Universidad Nacional Mayor de San Marcos
Universidad del Perú. Decana de América
Facultad de Química e Ingeniería Química
Escuela Profesional de Ingeniería Química

Evaluación de la correlación entre variables operativas sobre la concentración de manganeso en electroobtención de zinc mediante modelamiento con Machine Learning

TESIS

Para optar el Título Profesional de Ingeniera Química

AUTOR

Adriana Pilar CARDENAS CARBAJAL

ASESOR

Dr. Eder Clidio VICUÑA GALINDO

Lima, Perú

2026



Reconocimiento - No Comercial - Compartir Igual - Sin restricciones adicionales

https://creativecommons.org/licenses/by-nc-sa/4.0/

Usted puede distribuir, remezclar, retocar, y crear a partir del documento original de modo no comercial, siempre y cuando se dé crédito al autor del documento y se licencien las nuevas creaciones bajo las mismas condiciones. No se permite aplicar términos legales o medidas tecnológicas que restrinjan legalmente a otros a hacer cualquier cosa que permita esta licencia.

Referencia bibliográfica

Cardenas, A. (2026). *Evaluación de la correlación entre variables operativas sobre la concentración de manganeso en electroobtención de zinc mediante modelamiento con Machine Learning*. [Tesis de pregrado, Universidad Nacional Mayor de San Marcos, Facultad de Química e Ingeniería Química, Escuela Profesional de Ingeniería Química]. Repositorio institucional Cybertesis UNMSM.

Metadatos complementarios

Datos de autor	
Nombres y apellidos	Adriana Pilar Cardenas Carbajal
Tipo de documento de identidad	DNI
Número de documento de identidad	75882338
URL de ORCID	
Datos de asesor	
Nombres y apellidos	Eder Clidio Vicuña Galindo
Tipo de documento de identidad	DNI
Número de documento de identidad	09069542
URL de ORCID	https://orcid.org/0000-0003-4980-0490
Datos del jurado	
Presidente del jurado	
Nombres y apellidos	María Verónica Carranza Oropeza
Tipo de documento	DNI
Número de documento de identidad	10727849
Miembro del jurado 1	
Nombres y apellidos	José Fernandes de Oliveira Ugarte
Tipo de documento	DNI
Número de documento de identidad	48755147
Datos de investigación	
Línea de investigación	No aplica

Grupo de investigación	No aplica
Agencia de financiamiento	Sin financiamiento
Ubicación geográfica de la investigación	Edificio: Nexa Resources Cajamarquilla País: Perú Departamento: Lima Provincia: Lima Distrito: Lurigancho-Chosica Calle: Carretera Central km 9.5 Desvío a Huachipa Latitud: -11.96906 Longitud: -76.88569
Año o rango de años en que se realizó la investigación	Enero 2026 – Junio 2026
URL de disciplinas OCDE	2.04.02 – Ingeniería de procesos https://purl.org/pe-repo/ocde/ford#2.04.02

UNIVERSIDAD NACIONAL MAYOR DE SAN MARCOS
(Universidad del Perú, DECANA DE AMÉRICA)
FACULTAD DE QUÍMICA E INGENIERÍA QUÍMICA
Central Telefónica: 619-7000

ESCUELA PROFESIONAL DE INGENIERÍA QUÍMICA

ACTA DE SUSTENTACIÓN DE TESIS

Los suscritos miembros del Jurado Calificador nombrados por el señor Presidente del Comité Ejecutivo del "VI Programa Ciclo Taller de Titulación Profesional por la Modalidad de Tesis y Trabajo de Suficiencia Profesional para la Facultad de Química e Ingeniería Química 2026-I", bajo la Presidencia de la **Dra. María Verónica Carranza Oropeza;** el **Dr. José Fernandes De Oliveira Ugarte (Miembro)** y el **Dr. Eder Clidio Vicuña Galindo (Asesor);** habiendo presentado para tal efecto la **TESIS,** titulada **"Evaluación de la correlación entre variables operativas sobre la concentración de manganeso en electroobtención de zinc mediante modelamiento con Machine Learning",** después de **SUSTENTADA Y APROBADA** la tesis elaborada por la bachiller en Ingeniería Química: **CARDENAS CARBAJAL, ADRIANA PILAR;** para optar el **TÍTULO PROFESIONAL** de **INGENIERA QUÍMICA,** acordaron calificarla con la **NOTA** de:

Diecisiete (17)

(LETRAS) (NÚMEROS)

Ciudad Universitaria, 20 de junio del 2026

Dra. MARÍA VERÓNICA CARRANZA OROPEZA
Presidenta

Dr. JOSÉ FERNANDES DE OLIVEIRA UGARTE
Miembro

Dr. EDER CLIDIO VICUÑA GALINDO
Asesor

Dr. EDER CLIDIO VICUÑA GALINDO
Director de la E.P. de Ingeniería Química

UNIVERSIDAD NACIONAL MAYOR DE SAN MARCOS • ESCUELA ACADÉMICO PROFESIONAL DE INGENIERÍA QUÍMICA • Facultad de Química e Ingeniería Química

UNIVERSIDAD NACIONAL MAYOR DE
SAN MARCOS
Universidad del Perú, Decana de América

CERTIFICADO DE SIMILITUD

Yo, **Eder Clidio Vicuña Galindo** en mi condición de asesor con el Dictamen N°000008-2026-EPIQ-FQIQ/UNMSM de la Tesis, cuyo título es **Evaluación de la correlación entre variables operativas sobre la concentración de manganeso en electroobtención de zinc mediante modelamiento con Machine Learning,** presentado por la bachiller: **CARDENAS CARBAJAL, ADRIANA PILAR,** para optar el título Profesional de Ingeniera Química.

