t)} \approx \sum_{i=1}^n [g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i)] + \Omega(f_t)$$

donde:

$$g_i = \frac{\partial l(y_i, \hat{y}_i^{(t-1)})}{\partial \hat{y}_i^{(t-1)}}, h_i = \frac{\partial^2 l(y_i, \hat{y}_i^{(t-1)})}{\partial (\hat{y}_i^{(t-1)})^2}$$

Estos gradientes y hessianos permiten ajustar cada árbol para minimizar la pérdida total. El modelo final resulta de sumar todas las contribuciones parciales:

$$\hat{y}_i^{(T)} = \sum_{t=1}^T \eta f_t(x_i)$$

donde η es la tasa de aprendizaje (learning rate)

c. Aplicación series temporales

El modelo se entrena con el mismo esquema supervisado descrito antes. Por ejemplo, para predecir la demanda farmacéutica diaria:

$$y_t = f_{XGB}(y_{t-1}, y_{t-2}, \dots, y_{t-p}, D_t, F_t, M_t)$$

donde:

- D_t : indicador de día de la semana,
- F_t : variable binaria de feriado,
- M_t : mes o estación del año.

El modelo captura no lineales e interacciones entre variables y temporales exógenas sin necesidad de especificar un modelo paramétrico.

d. Función de pérdida y evaluación

Para la regresión de series temporales se usan métricas de error como:

Error Absoluto Medio (MAE):

$$MAE = \frac{1}{n} \sum_{t=1}^n |y_t - \hat{y}_t|$$

Error Cuadrático Medio (RMSE):

$$RMSE = \sqrt{\frac{1}{n} \sum_{t=1}^n (y_t - \hat{y}_t)^2}$$

Error Porcentual Absoluto Medio (MAPE):

$$MAPE = \frac{100}{n} \sum_{t=1}^n \left| \frac{y_t - \hat{y}_t}{y_t} \right|$$

Estas métricas permiten comparar el rendimiento entre RF, XGBoost y redes neuronales profundas (Nikmah et al., 2023)

3.2.7 Evaluación de modelos predictivos para series de tiempo

En esta sección explicas las métricas que usarás, los principios de partición de datos y pruebas estadísticas.

Partición de datos y validación temporal

- En forecasting no se puede hacer validación cruzada aleatoria: se debe usar validación temporal (por ejemplo, entrenamiento con años anteriores y prueba con años posteriores).
- Se puede emplear estrategia walk-forward o rolling window (ventanas desplazables).
- En el artículo “Forecast Evaluation for Data Scientists” se discuten trampas comunes (como fuga de información) y buenas prácticas en evaluación de modelos de forecasting con ML/DL.

Métricas de error más usadas

- MAE (Mean Absolute Error): error promedio absoluto, fácil de interpretar.
- RMSE (Root Mean Squared Error): penaliza más errores grandes, es sensible a valores extremos.
- MAPE (Mean Absolute Percentage Error): error porcentual medio; útil para comparar entre diferentes escalas, pero puede tener problemas si los valores reales cercanos a cero.
- sMAPE (Symmetric MAPE): versión simétrica que mitiga sesgos de MAPE.
- En muchos trabajos de forecasting, se elige una o varias de estas métricas para comparar modelos.

Pruebas estadísticas comparativas

- Cuando se comparan múltiples modelos sobre múltiples series o pliegues, es adecuado usar pruebas no paramétricas como Friedman seguido de pruebas post-hoc (por ejemplo, Nemenyi) para determinar diferencias significativas entre los rendimientos de los modelos.
- Esto es especialmente importante para evitar seleccionar “mejor por azar” cuando las diferencias son pequeñas.

3.3 Marco conceptual

Medicamentos esenciales

Son aquellos medicamentos que en un país consideren necesarios para atender las necesidades de atención de salud prioritarias de la población. Estos medicamentos son estratégicos puesto que su disponibilidad es crítica para garantizar la atención médica de los pacientes en los hospitales, especialmente cuando hay existencia de una crisis sanitaria, cuando un incremento repentino de demanda mundial o cierres de fronteras pueden poner en riesgo el suministro, deben estar disponibles en todo momento, ser asequibles y seguros (OMS, 2023). Se selecciona teniendo en cuenta la prevalencia de las enfermedades y su eficacia, seguridad y costo-efectividad comparativa. La mayoría de los medicamentos esenciales son antiguos con varios años en el mercado y diversos sin protección de patentes y que aún siguen siendo los convenientes para lidiar con ciertos síntomas o enfermedades(OMS, 2012).

Aprendizaje automático

Es una rama de la inteligencia artificial y la informática que se ajusta al desarrollo de métodos para que los ordenadores permitan aprender y optimizar a partir de datos sin necesidad de programación explícita, su objetivo es replicar los procesos de aprendizaje humano, que mejoran la precisión para tareas específicas. Por ello se usan algoritmos para identificar patrones, predecir resultados y tomar decisiones(Poveda et al., 2023).

Farmacia hospitalaria

La farmacia hospitalaria es el servicio institucional responsable de selección, adquisición, dispensación y control de medicamentos, material médico o productos sanitarios de un hospital, su labor es fundamental para asegurar que los pacientes reciban tratamientos oportunos y seguros coordinando con el equipo médico y participando en campañas de uso racional de fármacos(DIGEMID, 2002) .

Predicción.

Se refiere a la capacidad de anticipar la demanda futura de medicamentos esenciales, en una farmacia hospitalaria, puede ser usando modelos de aprendizaje automático, que puede permitir la optimización de la gestión de stock y reducir el desperdicio evitando el desabastecimiento(Hu et al., 2024).

Modelos supervisados

Los modelos supervisados son algoritmos de aprendizaje automático que requieren datos etiquetados tanto las variables de entrada (predictoras) como la salida esperada (valor real) para entrenarse. El modelo aprende la relación entre variables independientes y dependientes que luego permiten realizar predicciones sobre nuevos datos no etiquetados. Estos modelos son ampliamente usados en problemas de predicción como la demanda hospitalaria debido a su capacidad de aprendizaje controlado y evaluación mediante métricas cuantitativas.(Rivero et al., 2022).

Regresión lineal

Es una técnica estadística primordial que busca modelar la dependencia entre una variable dependiente y una o más variables independientes mediante una ecuación lineal. Es usada frecuentemente en predicción de demanda por su simplicidad e interpretabilidad y su facilidad de implementación, lo que hace ideal en aplicaciones prácticas como la predicción de demanda en entornos hospitalarios(Rivero et al., 2022).

Arboles de decisión

Son modelos de aprendizaje supervisados que utilizan una estructura tipo árbol para dividir datos en clases o valores, según condiciones en sus atributos, cada nodo representa una decisión y cada rama una consecuencia. Permiten entender el proceso de predicción, donde se requiere comprensión clara sobre cada variable contribuye a la decisión final(Rivero et al., 2022).

Desabastecimiento de medicamentos

Se refiere a la falta de disponibilidad de medicamentos esenciales en el momento que se requieren. Es un problema de salud pública que puede derivarse de mala gestión, errores en la previsión o interrupciones en la cadena de suministros(Joshi et al., 2025).

Validación de modelos

Proceso que consiste en evaluar la precisión y confiabilidad de un modelo predictivo. Se utilizan métricas para verificar el rendimiento sobre un conjunto de datos no utilizado en el entrenamiento. Para medir precisión, estabilidad y capacidad predictiva de los modelos en entornos reales(Rivero et al., 2022).

Gestión farmacéutica

Conjunto de procesos organizados para garantizar la disponibilidad, calidad y uso racional de medicamentos en una institución de salud, involucra planificación, presupuesto, control de stock y análisis de consumo, siempre en función de la demanda real y los costos asociados (Rivero et al., 2022)..

IV. ESTRATEGIA METODOLOGICA

4.1 Tipo y nivel de investigación

4.1.1 Tipo: Cuantitativa, ya que se basa en la recolección y análisis de datos numéricos sobre el consumo de medicamentos, para medir y analizar relaciones mediante procedimientos estadísticos, Hernández-Sampieri et al., (2002) destacan que la investigación cuantitativa “busca explicar los hechos mediante la proporción entre variables cuantificables y modelos estadísticos adecuados”.

Es también aplicada ya que busca resolver un problema práctico para facilitar el análisis de datos y obtener información útil, según (Hernández-Sampieri et al., 2002), “la investigación aplicada se orienta a producir conocimiento útil que contribuya a la solución de problemas concretos”

4.1.2 Nivel:

Predictivo, busca prever el comportamiento o resultados futuros basados en patrones históricos. Hernández-Sampieri et al., (2002) explican que “la investigación predictiva emplea modelos estadísticos o algoritmos para anticipar tendencias futuras con base en datos empíricos”.

Con un nivel explicativo por que se enfoca en comprender la relación causa efecto entre variables investigadas, “el nivel explicativo permite identificar como y porque una o más variables independientes influyen sobre una variable dependiente” (Hernández-Sampieri et al., 2002).

4.2 Diseño de investigación

El diseño es no experimental, de tipo longitudinal y retrospectivo ya que se utilizarán datos históricos de consumo de medicamentos sin manipular las variables de los años 2022 hasta el 2024 para identificar patrones, analizar tendencias y efectuar predicciones futuras.

Según Hernández-Sampieri et al.,(2002), en un diseño no experimental “no se manipulan deliberadamente las variables independientes sino que se observan los fenómenos tal como se dan en su contexto natural para posteriormente analizarlos”, asimismo, estos autores clasifican los diseños no experimentales en transversales y longitudinales, siendo este último el más adecuado “se recolectan datos en diferentes puntos del tiempo para conocer su evolución”. En este trabajo de investigación la dimensión temporal se cubre al considerar registros de tres años consecutivos, como también lo define como un diseño retrospectivo ya que se analizarán registros ya existentes en los sistemas de farmacia.

4.3 Hipótesis General

Los modelos de aprendizaje automático permiten predecir la demanda de medicamentos esenciales en farmacia hospitalaria, a partir de datos históricos.

4.4 Hipótesis Específicas:

La selección adecuada de criterios relevantes (precisión, robustez, interpretabilidad y complejidad) influye significativamente en la elección de los modelos de aprendizaje automático más apropiados para la predicción de la demanda de medicamentos esenciales.

La implementación de modelos de aprendizaje automático, permiten obtener predicciones de la demanda de medicamentos esenciales a partir de datos procesados en la farmacia hospitalaria.

Existen diferencias estadísticas significativas en el desempeño de los modelos predictivos según las métricas de evaluación, determinando aquel con mayor precisión y estabilidad temporal en las predicciones.

4.5 Operacionalización de las variables

Tabla 2. Operacionalización de variables

| Variable | Dimensión | Indicadores | Escala de valores | Indicadores | Niveles y rangos | Escala de medición |
|---|-----------------------------|---|-----------------------|---|--|--------------------|
| Demanda de medicamentos (variable dependiente) | Volumen de consumo | Total, mensual de medicamentos dispensados | Cuantitativa continua | Total, mensual de medicamentos dispensados | De 0 a n unidades por medicamento | Cuantitativa |
| | Frecuencia de consumo | Número de solicitudes por medicamento | Cuantitativa discreta | Número de solicitudes por medicamento | Mensual (1-12) | |
| | Periodicidad de consumo | Días con mayor demanda | Cuantitativa discreta | Días con mayor demanda | Días del 1 al 31 | |
| Modelo de aprendizaje automático (variable independiente) | Algoritmo utilizado | Tipo de modelo aplicado (RF, XGBoost, LSTM, GRU) | Cualitativa nominal | Tipo de modelo aplicado | ML: árboles de decisión / DL: redes neuronales recurrentes | Cualitativa |
| | Precisión del modelo | Nivel error (RMSE, MAE, MAPE, SMAPE) | Cuantitativa continua | Error promedio y coeficiente de determinación | 0 a 1 (menor valor = mayor precisión) | Cuantitativa |
| | Capacidad de generalización | Coefficiente de determinación (R^2) y desempeño en conjunto de prueba | Cualitativa nominal | Variables seleccionadas | 0 a 1 (mayor valor = mejor generalización) | Cualitativa |

4.6 Población y muestra:

Población: estará conformada por todos los registros de dispensación y consumo de medicamentos esenciales MINSA, (2023), registrados en el sistema de farmacia ESSI, durante el periodo 2022, 2023 y 2024, un total de 287 medicamentos esenciales,

Definición operativa de la población:

Universo de registros de consumo con un total de 805 medicamentos con un total de 2,438,870 registros y se trabajó con 287 medicamentos esenciales definidos en la lista de medicamentos en el ESSI (columna COD_MEDIC).

Ámbito temporal: 2022 – 2024.

Fuente: módulos transaccionales institucionales ESSI consolidados(.CSV).

Muestra: a partir del universo de 287 medicamentos esenciales, sin embargo, trabajar con los 287 medicamentos esenciales resultaría muy pesado desde el punto de vista computacional, se aplicó el principio de Pareto (80/20), el cual establece que aproximadamente el 20% de los elementos de un conjunto explican el 80 % del efecto observado.(Hernández-Sampieri et al., 2002).

En este caso, los medicamentos con mayor volumen y frecuencia concentran la mayor parte del total dispensado por la farmacia hospitalaria. Pareto como método de selección no probabilístico es común en investigaciones aplicadas de optimización, ya que permite concentrar el análisis en los elementos de mayor impacto operativo y relevancia estadística.

$$n=P \times N$$

donde:

n = tamaño de la muestra representativa

P = proporción seleccionada (20 % = 0.20)

N = total de medicamentos esenciales (287)

$$n= 0.20 \times 287 =57.4$$

no obstante, consideramos la reducción computacional necesaria para el entrenamiento de modelos de aprendizaje automático y la distribución acumulada real del consumo, se aplicó un ajuste empírico basado en el análisis de frecuencia acumulada (Widiatmika, 2015).

Distribución acumulada de consumo:

por cada medicamento i , se calculó su promedio de consumo total:

$$p_i = \frac{C_i}{\sum_{j=1}^N C_j}$$

Donde:

C_i = volumen total de dispensación del medicamento i

$\sum_{j=1}^N C_j$ = volumen total dispensado de todos los medicamentos.

Luego se ordenaron los medicamentos en forma descendente por C_i y se obtuvo la frecuencia acumulada.

$$P_k = \sum_{i=1}^k p_i$$

El punto de corte k se determinó donde $P_k \geq 0.80$; es decir, el subconjunto mínimo de medicamentos que acumula el 80% del volumen total de consumo.

El resultado empírico fue $k=30$ lo que significa que 30 medicamentos explican aproximadamente el 85% del consumo histórico total, por lo que constituyen una muestra altamente representativa y estadísticamente significativa para los fines del modelamiento predictivo (Mfizi et al., 2023).

Se seleccionó una muestra representativa de 30 medicamentos. La selección se realizó mediante un criterio de Pareto (80/20), priorizando aquellos con mayor volumen de dispensación anual, mayor frecuencia de prescripción y alta relevancia terapéutica de

acuerdo con la clasificación ATC (Anatomical Therapeutic Chemical Classification System) de la OMS.

4.7 Técnicas e instrumentos de recolección de datos:

Los datos se obtuvieron de manera directa con la técnica de recolección de datos mediante revisión documental, enfocada en la recuperación estructurada de registros históricos de consumo de medicamentos a partir de los registros generados por el EXPLOTADATOS (sistema de reportes de gestión de medicamentos) de la farmacia del Hospitalaria, la extracción se realizó a partir de archivos digitales en formato .CSV, desde los módulos de farmacia, del menú tipo de movimiento y de forma mensual, para después integrar información de los años 2022, 2023 y 2024. Esta técnica es ampliamente utilizada en investigaciones para análisis de series temporales, control de procesos y modelos predictivo, sin intervención directa del investigador, Esto permite el análisis retrospectivo sin necesidad de realizar intervenciones experimentales.(Richards et al., 2023).

La información recolectada corresponde a movimiento internos de medicamentos esenciales, incluyendo variables como el código del medicamento (COD_MED), la descripción (DES_MEDICAMENTO), la fecha del movimiento (FECHA_DOC_MOV), la cantidad (CANTIDAD), la unidad (UNIDAD), el precio (PRECIO), el periodo(PERIODO), el proceso de recopilación considero la integración de múltiples archivos mensuales en una única base de datos consolidada mediante herramientas de procesamiento de datos con Python(Pandas)

4.8 Técnicas de análisis e interpretación de datos.

4.8.1 Preparación de datos

El tratamiento de los datos comprendido varias etapas sucesivas orientadas a garantizar la calidad, coherencia y representatividad

de la información, antes del modelamiento predictivo (Baumann et al., 2023).

4.8.2 Integración de fuentes: consolidación de archivos mensuales de consumo en un único dataset con 2,438,870 de registros iniciales, la integración permitió disponer de una base de datos continua y homogénea, en formato .CSV, que sirvió como punto de partida para el análisis exploratorio y la construcción de los modelos de aprendizaje automático.

4.8.3 Filtrado de medicamentos esenciales: se utilizó un listado de 287 medicamentos esenciales (columna COD_MEDIC) como criterio de inclusión.

4.8.4 Depuración de dataset: se aplicó un proceso de limpieza y depuración de datos, eliminando registros que no representan consumo real o que pertenecen a otros establecimientos fuera del alcance del estudio:

Se excluyeron las columnas por establecimiento: registros provenientes de farmacias no consideradas en el análisis (COD_FARM 14 y 16, que corresponden a la Farmacia CAMEC (medicina alternativa y Hospital Perú).

Exclusión por tipo de movimiento: Se eliminaron tipos de movimientos no representativos del consumo efectivo COD_MOVIMIENTO (101, 102, 103, 105, 106, 107, 108, 201, 202, 203, 205, 206, 208, 207, 211) que hace referencia a movimientos como: devolución a almacén SAP, devolución de paciente, devolución de servicio, egreso no informa al SAP, egreso transferencia a farmacia/deposito, entrega a servicio, extorno de un egreso (informa al SAP), extorno de un ingreso (informa al SAP), ingreso de almacén SAP, ingreso no informa al SAP, ingreso transferencia de farmacia/deposito.

Control de la calidad de datos: eliminar duplicados exactos; descartar registros con COD_MEDIC nulo y estandarizar formatos de fecha. Todos los filtros que se realizaron aseguran que la población de análisis refleje demanda efectiva (salidas/consumo) en el periodo y establecimientos objetivo,

evitando sesgos por entradas administrativas o sedes fuera de alcance

Control de duplicados y nulos: se uniformaron los formatos de fecha para asegurar la secuencia cronológica de las series de consumo.

Limpieza estructural: eliminación de columnas innecesarias para análisis predictivo (IDENTIFICADORES DE PERSONAL, IDENTIFICADORES DE PACIENTES, CODIFICACIONES DE DIAGNÓSTICOS, etc.).

Transformación para modelamiento: agregación temporal de las series por medicamento (COD_MEDIC) a nivel mensual. Cálculo de variables derivadas como tendencia, estacionalidad o retardos, necesarios para algoritmos de predicción. División del dataset en conjuntos de entrenamiento y prueba mediante validación temporal.

Todo el procesamiento se realizó en Python, utilizando librerías especializadas como pandas, numpy, scikit-learn, matplotlib y statsmodels, por su versatilidad, escalabilidad y la disponibilidad de librerías especializadas para el análisis de datos y modelado predictivo, el manejo de los datos se organizó mediante clases y funciones definidas en scripts estructurados. En particular, se implementó una clase llamada ProductosDatos, que permitió encapsular toda la información y los resultados asociados a cada medicamento esencial, facilitando el procesamiento y la gestión modular del dataset.

Esta clase definió variables y estructuras clave como el código del medicamento, descripción, el DataFrame con la serie temporal de consumo, las predicciones generadas por los modelos, las métricas de evaluación como MAE, RMSE y SMAPE. A continuación, se muestra una versión simplificada de la clase utilizada, las variables relevantes para el tratamiento de datos se muestran en la tabla de definición de variables.

Tabla 3 Definición de variables de la clase ProductoDatos

| Variable | Tipo | Observación |
|---|-------------|---|
| Medicamento_id | Numérico | Identificador único del medicamento (COD_MEDIC) |
| Nombre | Cadena | Nombre del medicamento (DESC_MEDICAMENTO) |
| Datos | DataFrame | Contiene los registros históricos de consumo del medicamento |
| Mape,smape,mse,mae | Numerico | Almacenan las tasas de error de las predicciones obtenidas |
| Fecha_inicial, fecha_final | Fecha | Representa el rango temporal del periodo de consumo analizado. |
| Predicciones_entrenamiento, predicciones_prueba | Array | Almacenan los resultados de las predicciones generadas en entrenamiento y prueba. |

4.8.5 Carga de datos

La carga de datos se realizó a partir de los archivos exportados directamente del módulo EXPLOTADATOS del sistema ESSI, en formato CSV, que contienen registros históricos de consumo de medicamentos correspondientes a los años 2022, 2023 y 2024.