CERTIFICO que se ha cumplido con lo establecido en la Directiva de Originalidad y de Similitud de Trabajos Académicos, de Investigación y Producción Intelectual. Según la revisión, análisis y evaluación mediante la plataforma PIDS, el documento evaluado cuenta con el 10% de similitud, nivel **PERMITIDO** para continuar con los trámites correspondientes y para su **publicación en el repositorio institucional.**

Se emite el presente certificado en cumplimiento de lo establecido en las normas vigentes, como uno de los requisitos para la obtención del título Profesional de Ingeniera Química.

Lima, 13 de julio del 2026



Firma del Asesor

Nombres y Apellidos del asesor Eder Clidio Vicuña Galindo

DNI: 09069542

Agradecimientos

A mi asesor, Dr. Eder Clidio Vicuña Galindo, por su guía, paciencia y rigurosidad científica a lo largo de este proceso. Sus consejos y constante exigencia no solo hicieron posible la culminación de esta tesis, sino que contribuyó a mi formación profesional.

A Eder Francisco Vicuña Ara por su valioso aporte y orientación técnica; su disposición y eficiencia fueron fundamentales para el desarrollo de esta tesis.

Dedicatoria

A mi familia por su apoyo durante todo el proceso de mi carrera y la realización de esta tesis. Gracias por sus palabras de aliento en los momentos de cansancio y por creer en mí
cuando yo dudaba. Este logro es tan suyo como mío.

A mi mejor amiga y hermana de vida, Milan Alexandra Urcia Pérez. Gracias por tu apoyo incondicional en los momentos más críticos y no solo de este proyecto. Tu generosidad y fe en mí fueron el impulso que necesité para no rendirme. Este logro también es tuyo.

ÍNDICE DE CONTENIDO

I. INTRODUCCIÓN 9
1.1. Planteamiento del problema 9
1.2. Formulación del problema 10
1.3. Objetivos 10
1.4. Importancia y alcance de la investigación 11
1.5. Limitaciones de la investigación 12

II. REVISIÓN DE LA LITERATURA 14
2.1. Marco teórico 14
2.2. Antecedentes del estudio 18
2.3. Bases teóricas 20
2.4. Definición de términos 23

III. HIPÓTESIS Y VARIABLES 26
3.1. Hipótesis 26
3.2. Variables 26
3.3. Operacionalización de variables 27

IV. MATERIALES Y MÉTODOS 28
4.1. Área de estudio 28
4.2. Diseño de investigación 28
4.3. Procedimientos, Técnicas e instrumentos de recolección de información ... 29

V. RESULTADOS 32
5.1. Presentación y análisis de los resultados 32

VI. DISCUSIÓN DE RESULTADOS 43

VII. CONCLUSIONES Y RECOMENDACIONES 49
7.1. Conclusiones 49
7.2. Recomendaciones 50

VIII. BIBLIOGRAFÍA Y ANEXOS 52

Anexo 1: Proceso resumido de las reacciones de Zn y Mn en el ánodo 56

Anexo 2: Etapa preliminar de entrenamiento de modelamiento en Machine Learning …….. 57

Anexo 3: Código de modelo XGBoost con datos reales 63
Anexo 4: Código de modelo Random Forest con datos reales ……..... 77

ÍNDICE DE TABLAS

Tabla 1. Resumen de antecedentes del estudio..... 19

Tabla 2. Variable operativa independiente..... 27

Tabla 3. Variable operativa dependiente..... 27

Tabla 4. Información para el desarrollo del programa computacional 28

Tabla 5. Resultados por aspecto en XGBoost con datos teóricos 32

Tabla 6. Variables más importantes según modelo XGBoost con datos teóricos..... 34

Tabla 7. Resumen de correlación y promedios de casos extremos bajo y alto con Mn^{2+}_{EA} ….. 36

Tabla 8. Correlaciones con Mn^{2+}_{EA} g/L..... 37

Tabla 9. Correlaciones con Mn Sol Pura..... 37

Tabla 10. Correlaciones de moderada a alta y extremadamente baja..... 38

Tabla 11. Análisis comparativo entre XGBoost y Random Forest con datos reales 39

Tabla 12. Análisis de varianza..... 40

Tabla 13. Intercepto y pendiente de regresión..... 40

Tabla 14. Comparativo de regresión lineal y modelos de Machine Learning..... 41

ÍNDICE DE FIGURAS

Figura 1. Diagrama de flujo del proceso hidrometalúrgico del zinc.....14

Figura 2. Diagrama de flujo del proceso de electrodeposición de zinc con formación de dióxido de manganeso.....16

Figura 3. Diagrama de flujo de etapas sistemáticas del modelamiento en Machine Learning con datos reales 29

Figura 4. Predicciones vs variables operativas – XGBoost 33

Figura 5. Importancia de las variables operativas en la predicción de MnO_2 33

Figura 6. Predicción vs Real – Mn EA (XGBoost con datos reales) 35

Figura 7. Predicción vs Real – Mn EA (Random Forest con datos reales) 35

Figura 8. Matriz de correlación de variables operativas 36

Figura 9. Correlación de variables de entrada vs Mn EA predicho (XGBoost) 38

Figura 10. Correlación de variables de entrada vs Mn EA predicho (Random Forest) … 39