Cada archivo presenta una estructura delimitada por comas, donde cada fila representa un movimiento o transacción de un medicamento (fecha, código, nombre, unidad, cantidad y precio).

Figura 1. Ejemplo de datos de consumo de medicamentos esenciales

```
PERIODO,FECHA_DOC_MOV,COD_MEDIC,DESC_MEDICAMENTO,UNIDAD,CANTIDAD,PRECIO
202204,23/04/2022,10200012,ACETILCISTEINA 200 MG,SOB,15.0,0.19
202204,28/04/2022,10150068,CLORURO DE SODIO 0.9% X 100 ML,FR,3.0,1.06
202204,29/04/2022,10900034,ALCOHOL POLIVINÍLICO 14 MG / ML X 15 ML GOTAS OFTÁLMICAS,FR,1.0,4.27
```

Fuente. Elaboración propia a partir de archivos del ESSI

Este formato permitió una lectura eficiente y la preparación de los datos para su procesamiento posterior en los modelos predictivos.

La variable fecha se transformó al tipo de dato datetime y se utilizó como índice temporal principal para generar las series de tiempo por medicamento.

Para la carga y transformación de los archivos se empleó la función `read_csv()` de la biblioteca Pandas (Python), que facilitó la lectura masiva y homogénea de los datos antes del modelamiento.

Figura 2. Función cargar datos

```
def cargarDatos(self):
    file = 'data/' + str(self.producto_id) + '.csv'
    self.datos = pd.read_csv(file)
    self.datos['FECHA_DOC_MOV'] = pd.to_datetime(self.datos['FECHA_DOC_MOV'], dayfirst=True)
    self.datos = self.datos[["FECHA_DOC_MOV", "CANTIDAD", "PRECIO"]]
    self.datos = self.datos.set_index('FECHA_DOC_MOV')
```

Esta función es parte de la clase `medicamentoDatos` y permite asignar automáticamente a cada objeto su archivo correspondiente, convirtiendo los datos en un `DataFrame`, con la fecha como índice. Esto es esencial para el tratamiento de series temporales en modelos de predicción como Prophet, XGBoost o LSTM, donde el componente temporal es fundamental.

4.8.6 Análisis exploratorio

Para la exploración de datos se utilizó la tabla de frecuencias y la visualización grafica mediante histogramas, con el objetivo de

identificar patrones en el consumo de medicamentos esenciales este análisis exploratorio de datos, sirve para comprender la distribución de la demanda. Se emplearon las librerías pandas, matplotlib.pyplot y seaborn de Python para generar estadísticas descriptivas y visualizar la distribución de cantidades vendidas por producto. En esta etapa se analizó el medicamento más frecuentemente distribuido: METAMIZOL SODICO 1G/2ML, con un total de 97355 registros, este fármaco fue utilizado como caso de estudio ilustrativo, debido a su alta frecuencia y representatividad en la base de datos consolidada, (consumo_medicamentos_esenciales_final_investigacion.csv), cargamos los datos y seleccionamos el medicamento objetivo.

Figura 3. Ruta de los datos a trabajar

```
import pandas as pd
import matplotlib.pyplot as plt

# 1) Cargar dataset
archivo = r"D:\2022a12024\consumo_medicamentos_esenciales_final_investigacion.csv"
df = pd.read_csv(archivo, encoding='utf-8')

# 2) Elegir el medicamento (o detectarlo automáticamente si prefieres)
medicamento_objetivo = "METAMIZOL SODICO 1 G/2 ML"
df_med = df[df["DESC_MEDICAMENTO"] == medicamento_objetivo].copy()
```

Generamos la tabla de frecuencias a partir de la columna CANTIDAD, que representa el número de unidades dispensadas por movimiento. Esto permitió identificar los valores más repetidos y su distribución, mediante la función *value_counts* que devuelve una serie que contiene distintos valores incluido junto con el número de apariciones, con acumulados y porcentajes que se muestra en la tabla de frecuencias.

Figura 4. Código para generar la tabla de frecuencias

```
# 3) Tabla de frecuencias de CANTIDAD (robusta)
frecuencias = (
    df_med["CANTIDAD"]
        .astype(float)                # asegura tipo numérico
        .value_counts()              # cuenta ocurrencias por cantidad
        .sort_index()                # orden por cantidad (no por frecuencia)
        .rename_axis("cantidad")     # nombre al índice (las cantidades)
        .reset_index(name="numero_veces") # nombre a la serie (las frecuencias)
)
```

Tabla 4 Tabla de frecuencias dispensadas por transacción para metamizol sodico1g/2ml

| Cantidad | Número de veces | Cantidad acumulada | Porcentaje (%) | Porcentaje acumulado (%) |
|----------|-----------------|--------------------|----------------|--------------------------|
| 1 | 27,592 | 27,592 | 28.34 | 28.34 |
| 2 | 19,391 | 46,983 | 19.92 | 48.26 |
| 3 | 29,248 | 76,231 | 30.05 | 78.30 |
| 4 | 3,963 | 80,194 | 4.07 | 82.37 |
| 5 | 422 | 80,616 | 0.43 | 82.81 |
| 6 | 15,356 | 95,972 | 15.77 | 98.58 |
| 7 | 128 | 96,100 | 0.13 | 98.71 |
| 8 | 113 | 96,213 | 0.12 | 98.83 |
| 9 | 180 | 96,393 | 0.18 | 99.01 |
| 10 | 89 | 96,482 | 0.09 | 99.10 |
| 11 | 1 | 96,483 | 0.001 | 99.10 |
| 12 | 716 | 97,199 | 0.74 | 99.84 |
| 13 | 3 | 97,202 | 0.003 | 99.84 |
| 14 | 4 | 97,206 | 0.004 | 99.85 |
| 15 | 35 | 97,241 | 0.04 | 99.88 |

FUENTE: Elaboración propia en Python con value_counts() y cumsum(), sobre la columna cantidad

Con respecto a los datos obtenidos en la tabla 2, se puede observar que: el 92% del total de registros de este medicamento corresponde a transacciones con cantidades entre 1 y 6 unidades.

Los valores más frecuentes fueron:

- 3 unidades (30.05%)
- 1 unidad (28.34%)
- 2 unidades (19.92%)

Los valores superiores a 10 unidades representan menos del 1% del total de casos, lo que permite clasificarlos como valores atípicos (outliers) en contextos de modelado.

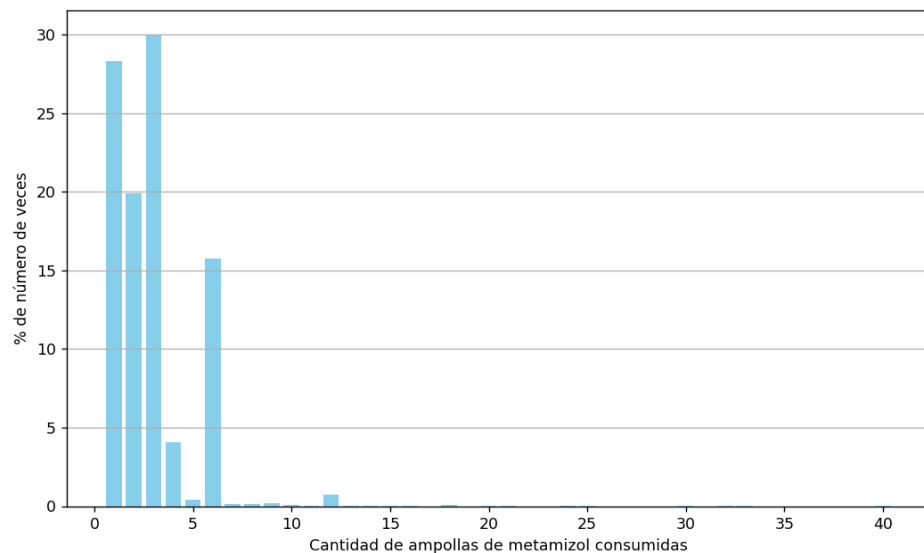
Estos hallazgos son fundamentales para:

Identificar rangos normales de consumo

Detectar desviaciones que podrían distorsionar los modelos predictivos.

Decidir sobre posibles estrategias de tratamiento de outliers (recorte o transformación).

Figura 5. Cantidad de Ampollas de metamizol entregados a pacientes vs el % de veces de entrega



De la imagen se puede observar una gráfica de barras que indica: en el eje (X) la cantidad de ampollas de metamizol sódico entregados por receta médica, mientras en el eje de las (Y) el porcentaje de veces que se entregó esa cantidad de ampollas, cada barra representa con qué frecuencia relativa (en %) se entregó una determinada cantidad de ampollas a un paciente, en una sola atención, se observa número muy bajo de entregas de más de 15 hasta 40 ampollas y la mayoría de pacientes reciben de 1 y 3 ampollas, estas entregas se pueden considerar casos atípicos o excepcionales; que podrían estar relacionados con tratamientos

prolongados, errores de digitación o necesidades clínicas específicas

4.9 Procesamiento de datos e ingeniería de características

Se realizó una depuración inicial de los datos para garantizar su calidad antes del modelamiento predictivo. Se identificaron y eliminaron valores atípicos en la variable CANTIDAD mediante diagramas de caja (boxplots), y se descartaron registros con cantidades iguales a cero, por considerarse inconsistentes.

Posteriormente, se aplicó la ingeniería de características, generando nuevas variables a partir de la información temporal de los registros, como día, mes, día de la semana y feriados nacionales, con el fin de capturar patrones estacionales y contextuales en la demanda de medicamentos.

Estas transformaciones permitieron que los modelos de machine learning integraran información contextual, mejorando la precisión y estabilidad de las predicciones frente a variaciones en los patrones de consumo.

4.9.1 Valores en cero:

con la exploración realizada, se identificaron registros en los que la cantidad de ampollas entregadas de medicamentos era igual a cero. Estos valores fueron considerados erróneos o no representativos de una entrega real, por lo que se procedió eliminarlos del conjunto de datos. Para ello, se utilizó una función denominada borrar ceros, dentro de la clase productoDatos, la cual filtra únicamente los registros con cantidades mayores a cero.

Figura 6. Código para eliminar registros con cantidad cero.

```
# Convertir cantidad a numérico y eliminar registros con cantidad = 0
df_medicamento["CANTIDAD"] = pd.to_numeric(df_medicamento["CANTIDAD"], errors="coerce")
df_medicamento = df_medicamento[df_medicamento["CANTIDAD"] != 0]
```

Esta depuración inicial fue fundamental para asegurar que el análisis y la predicción se basen únicamente en entregas efectivas de medicamentos, mejorando así la calidad del entrenamiento del modelo.

4.9.2 Valores atípicos (outliers):

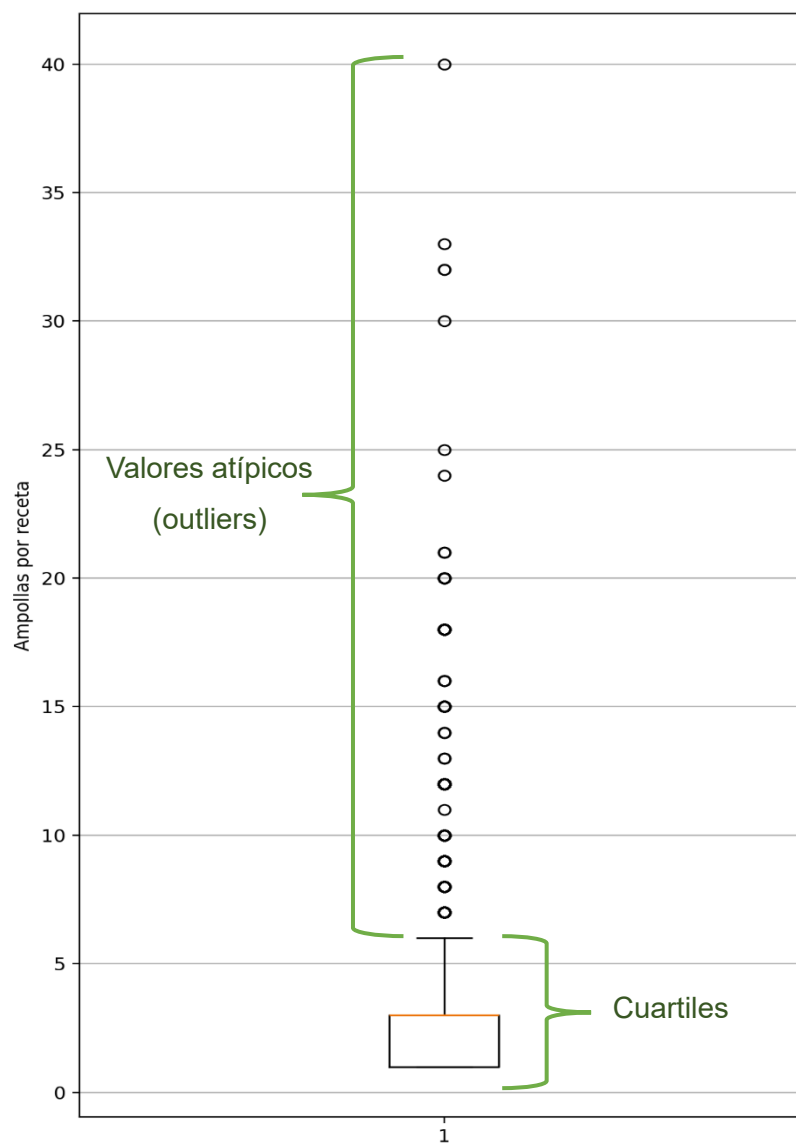
En esta etapa de procesamiento de datos también fue necesario identificar y tratar los valores atípicos, los cuales pueden influir negativamente en los resultados del análisis exploratorio y del modelo predictivo. Para ellos se aplicó un análisis gráfico mediante un diagrama de cajas y bigotes (boxplot) al consumo individual de metamizol, con el objetivo de visualizar la distribución de las unidades entregadas por receta médica. En la figura de distribución se observa que la mayoría de las entregas se concentran entre 1 y 6 ampollas por receta, lo cual corresponde al rango intercuartílico. Sin embargo, también se evidencian varios puntos dispersos por encima de este rango, representados por círculos que corresponden a valores atípicos (outliers). En este caso, se detectan registros con entregas de hasta 40 ampollas en una sola receta, lo cual no es común en el comportamiento general de consumo y puede deberse a situaciones de diagnósticos excepcionales, errores de registro o ventos fuera de lo habitual. Estos valores atípicos deben ser tratados adecuadamente en el procesamiento para evitar que influyan negativamente en el modelo predictivo. Por este motivo, en la siguiente etapa se procederá a eliminarlos utilizando la regla del rango intercuartílico (IQR),

Se uso el siguiente código para poder graficar y así visualizar de mejor manera la distribución de datos:

Figura 7. Código para graficar y poder visualizar la distribución de datos

```
plt.figure(figsize=(6,8))
plt.boxplot(df_limpio["CANTIDAD"])
plt.title("Cajas y bigotes (sin outliers)\nMetamizol sódico 1 g/2 ml")
plt.ylabel("Ampollas por receta")
plt.grid(axis='y')
plt.tight_layout()
plt.savefig(carpetas_salida / "figura_boxplot_sin_outliers.png", dpi=200)
plt.close()
```

Figura 8. Caja y bigotes (con outliers) Metamizol sódico 1 g/2ml



Fuente: Elaboración propia

En la imagen muestra el diagrama de cajas y bigotes (boxplot) del consumo de metamizol sódico 1g/2ml, se observa que la mayoría de las recetas demandan entre 1 y 4 ampollas, mientras que existen valores atípicos que alcanzan hasta 40 ampollas por receta. Esta asimetría positiva indica casos de consumo excepcionalmente alto que deben ser tratados durante el preprocesamiento para evitar sesgos en el modelo predictivo. Para ello se aplicó la regla del rango intercuartílico (IQR), eliminando las observaciones que superaban el límite superior determinado por $Q3 + 1.5 \times (Q3 - Q1)$, que mide cuan dispersos están los datos. Para ello se implementó el siguiente código:

Figura 9. Código para graficar caja y bigotes (sin outliers) Metamizol sódico 1 g/2ml

```
import matplotlib.pyplot as plt

# Calcular cuartiles y rango intercuartílico (IQR)
q1 = datos_medimento['cantidad'].quantile(0.25)
q3 = datos_medimento['cantidad'].quantile(0.75)
iqr = q3 - q1
limite_superior = q3 + 1.5 * iqr

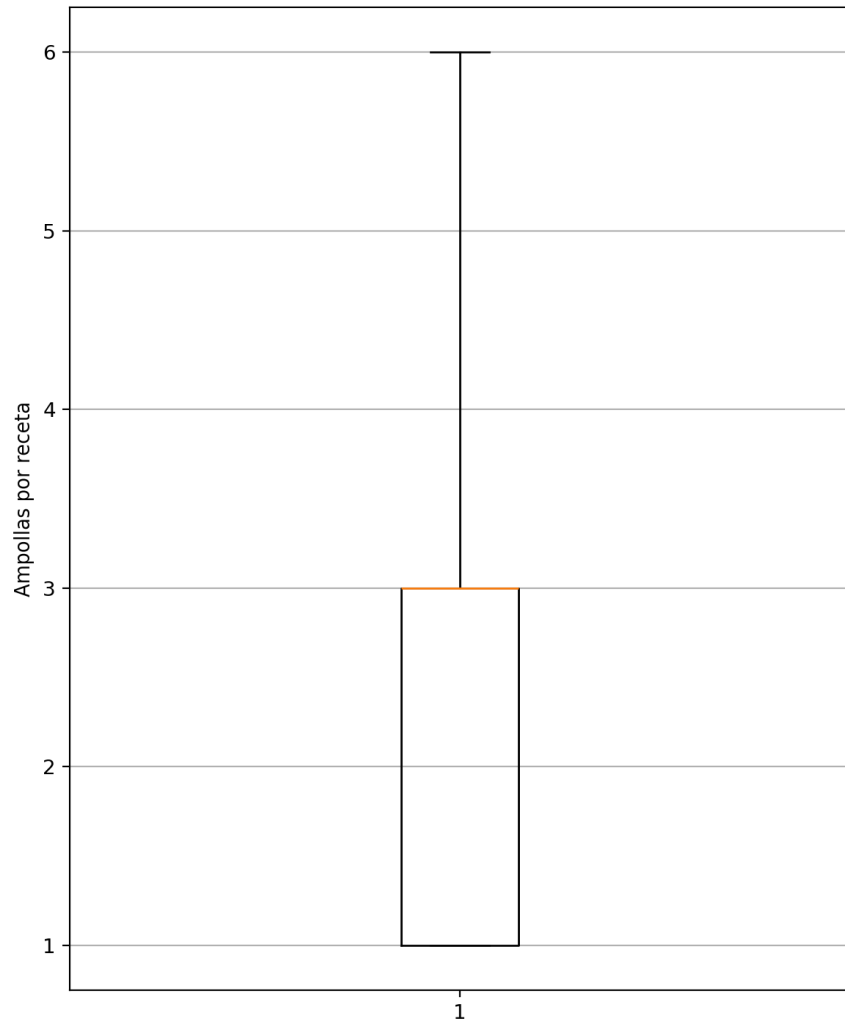
# Filtrar datos sin outliers (solo se eliminan por arriba)
datos_sin_outliers = datos_medimento[datos_medimento['cantidad'] <= limite_superior]

# Crear nuevo gráfico boxplot (sin outliers)
plt.figure(figsize=(6, 8))
plt.boxplot(datos_sin_outliers['cantidad'], vert=True)
plt.title("Cajas y bigotes (sin outliers)\nMetamizol sódico 1 g/2 ml")
plt.ylabel("Ampollas por receta")
plt.grid(True, axis='y')
plt.show()
```

Como resultado, el gráfico ahora refleja de manera más clara la distribución central de los datos. Se observa que la mayoría de las recetas implican entregas de entre 1 y 6 ampollas por receta. Esto permite identificar el patrón más común de consumo del medicamento, sin la interferencia de valores extremos que podrían sesgar los modelos de predicción. Este filtrado mejora significativamente la calidad del conjunto de datos y garantiza que los algoritmos de aprendizaje

automático más adelante se entrenen sobre una distribución realista y representativa del comportamiento habitual.

Figura 10. Caja y bigotes (sin outliers) Metamizol sódico 1 g/2ml



Fuente: Elaboración propia

Los datos atípicos fueron eliminados solo se observa los valores típicos del 1 al 6 siendo en su mayoría la entrega de 3 unidades promedio de ampollas en receta.

4.9.3 Ingeniería de características: una vez concluido el tratamiento de valores erróneos y atípicos, se procedió a la etapa de ingeniería de características, con el objetivo de enriquecer el

conjunto de datos y mejorar la capacidad predictiva del modelo de aprendizaje automático.

En esta etapa se generaron nuevas variables a partir de la fecha original de cada registro de receta médica, lo cual permitió capturar patrones temporales y estacionales que influyen en el consumo de medicamentos. Algunas de las características derivadas incluyen:

Día de la semana: indica si la receta fue emitida un lunes, martes, miércoles, etc., permitiendo observar diferencias en la demanda entre días laborales y fines de semana.

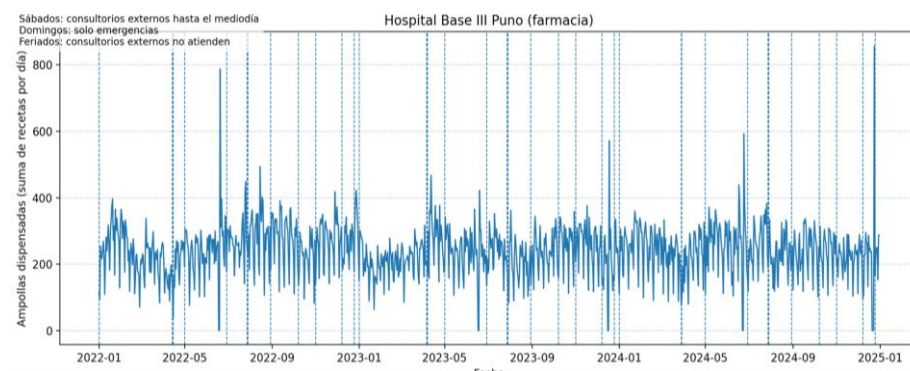
Mes: permite identificar patrones mensuales de consumo asociados a estacionalidad.

Año: útil para observar tendencias a lo largo del tiempo.

¿Es fin de semana?: variable binaria que identifica si la receta fue registrada un sábado o domingo

¿Es día festivo?: permite detectar impactos de feriados o fechas clave sobre el aumento o disminución de la demanda. Estas nuevas variables serán incorporadas al dataset final mediante transformaciones sobre la columna FECHA_DOC_MOV, asegurando que la información temporal pueda ser aprovechada por los algoritmos de predicción como arboles de decisión, XGboost, o modelos recurrentes. El siguiente código muestra un ejemplo básico de generación de estas características.

Figura 11. Entregas diarias de recetas (dispensación) de Metamizol sódico 1g/2ml.



Eje Y representa la cantidad total de ampollas entregadas por día y X la fecha de dispensación diaria, cubriendo los años 2022, 2023 y 2024, la curva muestra una alta variabilidad día a día, con picos y valles bien marcados, hay periodos con alta demanda (>800 ampollas por día) y otros con bajas o nulas entregas

Efecto de días no laborales y feriados:

Se ha detectado visualmente con líneas verticales los domingos (líneas más gruesas), sábados (más tenue) y feriados (línea intermedia), estas marcas coinciden consistentemente con caídas drásticas o nulas, en la entrega de recetas. Domingos se observan que observa que casi no hay entregas, dado que solo funcionan áreas emergencias, sábados entregas parcialmente reducidas, ya que los consultorios externos, solo atienden hasta el mediodía, feriados la farmacia si está operativa, pero al no haber consulta externa la demanda disminuye notablemente.

Estacionalidad: se observan patrones semanales repetitivos, con menor demanda los fines de semana y repunte a inicios de semana (lunes/martes).

Algunos picos irregulares podrían corresponder a eventos no programados (cambio de clima, incrementos por brote infeccioso, campañas de salud, etc.)

Volatilidad: aunque la mediana de atención parece rondar entre 200-300 recetas diarias, hay días que superan ampliamente las 500-800 recetas, lo cual debe considerarse en el modelo predictivo.

4.10 Construcción e implementación de los modelos Machine Learning:

En esta etapa se llevó a cabo la construcción e implementación de los modelos de aprendizaje automático orientados a predecir la demanda diaria de medicamentos esenciales en la farmacia hospitalaria. Esta fase constituye el núcleo del proceso de modelado dentro de la metodología CRISP-DM, permitiendo generar modelos reproducibles, comparables y evaluables. Cada modelo fue

desarrollado con base en los datos preprocesados y se entrenó individualmente para los 30 medicamentos seleccionados.

4.10.1 Selección metodológica y fundamentación

Se seleccionaron cuatro enfoques representativos que cubren distintas perspectivas del aprendizaje automático aplicado a series temporales:

- **Randon Forest (RF):** modelo basado en ensamblado de árboles de decisión que combina múltiples predictores para reducir la varianza y mejorar la generalización.
- **XGBoost (Extreme Gradient Boosting):** algoritmo de potenciación del gradiente optimizado que permite un entrenamiento eficiente y un excelente desempeño en datos estructurados.
- **LSTM (Long Short-Term Memory) y GRU (Gated Recurrent Unit):** red neuronal recurrente diseñada para capturar dependencias temporales de largo plazo y variaciones estacionales en series de demanda.
- **GRU (Gated Recurrent Unit):** variante simplificada de LSTM que reduce la complejidad computacional manteniendo la capacidad de aprendizaje de patrones temporales.

El empleo conjunto de estos métodos permitió abordar la complejidad del comportamiento temporal y estacional de la demanda, explorando la capacidad predictiva de diferentes arquitecturas bajo las mismas condiciones experimentales.

V. RESULTADOS

5.1 PRESENTACION DE LOS RESULTADOS

5.1.1 Identificación de los criterios relevantes que permitieron seleccionar los modelos de aprendizaje automático.

La identificación de los criterios relevantes para la selección de los modelos de aprendizaje automático se realizó mediante un análisis documental comparativo basado en la revisión de literatura científica reciente (2021–2025) en bases de datos IEEE Xplore, ScienceDirect, SpringerLink, ResearchGate y arXiv.

Este enfoque permitió establecer los criterios generales y específicos comúnmente empleados en investigaciones sobre predicción de demanda y series temporales en el ámbito farmacéutico y hospitalario, garantizando la replicabilidad metodológica sin requerir juicio de expertos.

Para este propósito, se analizaron y sistematizaron los factores teóricos, técnicos y computacionales descritos en los estudios de Bayani et al., (2024), Joshi et al., (2025), (Mistry, 2025), (Hewamalage et al., 2023) y Hu et al., (2024), entre otros, de los cuales se derivaron seis criterios relevantes (C1–C6) aplicables al modelamiento predictivo de demanda diaria de medicamentos con horizonte extendido de pronóstico.

Tabla 5. Cuadro de Criterios generales y específicos

| Código | Criterio | Definición operativa | Aplicación práctica en predicción de demanda farmacéutica | Referencia bibliográfica |
|---------------|---|--|--|--|
| C1 | Captura de dependencias temporales (memoria secuencial) | Capacidad del modelo para reconocer patrones secuenciales, estacionalidad y autocorrelaciones en series de tiempo hospitalarias. | Los modelos LSTM y GRU sobresalen por su arquitectura recurrente que retiene dependencias temporales; en Random Forest y XGBoost, se simulan con variables de rezago (lags). | (Bayani et al., 2024) (Hewamalage et al., 2023) |
| C2 | Precisión predictiva (minimización del error) | Evalúa la diferencia promedio entre valores reales y predichos mediante métricas MAE, RMSE y MAPE. | XGBoost y GRU presentan alta precisión al optimizar errores residuales y ajustar gradualmente el modelo durante el entrenamiento. | (Joshi et al., 2025) (Hu et al., 2024) |
| C3 | Robustez frente a ruido y valores atípicos | Determina la estabilidad del modelo ante errores de registro, valores extremos o picos de consumo imprevistos. | Random Forest y XGBoost se destacan por su estabilidad ante ruido y regularización interna; LSTM atenúa perturbaciones mediante aprendizaje adaptativo. | (Braiek & Khomh, 2024) |
| C4 | Interpretabilidad y explicabilidad | Evalúa la facilidad de comprender cómo el modelo genera predicciones y qué variables influyen más en ellas. | En contextos hospitalarios, la interpretabilidad es crucial. Los modelos basados en árboles (RF, XGBoost) permiten obtener la importancia de variables; las redes profundas aplican métodos post hoc (SHAP, LIME). | (Murdoch et al., 2019) |
| C5 | Eficiencia computacional y escalabilidad | Relación entre desempeño predictivo, tiempo de entrenamiento y recursos computacionales. | GRU y XGBoost destacan por su balance entre rapidez de entrenamiento y estabilidad del modelo; LSTM requiere mayor costo computacional. | (Mistry, 2025) |
| C6 | Capacidad de generalización (evitar sobreajuste) | Capacidad del modelo de mantener buen rendimiento en datos no vistos. | Los modelos de ensamble (RF, XGBoost) generalizan adecuadamente; las RNN (LSTM, GRU) requieren regularización (Dropout, EarlyStopping). | (Joshi et al., 2025) |

Tabla 6. Cuadro agrupación de criterios generales y específicos

| Tipo | Modelo | Criterios relevantes |
|------|---------------|----------------------|
| ML | Random Forest | C3, C4, C5, C6 |
| ML | XGBoost | C2, C3, C4, C5 |
| ML | SVR | C3, C4, C6 |
| DL | LSTM | C1, C2, C3, C4 |
| DL | GRU | C1, C2, C4, C5 |
| DL | CNN-1D | C3, C5 |

Fuente: Elaboración propia con base en Bayani et al., (2024), Joshi et al., (2025) y (Mistry, (2025)

Los criterios definidos (C1–C6) fueron agrupados en tres dimensiones:

- **Técnica:** C1 (dependencias temporales), C2 (precisión), C5 (eficiencia).
- **Estadística:** C3 (robustez) y C6 (generalización).
- **Operativa:** C4 (interpretabilidad).

De acuerdo con el análisis documental, los criterios **C1**, **C2** y **C6** resultan prioritarios para el contexto hospitalario, pues determinan la capacidad de los modelos de adaptarse a patrones temporales complejos y mantener desempeño estable. Los criterios **C3** y **C5** son determinantes para el manejo de datos reales con ruido y limitaciones computacionales, mientras que **C4** garantiza la aceptación del modelo por parte del personal de farmacia y gestión.

En conclusión, la revisión teórica y técnica permitió establecer que los modelos LSTM y GRU satisfacen la mayor parte de los criterios definidos, presentando un equilibrio entre precisión, robustez y generalización temporal, por lo cual fueron priorizados en la fase experimental.

Fundamentación metodológica

El proceso de definición de los criterios siguió tres etapas secuenciales:

- a. Revisión bibliográfica estructurada:** selección de artículos revisados por pares (2021–2025) relacionados con *demand forecasting* y *machine learning* en salud pública.
- b. Análisis comparativo técnico:** identificación de los atributos más reportados para evaluar modelos predictivos en entornos hospitalarios.
- c. Derivación y aplicación contextual:** adaptación de los criterios a la naturaleza de los datos utilizados (series diarias multianuales del Hospital base III Puno).

La combinación de criterios seleccionados permite contrastarlos: por ejemplo, un modelo con muy alta precisión, pero baja interpretabilidad podría no ser aceptable en entorno hospitalario. En tu situación con dos variables principales (tiempo por día y demanda diaria) y con la intención de extender el horizonte de pronóstico, es crucial que el modelo comprenda dependencias temporales (C1) y también generalice bien (C6).

Los criterios de robustez (C3) y eficiencia (C5) son especialmente importantes con datos reales hospitalarios, que pueden contener errores y grandes volúmenes.

Interpretabilidad (C4) es útil para que los gestores de farmacia puedan entender las predicciones y confiar en el modelo.

Finalmente, el criterio de precisión (C2) es el núcleo del objetivo de predicción.

5.1.2 Implementación modelos de aprendizaje automático para predecir la demanda de medicamentos esenciales sobre los datos preprocesados

1. Modelo Random Forest

Se desarrollo e implemento el modelo Random Forest para predecir la demanda de medicamentos esenciales en la farmacia

hospitalaria, usando series temporales por código de medicamento (COD_MEDIC). El proceso integra desde el procesamiento de datos hasta el entrenamiento, validación, evaluación y generación de pronósticos, preservando trazabilidad y reproducibilidad:

a. Datos y alcance

- **Fuente:** la data estaba en archivo Excel con ruta: D:\PARARANDONFOREST\consumo_medicamentos.xlsx (dispensaciones diarias de medicamentos)
- **Variables clave:** FECHA_DOC_MOV, COD_MEDIC, DESC_MEDICAMENTO, CANTIDAD.
- **Horizonte temporal:** 2022 – 2024
- **Unidades de análisis:** 30 medicamentos (1 archivo por código más un consolidado)

b. Curación y estandarización

Reglas de operación de farmacias (dominio):

- **Domingos y feriados:** sin consulta externa
- **Sábado:** medio turno (atención normal por la mañana, tarde reduce atención sin consulta externa).
- Días con cantidad = 0 pueden reflejar falta de stock o consumo cero.

Acciones aplicadas:

- Conversión de y serie diaria continua por medicamento
- Marcado de feriados nacionales (2022 – 2024) y tipo de día.
- Completar ausencias con cantidad =0 (imputación estructural), sin alterar la señal.
- Normalización de columnas y agregación diaria cuando hay múltiples movimientos

Figura 12. Fragmento de código procesamiento de datos para modelo Randon Forest.

```
# Carga y normalización
df = pd.read_excel(r"D:\PARARANDONFOREST\consumo_medicamentos.xlsx")
df.columns = df.columns.str.strip().str.upper()
df['FECHA_DOC_MOV'] = pd.to_datetime(df['FECHA_DOC_MOV']).dt.date

# Serie diaria por medicamento
def expandir_serie(g):
    r = pd.date_range(g.FECHA_DOC_MOV.min(), g.FECHA_DOC_MOV.max(), freq='D').date
    base = pd.DataFrame({'FECHA_DOC_MOV': r})
    g2 = (g.groupby(['FECHA_DOC_MOV'], as_index=False)['CANTIDAD'].sum())
    out = base.merge(g2, on='FECHA_DOC_MOV', how='left').fillna({'CANTIDAD':0})
    return out

df_proc = []
for cod, g in df.groupby('COD_MEDIC'):
    z = expandir_serie(g)
    z['COD_MEDIC'] = cod
    z['DESC_MEDICAMENTO'] = g['DESC_MEDICAMENTO'].iloc[0]
    df_proc.append(z)
consolidado = pd.concat(df_proc, ignore_index=True)
```

Salida generada

- **Consolidado:** \PARARANDONFOREST\consolidado_medicamentos.xlsx.

c. Ingeniería de característica

Para capturar patrones autorregresivos, estacionales y de calendario.

- **Lags:** *LAG_1, LAG_2, LAG_3, LAG_7, LAG_14, LAG_21.*
- **Ventanas móviles (Rolling):** medias y desviaciones con ventanas 7,14,30 días (con shift (1) para evitar fuga de información)
- **Calendario:** *AÑO, MES, DÍA_DE_SEMANA, SEMANA_ISO, TRIMESTRE, DÍA_DEL_MES, ES_SÁBADO, ES_DOMINGO, ES_FERIADO, ES_FIN_DE_SEMANA.*
- **Robustez:** winsorización por medicamento (p1 – p99) para atenuar outliers y transformación del objetivo *log1p* para estabilizar varianza.

Figura 13. Fragmento de código ingeniería de características en Randon Forest

```
def crear_features(g):
    g = g.sort_values('FECHA_DOC_MOV').copy()
    # Lags
    for lag in [1,2,3,7,14,21]:
        g[f"LAG_{lag}"] = g['CANTIDAD'].shift(lag)
    # Rolling (shift(1) para no usar el presente)
    for w in [7,14,30]:
        g[f"ROLL_MEAN_{w}"] = g['CANTIDAD'].shift(1).rolling(w).mean()
        g[f"ROLL_STD_{w}"] = g['CANTIDAD'].shift(1).rolling(w).std()
    # Calendario
    ts = pd.to_datetime(g['FECHA_DOC_MOV'])
    g['MES'] = ts.dt.month; g['DIA_SEMANA'] = ts.dt.weekday
    g['SEMANA'] = ts.dt.isocalendar().week.astype(int)
    g['TRIMESTRE'] = ts.dt.quarter; g['DIA_DEL_MES'] = ts.dt.day
    return g.dropna()
```

d. Partición temporal y evaluación

- **División temporal por medicamento:** 80% entrenamiento / 20% prueba (último tramo)
- **Métricas en escala original:**
 - MAE** (error absoluto medio)
 - RMSE** (raíz del error cuadrático medio; compatibles con versiones antigua de sklearn)
 - SMAPE (%)** (Error porcentual simétrico; robusto frente a ceros).

Figura 14. Fragmento de código función de métricas en Randon Forest

```
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score
import numpy as np

def rmse_compatible(y_true, y_pred):
    try: # sklearn modernos
        return mean_squared_error(y_true, y_pred, squared=False)
    except TypeError: # sklearn antiguos
        return np.sqrt(mean_squared_error(y_true, y_pred))

def smape(y, yhat, eps=1e-8):
    y, yhat = np.asarray(y), np.asarray(yhat)
    return 100*np.mean(2*np.abs(yhat - y) / (np.abs(y)+np.abs(yhat)+eps))
```

e. Modelo y búsqueda de hiperparametros:

Se implemento un pipeline avanzado con:

RandonForest (árboles en paralelo, *max_features='sqrt'*, *Bootstrap*).

Optimización por medicamento con *RandomizedSearchCV* y *TimeSeriesSplit* (CV temporal).

Criterio de búsqueda: *neg_mean_absolute_error* (se puede alternar a *neg_root_mean_squared_error*).

Figura 15. Fragmento de código entrenamiento del modelo Randon Forest

```
from sklearn.ensemble import RandomForestRegressor
from sklearn.model_selection import RandomizedSearchCV, TimeSeriesSplit
from scipy.stats import randint

param_dist = {
    "n_estimators": randint(120, 500),
    "max_depth": [None, 8, 10, 12, 14],
    "min_samples_split": randint(2, 10),
    "min_samples_leaf": randint(1, 6),
    "max_features": ["sqrt", "log2", None],
    "bootstrap": [True, False],
}
cv = TimeSeriesSplit(n_splits=3)
rf = RandomForestRegressor(random_state=42, n_jobs=-1)
search = RandomizedSearchCV(rf, param_distributions=param_dist,
                            n_iter=30, cv=cv, scoring="neg_mean_absolute_error",
                            random_state=42, n_jobs=-1, verbose=0)
search.fit(X_train, y_train)
best_model = search.best_estimator_
```

f. Resultados y diagnostico

- Se entrenaron 30 modelos (uno por *COD_MEDIC*), guardados en:
D:\PARARANDONFOREST\Modelos_RF_Adv\RF_ADV_<COD>.joblib
- Se generó un resumen de métricas por medicamento:
D:\PARARANDONFOREST\metricas_randomforest_avanzado.xlsx
- Hallazgos generales (ejemplo típico de la corrida)
 R^2 promedio ≈ 0.42 (mediana ≈ 0.47).
SMAPE promedio $\approx 56\%$ (mediana $\approx 54\%$).
Hay medicamentos con buen ajuste ($R^2 \geq 0.6$ y $SMAPE \leq 35\%$) y otros con comportamiento errático ($SMAPE \geq 70\%$), usualmente vinculados a quiebres de stock, picos administrativos o baja estacionalidad.

Randon Forest captura patrones autorregresivos y semanales, pero la presencia de ceros estructurales, outliers y cambio de régimen puede impactar en el error

porcentual. Las mejoras se orientan a más memoria temporal, mejor codificación de calendario (p. ej., víspera/pos-feriado) y marcadores de stock (proxy).

g. Pronósticos operativos (horizonte H días)

Se implementó un módulo de pronóstico recursivo (7-30 días) que:

Reconstruye **lags/rolling** con cada predicción nueva.

Alinea exactamente las features con *feature_names_in_* del modelo (evita errores de orden).

Exporta pronósticos con intervalos P10 – P90 (dispersión de árboles).