RESUMEN

Esta tesis determinó la influencia de variables operativas sobre la concentración de manganeso en electrolito agotado de la electroobtención de zinc, aplicando modelos de Machine Learning (XGBoost y Random Forest) a 5040 registros reales de la Refinería Cajamarquilla. El análisis exploratorio en los modelos de Machine Learning, reveló que la concentración de manganeso en solución pura (Mn^{2+}) presenta una excelente correlación directamente proporcional (r = 1,00) con la concentración de manganeso en electrolito agotado (Mn^{2+}), mientras que el resto de variables operativas (temperatura de electrolito (°C), acidez (%), densidad (g/L), impurezas (Fe, Cu, Sb mg/L), horas de depósito y peso de depósito (kg)) mostraron correlaciones prácticamente nulas ($|r| \leq 0{,}002$). La regresión lineal estableció, mediante la relación $Mn^{2+}_{EA} = 2{,}8 Mn^{2+}_{SP}$, que el manganeso se comporta como elemento conservativo bajo las condiciones estudiadas; asimismo, las variables operativas de electroobtención de zinc no influyen en el control de la concentración de manganeso en electrolito agotado. Se concluye que durante la electroobtención de zinc no se elimina significativamente el manganeso y la predicción computacional para esta impureza es solo aplicable como monitoreo en esta etapa; siendo esta la razón por la que se recomienda desplazar el esfuerzo de control de la electroobtención hacia la purificación.

Palabras clave: Remoción de manganeso, XGBoost, Random Forest, Predicción computacional.

Abstract

This thesis determined the influence of operating variables on the manganese concentration in spent electrolyte from zinc electrowinning by applying machine learning models (XGBoost and Random Forest) to 5040 real data records from the Cajamarquilla Refinery. Exploratory analysis within the machine learning models revealed that the manganese concentration in the purified solution (Mn^{2+}_{SP}) shows an excellent directly proportional correlation (r = 1.00) with the manganese concentration in the spent electrolyte (Mn^{2+}_{EA}), whereas the remaining operating variables (electrolyte temperature (°C), acidity (%), density (g/L), impurities (Fe, Cu, Sb mg/L), deposition time, and deposit weight (kg)) showed virtually zero correlation ($|r| \leq 0.002$). Linear regression established, via the relationship $Mn^{2+}_{EA} = 2{,}8 Mn^{2+}_{SP}$, that manganese behaves as a conservative element under the conditions studied; furthermore, zinc electrowinning operating variables do not influence the control of manganese concentration in the spent electrolyte. It is concluded that manganese is not significantly removed during zinc electrowinning and that computational prediction for this impurity is applicable only for monitoring purposes at this stage; consequently, it is recommended that control efforts be shifted from the electrowinning stage to the purification stage.

Keywords: Manganese removal, XGBoost, Random Forest, computational prediction

I. INTRODUCCIÓN

1.1. Planteamiento del problema

El proceso de electroobtención de zinc a partir de soluciones acidificadas enfrenta desafíos críticos debido a la presencia de impurezas metálicas (Wark, 1978). Entre estas, el manganeso destaca por su complejidad química; concentraciones elevadas de este generan efectos despolarizantes a través de iones y otros productos oxidados, lo que deriva en la modificación de la morfología superficial del depósito, pérdida de calidad del cátodo y una disminución directa de la eficiencia de corriente (Zhang, 2009).

El control del manganeso en el electrolito circulante es un equilibrio técnico delicado. Según Habashi (1997) y Sinclair (2005), la concentración ideal debe mantenerse bajo los 5 g/L. Menor a 2 g/L provoca que el plomo del ánodo contamine las láminas de zinc. Por el contrario, niveles superiores a 8 g/L incrementan la resistencia eléctrica y saturan el sistema con lodos anódicos, los cuales precipitan al fondo de las celdas y generan cortocircuitos. A pesar de esto, muchas plantas que no logran una remoción eficiente en la etapa de purificación operan en rangos críticos de 8 a 12 g/L, viéndose obligadas a ajustar la acidez libre para intentar mantener la conductividad del sistema (Zhang & Cheng, 2007).

Esta problemática no es aislada; en refinerías de China, los niveles excesivos de manganeso elevan los costos operativos entre un 15% y 20% debido a la limpieza de celdas y pérdida de electrolito (Chen et al., 2019). En Latinoamérica, la situación es similar: plantas en el norte de México reportan niveles superiores a 10 g/L, optando por promover la precipitación de lodos en lugar de la remoción previa (Bautista, 2010).

En el Perú, la industria de recuperación de zinc enfrenta un desafío geológico particular. Los concentrados nacionales son frecuentemente de baja ley, con altas concentraciones de manganeso presentes en minerales como la franklinita ($ZnO\ (Fe, Mn)_2O_3$) (INGEMNET, 2000; Bonilla et al., 2023). Durante el proceso de tostación, el reemplazo de sulfuros por óxidos genera una reducción de la masa total de la calcina, lo que provoca un aumento proporcional en el ensayo de metales no volátiles y, críticamente, el manganeso (Dutta y Chandler, 1980).

En consecuencia, las refinerías peruanas operan con un electrolito cargado de

manganeso que compromete la eficiencia del proceso. Los métodos de control actuales basados en ajustes manuales de acidez y limpiezas correctivas resultan insuficientes y costosos. Existe, por lo tanto, una necesidad tecnológica de explorar si los modelos de

Machine Learning, basados en las variables operativas rutinarias, pueden anticipar el comportamiento de esta impureza. Esta tesis evalúa dicha posibilidad y, en caso de no ser viable, identifica las limitaciones y recomienda líneas futuras.

1.2. Formulación del problema

En esta sección se definen las interrogantes principales y específicas que permitirán evaluar la eficacia del modelo de Machine Learning propuesto para mejorar la remoción de manganeso del electrolito agotado en el proceso de electroobtención.

1.2.1. Determinación del problema

¿Cómo influyen las variables operativas del proceso de electroobtención de zinc sobre la concentración de manganeso en el electrolito agotado, y qué modelo de Machine Learning permite estimar dicha concentración con mayor precisión?

1.2.2. Formulación del problema

- ¿Qué variables operativas presentan mayor influencia sobre la concentración de manganeso en el electrolito agotado durante la electroobtención de zinc?
- ¿Qué modelo de Machine Learning ofrece mejor capacidad predictiva para estimar la concentración de manganeso en el electrolito agotado en electroobtención de zinc?
- ¿Cuáles son las implicancias operativas de utilizar modelos de Machine Learning para el control del proceso de electroobtención de zinc?

1.3. Objetivos

En este apartado se establece el propósito central del estudio y los pasos operativos necesarios para validar el uso de Machine Learning como herramienta para predecir la concentración de manganeso del electrolito agotado en el proceso de electroobtención de zinc.

1.3.1. Objetivo general

Determinar la influencia de las variables operativas sobre la concentración de manganeso en el electrolito agotado en electroobtención de zinc, mediante la

aplicación de Machine Learning.

1.3.2. Objetivos específicos

- Identificar las variables operativas con mayor influencia sobre la concentración de manganeso en el electrolito agotado, a través del análisis de importancia de variables en modelos de Machine Learning.

- Comparar el desempeño predictivo de los modelos de Machine Learning en términos de MAE, R^2 y tiempo de entrenamiento, aplicados a datos de electroobtención de zinc.

- Evaluar las implicancias operativas del empleo de modelos de Machine Learning para la predicción y control de la concentración de manganeso en la electroobtención de zinc.

1.4. Importancia y alcance de la investigación

La relevancia de esta investigación radica en la necesidad crítica de resolver la inestabilidad operativa en el proceso de electroobtención de zinc, la cual se ve gravemente afectada por la complejidad química del manganeso. Como señala Wark (1978), la presencia de impurezas metálicas en soluciones acidificadas representa uno de los desafíos más persistentes para la industria. La importancia técnica de este estudio se justifica en que niveles elevados de manganeso generan efectos despolarizantes que modifican la morfología superficial del depósito, degradan la calidad del cátodo y reducen drásticamente la eficiencia de corriente (Zhang, 2009).

Desde una perspectiva económica y operativa, la investigación es fundamental para mitigar los sobrecostos que enfrentan las refinerías. Mientras que el equilibrio ideal de manganeso debe situarse entre los 2 y 5 g/L para evitar tanto la contaminación por plomo como la saturación por lodos anódicos (Habashi, 1997; Sinclair, 2005), la realidad de muchas plantas es la operación en rangos críticos de 8 a 12 g/L (Zhang & Cheng, 2007). Esta ineficiencia se traduce en un incremento de costos operativos de hasta un 20% (Chen et al., 2019), una situación que afecta directamente a la industria peruana debido a la naturaleza geológica de sus concentrados, ricos en minerales como la franklinita (INGEMNET, 2000; Bonilla et al., 2023). Por tanto, este trabajo se justifica al proponer una transición desde los ajustes manuales y correctivos —insuficientes y costosos— hacia una solución tecnológica de vanguardia basada en

modelos predictivos y algoritmos de Machine Learning, capaces de anticipar y optimizar el control de esta impureza en tiempo real.

El alcance de este estudio se delimita al análisis y optimización del control del manganeso específicamente en la etapa de electroobtención de zinc. El trabajo se enfoca en el comportamiento de esta impureza dentro del electrolito agotado, considerando las condiciones operativas típicas de las refinerías que procesan minerales complejos con altas concentraciones de metales no volátiles, producto de los procesos de tostación (Dutta y Chandler, 1980).

Geográficamente, la investigación toma como referencia la problemática de las plantas de recuperación de zinc en el Perú, las cuales lidian con concentrados de baja ley y niveles de manganeso que superan los límites técnicos recomendados. En cuanto al alcance tecnológico, la investigación se centra en el desarrollo y validación de modelos predictivos mediante herramientas de Machine Learning. El estudio no pretende abarcar la reingeniería física de las etapas de purificación o lixiviación, sino que se restringe a la predicción de variables críticas para estabilizar el sistema y mantener la conductividad a través del ajuste de la acidez libre. El resultado final del alcance será un modelo capaz de prever fluctuaciones en la concentración de manganeso, proporcionando una base técnica para reducir la dependencia de medidas reactivas como la limpieza correctiva de lodos anódicos descrita en casos similares en la región (Bautista, 2010).

1.5. Limitaciones de la investigación

Existe una limitación operativa y química relacionada con la variabilidad de la alimentación. Aunque el modelo busca predecir el comportamiento del manganeso en electrolito agotado a partir de concentrados nacionales, fluctuaciones extremas o no registradas en la etapa de tostación y lixiviación podrían introducir variables externas que el modelo no ha sido entrenado para identificar. Asimismo, debido a que el estudio se centra en el desarrollo de un modelo predictivo, la validación de los resultados se encuentra sujeta a los protocolos de seguridad y disponibilidad de las celdas electrolíticas, lo cual limita la posibilidad de realizar pruebas experimentales.

La base de datos no incluye columnas de fecha u hora de muestreo, lo que impide realizar un análisis de series temporales y evaluar la dependencia del manganeso con el tiempo (rezagos). Esta limitación impide determinar si existe una relación causal temporal entre el Mn en solución pura y el Mn EA, más allá de la

correlación contemporánea observada.

El estudio no incluye mediciones de potencial anódico o estado de los ánodos en la Refinería Cajamarquilla, que podrían ser relevantes para la oxidación del Mn^{2+}. Además, los datos corresponden a un rango operativo específico de esta planta, por lo que los resultados podrían ser diferentes en otros rangos de operación.

II. REVISIÓN DE LITERATURA

2.1. Marco teórico

Se presenta toda la fundamentación necesaria para este trabajo de investigación, empezando por una apreciación general del proceso bajo estudio.