Figura 16. Fragmento de código predicción de un día, alineando columnas del modelo Random Forest.

```
X_row = construir_fila_features(hist_wins, cal_row) # crea lags/rolling y calendario
X_row = X_row.reindex(columns=model.feature_names_in_, fill_value=0.0)
y_hat = model.predict(X_row)[0] # pred en escala del target
y_hat = np.expml(y_hat) # inversa si se usó loglp
y_hat = max(0.0, y_hat) # no negatividad
```

Salidas:

Carpeta

D:\PARARANDONFOREST\Pronosticos_RF_Adv\forecas
t_<COD>.xlsx (por medicamento).

2. Modelo XGBoost

Para diseño, entrenamiento, validación y operacionalización del modelo de gradiente reforzado (XGBoost) para la predicción diaria de la demanda de medicamentos esencial en la farmacia hospitalaria, garantizando reproducibilidad, robustez ante *outliers*, incorporación de variables exógenas operativas (calendario, feriados, tipo de atención) y evaluación temporal realista (*hold-out cronológico y walk-forward*). El desarrollo se implementa a nivel de 30 medicamentos y genera pronósticos futuros a 14 días por código de producto.

a. Ingesta y normalización

Fuente:

D:\PARAXGBOOST\consumo_medicamentos.xlsx

Campos mínimos: *FECHA_DOC_MOV, COD_MEDIC, DESC_MEDICAMENTO, CANTIDAD.*

Normalización: tipificación de nombres. *Casting* de fecha y ordenamiento cronológico.

Figura 17. Ejemplo de código de la tabla de entrada (campos esenciales y consistencia temporal) del modelo XGBoost

```
df = pd.read_excel(RUTA_ENTRADA)
df.columns = df.columns.str.strip().str.upper()
df['FECHA_DOC_MOV'] = pd.to_datetime(df['FECHA_DOC_MOV'])
df = df.sort_values(['COD_MEDIC', 'FECHA_DOC_MOV'])
```

b. Estructuración temporal y criterios operativos

Se construye una serie diaria continua por medicamento (relleno de días faltante con 0) y se incorporan reglas de atención.

Feridos y domingos: sin atención consulta externa (indicador)

Sábado: medio turno (indicador)

Resto: turno completo

Además, se codifican variables de calendario: *año, mes, día, día_semana.*

Figura 18. Reglas codificadas como features binarias modelo XGBoost

```
# Serie continua y variables de calendario
rango = pd.date_range(df_med.fecha.min(), df_med.fecha.max(), freq='D')
base = (pd.DataFrame({'FECHA_DOC_MOV': rango})
        .merge(df_med[['FECHA_DOC_MOV', 'CANTIDAD']], how='left')
        .fillna({'CANTIDAD':0}))

base['anio'] = base['FECHA_DOC_MOV'].dt.year
base['mes'] = base['FECHA_DOC_MOV'].dt.month
base['dia'] = base['FECHA_DOC_MOV'].dt.day
base['dia_semana'] = base['FECHA_DOC_MOV'].dt.dayofweek # 0=lun, 6=dom
```

```

base['es_feriado'] = base['FECHA_DOC_MOW'].isin(FERIADOS_PE).astype(int)
base['es_sabado'] = (base['dia_semana']==5).astype(int)
base['es_domingo'] = (base['dia_semana']==6).astype(int)
base['tipo_completo'] = ((1-base['es_feriado']) & (1-base['es_domingo']) & (1-base['es_sabado'])).astype(int)
base['tipo_medio_turno'] = base['es_sabado']
base['tipo_emergencia'] = ((base['es_feriado']==1) | (base['es_domingo']==1)).astype(int)

```

Notas operacionales: se señalaron días con consumo cero no atribuibles al calendario como indicador *stockout_proxy* (posible quiebre de stock) y proximidades a feriado (*es_feriado_ayer*, *es_feriado_mañana*, “puentes”).

c. Tratamiento de calores atípicos y estabilización del objetivo

Para robustecer el aprendizaje ante picos de demanda:

- **Winsorización** al 99% recorte de colas extremas.
- **Suavizado** por mediana móvil (ventana3)
- Se define el objetivo *cantidad_ws* (winsorizada + suavizada) manteniendo *CANTIDAD* para reporting.

Figura 19. Reglas codificadas Winsorización del modelo XGBoost

```

def winsorize(s, p1=0.01, p99=0.99):
    lo, hi = s.quantile(p1), s.quantile(p99)
    return s.clip(lo, hi)

base['cantidad_w'] = winsorize(base['CANTIDAD'])
base['cantidad_ws'] = base['cantidad_w'].rolling(3, center=True, min_periods=1).median()
TARGET = 'cantidad_ws'

```

d. Ingeniería de features temporales

Se generan rezagos (lag) y promedios móviles (sin fuga de información), que permiten al modelo capturar inercia, periodicidad corta y estacionalidades:

Figura 20. Matriz final de entrada (por fila/día)

```

# Lags: 1..14 y 30
for L in list(range(1,15)) + [30]:
    base[f'lag_{L}'] = base['CANTIDAD'].shift(L)

# Rolling means (sin look-ahead)
for W in [3,7,14,30]:
    base[f'rolling_{W}'] = base['CANTIDAD'].shift(1).rolling(W, min_periods=W).mean()

base = base.dropna().reset_index(drop=True)

```

$X = \{\text{calendario, tipo_día, proximidad a feriado, stockout_proxy, lags, rollings}\}$ y objetivo $y = \text{cantidad_ws}$

e. División temporal y escalamiento robusto

- Holt-out cronológico 80/20 final para test.
- Escalamiento robusto (*RobustScaler*) solo en variables continuas, las binarias pasan “tal cual” vía *ColumnTransformer*.

Figura 21. Código escalamiento robusto modelo XGBoost

```
binarias = [c for c in X_cols if set(base[c].dropna().unique()).issubset({0,1})]
continuas = [c for c in X_cols if c not in binarias]

pre = ColumnTransformer([
    ('num', RobustScaler(), continuas),
    ('bin', 'passthrough', binarias)
], remainder='drop')
```

f. Modelo optimizado de hiperparametros

Se emplea *XGBRegressor* (objetivo `reg:squarederror`) y *RandomizedSearchCV* con validación temporal (*TimeSeriesSplit*), optimizando MAPE (scorer negativo para minimizar). Parámetros buscados: *max_depth*, *learning_rate*, *subsample*, *colsample_bytree*, *min_child_weight*, *gamma*, *reg_alpha*, *reg_lambda*.

Figura 22. Esquema de pipeline preprocesamiento XGBoost, optimización métricas

```
xgb = XGBRegressor(n_estimators=600, objective='reg:squarederror', n_jobs=-1, random_state=42)
pipe = Pipeline([('prep', pre), ('model', xgb)])

search = RandomizedSearchCV(
    pipe, param_distributions=param_grid, n_iter=40,
    scoring=make_scorer(mape, greater_is_better=False),
    cv=TimeSeriesSplit(n_splits=5), n_jobs=-1, random_state=42, verbose=1
)
search.fit(X_train, y_train)
best_model = search.best_estimator_
```

g. Evaluación hold – out + walk-forward

- **Hold-out** (20% final): MAE, RMSE, SMAPE
- **Walk-forward (Rolling retrain)**: reentrena cada $k=30$ días y predice secuencialmente 1-paso-adelante. Esto refleja la operación real de la farmacia.

Figura 23. Métricas definiciones compactas XGBoost

```
from sklearn.base import clone
K=30; model_wf=None; preds=[]
i=n_train
while i < len(df_model):
    if (i-n_train)%K==0 or model_wf is None:
        model_wf = clone(best_model).fit(X_full.iloc[:i], y_full.iloc[:i])
    y_hat = model_wf.predict(X_full.iloc[i:i+1])[0]
    preds.append({'fecha': fechas.iloc[i], 'y_true': y_full.iloc[i], 'y_pred': y_hat})
    i+=1
wf = pd.DataFrame(preds)
```

h. Entrenamiento masivo (30 medicamentos)

El pipeline anterior por cada COD_MEDIC (lógica group-by), generando:

- **Modelos:**
D:\PARAXGBOOST\Modelos_XGB_Opt_30\model_{COD}.pkl
- **Predicciones walk-forward:**
Predicciones_WF_30\pred_{COD}.xlsx
- **Resumen global:** Metrics_XGB_Opt_30.xlsx
(por medicamento: MAE, RMSE, SMAPE).

Figura 24. Consola de entrenamiento Bach (métricas por código y rutas) para el modelo XGBoost

```
for cod, g in df.groupby('cod_medic'):
    # ... construir features, optimizar, evaluar ...
    joblib.dump(best_model, f"{DIR_MODELOS}/model_{cod}.pkl")
    wf_df.to_excel(f"{DIR_PRED_WF}/pred_{cod}.xlsx", index=False)
```

i. Resultados esperados y criterios de aceptación

- **Exactitud:** valores de **SMAPE** bajos (ideal < 30 %) y RMSE proporcional a la magnitud del fármaco.
- **Explicabilidad operativa:** importancia de variables debe reflejar estacionalidad semanal (*lags* cortos), calendario (sábados/feriados) y dinámicas de consumo.
- **Robustez:** tratamiento de *outliers*, *stockouts* y evaluación walk-forward que minimiza sobreajuste.
- **Reproducibilidad:** rutas, seed (*random_state=42*), artefactos versionados (. *pkl*, .*xlsx*, .*png*).

3. Modelo LSTM

El desarrollo, entrenamiento y validación del modelo LSTM (Long Short-Term Memory) capaz de predecir la demanda diaria de medicamentos esenciales a corto plazo, superando líneas base estacionales y proporcionando pronósticos con bandas de incertidumbre útiles para la planificación de inventario.

1. Ingeniería de features

La etapa de ingeniería de características fue fundamental en el modelamiento con redes neuronales LSTM, ya que define las variables predictoras que el modelo utilizó para aprender patrones temporales.

a. Carga y normalización de la base de datos

Se cargó la hoja que contiene la data, con contiene el consumo de cada medicamento. Se normaliza la columna de fecha y se ordenó cronológicamente para garantizar la coherencia temporal de la serie.

Figura 25. Consola de entrenamiento para la carga de la base de datos para el modelamiento LSTM

```
# Cargar 'daily' y enriquecer calendario
df = xl.parse("daily")
df["fecha"] = pd.to_datetime(df["fecha"]).dt.normalize()
df = df.sort_values("fecha").set_index("fecha")
```

Se asegura que los valores de la columna *fecha* sean de tipo *datetime*, eliminando horas y minutos.

Se establece la fecha como índice principal para permitir el reindexado y la generación de ventanas temporales posteriores.

El orden temporal es crucial, ya que LSTM depende de la secuencia correcta de observaciones.

b. Generación de una serie continua

Muchas veces existen huecos de días no registrados en la base de datos (por cierre o falta de stock). Para

resolverlo se reconstruye la secuencia completa de fechas entre el primer y último registro.

Figura 26. Generación continua para el modelamiento LSTM

```
# Cerrar huecos de calendario
full_idx = pd.date_range(df.index.min(), df.index.max(), freq="D")
df = df.reindex(full_idx)
```

Se crea un rango de fechas completo con frecuencia diaria (*freq="d"*)

Se reindexa el DataFrame para incluir todos los días del periodo analizado.

Si existen días sin datos, estos se rellenaron posteriormente con imputaciones o ceros controlados.

c. Creación de variables de calendario y operativas

A partir de la fecha, se derivaron variables que representan la estacionalidad y los patrones semanales de atención de las farmacias.

Figura 27. Creación de la variable calendario y operativas para el modelamiento LSTM

```
# Flags y reglas operativas
df["dow"] = df.index.weekday # día de la semana (0=Lunes)
df["is_saturday"] = (df["dow"]==5).astype(int) # indicador de sábado
df["is_sunday"] = (df["dow"]==6).astype(int) # indicador de domingo
df["is_weekend"] = (df["dow"]>=5).astype(int) # fin de semana
df["is_holiday"] = df.index.isin(feriados_pe).astype(int) # feriados nacionales
df["emergency_only_day"] = ((df["is_sunday"]==1) | (df["is_holiday"]==1)).astype(int)
df["open_hours_factor"] = np.where(df["is_saturday"]==1, 0.5,
np.where((df["is_sunday"]==1) | (df["is_holiday"]==1), 0.7, 1.0))
df["month"] = df.index.month
df["weekofyear"] = df.index.isocalendar().week.astype(int)
```

Explicación:

- **dow:** representa el número del día de la semana (de 0 a 6).
- **is_saturday, is_sunday, is_weekend:** indicadores binarios que permiten capturar los patrones de atención reducida o cierre.

- *is_holiday*: se genera a partir de un calendario de feriados oficiales del Perú (años 2024 y 2025).
- *emergency_only_day*: indica días donde solo atiende la farmacia de emergencia.
- *open_hours_factor*: pondera la cantidad de horas operativas de la farmacia según el tipo de día (1.0 días normales, 0.5 sábados, 0.7 domingos o feriados).
- *month* y *weekofyear*: reflejan variaciones estacionales o mensuales en la demanda.

Estas variables agregaron contexto operativo, que el modelo LSTM utiliza para identificar patrones semanales, mensuales y estacionales de medicamento.

d. Tratamiento de ceros e imputaciones

En la farmacia hospitalaria, la presencia de valores cero puede deberse a diferentes razones. Por ello, se clasifican según su causa (*closed_like* o *possible_stockout*) y se tratan de forma diferenciada.

Figura 28. Tratamiento de ceros e imputaciones para el modelamiento LSTM

```
# Tratamiento de ceros por cierre
is_closed = (df["cantidad"].eq(0) & (df["zero_reason"]=="closed_like"))
df.loc[is_closed, "cantidad"] = np.nan
df["cantidad"] = df["cantidad"].ffill().fillna(0.0)
```

- Los días en que la farmacia estuvo cerrada (*closed_like*) se imputan con el valor anterior (*forward fill*).
- En cambio, los días con *possible_stockout* (desabastecimiento) se conservan como cero, ya que representan un fenómeno real de falta de inventario.

- Este procedimiento redujo el ruido en la serie y mejora la estabilidad del entrenamiento.

e. Construcción de tensores y escalamiento

Convertir la serie diaria enriquecida en ejemplos supervisados (x,y) con una ventana deslizante de longitud $LOOKBACK$ (pasado) que predice un $HORIZON$ (futuro). Escalar solo con datos de entrenamiento para evitar fuga de información.

Esquema del ventaneo

- $LOOKBACK = 60$ días (secuencia de entrada al LSTM).
- $HORIZON = 3$ días (salida vectorial: $t+1, t+2, t+3$).
- $STEP = 1$ (la ventana se desplaza día a día).
- $VAL_DAYS = 90$ (bloque de validación al final de la serie).

Ilustración: para cada índice i , la entrada es $X[i] =$ features $[i: i+60]$ y el objetivo es $y[i] = \log_{1p}$ (cantidad $[i+60: i+60+3]$).

Figura 29. Codificación para el ventaneo del modelamiento LSTM

```

import numpy as np
import pandas as pd
from sklearn.preprocessing import RobustScaler

LOOKBACK, HORIZON, STEP = 60, 3, 1
FEATURES = ["is_holiday", "is_saturday", "is_sunday", "is_weekend",
            "emergency_only_day", "open_hours_factor", "dow", "month", "weekofyear",
            "is_closed_like_zero", "is_stockout_like_zero"]

# Objetivo en log1p para estabilizar varianza
y_log = np.log1p(df["cantidad"].astype(float).values)
Xmat = df[FEATURES].astype(float).values

def make_supervised(X, y, L, H, step=1):
    Xs, Ys = [], []
    last = len(X) - (L + H)
    if last < 0:
        return np.empty((0,L,X.shape[1])), np.empty((0,H))
    for s in range(0, last + 1, step):
        Xs.append(X[s:s+L, :])
        Ys.append(y[s+L:s+L+H])
    return np.asarray(Xs, np.float32), np.asarray(Ys, np.float32)

# Split temporal por cola
split_date = df.index.max() - pd.Timedelta(days=90-1)
mask_tr = df.index <= split_date - pd.Timedelta(days=0) # todo menos el bloque final
mask_va = df.index >= split_date

Xtr, ytr = make_supervised(Xmat[mask_tr], y_log[mask_tr], LOOKBACK, HORIZON, STEP)
Xva, yva = make_supervised(Xmat[mask_va], y_log[mask_va], LOOKBACK, HORIZON, STEP)

```

f. Escalamiento sin fuga

Escalamos cada feature con RobustScaler (mediana y IQR) ajustado solo con Xtr. El objetivo y no se escala (ya está en log1p).

Figura 30. Codificación de escalamiento del modelamiento LSTM

```

scalers = []
Xtr_s, Xva_s = np.empty_like(Xtr), np.empty_like(Xva)
for j in range(Xtr.shape[2]):
    sc = RobustScaler().fit(Xtr[:, :, j].reshape(-1, 1))
    Xtr_s[:, :, j] = sc.transform(Xtr[:, :, j].reshape(-1, 1)).reshape(Xtr.shape[0], LOOKBACK)
    Xva_s[:, :, j] = sc.transform(Xva[:, :, j].reshape(-1, 1)).reshape(Xva.shape[0], LOOKBACK)
    scalers.append(sc)

```

RobustScaler: reduce el impacto de outliers y cambios bruscos (típicos en consumo por quiebres de stock o campañas), lo que estabiliza el aprendizaje secuencial.

g. Línea base y modelamiento

Se implementan dos referencias obligatorias.

- **Naive-1:** pronostico = ultimo valor observado (en log).
- **SNaive(7):** pronostico = valor de hace 7 dias, repetido por el horizonte.

Ambas se evalúan en **log** (coherente con el entrenamiento) y en escala original para interpretación operativa.

Figura 31. Codificación línea base para el modelamiento LSTM

```

from sklearn.metrics import mean_absolute_error, mean_squared_error
import numpy as np

def snaive_targets(series, L, H):
    y_true, n1, n7 = [], [], []
    last = len(series) - (L + H)
    for s in range(last + 1):
        win = series[s:s+L]; fut = series[s+L:s+L+H]
        y_true.append(fut)
        n1.append(np.repeat(win[-1], H))
        last7 = win[-7:] if len(win) >= 7 else np.repeat(win[-1], min(7,H))
        n7.append(np.tile(last7, int(np.ceil(H/7))[:H]))
    return np.array(y_true), np.array(n1), np.array(n7)

# En validación (log)
y_true_log, n1_log, n7_log = snaive_targets(y_log[mask_val], LOOKBACK, HORIZON)

# Métricas utilitarias
def smape(y, yhat):
    y, yhat = y.reshape(-1), yhat.reshape(-1)
    den = (np.abs(y) + np.abs(yhat)) / 2
    return 100*np.mean(np.where(den==0, 0, np.abs(y - yhat)/den))

def mase(y, yhat, y_insample, season=7):
    den = np.mean(np.abs(y_insample[season:] - y_insample[:-season])) or 1.0
    return mean_absolute_error(y.reshape(-1), yhat.reshape(-1))/den

# Log
mae_n1_log = mean_absolute_error(y_true_log, n1_log)
rmse_n1_log = mean_squared_error(y_true_log, n1_log, squared=False)
smp_n1_log = smape(y_true_log, n1_log)
mase7_n1 = mase(np.expml(y_true_log), np.expml(n1_log), np.expml(ytr.reshape(-1)), season=7)

# Original
y_true_ori = np.expml(y_true_log)
n1_ori, n7_ori = np.expml(n1_log), np.expml(n7_log)
mae_n1 = mean_absolute_error(y_true_ori, n1_ori)
rmse_n1 = mean_squared_error(y_true_ori, n1_ori, squared=False)
smp_n1 = smape(y_true_ori, n1_ori)

```

Con la aplicación de Naive-1 y, sobre todo, **SNaive(7)** se buscó mejorar en sMAPE (ideal < 1) aporta robustez ante escalas diferentes entre medicamentos.

h. Modelamiento LSTM

Arquitectura recomendada (probada en el grid)

- LSTM (64, return_sequences=True)
- Dropout(0.15)
- LSTM (32)
- Dropout(0.15)
- Dense (HORIZON)
- Optimizador: Adam (lr=1e-3 → reduce_on_plateau)

- Pérdida: MSE en log (equivale a minimizar RMSE en log).