2.1.1. Proceso hidrometalúrgico del zinc

La producción de zinc mediante rutas hidrometalúrgicas es ampliamente utilizada a nivel industrial debido a su elevada eficiencia energética y a su capacidad para tratar concentrados sulfurados complejos. Este proceso comprende, de manera general, las etapas de tostación, lixiviación, purificación de soluciones y electrólisis, siendo esta última la responsable de la obtención del zinc metálico de alta pureza (Habashi, 1997; Davenport et al., 2011). El proceso como tal se presenta en la figura 1.

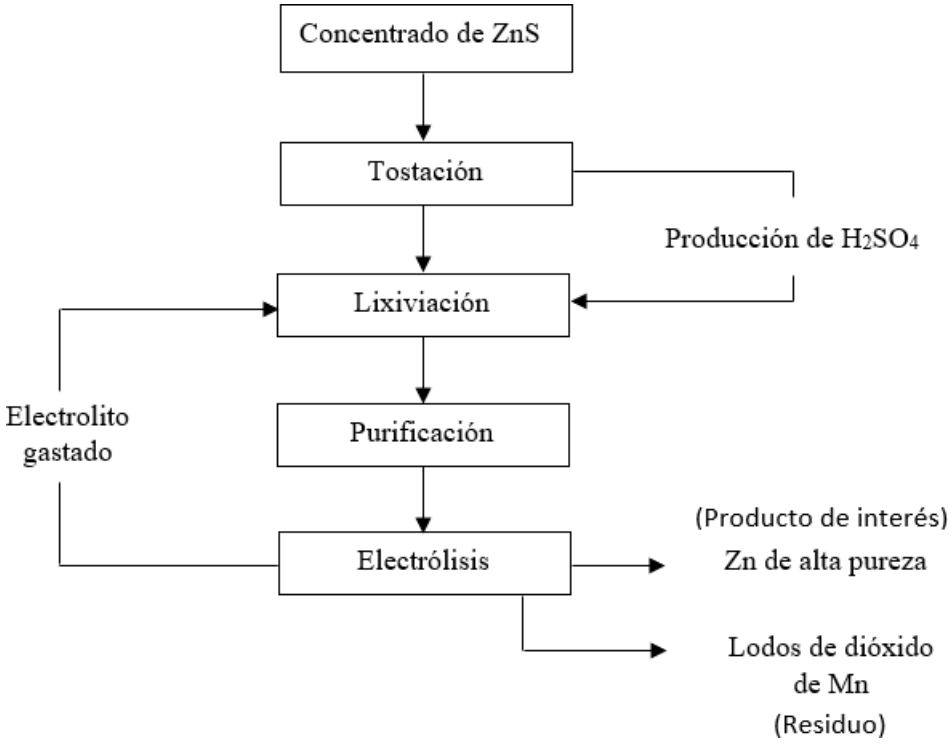

Figura 1. *Diagrama de flujo del proceso hidrometalúrgico del zinc.* Basado en González et al. (2025).

Durante la lixiviación, el zinc se disuelve principalmente como Zn^{2+} en soluciones de sulfato; sin embargo, junto con el zinc se incorporan diversas impurezas metálicas, tales

como hierro, cobre, cobalto, níquel y manganeso. El control de estas impurezas es fundamental, ya que su

acumulación afecta la eficiencia del proceso, incrementa el consumo energético y puede deteriorar la calidad del zinc depositado. El manganeso es un residuo de la electroobtención en estado de sólido como lodos. (Fleming, 2010).

2.1.2. Electrólisis de zinc en medio sulfato

La electrólisis de zinc se lleva a cabo en celdas electrolíticas que operan con un electrolito ácido compuesto principalmente por sulfato de zinc y ácido sulfúrico. En el cátodo ocurre la reducción del ion Zn^{2+} para formar zinc metálico, mientras que en el ánodo se produce predominantemente la evolución de oxígeno mediante la oxidación del agua (Biswas & Davenport, 1994).

Desde el punto de vista de la ingeniería química, la electrólisis de zinc constituye un proceso de transferencia de carga acoplado a fenómenos de transporte de masa y energía. Variables como la densidad de corriente, la temperatura, la composición del electrolito y el material del ánodo influyen directamente en la eficiencia de corriente y en la selectividad de las reacciones electroquímicas (Newman & Thomas-Alyea, 2012).

2.1.3. Impurezas en el electrolito de zinc

En la operación continua de una refinería de zinc, las impurezas presentes en el electrolito tienden a acumularse si no se implementan mecanismos adecuados de control. Algunas impurezas pueden ser removidas eficazmente durante las etapas de purificación, mientras que otras, como el manganeso, presentan mayor dificultad debido a su estabilidad química en soluciones ácidas (Davenport et al., 2011).

El manganeso se encuentra predominantemente como Mn^{2+} en el electrolito y, a diferencia de otras impurezas metálicas, no se deposita en el cátodo ni precipita fácilmente bajo las condiciones típicas de purificación. Por ello, su control depende en gran medida de su comportamiento electroquímico durante la electrólisis (Fleming, 2010).

2.1.4. Comportamiento electroquímico del manganeso en solución

El manganeso presenta múltiples estados de oxidación, siendo Mn^{2+} la especie más estable en soluciones ácidas de sulfato. Bajo condiciones anódicas suficientemente oxidantes, el Mn^{2+} puede oxidarse para formar óxidos de manganeso,

principalmente dióxido de manganeso (MnO_2) (Pourbaix, 1974).

La oxidación electroquímica del manganeso es una reacción compleja que involucra múltiples transferencias electrónicas y cuya cinética depende del potencial anódico, la

concentración de Mn^{2+}, la acidez del medio y la temperatura. En sistemas industriales, esta reacción ocurre simultáneamente con la evolución de oxígeno, lo que limita su eficiencia electroquímica (Bockris & Reddy, 2000).