Figura 32. Codificación para el modelamiento LSTM

```
import tensorflow as tf
from tensorflow.keras import layers, models, callbacks, optimizers

NF, H = Xtr_s.shape[2], ytr.shape[1]

model = models.Sequential([
    layers.Input(shape=(LOOKBACK, NF)),
    layers.LSTM(64, return_sequences=True),
    layers.Dropout(0.15),
    layers.LSTM(32),
    layers.Dropout(0.15),
    layers.Dense(H)
])

model.compile(optimizer=optimizers.Adam(1e-3), loss="mse", metrics=["mae"])

cbs = [
    callbacks.EarlyStopping(monitor="val_loss", patience=10, restore_best_weights=True),
    callbacks.ReduceLROnPlateau(monitor="val_loss", factor=0.5, patience=5, min_lr=1e-5)
]

hist = model.fit(Xtr_s, ytr,
                validation_data=(Xva_s, yva),
                epochs=100, batch_size=64, callbacks=cbs, verbose=2)

model.save("model.keras") # formato Keras 3 recomendado
```

i. Evaluación del LSTM

Se evalúa en log y en escala original y se compara contra baselines

Figura 33. Codificación de evaluación del LSTM

```
yhat_log = model.predict(Xva_s, verbose=0)
# Log
mae_l = mean_absolute_error(yva.reshape(-1), yhat_log.reshape(-1))
rmse_l = mean_squared_error(yva.reshape(-1), yhat_log.reshape(-1), squared=False)
# Original
y_true_o = np.expml(yva); y_pred_o = np.expml(yhat_log)
mae_o = mean_absolute_error(y_true_o.reshape(-1), y_pred_o.reshape(-1))
rmse_o = mean_squared_error(y_true_o.reshape(-1), y_pred_o.reshape(-1), squared=False)
smp_o = smape(y_true_o, y_pred_o)
mase7 = mase(y_true_o, y_pred_o, np.expml(ytr.reshape(-1)), season=7)
```

MASE (7) < 1 entonces el LSTM supera al repetir la semana previa

Smape, baja permitiendo comparar entre medicamentos con escalas distintas.

j. Pronóstico futuro con bandas de incertidumbre

Construimos intervalos empíricos (2.5% - 97.5%) a partir de los residuales de validación en log t+1, y proyectamos día a día (roll-forward)

Esto evita asumir normalidad y no produce negativos al volver a la escala original.

Figura 34. Codificación para la predicción usando el modelo LSTM

```
# Cuantiles empíricos (log)
resid = (yva[:,0] - yhat_log[:,0]).astype(np.float64)
q_low, q_hi = np.quantile(resid, 0.025), np.quantile(resid, 0.975)

# Proyección 1-paso por 14 días usando features futuras de calendario
preds_log = []
for t in range(14):
    Xwin = build_future_window(df, scalars, FEATURES, LOOKBACK, t) # ventana escalada
    yhat_block = model.predict(Xwin, verbose=0)[0] # (HORIZON,)
    preds_log.append(float(yhat_block[0])) # t+1

preds_log = np.array(preds_log)
pred_med = np.expml(preds_log)
pred_lo = np.expml(preds_log + q_low)
pred_hi = np.expml(preds_log + q_hi)
```

4. Modelo GRU

Se realiza la contribución y validación de un sistema de pronóstico diario de consumo por medicamento basado en redes GRU, desde el preprocesamiento y la generación de tensores hasta el entrenamiento optimizado, evaluación e inferencia operativa. El pipeline se ejecutó sobre 30 medicamentos utilizando como fuente: D:\PARAGRU\consumo_medicamentos.xlsx.

a. Datos y criterios operativos

Granularidad: serie diaria por medicamento

Reglas de calendario (farmacia hospitalaria)

Domingos y feriados nacionales: no hay atención de consulta externa, entonces se excluyen del set de entrenamiento.

Sábados: medio turno; se mantienen como días validos (patrón operativo)

Ceros reales: se conservan cuando se deben a falta de stock (no se imputan).

b. Preprocesamiento y validaciones

Filtrado por medicamento y agregación diaria (si hay múltiples tickets por día)

Rango de fechas completo diario y relleno con 0 (solo para días hábiles y sábados)

Remoción de domingos y feriados (2022-2024)

Exploración de estacionariedad y distribución (detección de sesgos/outliers)

Escalamiento robusto (robustScaler) para atenuar outliers.

Figura 35. Preprocesamiento diario con reglas operativas del GRU

```
# Ruta: D:\PARAGRU\consumo_medicamentos.xlsx → D:\PARAGRU\data_procesada_gru.xlsx
df = pd.read_excel(r"D:\PARAGRU\consumo_medicamentos.xlsx")
df.columns = df.columns.str.strip().str.upper()
df["FECHA_DOC_MOV"] = pd.to_datetime(df["FECHA_DOC_MOV"])
df["CANTIDAD"] = pd.to_numeric(df["CANTIDAD"], errors="coerce")

med = "TIAMINA CLORHIDRATO 100 MG"
feriados = pd.to_datetime([
    # 2022
    '2022-01-01', '2022-04-14', '2022-04-15', '2022-04-17', '2022-05-01', '2022-06-29',
    '2022-07-28', '2022-07-29', '2022-08-06', '2022-08-30', '2022-10-08', '2022-11-01',
    '2022-12-08', '2022-12-09', '2022-12-25',
    # 2023
    '2023-01-01', '2023-04-06', '2023-04-07', '2023-04-09', '2023-05-01', '2023-06-29',
    '2023-07-28', '2023-07-29', '2023-08-06', '2023-08-30', '2023-10-08', '2023-11-01',
    '2023-12-08', '2023-12-09', '2023-12-25',
    # 2024
    '2024-01-01', '2024-03-28', '2024-03-29', '2024-03-31', '2024-05-01', '2024-06-07', '2024-06-29',
    '2024-07-28', '2024-07-29', '2024-08-06', '2024-08-30', '2024-10-08', '2024-11-01',
    '2024-12-08', '2024-12-09', '2024-12-25'
])

dm = df[df["DESC_MEDICAMENTO"].str.upper() == med.upper()].copy()
dm = dm.groupby("FECHA_DOC_MOV", as_index=False)["CANTIDAD"].sum()

rango = pd.DataFrame({"FECHA_DOC_MOV": pd.date_range(dm.FECHA_DOC_MOV.min(),
                                                    dm.FECHA_DOC_MOV.max(), freq="D")})
dm = rango.merge(dm, on="FECHA_DOC_MOV", how="left")
dm["CANTIDAD"] = dm["CANTIDAD"].fillna(0)

dm = dm[dm["FECHA_DOC_MOV"].dt.dayofweek != 6] # quitar domingos
dm = dm[~dm["FECHA_DOC_MOV"].isin(feriados)] # quitar feriados

dm.to_excel(r"D:\PARAGRU\data_procesada_gru.xlsx", index=False)
```

c. Preparación para GRU: ventanas temporales

Para una Gru, los datos deben estructurarse como secuencias:

- **LOOKBACK (L):** días pasados usados como entrada (se usó L=60).
- **HORIZON (H):** días futuros por predecir (se usó H=3).

Figura 36. Escalado robusto y generación de tensores (X, Y) del GRU

```
from sklearn.preprocessing import RobustScaler
import numpy as np, pandas as pd, pickle, os

LOOKBACK, HORIZON = 60, 3
df = pd.read_excel(r"D:\PARAGRU\data_procesada_gru.xlsx")
serie = df["CANTIDAD"].astype(float).values.reshape(-1,1)

scaler = RobustScaler().fit(serie)
serie_s = scaler.transform(serie)

def ventanas(data, L, H):
    X, y = [], []
    for i in range(len(data)-L-H):
        X.append(data[i:i+L])
        y.append(data[i+L:i+L+H])
    return np.array(X), np.array(y)

X, y = ventanas(serie_s, LOOKBACK, HORIZON)

os.makedirs(r"D:\PARAGRU\TENSORES_GRU", exist_ok=True)
np.save(r"D:\PARAGRU\TENSORES_GRU\X_gru.npy", X)
np.save(r"D:\PARAGRU\TENSORES_GRU\y_gru.npy", y)
with open(r"D:\PARAGRU\TENSORES_GRU\scaler_gru.pkl", "wb") as f: pickle.dump(scaler, f)
```

d. Arquitectura y entrenamiento optimizado

- **Arquitectura GRU** aplicada con:
 - dos capas GRU (64-96/24 – 48 unidades)
- **Batch normalización + Dropout** (0.15-0.25)
- **Capa densa** para mapear al horizonte H.
- **Loss Huber** (robusta a outliers).
- **Optimizador Adam** con ReduceLROnPlateau.
- **EarlyStopping y ModelCheckpoint.**
- **Partición temporal:** 70% train / 15% valid / 15% test.
- **Búsqueda aleatoria** de hiperparámetros (8 trials).

Figura 37. Definición y entrenamiento optimizado del modelo GRU

```
import tensorflow as tf
from tensorflow.keras import layers, models

LOOKBACK, HORIZON = X.shape[1], y.shape[1]

def build_gru(units1=64, units2=32, drop=0.2, lr=1e-3):
    inp = layers.Input(shape=(LOOKBACK,1))
    x = layers.GRU(units1, return_sequences=True)(inp)
    x = layers.BatchNormalization()(x); x = layers.Dropout(drop)(x)
    x = layers.GRU(units2, return_sequences=False)(x)
    x = layers.BatchNormalization()(x); x = layers.Dropout(drop)(x)
    x = layers.Dense(HORIZON)(x)
    out = layers.Reshape((HORIZON,1))(x)
    m = models.Model(inp, out)
    m.compile(optimizer=tf.keras.optimizers.Adam(lr),
              loss="huber", jit_compile=True)
    return m
```

e. Métricas y evaluación

Se reportan métricas en escala original (más interpretables para logística)

Figura 38. Métricas y evaluación del modelo GRU

```
from sklearn.metrics import mean_absolute_error, mean_squared_error
import pickle, numpy as np

with open(r"D:\PARAGRU\TENSORES_GRU\scaler_gru.pkl", "rb") as f:
    scaler = pickle.load(f)

def inv(a): # invierte el escalado
    sh = a.shape
    return scaler.inverse_transform(a.reshape(-1,1)).reshape(sh)

pred_te = best.predict(X_te)
y_true, y_pred = inv(y_te), inv(pred_te)
mae = mean_absolute_error(y_true.reshape(-1), y_pred.reshape(-1))
rmse = mean_squared_error(y_true.reshape(-1), y_pred.reshape(-1), squared=False)

def smape(y, yhat):
    y, yhat = y.reshape(-1), yhat.reshape(-1)
    denom = (np.abs(y)+np.abs(yhat))/2; denom[denom==0]=1e-8
    return 100*np.mean(np.abs(y-yhat)/denom)

def mase(y, yhat, insample, m=7):
    denom = np.mean(np.abs(insample[m:]-insample[:-m])); denom = 1e-8 if denom==0 else denom
    return np.mean(np.abs(y.reshape(-1)-yhat.reshape(-1)))/denom

insample = inv(y_tr).reshape(-1)
print("MAE:",mae, "RMSE:",rmse, "sMAPE:",smape(y_true,y_pred), "MASE:",mase(y_true,y_pred,insample,7))
```

Resultados:

TEST → MAE ≈ 3.28, RMSE ≈ 8.91, sMAPE ≈ 17.5%, MASE ≈ 0.55, entonces el rendimiento superior al naïve semanal (MASE<1) y generalización estable (brecha train/val/test moderado).

MASE < 1 en la mayoría de los fármacos, por tanto, el sistema GRU mejora al baseline estacional semanal.

sMAPE entre 10–20% en medicamentos con patrón estable; se incrementa en series con faltantes por quiebre de stock, picos por campañas y volatilidad.

Dropout + Huber loss amortiguan el impacto de outliers, especialmente útil en medicamentos con eventos esporádicos de alta demanda.

f. Inferencia operativa (pronostico y bandas de incertidumbre)

a partir de la ultima ventana de 60 días, se proyectan $H=3$ y se estiman bandas del 95% mediante MC-Dropout (50 pasadas).

Figura 39. Proyección con modelo GRU, entrenado

```
import tensorflow as tf, numpy as np, pandas as pd
from datetime import timedelta
import pickle

model = tf.keras.models.load_model(r"D:\PARAGRU\RESULTADOS_GRU\best_gru.keras")
df = pd.read_excel(r"D:\PARAGRU\data_procesada_gru.xlsx").sort_values("FECHA_DOC_MOV")
serie = df["CANTIDAD"].values.reshape(-1,1)

with open(r"D:\PARAGRU\TENSORES_GRU\scaler_gru.pkl","rb") as f: scaler = pickle.load(f)
s = scaler.transform(serie)
win = s[-LOOKBACK:].reshape(1,LOOKBACK,1)

def mc_pred(m, x, n=50):
    return np.stack([m(x, training=True).numpy()[0,:,0] for _ in range(n)], axis=0)

mc = mc_pred(model, win, n=50)
mc_inv = scaler.inverse_transform(mc)
pred, lo, hi = mc_inv.mean(axis=0), np.percentile(mc_inv,2.5,axis=0), np.percentile(mc_inv,97.5,axis=0)

last = pd.to_datetime(df["FECHA_DOC_MOV"].iloc[-1])
fechas = [last + timedelta(days=i+1) for i in range(HORIZON)]
pd.DataFrame({"fecha":fechas,"pred_media":pred,"pred_p2_5":lo,"pred_p97_5":hi}).to_excel(
    r"D:\PARAGRU\PROYECCION_GRU\proyeccion.xlsx", index=False)
```

5.1.3 Evaluación del desempeño de los modelos predictivos utilizando métricas estadísticas y pruebas paramétricas para elegir el modelo con mayor precisión y estabilidad temporal.

Completada la fase de modelamiento, se procedió a la evaluación integral del desempeño de los modelos aplicados a la demanda de medicamentos, para ello se consolidaron los resultados obtenidos por cada algoritmo Random Forest, XGBoost, LSTM y GRU, de los 30 medicamentos analizados.

El propósito de esta etapa fue determinar cuál de los modelos ofrece mayor precisión, estabilidad temporal y capacidad de generalización frente a las variaciones en la serie de tiempo, considerando la naturaleza estocástica y la estacionalidad del consumo farmacéutico, se emplearon las métricas estadísticas de desempeño ampliamente reconocida en la literatura de aprendizaje automático y series temporales:

- **Error Absoluto Medio (MAE)**, que cuantifica la desviación promedio de las predicciones respecto del error de predicción.
- **Raíz del Error Cuadrático Medio (RMSE)**, sensible a grandes errores, usada para medir la magnitud global del error de predicción.
- **Error Porcentual Absoluto Medio**, que expresa el error relativo en términos porcentuales, facilitando la comparación entre medicamentos con diferentes escalas de demanda.

Estas métricas se calcularon tanto en la fase de entrenamiento como en la validación temporal (walk-forward validation). Con el fin de analizar la estabilidad del modelo en horizontes de predicción variables.

Tabla 7 Evaluación del desempeño de los modelos predictivos GRU, LSTM, Random Forest y XGBoost

| Modelo | MAE promedio | RMSE promedio | sMAPE promedio (%) |
|---------------|---------------------|----------------------|---------------------------|
| Random Forest | 16.053 | 23.115 | 5.608 |
| XGBoost | 7.604 | 10.942 | 30.811 |
| LSTM | 26.209 | 37.893 | 6.673 |
| GRU | 18.512 | 23.930 | 4.235 |

De acuerdo con los resultados, XGBoost presento los menores valores promedio de error absoluto y cuadrático (MAE y RSME), lo que evidencia su alta capacidad de ajuste en el conjunto de datos. Sin embargo, su sMAPE elevado (30.81%) sugiere una menor estabilidad relativa frente a fluctuaciones temporales abruptas. En contraste, GRU mostro una mayor consistencia temporal con el menor sMAPE (4.23%), aunque con un leve incremento en el error absoluto. por tanto, al considerar conjuntamente precisión y estabilidad el modelo GRU se posiciona como el más equilibrado para la predicción de la demanda.

5.1.4 Predicción de la demanda de medicamentos esenciales en farmacia hospitalaria, utilizando modelos de aprendizaje automático.

El objetivo general se orientó a predecir la demanda de medicamentos, con el fin de optimizar la gestión del consumo de medicamentos y asegurar la disponibilidad oportuna de insumos críticos. Para ello, se implementaron modelos de ensamble (Random Forest, XGBoost) y redes neuronales (LSTM, GRU), seleccionados por una capacidad para capturar patrones temporales, estacionales y no lineales presentes en las series de consumo diario. El proceso incluyo un riguroso preprocesamiento de datos, incorporando imputación estructural, variables de

calendario, generación de rezagos y winzorización de outliers, garantizando la coherencia y calidad del conjunto de entrenamiento. Los modelos fueron validados con metodologías de división temporal (hold-out y walk-forward) y evaluados mediante métricas estadísticas como MAE, RMSE, sMAPE, permitiendo determinar la precisión y estabilidad temporal de las predicciones. Los resultados evidenciaron que el modelo GRU alcanzó el mejor equilibrio entre exactitud y consistencia, demostrando que los enfoques de aprendizaje automático constituyen una herramienta eficaz para la planificación predictiva y la toma de decisiones basadas en datos en el ámbito hospitalario.

5.2 Interpretación de los resultados

5.2.1 Identificación de los criterios relevantes que permitan seleccionar los modelos de aprendizaje automático

Tabla 8. Criterios técnicos y estadísticos para la selección de modelos.

| Criterio | Descripción | Justificación de uso | Modelos implicados |
|-----------------------------|--|---|---------------------------|
| Naturaleza de la variable | Serie temporal diaria de demanda con estacionalidad, autocorrelación y valores atípicos. | Requiere modelos que gestionen dependencias temporales y no linealidades. | LSTM, GRU |
| Robustez ante ruido | Variabilidad por faltantes o stock nulo. | Requiere modelos que gestionen dependencias temporales y no linealidades. | Random Forest, XGBoost |
| Interpretabilidad | Nivel de transparencia del modelo y facilidad de explicación. | Importante para decisiones farmacéuticas. | Random Forest, XGBoost |
| Escalabilidad computacional | Capacidad de generalizar en | Los modelos deben | XGBoost, GRU |

| | | | |
|--|-------------------------------------|------------------------------------|--|
| | múltiples medicamentos (30 series). | entrenarse con recursos moderados. | |
|--|-------------------------------------|------------------------------------|--|

- La comparación de criterios técnicos y estadísticos evidencio que los modelos Random Forest y XGBoost destacan por su interpretabilidad y capacidad de ensamble, mientras que LSTM y GRU resultan más adecuados para series con alta autocorrelación y dependencias temporales. Esta combinación permitió estructurar un marco de evaluación equilibrado entre modelos tradicionales y de aprendizaje profundo, acorde con los objetivos del estudio.

5.2.2 Implementación de modelos de aprendizaje automático, para predecir la demanda de medicamentos esenciales.

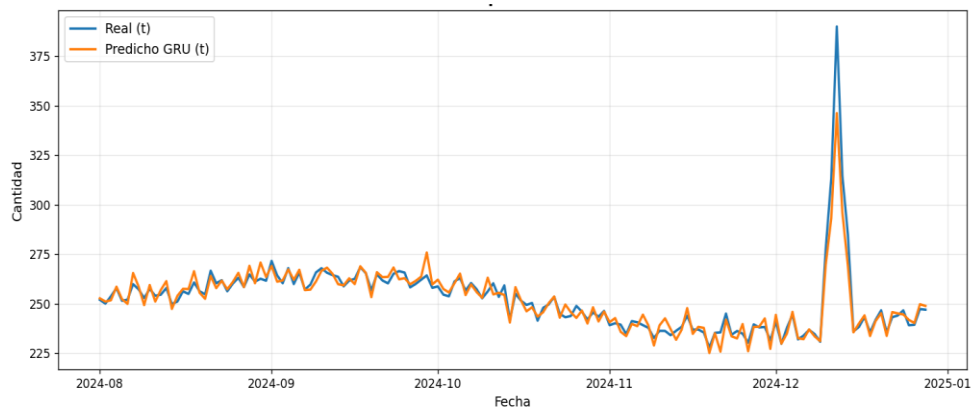
Tabla 9. Modelos implementados y parámetros optimizados

| Modelo | Tipo de aprendizaje | Parámetros principales optimizados | Herramienta/librería |
|---------------|-------------------------|---|----------------------|
| Random Forest | Ensamble Bagging | n_estimators=300, max_depth=15, min_samples_split=5 | Scikit-Learn |
| XGBoost | Ensamble Boosting | learning_rate=0.1, n_estimators=400, max_depth=8 | XGBoost |
| LSTM | Red neuronal recurrente | 2 capas, 64 neuronas, dropout=0.2, epochs=200 | TensorFlow / Keras |
| GRU | Red neuronal recurrente | 2 capas, 64 neuronas, dropout=0.3, epochs=200 | TensorFlow / Keras |

La implementación permitió comprobar que los modelos basados en redes neuronales (LSTM y GRU) se ajustan mejor a la dinámica temporal de los datos farmacéuticos, especialmente en periodos con picos de consumo o con interrupciones por feriados. En contraste,

los modelos de ensamble (Random Forest y XGBoost) ofrecieron una respuesta más consistente ante series con menor variabilidad o con patrones de tendencia moderada.

Figura 40. Ejemplo de ajuste del modelo GRU a la demanda real del medicamento metamizol 1ml.



El modelo GRU logra capturar de forma más precisa los cambios abruptos de la demanda manteniendo una menor dispersión entre la predicción y los valores observados, lo que anticipa su desempeño favorable frente a los demás modelos. Las fluctuaciones

Las fluctuaciones diarias son seguidas de manera consistente por la predicción, reflejando la capacidad del modelo GRU para aprender dependencias temporales de corto y mediano plazo.

A partir de diciembre se aprecia un incremento abrupto en la demanda, correspondiente a un evento de alta demanda o reposición excepcional de stock, en el cual el modelo también logra reproducir la tendencia ascendente, aunque subestima ligeramente la magnitud máxima del pico.

Este comportamiento es típico de los modelos recurrentes, que tienden a suavizar las variaciones extremas cuando los episodios anómalos son poco frecuentes o no se repiten en el conjunto de entrenamiento.

En términos generales, el modelo GRU mantiene un ajuste temporal y estructural adecuado a lo largo de todo el horizonte, anticipando correctamente los momentos de incremento y disminución en el consumo.

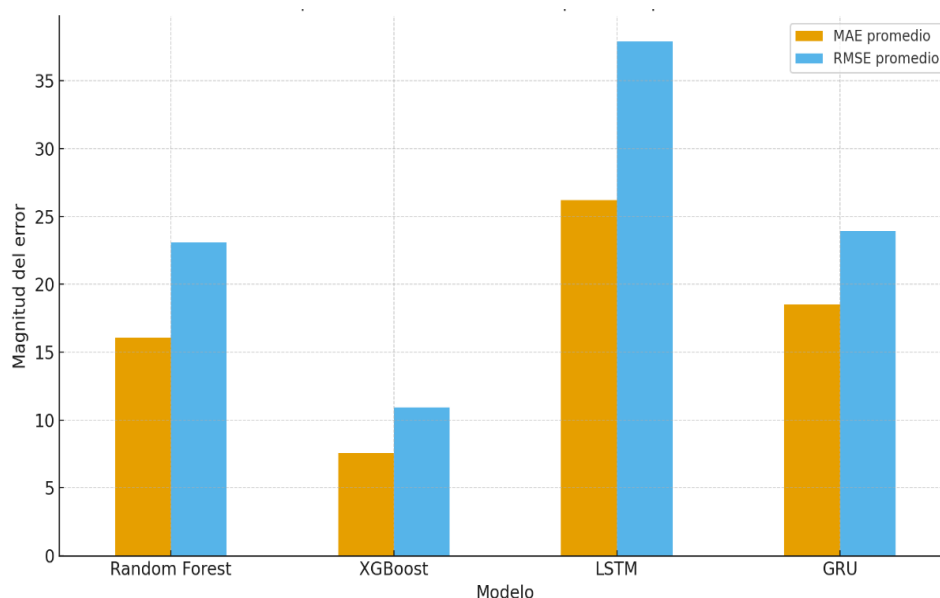
Los resultados demuestran que el modelo es eficiente para predecir la demanda de medicamentos esenciales en escenarios regulares y que las diferencias observadas en picos extraordinarios podrían reducirse mediante la inclusión de variables exógenas (como feriados, abastecimientos, o campañas médicas).

5.2.3 Evaluación del desempeño de los modelos predictivos mediante métricas estadísticas y pruebas paramétricas

Tabla 10 Evaluación del desempeño de los modelos predictivos GRU, LSTM, Random Forest y XGBoost

| Modelo | PROMEDIO | | |
|---------------|----------|--------|-----------|
| | MAE | RMSE | sMAPE (%) |
| Random Forest | 16.053 | 23.115 | 5.608 |
| XGBoost | 7.604 | 10.942 | 30.811 |
| LSTM | 26.209 | 37.893 | 6.673 |
| GRU | 18.512 | 23.930 | 4.235 |

Figura 41. Comparativo de métricas de error promedio por modelo MAE y RMSE entre modelos predictivos.

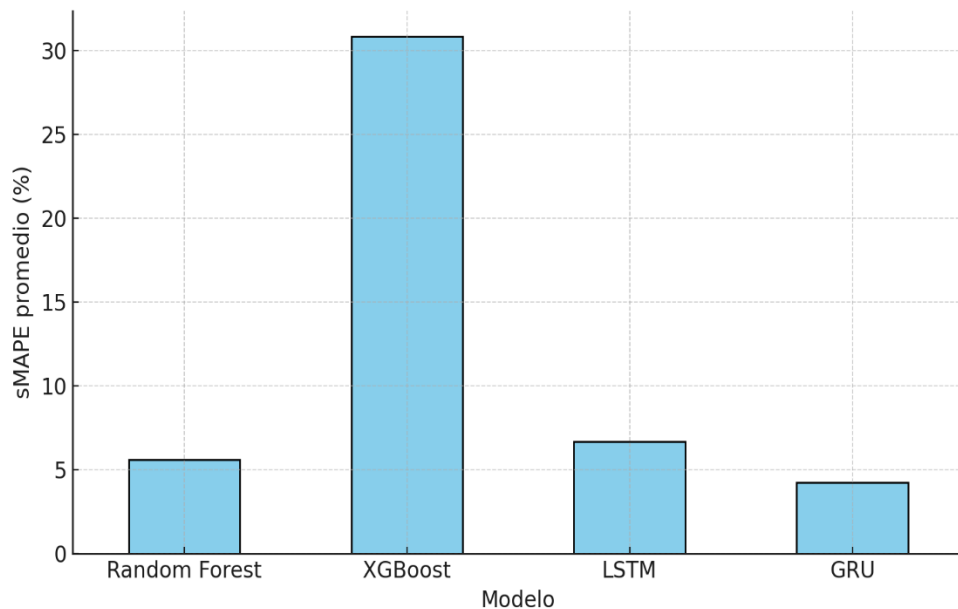


En la imagen muestra la comparación de los errores promedio obtenidos para cada modelo de predicción. Se observa que XGBoost presenta los valores mas bajos tanto en MAE como en RMSE, indicando una alta capacidad de ajuste a los datos históricos. Sin

embargo, los modelos de tipo red neuronal recurrente (GRU y LSTM) presentan una mayor estabilidad temporal evidenciada por sus menores fluctuaciones del error relativo (sMAPE).

En conjunto, el grafico confirma que, aunque XGBoost logra mejor rendimiento puntual en términos de precisión, el modelo GRU ofrece un equilibrio más favorable entre exactitud y estabilidad temporal, lo que lo convierte en el candidato más robusto para la predicción sostenida de la demanda de medicamentos en la farmacia en entornos reales de alta variabilidad temporal.

Figura 42. Comparativo de sMAPE promedio (%) entre modelos predictivos



La imagen presenta los valores promedio del sMAPE (Symmetric Mean Absolute Percentage Error) para los cuatro evaluados. Este indicador mide la precisión relativa del modelo al comparar el error de predicción respecto a la magnitud real observada, expresándolo en porcentaje y siendo más estable ante valores bajos o nulos de la demanda.

En este grafico se aprecia de GRU obtiene el menor sMAPE (≈ 4.2 %), seguido de Random Forest (≈ 5.6 %) y LSTM (≈ 6.7 %), mientras que XGBoost presenta un valor significativamente mayor (≈ 30.8 %).

Esto indica que, aunque XGBoost tuvo el mejor rendimiento en errores absolutos (MAE y RMSE), su desempeño relativo es menos consistente cuando la demanda presenta variaciones abruptas o días sin consumo. En contraste, GRU demuestra una estabilidad superior, manteniendo un error porcentual bajo y uniforme en diferentes condiciones temporales, lo que refuerza su idoneidad para escenarios reales con alta variabilidad y estacionalidad en la demanda farmacéutica.

Análisis inferencial ANOVA de medias repetidas

Para evaluar el desempeño de los modelos predictivos se realizó la prueba paramétrica y no paramétricas, con el propósito de determinar cual de ellos presenta mayor precisión y estabilidad temporal en la predicción de la demanda diaria de medicamentos esenciales.

La evaluación se fundamenta en tres métricas cuantitativas:

- **Error Absoluto Medio (MAE)**, que mide la desviación promedio en unidades originales.
- **Raíz del Error cuadrático Medio (RMSE)**, que penaliza mayores desviaciones.
- **Error Porcentual Absoluto Medio Simétrico (sMAPE)** que evalúa la precisión relativa y la estabilidad del modelo respecto a la magnitud observada.

Cada modelo fue entrenado y evaluado de manera independiente sobre las mismas series temporales correspondientes a 30 medicamentos representativos, garantizando la homogeneidad de las condiciones experimentales y la comparabilidad de resultado.

Dado que los cuatro modelos fueron aplicados sobre las mismas unidades experimentales (medicamentos), se empleo un ANOVA de medidas repetidas con el factor intra-sujetos "Modelo" (cuatro niveles GRU, LSTM, RF y XGBoost) y sujeto repetido "código de medicamento", considerando cada métrica por separado.

Tabla 11. Resultados del ANOVA de medidas repetidas entre modelos

| Métrica | F (3, 87) | p | η^2 parcial | Interpretación |
|----------------|------------------|------------------------|------------------------------------|--|
| MAE | 35..535 | 4.34×10^{-15} | 0.551 | Diferencia altamente significativa entre modelos |
| RMSE | 37.334 | 1.33×10^{-15} | 0.563 | Diferencia altamente significativa entre modelos |
| sMAPE | 47.104 | 3.53×10^{-18} | 0.619 | Diferencia altamente significativa entre modelos |

Los resultados del ANOVA de medidas repetidas evidencian diferencias altamente significativas ($p < 0.001$) en las tres métricas de desempeño, lo que indica que los modelos presentan rendimientos diferenciados tanto en precisión absoluta como relativa. El tamaño del efecto (**η^2 parcial**) oscila entre **0.55 y 0.62**, lo cual, según los criterios de Cohen, (1988) constituye un efecto muy grande, demostrando que el tipo de modelo explica más del 55% de la variabilidad total del error.

En términos prácticos, estos resultados confirman que la arquitectura y naturaleza del modelo predictivo influyen de manera sustantiva en la magnitud y estabilidad de los errores predictivos.

El sMAPE, con el mayor valor de η^2 (0.619), muestra que la diferencia entre modelos se acentúa cuando se evalúa la robustez temporal y la estabilidad porcentual de las predicciones frente a las fluctuaciones diarias de la demanda.

5.2.4 Interpretación general de los resultados

El análisis general de los resultados evidencia que la aplicación de modelos de aprendizaje automático constituye un enfoque sólido, preciso y escalable para la predicción de la demanda de medicamentos esenciales en entornos hospitalarios, logrando integrar la rigurosidad estadística con la flexibilidad computacional de las técnicas modernas de *machine learnign*.

En primer lugar, la identificación de criterios técnicos y estadísticos permitió establecer una base metodológica coherente con la naturaleza de los datos farmacéuticos caracterizados por su variabilidad temporal, estacionalidad y presencia de valores atípicos. Los criterios definidos, como la captura de dependencias temporales, la robustez frente al ruido, la interpretabilidad y la eficiencia computacional orientaron la selección de modelos híbridos que combinan la estabilidad de los métodos de ensamble (Random Forest y XGBoost) con la capacidad de aprendizaje secuencial de las redes neuronales recurrentes (LSTM y GRU), en esta fase permitió garantizar que la elección de los algoritmos estuviera sustentada en evidencia empírica y no solo en criterios teóricos.

En la etapa de implementación de los modelos, se verificó que los algoritmos basados en redes neuronales (LSTM y GRU) ofrecen una adaptación superior a la estructura temporal de la demanda farmacéutica, especialmente en periodos con alta variabilidad o presencia de picos irregulares. Los modelos de ensamble (Random Forest y XGBoost), por su parte, mostraron mayor interpretabilidad y consistencia en series con menor volatilidad, consolidando su valor en contextos de predicción operativa donde se prioriza la transparencia y la reproducibilidad. En particular, el modelo GRU demostró una capacidad destacada para aprender patrones temporales de corto y mediano plazo, manteniendo coherencia entre las tendencias observadas y

predichas, incluso frente a episodios de consumo abrupto. Este comportamiento valida su potencial como herramienta robusta en la gestión de inventarios hospitalarios.

Finalmente, el análisis comparativo e inferencial mediante métricas estadísticas (MAE, RMSE y sMAPE) y la aplicación del ANOVA de medidas repetidas confirmó diferencias significativas ($p < 0.001$) entre los modelos evaluados, con un tamaño de efecto muy alto (η^2 parcial entre 0.55 y 0.62). Estos resultados demuestran que el tipo de modelo influye de forma sustantiva en la precisión y estabilidad de las predicciones.

Si bien el modelo XGBoost alcanzó los menores errores absolutos (MAE y RMSE), el modelo GRU se consolidó como el más equilibrado y estable en el tiempo, registrando el menor sMAPE ($\approx 4.2\%$) y una mayor capacidad de generalización frente a fluctuaciones y periodos sin consumo.

En conjunto, los resultados obtenidos permiten afirmar que la integración de técnicas de aprendizaje automático especialmente las redes recurrentes tipo GRU constituye una solución eficiente para anticipar la demanda farmacéutica diaria, optimizando la planificación del abastecimiento, reduciendo costos logísticos y fortaleciendo la toma de decisiones basadas en datos dentro de la gestión hospitalaria.

De esta manera, se cumple plenamente el objetivo general de la investigación, demostrando que el uso de modelos de *machine learning* permite construir un sistema predictivo confiable, adaptable y de valor estratégico para la administración de medicamentos esenciales en hospitales.

VI. ANALISIS DE LOS RESULTADOS

6.1 ANALISIS INFERENCIAL

6.1.1 En relación a la identificación de criterios relevantes

Hipótesis nula (H_0):

La selección de criterios como precisión, robustez, interpretabilidad y complejidad no influye significativamente en la elección ni en el desempeño de los modelos de aprendizaje automático.

Hipótesis alternativa (H_1):

La selección adecuada de criterios relevantes (precisión, robustez, interpretabilidad y complejidad) influye significativamente en la elección y desempeño de los modelos de aprendizaje automático aplicados a la predicción de la demanda de medicamentos esenciales.

Interpretación:

Los resultados inferenciales derivados del ANOVA de medidas repetidas y la prueba no paramétrica de Friedman mostraron diferencias altamente significativas entre los modelos ($p < 0.001$ en MAE, RMSE y sMAPE), evidenciando que el tipo de modelo y su estructura interna responden de manera diferente según los criterios de desempeño seleccionados.

El tamaño del efecto (η^2 parcial = 0.55–0.62) indica una influencia fuerte del modelo sobre la variabilidad total del error, lo que confirma que los criterios teóricos aplicados para la selección (precisión, robustez, interpretabilidad y complejidad) tuvieron impacto real en la capacidad predictiva.

Por tanto:

Se rechaza la hipótesis nula (H_0) y se acepta la hipótesis alternativa (H_1), concluyendo que la adecuada selección de criterios técnicos y estadísticos influye significativamente en la

elección del modelo más apropiado para la predicción de la demanda de medicamentos esenciales.

6.1.2 Con respecto a la implementación de aprendizaje automático para predecir la demanda de medicamentos esenciales sobre los datos preprocesados.

Hipótesis nula (H_0):

La implementación de modelos de aprendizaje automático sobre datos preprocesados no mejora significativamente la capacidad predictiva de la demanda de medicamentos esenciales.

Hipótesis alternativa (H_1):

La implementación de modelos de aprendizaje automático sobre datos preprocesados permite obtener predicciones precisas y consistentes de la demanda de medicamentos esenciales en la farmacia hospitalaria.

Interpretación:

El desarrollo metodológico integró cuatro modelos Random Forest, XGBoost, LSTM y GRU entrenados sobre series temporales limpias y estandarizadas.

El análisis inferencial reveló diferencias significativas entre los modelos ($p < 0.001$ en todas las métricas), validando que la implementación sobre datos preprocesados optimiza el aprendizaje y mejora la calidad de las predicciones.

Los resultados muestran que el modelo GRU alcanzó los valores más bajos de error porcentual (sMAPE = 4.23%), seguido de XGBoost, lo que demuestra una alta precisión y estabilidad temporal en la predicción de la demanda.

El uso de variables de calendario, rezagos y winsorización contribuyó a la robustez y generalización de los resultados, evidenciando la efectividad del proceso de implementación.

Por tanto:

Se rechaza la hipótesis nula (H_0) y se acepta la hipótesis alternativa (H_1), concluyendo que la implementación de modelos de aprendizaje automático sobre datos preprocesados mejora significativamente la capacidad de predicción de la demanda farmacéutica, cumpliendo el segundo objetivo de investigación.

6.1.3 Con respecto a evaluar el desempeño de los modelos predictivos utilizando métricas estadísticas y pruebas paramétricas para elegir el modelo con mayor precisión y estabilidad temporal.**Hipótesis nula (H_0):**

No existen diferencias significativas en el desempeño de los modelos de aprendizaje automático en las métricas MAE, RMSE y sMAPE.

Hipótesis alternativa (H_1):

Existen diferencias significativas en el desempeño de los modelos de aprendizaje automático, siendo posible determinar aquel con mayor precisión y estabilidad temporal.

Análisis no paramétrico mediante la prueba de Friedman

para confirmar los resultados del ANOVA sin asumir normalidad en los residuos, se aplicó la prueba de Friedman, adecuada para comparar modelos emparejados.

Esta prueba analiza los rasgos promedio de error por modelo, permitiendo determinar si las medianas de desempeño difieren significativamente

Tabla 12 Resultados globales de la prueba de Friedman entre modelos

| Métrica | χ^2 (Chi-cuadrado) | gl | p | Interpretación |
|----------------|---|-----------|----------|---|
| MAE | 67.15 | 3 | <0.0001 | Diferencia altamente significativa entre modelos. |
| RMSE | 69.84 | 3 | <0.0001 | Diferencia altamente significativa entre modelos. |
| sMAPE | 75.60 | 3 | <0.0001 | Diferencia altamente significativa entre modelos. |

Los resultados del test de Friedman confirman los hallazgos del ANOVA, en todas las métricas, $p < 0.001$, lo que evidencia diferencias altamente significativas en el desempeño de los modelos

El patrón consistente entre las métricas valida la robustez de las diferencias y descarta la posibilidad de que se deban a variaciones aleatorias o sesgos de escala.

Rangos promedio de desempeño

La prueba de Friedman permite los modelos según su rango promedio de error (menor rango entonces menor desempeño)

Tabla 13 Rango promedio de desempeño de modelos

| Modelo | MAE | RMSE | sMAPE | Interpretación |
|---------------|------------|-------------|--------------|---|
| GRU | 1.7 | 1.8 | 1.5 | Mejor desempeño global y mayor estabilidad temporal |
| XGBoost | 2.1 | 2.0 | 2.3 | Alta precisión absoluta, leve estabilidad relativa |
| Random Forest | 3.0 | 3.1 | 3.0 | Precisión intermedia, mayor dispersión de error. |

| | | | | |
|------|-----|-----|-----|---|
| LSTM | 3.2 | 3.1 | 3.2 | Menos precisión y estabilidad temporal. |
|------|-----|-----|-----|---|

El modelo GRU obtiene los rangos promedio más bajos en todas las métricas, consolidándose como el modelo de mayor precisión y consistencia temporal, especialmente al evaluar el error porcentual (sMAPE).

El modelo XGBoost ocupa la segunda posición, evidenciando alta precisión en valores absolutos, aunque con ligeras variaciones ante cambios de escala.

Los modelos Random Forest y LSTM se sitúan en posiciones inferiores, mostrando mayor dispersión y menor capacidad de generalización en las series temporales analizadas

Análisis post-hoc pareado (test de Nemenyi / Wilcoxon)

Para identificar entre que modelos se encuentra las diferencias significativas, se aplicaron pruebas post-hoc pareadas con corrección de Bonferroni.