2.1.5. Formación anódica de dióxido de manganeso (MnO_2)

El dióxido de manganeso formado durante la electrólisis se deposita generalmente sobre la superficie del ánodo o se desprende en forma de lodos que se acumulan en el fondo de la celda electrolítica. Este fenómeno constituye un mecanismo efectivo de remoción del manganeso del electrolito, dado que el MnO_2 es insoluble bajo las condiciones ácidas del proceso (Gamburg & Zangari, 2011).

Diversos estudios indican que la formación de MnO_2 está influenciada por la densidad de corriente aplicada, el potencial del ánodo y la disponibilidad de Mn^{2+} en la interfase electrodo-electrolito. Asimismo, el crecimiento del depósito puede verse limitado por fenómenos de difusión y por la baja conductividad eléctrica del propio óxido formado (Brenet et al., 2014).

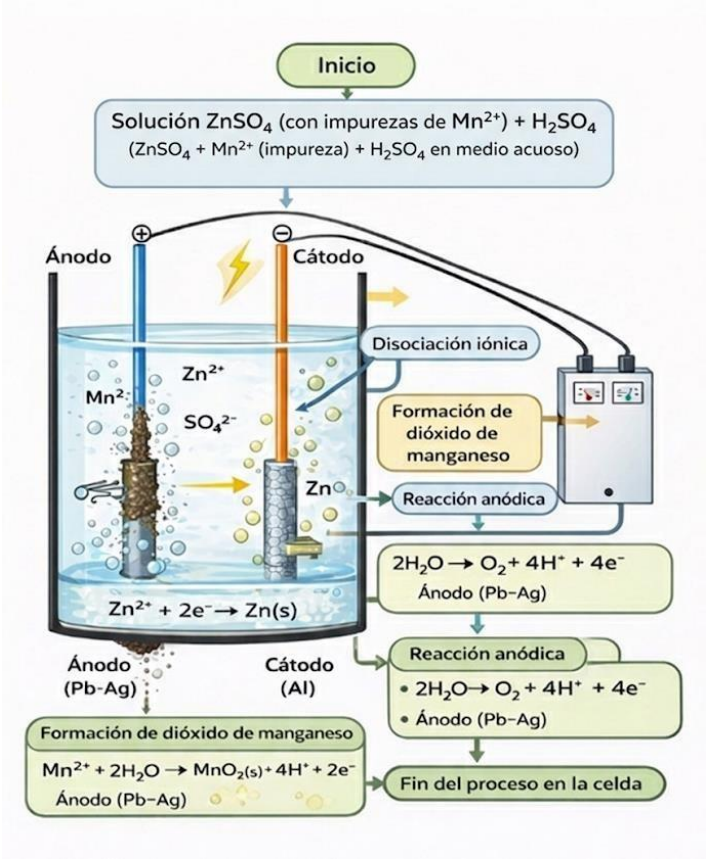

Figura 2. *Diagrama de flujo del proceso de electrodeposición de zinc con formación de dióxido de manganeso* [Imagen generada por inteligencia artificial]. Gemini AI.

(2026).

2.1.6. Rol de la electrólisis en el control del manganeso

Si bien el control del manganeso puede abordarse en distintas etapas del proceso hidrometalúrgico, la electrólisis representa una de las pocas etapas donde es posible removerlo directamente mediante un mecanismo electroquímico. El análisis de esta etapa como herramienta de control de impurezas permite identificar oportunidades de optimización sin necesidad de modificaciones estructurales del proceso (Davenport et al., 2011). La formación controlada de MnO_2 en el ánodo puede considerarse una estrategia complementaria para reducir la acumulación de manganeso en el electrolito y mejorar la estabilidad operativa de la refinería.

2.1.7. Métodos de ensamble

Los métodos de ensamble combinan múltiples modelos predictivos (denominados "aprendices base") para obtener un modelo final con mejor rendimiento que cualquiera de los modelos individuales. La intuición es que, si los errores de los aprendices base no están correlacionados, el promedio de ellos reduce la varianza total sin aumentar excesivamente el sesgo (Dietterich, 2000). En la práctica, los ensambles han demostrado ser superiores a los modelos individuales en una amplia variedad de tareas de clasificación y regresión, incluyendo problemas químicos y electroquímicos (Wang et al., 2025).

2.1.8. Algoritmos de ensamble: Random Forest y XGBoost

Random Forest, propuesto por Breiman (2001), construye un conjunto de árboles de decisión a partir de diferentes submuestras del conjunto de entrenamiento, obtenidas mediante muestreo bootstrap. Además, en cada división de un árbol, solo se considera un subconjunto aleatorio de las variables predictoras, lo que reduce la correlación entre los árboles. La predicción final se obtiene promediando las predicciones de todos los árboles. Esta estrategia disminuye la varianza del modelo sin aumentar significativamente el sesgo, lo que lo hace muy efectivo para conjuntos de datos con ruido o con alta dimensionalidad. Random Forest maneja bien relaciones no lineales y no requiere una estandarización estricta de las variables, aunque su interpretabilidad es limitada.

XGBoost (eXtreme Gradient Boosting) es una implementación optimizada del gradient boosting (Chen & Guestrin, 2016). A diferencia del bagging, el boosting construye los árboles de forma secuencial: cada nuevo árbol intenta corregir los errores

residuales del conjunto anterior. XGBoost incorpora regularización L1 (lasso) y L2 (ridge) en la función de coste, lo que ayuda a prevenir el sobreajuste. Además, utiliza una aproximación de segundo orden (hessiana) para acelerar la optimización. Otras ventajas incluyen el manejo

nativo de valores faltantes, la posibilidad de evaluar la importancia de las variables y la paralelización del cálculo. Los hiperparámetros críticos son: el número de estimadores (árboles), la tasa de aprendizaje (learning rate) que reduce la contribución de cada árbol, y la profundidad máxima de cada árbol.