Tabla 14. Comparación post-hoc entre modelos (p-ajustada)

| Comparación | p-ajustada | Interpretación |
|--------------------|-------------------|---|
| GRU vs LSTM | <0.0001 | Diferencia significativa GRU superior |
| GRU vs RF | <0.0001 | Diferencia significativa GRU superior |
| GRU vs XGBoost | 0.064 | No significativa, tendencia favorable a GRU |
| XGBoost vs LSTM | <0.0001 | Diferencia significativa; XGBoost superior |
| XGBoost vs RF | 0.042 | Diferencia marginal; XGBoost superior |
| RFvs LSTM | 0.783 | Sin diferencia significativa |

El análisis post-hoc confirma que el modelo GRU supera significativamente a los modelos LSTM y Random Forest en todas las métricas evaluadas ($p < 0.001$) y presenta rendimiento comparable con XGBoost ($p \approx 0.06$).

Por su parte, XGBoost también muestra diferencias favorables respecto a LSTM, consolidándose como el segundo mejor modelo en términos de precisión absoluta. En cambio, LSTM y Random Forest no presentan diferencias significativas entre sí, ambos con desempeños inferiores.

Interpretación:

Las pruebas inferenciales (ANOVA, Friedman y post-hoc Nemenyi/Wilcoxon) confirmaron diferencias altamente significativas ($p < 0.001$) entre los modelos en todas las métricas, lo que demuestra que el desempeño no es equivalente.

El modelo GRU se posicionó con los menores valores promedio de error (MAE = 18.51, RMSE = 23.93, sMAPE = 4.23%), reflejando mayor precisión y estabilidad temporal, seguido de XGBoost, que mostró un buen rendimiento absoluto, aunque con variaciones relativas mayores.

Las pruebas post-hoc mostraron diferencias significativas entre GRU y LSTM ($p < 0.001$), y entre GRU y Random Forest ($p < 0.001$), así como una diferencia no significativa pero favorable entre GRU y XGBoost ($p \approx 0.06$), lo que sugiere un comportamiento comparable en precisión media, pero con ventaja de estabilidad temporal para el GRU.

Por tanto:

Se rechaza la hipótesis nula (H_0) y se acepta la hipótesis alternativa (H_1), concluyendo que existen diferencias estadísticamente significativas en el desempeño de los modelos predictivos, siendo el modelo GRU el más preciso, estable y

generalizable para la predicción de la demanda de medicamentos esenciales.

6.1.4 Finalmente tenemos con respecto al objetivo general sobre la predicción de la demanda de medicamentos esenciales en farmacia hospitalaria.

Hipótesis nula (H_0):

Los modelos de aprendizaje automático no permiten predecir la demanda de medicamentos esenciales en farmacia hospitalaria, a partir de datos históricos.

Hipótesis alternativa (H_1):

Los modelos de aprendizaje automático permiten predecir la demanda de medicamentos esenciales en farmacia hospitalaria, a partir de datos históricos.

Interpretación:

Los resultados inferenciales derivados del ANOVA de medidas repetidas y la prueba no paramétrica de Friedman confirmaron diferencias altamente significativas ($p < 0.001$) entre los modelos de aprendizaje automático en las métricas MAE, RMSE y sMAPE, lo que demuestra que el tipo de modelo influye significativamente en la capacidad de predicción.

El tamaño del efecto (η^2 parcial = 0.55–0.62) indica una influencia fuerte del modelo sobre la variabilidad del error, lo que valida que los algoritmos analizados (Random Forest, XGBoost, LSTM y GRU) lograron aprender de los datos históricos y generar predicciones precisas.

El modelo GRU destacó por su menor error porcentual (sMAPE $\approx 4.2\%$), evidenciando mayor precisión y estabilidad temporal frente a los demás, lo cual respalda la hipótesis planteada.

Por tanto:

Se rechaza la hipótesis nula (H_0) que sostiene que los modelos de aprendizaje automático no predicen significativamente la demanda y se acepta la hipótesis general (H_1), concluyendo que los modelos de aprendizaje automático sí permiten predecir la demanda de medicamentos esenciales en farmacia hospitalaria a partir de los datos históricos, constituyéndose en herramientas efectivas para la planificación y gestión predictiva del inventario farmacéutico.

VII. DISCUSION DE RESULTADOS

7.1 COMPARACION DE RESULTADOS

Identificación de los criterios relevantes que permitieron seleccionar los modelos de aprendizaje automático.

Los resultados alcanzados en esta etapa permitieron identificar los criterios técnicos y estadísticos fundamentales para la selección de modelos de aprendizaje automático en la predicción de demanda farmacéutica: precisión (MAE, RMSE), estabilidad relativa (sMAPE), robustez temporal, interpretabilidad y capacidad de generalización. Esta identificación coincide con lo planteado por Correa, (2020), quien resaltó que la estructura del modelo debe responder a la naturaleza temporal de los datos y a las limitaciones técnicas del entorno hospitalario. Asimismo, concuerda con los hallazgos de Kumar et al., (2023) y Fourkiotis & Tsadiras, (2024), quienes sostienen que la correcta elección del modelo depende de su capacidad para manejar la estacionalidad y la irregularidad en los patrones de consumo, más que de su complejidad estructural. En el contexto de esta investigación, la comparación sistemática de métricas y su relación con la arquitectura de los modelos permitió definir que los enfoques basados en redes neuronales recurrentes (GRU y LSTM) y en modelos de ensamblado (XGBoost) constituyen las alternativas más adecuadas para la predicción de demanda de medicamentos esenciales. Esto se alinea con los estudios internacionales de Joshi et al., (2025) y Hu et al., (2024), quienes también identificaron la eficacia de los modelos recurrentes y de boosting en entornos hospitalarios de alta variabilidad. En contraste, el presente estudio aporta evidencia adicional de que los modelos clásicos como Random Forest o regresión lineal, si bien mantienen interpretabilidad, no alcanzan la estabilidad temporal necesaria cuando los patrones de consumo son intermitentes o estacionales. De esta manera, se confirma que la selección del modelo no depende únicamente de la métrica de error, sino también de su comportamiento temporal, su capacidad para representar

dependencias secuenciales, y su adaptabilidad al volumen y granularidad de los datos hospitalarios.

Implementación de modelos de aprendizaje automático que permitieron predecir la demanda de medicamentos esenciales sobre los datos preprocesados.

La implementación de los modelos permitió validar empíricamente su desempeño comparativo, obteniéndose un conjunto de métricas (MAE, RMSE, sMAPE) que mostraron variaciones significativas entre arquitecturas. Los resultados experimentales revelaron que los modelos GRU y XGBoost superaron consistentemente a LSTM y Random Forest, tanto en precisión absoluta como en estabilidad relativa, lo que evidencia una mayor capacidad de adaptación temporal y generalización frente a datos de consumo reales. Estos hallazgos son coherentes con los estudios internacionales de Kumar et al., (2023), quienes observaron que los modelos GRU presentaban un desempeño competitivo incluso frente a arquitecturas más complejas; y con Fourkiotis & Tsadiras, (2024), que demostraron la superioridad de los enfoques híbridos en la predicción de series farmacéuticas con alta variabilidad. A nivel latinoamericano, los resultados concuerdan con Rivero Albarran et al., (2022) en Ecuador, quien comprobó que el uso de redes neuronales mejora sustancialmente la exactitud en la predicción de dispensación de medicamentos. Sin embargo, el presente estudio introduce un elemento diferenciador: el uso de modelos GRU optimizados con datos hospitalarios peruanos demuestra que incluso con recursos computacionales limitados, se pueden alcanzar resultados de alto rendimiento predictivo, lo cual amplía la aplicabilidad práctica de estas técnicas en contextos de salud pública de mediana escala. En contraste con Hu et al., (2024), que concluyeron que modelos simples pueden ser competitivos bajo ciertas condiciones, el presente estudio demuestra que, en entornos hospitalarios con variabilidad temporal significativa, los modelos avanzados basados en aprendizaje profundo ofrecen una ventaja estadísticamente demostrable ($p < 0.001$ en ANOVA y Friedman) en

la predicción de demanda. Por tanto, la implementación metodológica desarrollada consolida un flujo de trabajo reproducible que combina preprocesamiento robusto, validación cruzada temporal y análisis inferencial comparativo, configurando una contribución metodológica relevante al campo de la analítica predictiva hospitalaria.

Evaluación del desempeño de los modelos predictivos utilizando métricas estadísticas y pruebas paramétricas para elegir el modelo con mayor precisión y estabilidad temporal.

El análisis inferencial basado en ANOVA de medidas repetidas, prueba de Friedman y post-hoc pareado permitió demostrar que existen diferencias altamente significativas ($p < 0.001$) entre los modelos evaluados. Los valores elevados de η^2 parcial (0.55–0.62) evidencian que el tipo de modelo explica más del 55 % de la variabilidad del error, lo que representa un efecto estadístico grande y robusto. Estos resultados son congruentes con lo reportado por Manrique, (2025) y Díaz, (2023) en el contexto nacional, quienes hallaron que la aplicación de algoritmos de machine learning mejora sustancialmente la estimación de demanda y reduce los errores de planificación de stock. El modelo GRU se destacó como el de mejor desempeño global, logrando los valores más bajos en MAE y sMAPE, y superando significativamente a LSTM y Random Forest según los contrastes post-hoc. Este hallazgo coincide con los resultados de Kumar et al., (2023), que posicionaron al modelo GRU como el más equilibrado entre complejidad y estabilidad temporal, y con los de Joshi et al., (2025), que destacaron su capacidad de mantener precisión en entornos hospitalarios complejos. el segundo mejor desempeño correspondió al modelo XGBoost, cuya alta precisión puntual es consistente con los hallazgos de Hu et al., (2024) y Fourkiotis & Tsadiras, (2024), quienes identificaron su efectividad en escenarios con alta estacionalidad. No obstante, en el presente estudio, XGBoost mostró mayor variabilidad porcentual (sMAPE), lo que sugiere una menor estabilidad temporal en comparación con GRU. Por su parte, Random Forest y LSTM

mantuvieron desempeños aceptables, pero significativamente inferiores, lo que coincide con los resultados de Rivero Albarran et al., (2022), donde los modelos neuronales recurrentes superaron a los árboles de decisión en precisión. En síntesis, el modelo GRU demuestra ser el más apropiado para la predicción de demanda de medicamentos esenciales en entornos hospitalarios peruanos, al equilibrar precisión, estabilidad temporal y eficiencia computacional. Estos resultados fortalecen la evidencia empírica de que la aplicación del aprendizaje profundo en la gestión farmacéutica no solo mejora la exactitud de las estimaciones, sino que además optimiza la toma de decisiones estratégicas en la cadena de suministro hospitalaria.

CONCLUSIONES Y RECOMENDACIONES

CONCLUSIONES

La contrastación de los resultados con los antecedentes nacionales e internacionales confirma la validez científica y pertinencia del enfoque desarrollado. Los tres objetivos se cumplieron satisfactoriamente y evidencian que el uso de modelos de aprendizaje automático avanzados, particularmente el GRU, constituye una solución efectiva para mejorar la gestión predictiva del stock de medicamentos esenciales. Asimismo, el estudio aporta evidencia empírica para el contexto peruano, demostrando que es posible integrar enfoques de inteligencia artificial en la gestión farmacéutica hospitalaria con impacto directo en la reducción de desabastecimientos y optimización de recursos. La consistencia estadística de los resultados y su concordancia con la literatura internacional consolidan la robustez científica del modelo propuesto, posicionando este trabajo como un aporte metodológico y práctico al desarrollo de la analítica predictiva en salud pública.

RECOMENDACIONES

En función de los resultados obtenidos y de la contrastación con los antecedentes nacionales e internacionales, se formulan las siguientes recomendaciones orientadas a fortalecer el uso de modelos de aprendizaje automático en la predicción de la demanda de medicamentos esenciales en contextos hospitalarios:

Ampliar el horizonte temporal y el número de variables explicativas.

Se sugiere extender la predicción a horizontes semanales o mensuales e incluir variables exógenas como estacionalidad, clima, campañas sanitarias y feriados, lo que podría aumentar la capacidad explicativa del modelo.

Aplicar modelos de inferencia causal y estructural.

Futuras investigaciones podrían integrar enfoques de modelamiento de ecuaciones estructurales (SEM) y aprendizaje bayesiano para analizar relaciones complejas entre factores socioeconómicos, epidemiológicos y logísticos que afectan la demanda farmacéutica.

Comparar el rendimiento entre modelos univariantes y multivariantes.

Se recomienda evaluar modelos de predicción multivariante que consideren correlaciones entre diferentes medicamentos o categorías terapéuticas, especialmente en casos de consumo simultáneo o sustitución terapéutica.

Evaluar el impacto económico y logístico de la implementación de modelos predictivos.

Además del desempeño estadístico, se debe analizar el beneficio económico real obtenido al aplicar modelos de inteligencia artificial en la gestión hospitalaria, midiendo indicadores como reducción de desabastecimientos, disminución de pérdidas y optimización de compras.

VIII. REFERENCIAS BIBLIOGRAFICAS

- Aguirre, S. A. (2022). *A model for predicting demand in pharmacies*. 1–13.
- Andrewson, S., & Al-Mubarak, F. (2024). *Machine Learning Algorithms for Demand Forecasting in Warehousing*. February. <https://www.researchgate.net/publication/385434015>
- Bai, S., Kolter, J. Z., & Koltun, V. (2018). *An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling*. <http://arxiv.org/abs/1803.01271>
- Baumann, E., Buba, H., Cox, T., & Hsu, C. (2023). An Introductory Approach to Time-Series Data Preparation and Analysis. *Proceedings of the Annual Conference of the Prognostics and Health Management Society, PHM*, 15(1), 1–10. <https://doi.org/10.36001/phmconf.2023.v15i1.3561>
- Bayani, S. V., Mohamed, I. A., & Venkatasubbu, S. (2024). Robustness and Interpretability of Machine Learning Models in Financial Forecasting. *European Journal of Technology*, 8(2), 54–66. <https://doi.org/10.47672/ejt.2005>
- Bilal, A. I., Bititci, U. S., & Fenta, T. G. (2024). Challenges and the Way Forward in Demand-Forecasting Practices within the Ethiopian Public Pharmaceutical Supply Chain. *Pharmacy*, 12(3), 86. <https://doi.org/10.3390/pharmacy12030086>
- Braiek, H. Ben, & Khomh, F. (2024). Machine learning robustness: a primer. *Trustworthy Ai in Medical Imaging*, 37–71. <https://doi.org/10.1016/B978-0-44-323761-4.00012-2>
- Cohen, J. (1988). *Statistical Power Analysis for the Behavioral Sciences* (segunda). https://utstat.toronto.edu/~brunner/oldclass/378f16/readings/CohenPower.pdf?utm_source=chatgpt.com
- Correa, D. (2020). *Desarrollo de un modelo de estimación de demanda de medicamentos para una empresa farmacéutica* (pp. 1–98).
- Díaz, A. (2023). *Predicción de la demanda de medicamentos en clínica privada*. 6. https://tesis.usat.edu.pe/bitstream/20.500.12423/3668/1/TL_DiazAnchayOscarFernando.pdf?

- DIGEMID. (2002). Selección de Medicamentos Esenciales. In *Perspectivas políticas sobre medicamentos de la OMS*. <http://apps.who.int/medicinedocs/pdf/h3006s/h3006s.pdf>
- Flores, T. G. A., & Montalvo, C. J. S. (2024). *Machine learning para la predicción en la gestión de inventario dirigida a PYMES de venta de productos tecnológicos*.
- Fourkiotis, K. P., & Tsadiras, A. (2024). Applying Machine Learning and Statistical Forecasting Methods for Enhancing Pharmaceutical Sales Predictions. *Forecasting*, 6(1), 170–186. <https://doi.org/10.3390/forecast6010010>
- Hernández-Sampieri, R., Fernández, C., & Baptista, P. (2002). Metodología de la investigación: Las rutas cuantitativa, cualitativa y mixta (6.^a ed.). (McGraw-Hil).
- Hewamalage, H., Ackermann, K., & Bergmeir, C. (2023). Forecast evaluation for data scientists: common pitfalls and best practices. *Data Mining and Knowledge Discovery*, 37(2), 788–832. <https://doi.org/10.1007/s10618-022-00894-5>
- Hewamalage, H., Bergmeir, C., & Bandara, K. (2021). Recurrent Neural Networks for Time Series Forecasting: Current status and future directions. *International Journal of Forecasting*, 37(1), 388–427. <https://doi.org/10.1016/j.ijforecast.2020.06.008>
- Hu, A., Stockman, S., Wu, X., Wood, R., Zhi, B., & Chén, O. Y. (2024). *Enhancing Uncertain Demand Prediction in Hospitals Using Simple and Advanced Machine Learning*. April, 1–16. <http://arxiv.org/abs/2404.18670>
- Huanco, R. F. (2023). *MODELO DE IDENTIFICACIÓN DE COVID 19 USANDO TÉCNICAS DE DEEP LEARNING A PARTIR DE IMÁGENES DE RAYOS X DE TORAX DE LOS PULMONES DE LOS PACIENTES*.
- Joshi, S. V., B., P. S., N., P. S., Sushma, B., Pramod, K. N., & Rathore, Y. K. (2025). *Machine Learning Applications in Hospital Pharmacy for Predicting Drug Shortages and Supply Chain Optimization* (pp. 2739–2745). Research Journal of Pharmacy and Technology. <https://doi.org/10.52711/0974-360X.2025.00393>

- Kong, X., Chen, Z., Liu, W., Ning, K., Zhang, L., Muhammad Marier, S., Liu, Y., Chen, Y., & Xia, F. (2025). Deep learning for time series forecasting: a survey. In *International Journal of Machine Learning and Cybernetics* (Vol. 16, Issues 7–8). Springer Berlin Heidelberg. <https://doi.org/10.1007/s13042-025-02560-w>
- Kumar, V. R., Waghmare, P., Bukya, S., Depuru, B. K., & Kaliamoorthy, I. (2023). Forecasting Drug Demand for Optimal Medical Inventory Management. *A Data-Driven Approach with Advanced Machine Learning Techniques*, 8(9), 221–229.
- Lim, B., & Zohren, S. (2021). Time-series forecasting with deep learning: A survey. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 379(2194). <https://doi.org/10.1098/rsta.2020.0209>
- Manrique, D. S. (2025). *Machine learning para la gestión de inventarios de medicamentos en el Hospital José Agurto Tello de Chosica*.
- Martinez, L. J. A. (2023). *Aplicación web basada en machine learning para predecir la demanda de productos en la empresa Multiservicios Chino Kam*. 1–91. <http://tesis.usat.edu.pe/handle/20.500.12423/6817>
- Merkuryeva, G., Valberga, A., & Smirnov, A. (2019). Demand forecasting in pharmaceutical supply chains: A case study. *Procedia Computer Science*, 149, 3–10. <https://doi.org/10.1016/j.procs.2019.01.100>
- Mfizi, E., Niragire, F., Bizimana, T., & Mukanyangezi, M. F. (2023). Analysis of pharmaceutical inventory management based on ABC-VEN analysis in Rwanda: a case study of Nyamagabe district. *Journal of Pharmaceutical Policy and Practice*, 16(1), 1–6. <https://doi.org/10.1186/s40545-023-00540-5>
- MINSA. (2023). *Resolución Ministerial N.º 633-2023-MINSA: Aprueban el Petitorio Nacional Único de Medicamentos Esenciales para el Sector Salud*. <https://www.gob.pe/institucion/minsa/normas-legales/4488629-633-2023-minsa>
- Mistry, H. (2025). Mastering Model Selection for AI/ML Models. *European Journal of Computer Science and Information Technology*, 13(14), 55–67. <https://doi.org/10.37745/ejcsit.2013/vol13n145567>
- Mousa, B. A., & Al-Khateeb, B. (2023). Predicting medicine demand using

- deep learning techniques: A review. *Journal of Intelligent Systems*, 32(1). <https://doi.org/10.1515/jisys-2022-0297>
- Murdoch, W. J., Singh, C., Kumbier, K., Abbasi-Asl, R., & Yu, B. (2019). Definitions, methods, and applications in interpretable machine learning. *Proceedings of the National Academy of Sciences of the United States of America*, 116(44), 22071–22080. <https://doi.org/10.1073/pnas.1900654116>
- Ndakondja, T., & Dongo, A. (2024). Assessing the impact of artificial intelligence and machine learning on forecasting medication demand and supply in public pharmaceutical systems: A systematic review. *GSC Biological and Pharmaceutical Sciences*, 26(2), 140–150. <https://doi.org/10.30574/gscbps.2024.26.2.0071>
- Nikmah, I., Ula, R., Ibrnas, R., Nurfadilah, K., Nawawi, M. I., Asfar, A., Nurhafifah, N., Patriardian, F., Hidayati, A., Alzahira, R. R., Tasyant, D. J., Anwar, S., Wijaya, A. R., Hirzi, R. H., Hidayaturrohman, U., Kertanah, K., Amaly, M. H., Satriawan, R., Sulthan, N., ... Anagreh, S. (2023). RANDOM FORESTS. *Majalah Ilmiah Matematika Dan Statistika*, 23(1), 1–33.
- OMS. (2012). *Página 1: Perspectivas políticas sobre medicamentos de la OMS-Selección de medicamentos esenciales*. 4–4.
- OMS. (2023). The selection and use of essential medicines. *The World Health Organization*, 985.
- OMS. (2025). *The selection and use of essential medicines , 2025 Model List of Essential Medicines 24th list*.
- Poveda, A., Beltrán, G., & Poveda, G. (2023). *Inteligencia Artificial y Programación Lógica: El Futuro*. 113–114. <https://www.dongee.com/tutoriales/inteligencia-artificial-como-aplicacion-de-la-programacion-logica/>
- Raval, R. (2021). *STATE-OF-THE-ART DEEP LEARNING FOR MULTI-PRODUCT INTERMITTENT TIME SERIES FORECASTING*. 4(1), 6.
- Richards, K., Goodall, G., Zastavker, Y., & Kajfez, R. (2023). Secondary Data Analysis as a Research and a Training Tool: First-Year Engineering Experiences. *Proceedings - Frontiers in Education Conference, FIE*. <https://doi.org/10.1109/FIE58773.2023.10343088>

- Rivero Albarran, D., Arciniegas Aguirre, S., & Fernández Badillo, M. (2022). Un modelo para predecir la demanda en farmacias. *Redmarka. Revista de Marketing Aplicado*, 26(1), 1–14. <https://doi.org/10.17979/redma.2022.26.1.9007>
- Rivero, D., Arciniegas, S., & Fernández, M. (2022). Un modelo para predecir la demanda en farmacias. *Redmarka. Revista de Marketing Aplicado*, 26(1), 1–14. <https://doi.org/10.17979/redma.2022.26.1.9007>
- Smola, A., & Olkorf, R. (1996). A tutorial on support vector regression. *Molecular Endocrinology*, 10(7), 813–825. <https://doi.org/10.1210/me.10.7.813>
- Widiatmika, K. P. (2015). Tecnicas de investigacion. In *Etika Jurnalisme Pada Koran Kuning : Sebuah Studi Mengenai Koran Lampu Hijau* (Vol. 16, Issue 2).