2.2. Antecedentes del estudio

La aplicación de técnicas de Machine Learning en procesos hidrometalúrgicos y electroquímicos ha cobrado relevancia en los últimos años, como lo demuestran los trabajos recientes de Wang et al. (2025), Shang et al. (2026), Zhu et al. (2026) y De Oliveira et al. (2025). Estos estudios abordan problemas diversos –desde el modelado de reactores hasta la predicción de impurezas y el diagnóstico de fallas– y proporcionan un contexto valioso para la presente tesis, que busca predecir la concentración de manganeso en el electrolito agotado de la electroobtención de zinc.

Wang et al. (2025) publicaron una revisión tutorial en Digital Chemical Engineering sobre el uso de Machine Learning en el modelado, análisis y control de reactores electroquímicos. Los autores examinan algoritmos supervisados (redes neuronales, Random Forest, XGBoost) y su capacidad para aproximar dinámicas no lineales, estimar variables no medidas y diseñar estrategias de control predictivo. Concluyen que, gracias a la creciente disponibilidad de datos operativos, el Machine Learning se perfila como una herramienta complementaria –y en ocasiones superior– a los modelos fenomenológicos tradicionales. Este trabajo justifica la elección de XGBoost y Random Forest en nuestra tesis y alerta sobre la necesidad de preprocesar correctamente los datos para evitar fugas de información.

Shang et al. (2026) desarrollaron un método híbrido que combina análisis espectral singular (SSA) y aprendizaje profundo (deep learning) para predecir la concentración de iones cadmio (Cd^{2+}) en la etapa de purificación de la hidrometalurgia del zinc. Publicado en Chemical Engineering Science, el estudio logró predicciones precisas al extraer componentes de tendencia y estacionalidad de las series temporales de concentración. Aunque su enfoque emplea redes neuronales en lugar de ensambles de árboles, demuestra que las impurezas disueltas en licores de zinc pueden ser modeladas con técnicas de Machine Learning a partir de datos históricos. Este antecedente es especialmente relevante porque aborda una impureza metálica (cadmio) en un proceso

similar al nuestro (zinc), validando la hipótesis de que es posible anticipar concentraciones de contaminantes mediante herramientas computacionales.

Zhu et al. (2026) presentaron en Measurement un método de diagnóstico inteligente en línea para detectar cortocircuitos entre electrodos en celdas de electroobtención de zinc. Utilizando sensores magnéticos y algoritmos de clasificación (máquinas de vectores de soporte y árboles de decisión), lograron identificar fallas con alta precisión. Aunque su objetivo es el diagnóstico de cortocircuitos y no la predicción de impurezas, su trabajo confirma la viabilidad de integrar inteligencia artificial al monitoreo de celdas electrolíticas. Asimismo, subraya la importancia de variables operativas como la densidad de corriente y la temperatura, que también son utilizadas en nuestro modelo.

De Oliveira et al. (2025) aplicaron inteligencia artificial (Random Forest y redes neuronales) para predecir el espesor del recubrimiento de zinc sobre acero bajo carbono SAE 1008 en un proceso de electrodeposición. Publicado en Química Nova, el estudio empleó parámetros como densidad de corriente, temperatura, tiempo de depósito y concentración de iones. Los mejores modelos alcanzaron errores relativos inferiores al 5%, demostrando que el Machine Learning puede capturar relaciones entre condiciones operativas y resultados de calidad. Metodológicamente, este trabajo es análogo al nuestro: se entrenaron modelos de regresión con datos de proceso para predecir una variable de salida continua (espesor de zinc). Esta analogía refuerza la pertinencia de nuestro enfoque.

Ninguno de los artículos usados para este segmento de antecedentes aborda la predicción específica de la concentración de manganeso en el electrolito agotado de la electroobtención de zinc utilizando variables operativas rutinarias. Por lo tanto, esta tesis llena este vacío y, al mismo tiempo, documenta las limitaciones encontradas cuando las variables disponibles presentan baja correlación con la impureza objetivo. Los antecedentes revisados también sugieren líneas futuras (incorporación de análisis de series temporales, mediciones electroquímicas adicionales) que se discute en las conclusiones.

A continuación, en la Tabla 1, se resumen los principales hallazgos de los cuatro trabajos más relevantes para la presente investigación.

Tabla 1. Resumen de antecedentes del estudio

Autor	Técnica empleada	Variable predicha	Resultado principal
Wang et al. (2025)	Revisión de Machine Learning (Random Forest, XGBoost, redes neuronales) aplicado a reactores electroquímicos	Variables de estado de reactores (concentraciones, temperatura, potencial)	El ML permite modelar dinámicas no lineales y estimar variables no medidas; se recomienda su uso complementario a modelos fenomenológicos.
Shang et al. (2026)	Análisis espectral singular (SSA) + aprendizaje profundo (deep learning)	Concentración de cadmio (Cd^{2+}) en etapa de purificación del zinc	Predicción precisa de la concentración de cadmio, demostrando que la descomposición espectral mejora el rendimiento al reducir ruido.
Zhu et al. (2026)	Árboles de decisión y máquinas de vectores de soporte (SVM) sobre señales magnéticas	Detección de cortocircuitos entre electrodos en celdas de electroobtención de zinc	Método de diagnóstico en línea con alta precisión, validando el uso de inteligencia artificial para monitoreo de celdas.
De Oliveira et al. (2025)	Random Forest y redes neuronales	Espesor de recubrimiento de zinc sobre acero SAE 1008	Modelos con error relativo inferior al 5%, demostrando que ML puede predecir calidad de electrodeposición a partir de variables operativas.