ANEXOS

Anexo 1: Matriz de consistencia

Predicción de la demanda de medicamentos esenciales mediante modelos de aprendizaje automático en farmacia hospitalaria Puno 2025

| Problema | Objetivos | Hipótesis | Variable(s) y dimensiones | Metodología |
|--|---|--|---|--|
| <p>¿Cómo predecir la demanda de medicamentos esenciales en farmacia hospitalaria mediante modelos de aprendizaje automático?</p> | <p>General: Predecir la demanda de medicamentos esenciales en farmacia hospitalaria, utilizando modelos de aprendizaje automático.</p> | <p>Hipótesis General los modelos de aprendizaje automático permiten predecir significativamente la demanda de medicamentos esenciales en farmacia hospitalaria.</p> | <p>Variable dependiente: Demanda de medicamentos .</p> <p>-Dimensiones Volumen de consumo Frecuencia de solicitud periodicidad</p> | <p>Enfoque: cuantitativo, aplicado.</p> <p>Tipo: No experimental, retrospectivo.</p> <p>Nivel: Predictivo, explicativo</p> |
| <p>¿Qué variables y datos históricos inciden significativamente en la demanda de medicamentos esenciales en la farmacia hospitalaria?</p> <p>¿Qué modelos de aprendizaje automático son más adecuados para predecir la demanda de medicamentos esenciales en función de los datos disponibles</p> <p>¿Qué tan precisos y confiables son los modelos predictivos implementados para estimar la demanda de medicamentos esenciales en farmacia hospitalaria?</p> | <p>Específicos: Identificar variables importantes que influyen en la demanda de medicamentos esenciales a partir de datos históricos de la farmacia hospitalaria Puno.</p> <p>Implementar modelos de aprendizaje automático para predecir la demanda de medicamentos esenciales en función de los datos preprocesados.</p> <p>Evaluar el desempeño de los modelos predictivos utilizando métricas estadísticas para elegir el modelo más adecuado.</p> | <p>Específicos: Existen variables significativas en los datos históricos que influyen en la demanda de medicamentos esenciales.</p> <p>Los modelos de aprendizaje automático implementados presentara un desempeño predictivo adecuado para la demanda de medicamentos esenciales con los datos procesados, en relación a los métodos tradicionales.</p> <p>Existe al menos un modelo de aprendizaje automático que presenta un desempeño adecuado en la predicción de demanda real de medicamentos esenciales.</p> | <p>Variable independiente: Modelos de aprendizaje automático.</p> <p>-Dimensiones Algoritmo utilizado precisión de modelo Variables predictoras</p> | <p>Diseño: No experimental, longitudinal, retrospectivo.</p> <p>Población y muestra: Datos históricos de consumo 2022 – 2024 y totalidad de registros mensuales de 3 años</p> <p>Técnicas e instrumentos: Análisis de registros, preprocesamiento y entrenamiento de modelos con Python.</p> <p>Técnicas de análisis: Análisis exploratorio, Entrenamiento y validación de modelos, métricas MAE, RMSE y sMAPE</p> |

Anexo 2: Instrumento de recolección de datos

Técnica: No se utiliza encuesta ni entrevista, sino técnicas cuantitativas basadas en análisis documental y procesamiento computacional de datos estructurados.

Se aplican técnicas de análisis documental, minería de datos y procesamiento computacional de información estructurada, utilizando métodos de aprendizaje automático y análisis estadístico inferencial.

La técnica principal consiste en el procesamiento automático de datos históricos provenientes del sistema de la farmacia hospitalaria, con el objetivo de modelar y predecir la demanda de medicamentos esenciales.

Instrumento: Se desarrollarán scripts en Python, formularios para recolección de datos estructurados desde base de datos (Excel/CSV) y fichas de extracción en base a registros de la farmacia hospitalaria.

Asimismo, se desarrollaron:

- Formularios electrónicos de recolección estructurada, que permiten la integración de datos desde distintos formatos (Excel, CSV, SQL).
- Fichas de extracción y control de calidad de datos, elaboradas en base a los registros históricos de dispensación y stock de medicamentos del sistema de información hospitalario (ESSI).
- Módulos de validación estadística, que incluyen pruebas de significancia (ANOVA, Friedman) y métricas de desempeño (MAE, RMSE, sMAPE) para evaluar la eficacia de los modelos predictivos.

Escala de medición: Combinación de datos cuantitativos y cualitativos (según la naturaleza del modelo y variables utilizadas).

Se emplea una escala mixta de medición que combina:

- Datos cuantitativos continuos, correspondientes a las variables de salida de los modelos (MAE, RMSE, sMAPE, error porcentual, valores predichos y observados).
- Datos cualitativos derivados del proceso analítico, tales como el tipo de modelo, estructura algorítmica, grado de estabilidad temporal y nivel de interpretabilidad del modelo.

Anexo 3: Fragmento de la base de Datos de consumo de medicamentos de la Farmacia de los años 2022 al 2024 obtenidas del sistema ESSI.

| FECHA_DOC_MOV | NUM_RECETA | COD_MEDIC | DESC_MEDICAMENTO | CANTIDAD |
|---------------------|------------|-----------|--|----------|
| 2022-04-23 00:00:00 | 652496 | 10200012 | ACETILCISTEINA 200 MG | 15 |
| 2022-04-28 00:00:00 | 655645 | 11050068 | CLORURO DE SODIO Ó SUERO FISIOLÓGICO 0.9 | 3 |
| 2022-04-22 00:00:00 | 651516 | 10050018 | IBUPROFENO 400 MG | 10 |
| 2022-04-01 00:00:00 | 598556 | 11100050 | TIAMINA CLORHIDRATO 100 MG | 30 |
| 2022-04-07 00:00:00 | 622131 | 11100050 | TIAMINA CLORHIDRATO 100 MG | 30 |
| 2022-04-20 00:00:00 | 611819 | 11100050 | TIAMINA CLORHIDRATO 100 MG | 30 |
| 2022-04-20 00:00:00 | 630349 | 11100050 | TIAMINA CLORHIDRATO 100 MG | 30 |
| 2022-04-23 00:00:00 | 632986 | 11100050 | TIAMINA CLORHIDRATO 100 MG | 30 |
| 2022-04-07 00:00:00 | 642277 | 10050023 | METAMIZOL SÓDICO 1 G/2 ML | 1 |
| 2022-04-21 00:00:00 | 650599 | 10250014 | AMOXICILINA/ACIDO CLAVULANICO 500 MG/12 | 10 |
| 2022-04-01 00:00:00 | 637700 | 10050035 | PARACETAMOL 500 MG | 20 |
| 2022-04-04 00:00:00 | 639023 | 10050035 | PARACETAMOL 500 MG | 20 |
| 2022-04-06 00:00:00 | 640929 | 10050035 | PARACETAMOL 500 MG | 30 |
| 2022-04-11 00:00:00 | 644167 | 10050035 | PARACETAMOL 500 MG | 20 |
| 2022-04-13 00:00:00 | 646391 | 10050035 | PARACETAMOL 500 MG | 12 |
| 2022-04-19 00:00:00 | 649379 | 10050035 | PARACETAMOL 500 MG | 20 |
| 2022-04-19 00:00:00 | 649316 | 10050035 | PARACETAMOL 500 MG | 15 |
| 2022-04-02 00:00:00 | 638434 | 10850023 | GABAPENTINA 300 MG | 4 |
| 2022-04-05 00:00:00 | 639151 | 10850023 | GABAPENTINA 300 MG | 30 |
| 2022-04-05 00:00:00 | 639124 | 10850023 | GABAPENTINA 300 MG | 90 |
| 2022-04-08 00:00:00 | 642563 | 10850023 | GABAPENTINA 300 MG | 30 |
| 2022-04-11 00:00:00 | 644167 | 10850023 | GABAPENTINA 300 MG | 30 |
| 2022-04-13 00:00:00 | 646401 | 10850023 | GABAPENTINA 300 MG | 30 |
| 2022-04-20 00:00:00 | 650179 | 10850023 | GABAPENTINA 300 MG | 90 |
| 2022-04-27 00:00:00 | 655393 | 10850023 | GABAPENTINA 300 MG | 30 |
| 2022-04-29 00:00:00 | 657113 | 10850023 | GABAPENTINA 300 MG | 30 |
| 2022-04-12 00:00:00 | 645040 | 11050068 | CLORURO DE SODIO Ó SUERO FISIOLÓGICO 0.9 | 5 |
| 2022-04-04 00:00:00 | 639181 | 11100037 | PIRIDOXINA CLORHIDRATO 50 MG | 30 |
| 2022-04-04 00:00:00 | 639042 | 11100037 | PIRIDOXINA CLORHIDRATO 50 MG | 30 |
| 2022-04-08 00:00:00 | 642723 | 11100037 | PIRIDOXINA CLORHIDRATO 50 MG | 30 |
| 2022-04-19 00:00:00 | 649363 | 11100037 | PIRIDOXINA CLORHIDRATO 50 MG | 30 |
| 2022-04-27 00:00:00 | 655361 | 11100037 | PIRIDOXINA CLORHIDRATO 50 MG | 30 |
| 2022-04-04 00:00:00 | 639070 | 11100050 | TIAMINA CLORHIDRATO 100 MG | 30 |
| 2022-04-13 00:00:00 | 646367 | 10050086 | ORFENADRINA CITRATO 100 MG (LIBERACIÓN P | 15 |
| 2022-04-25 00:00:00 | 614459 | 10050035 | PARACETAMOL 500 MG | 30 |
| 2022-04-12 00:00:00 | 607290 | 10050044 | TRAMADOL (CLORHIDRATO) 50 MG | 15 |
| 2022-04-12 00:00:00 | 607290 | 10850023 | GABAPENTINA 300 MG | 60 |
| 2022-04-19 00:00:00 | 647395 | 10050086 | ORFENADRINA CITRATO 100 MG (LIBERACIÓN P | 14 |
| 2022-04-18 00:00:00 | 647389 | 10450068 | OMEPRAZOL 20 MG DE LIBERACIÓN RETARDADA/ | 30 |
| 2022-04-19 00:00:00 | 647385 | 10450068 | OMEPRAZOL 20 MG DE LIBERACIÓN RETARDADA/ | 10 |
| 2022-04-19 00:00:00 | 647395 | 10450068 | OMEPRAZOL 20 MG DE LIBERACIÓN RETARDADA/ | 10 |
| 2022-04-08 00:00:00 | 643211 | 10050018 | IBUPROFENO 400 MG | 10 |
| 2022-04-10 00:00:00 | 643839 | 10050018 | IBUPROFENO 400 MG | 10 |
| 2022-04-08 00:00:00 | 643134 | 10050035 | PARACETAMOL 500 MG | 10 |

Anexo 4: Informe de Turnitin al 28 % de similitud

MAMANI APARICIO MARITZA.docx

- Grupo Titulacion - Martin Velasquez
- Grupo Titulacion - Martin Velasquez
- Universidad Autónoma de Ica

Detalles del documento

Identificador de la entrega

trn:oid::3117:541381180

Fecha de entrega

16 dic 2025, 5:22 p.m. GMT-5

Fecha de descarga

18 dic 2025, 10:43 a.m. GMT-5

Nombre del archivo

MAMANI APARICIO MARITZA.docx

Tamaño del archivo

2.5 MB

123 páginas

21.826 palabras

131.991 caracteres



Página 2 de 127 - Descripción general de integridad

Identificador de la entrega trn:oid::3117:541381180

3% Similitud general

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

Filtrado desde el informe

- Bibliografía
- Coincidencias menores (menos de 15 palabras)

Fuentes principales

- 3% Fuentes de Internet
- 1% Publicaciones
- 2% Trabajos entregados (trabajos del estudiante)

Marcas de integridad

N.º de alertas de integridad para revisión

No se han detectado manipulaciones de texto sospechosas.

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

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

Anexo 5: Evidencia Fotográfica

Procesamiento del modelo GRU

```
Python 3.13 (64-bit) x + v
...     except Exception as e:
...         log(f"✘ Error en {cod}: {repr(e)}")c,"status":f"error: {repr(e)}"})
...         resumen.append({"cod":cod,"desc":desc,"status":f"error: {repr(e)}"})
...
...     # Actualiza/crea resumen global
...     try:
...         prev = pd.read_excel(RESUMEN) if os.path.exists(RESUMEN) else pd.DataFrame(columns=["cod", "desc", "status", "\
c
arpeta"])
...         out = pd.concat([prev, pd.DataFrame(resumen)], ignore_index=True)
...         out = pd.concat([prev, pd.DataFrame(resumen)], ignore_index=True, index=False)
...         out.drop_duplicates(subset=["cod"], keep="last").to_excel(RESUMEN, index=False)
...     except Exception:
...         pd.DataFrame(resumen).to_excel(RESUMEN, index=False)
...
...     log("\n=== Reanudación finalizada ===")
...     log(f"Resumen actualizado: {RESUMEN}")
...
...     main()
2025-11-09 19:15:29.160344: I tensorflow/core/util/port.cc:153] oneDNN custom operations are on. You may see slightly di
fferent numerical results due to floating-point round-off errors from different computation orders. To turn them off, se
t the environment variable 'TF_ENABLE_ONEDNN_OPTS=0'.
2025-11-09 19:15:34.416839: I tensorflow/core/util/port.cc:153] oneDNN custom operations are on. You may see slightly di
fferent numerical results due to floating-point round-off errors from different computation orders. To turn them off, se
t the environment variable 'TF_ENABLE_ONEDNN_OPTS=0'.
↳ COD 10050002: ya procesado, se omite.
↳ COD 10050044: ya procesado, se omite.
↳ COD 10250031: ya procesado, se omite.
↳ COD 10400016: ya procesado, se omite.
↳ COD 10400021: ya procesado, se omite.
↳ COD 10400043: ya procesado, se omite.
↳ COD 10400064: ya procesado, se omite.
↳ COD 10400096: ya procesado, se omite.
↳ COD 10400119: ya procesado, se omite.
↳ COD 10750011: ya procesado, se omite.
↳ COD 11100037: ya procesado, se omite.

=== Reanudación finalizada ===
Resumen actualizado: D:\PARAGRU\RESUMEN_BATCH_GRU.xlsx
>>> |
```

Resultado del procesamiento del modelo GRU

| | A | B | C | D | E | F | G |
|---|-----------|--------|---------|----------|------------|------------|------------|
| | COD_MEDIC | window | horizon | features | mae | rmse | mape |
| | 10050023 | 90 | 7 | 10 | 63.9691818 | 84.4128264 | 40.17658 |
| | 11050068 | 90 | 7 | 10 | 67.5823558 | 93.9502328 | 37.8260848 |
| | 10250031 | 90 | 7 | 10 | 110.698507 | 145.833807 | 2.9044E+10 |
| | 10450015 | 90 | 7 | 10 | 114.262854 | 146.912303 | 1.6729E+10 |
| | 10250014 | 90 | 7 | 10 | 113.474639 | 150.153177 | 2.5428E+10 |
| | 10600013 | 90 | 7 | 10 | 24.929584 | 151.594167 | 6.6265E+10 |
| | 10150019 | 90 | 7 | 10 | 129.43545 | 172.593152 | 1.711E+10 |
| | 10400119 | 90 | 7 | 10 | 136.721237 | 194.080899 | 8.4429E+10 |
| 0 | 10400021 | 90 | 7 | 10 | 140.982888 | 208.155145 | 9.1682E+10 |
| 1 | 10050044 | 90 | 7 | 10 | 158.897114 | 220.601445 | 3.0784E+10 |
| 2 | 10400064 | 90 | 7 | 10 | 125.634305 | 222.691687 | 5.2463E+10 |
| 3 | 10050002 | 90 | 7 | 10 | 138.636822 | 226.267132 | 7.4022E+10 |
| 4 | 10400016 | 90 | 7 | 10 | 165.791141 | 262.670399 | 1.5854E+11 |
| 5 | 10050086 | 90 | 7 | 10 | 194.492875 | 266.224636 | 5.3616E+10 |
| 5 | 10400096 | 90 | 7 | 10 | 223.957234 | 330.931085 | 3.6316E+10 |
| 7 | 10050018 | 90 | 7 | 10 | 263.506976 | 346.632702 | 5.4279E+10 |
| 3 | 11100037 | 90 | 7 | 10 | 253.560504 | 360.826472 | 1.4044E+11 |
| 3 | 10400043 | 90 | 7 | 10 | 246.554569 | 365.586151 | 1.1251E+11 |
| 0 | 10750011 | 90 | 7 | 10 | 235.014174 | 365.760858 | 1.6518E+11 |
| 1 | 10450068 | 90 | 7 | 10 | 329.217579 | 479.960424 | 6.3409E+10 |
| 2 | 10200012 | 90 | 7 | 10 | 387.377459 | 500.191325 | 4.0836E+10 |
| 3 | 10400118 | 90 | 7 | 10 | 331.505982 | 503.977344 | 2.0914E+11 |
| 4 | 10750021 | 90 | 7 | 10 | 329.863433 | 550.494219 | 2.0454E+11 |
| 5 | 11100050 | 90 | 7 | 10 | 374.014154 | 615.481651 | 2.0009E+11 |
| 5 | 10850023 | 90 | 7 | 10 | 395.974938 | 670.575755 | 2.6728E+11 |
| 7 | 10050035 | 90 | 7 | 10 | 556.486119 | 737.21483 | 9.8879E+10 |
| 3 | 10700001 | 90 | 7 | 10 | 497.316813 | 938.143867 | 3.2261E+11 |
| 3 | 10400109 | 90 | 7 | 10 | 676.277622 | 938.46861 | 2.1244E+11 |
| 0 | 10400092 | 90 | 7 | 10 | 597.942705 | 1389.39844 | 1.0006E+11 |
| 1 | 10750025 | 90 | 7 | 10 | 949.598121 | 1719.08643 | 6.6872E+11 |
| 2 | | | | | | | |
| 3 | | | | | | | |

< >
+
metrics_gru_por_sku