2.3. Bases teóricas

Las bases teóricas se sostienen en el entendimiento del proceso hidrometalúrgico del zinc, la electrólisis y los parámetros y variables involucrados en el proceso

2.3.1. Proceso hidrometalúrgico del zinc

La producción industrial de zinc se sustenta mayoritariamente en la ruta de lixiviación-purificación-electroobtención (LPE). Esta vía es preferida por su capacidad para procesar concentrados con mineralogías complejas y por la alta pureza del

producto final. El proceso se inicia con la tostación para transformar sulfuros en óxidos (calcinas), seguido de una lixiviación en ácido sulfúrico donde el zinc se solubiliza como $ZnSO_4$. Sin embargo, la disolución no es selectiva; junto al zinc, el electrolito se carga de impurezas metálicas. Según Davenport et al. (2011), el control de estos elementos es el desafío técnico más crítico, ya que incluso en concentraciones de partes por millón (ppm), pueden inhibir la deposición del zinc o aumentar drásticamente el sobrevoltaje de la celda (Habashi, 1997; Fleming, 2010).

2.3.2. Electrólisis de zinc en medio sulfato

La electrólisis es un proceso de reducción electrolítica en el cual se aplica una diferencia de potencial para forzar la deposición de zinc metálico sobre cátodos (generalmente de aluminio). La reacción catódica principal es $Zn^{2+} + 2e^{-} \rightarrow Zn^{0}$.

Paralelamente, en el ánodo de aleación de plomo, ocurre la descomposición del agua para liberar oxígeno y protones. Biswas & Davenport (1994) señalan que este sistema opera en un equilibrio precario: el alto sobrevoltaje del hidrógeno sobre el zinc es lo que permite que el zinc se deposite antes que el gas hidrógeno en un medio tan ácido. Desde la termodinámica y la ingeniería de procesos, variables como la temperatura y la acidez deben controlarse estrictamente para mantener la eficiencia de corriente por encima del 90% (Newman & Thomas-Alyea, 2012).

2.3.3. Impurezas en el electrolito de zinc

Las impurezas se clasifican según su comportamiento electroquímico. Aquellas con potenciales de reducción más nobles que el zinc (como Cu, Co, Ni) tienden a codepositarse, contaminando el producto. A diferencia de los otros metales, el manganeso tiende a permanecer en solución o a oxidarse en el ánodo en lugar de reducirse en el cátodo (Davenport et al., 2011). Su acumulación en el circuito cerrado de la refinería incrementa la viscosidad del electrolito y puede pasivar los ánodos, por lo que su monitoreo es vital para la integridad de los componentes de la celda (Fleming, 2010).

2.3.4. Comportamiento electroquímico del manganeso en solución

El manganeso es un elemento de transición con una química redox compleja. En el electrolito ácido de zinc, la especie predominante es el ion manganoso (Mn^{2+}). De acuerdo con el diagrama de Pourbaix (1974), la estabilidad del Mn^{2+} se ve interrumpida a medida que el potencial anódico aumenta hacia la zona de estabilidad del dióxido de manganeso (MnO_2). La transición de Mn^{2+} a MnO_2 implica una transferencia de dos electrones y es altamente sensible al pH local de la interfase electrodo-solución. Bockris & Reddy (2000) enfatizan que la cinética de esta reacción es lenta y está influenciada por la formación previa de intermediarios como el Mn^{3+}, lo cual añade una capa de complejidad al control operativo.

Mahon et al. (2012) desarrollaron una simulación dinámica del proceso que integra balances de masa y energía para predecir la concentración de iones en el electrolito, demostrando que el aumento de concentración de especies inertes como el Mn^{2+} es una consecuencia directa de la contracción volumétrica del agua que se debe a la electrólisis del

agua en el ánodo. Investigaciones recientes confirman que, al emplear ánodos con materiales selectivos para la evolución de oxígeno, como los basados en óxido de manganeso, se puede favorecer cinéticamente esta reacción, inhibiendo otras oxidaciones y validando el comportamiento inerte del Mn^{2+} en el sistema (Shang et al., 2025)

2.3.5. Formación anódica de dióxido de manganeso (MnO2)

La formación de MnO_2 sólido es el único mecanismo natural de purificación de manganeso dentro de las celdas. Este compuesto se presenta en diversas fases polimórficas, siendo la Υ-MnO_2 la más común en condiciones electrolíticas. Gamburg & Zangari (2011) explican que el óxido puede adherirse al ánodo, formando una capa protectora que reduce la corrosión del plomo, o desprenderse como lodos debido a la turbulencia generada por el desprendimiento de burbujas de oxígeno. Sin embargo, un exceso de formación de lodos puede obstruir los sistemas de circulación y requerir paradas de mantenimiento no programadas (Brenet et al., 2014).

2.3.6. Rol de la electrólisis en el control del manganeso

La electrólisis se veía como la etapa de producción única e importante; sin embargo, bajo la visión moderna de la ingeniería química, se reconoce como un reactor electroquímico de purificación continua. Davenport et al. (2011) sostienen que la optimización de esta etapa permite un "auto purgado" del manganeso, pues de la relación entre las variables de entrada y la masa de MnO_2 precipitado, es posible gestionar el balance de masa de manganeso en toda la planta y. así, prescindir de etapas de precipitación química externas, reduciendo costos operativos y mejorando la sostenibilidad del proceso global de refinación de zinc.

2.3.7. Aprendizaje profundo para predicción de impurezas

Shang et al. (2026) proporcionan una base teórica para el tratamiento de series temporales en la predicción de concentraciones de iones en procesos hidrometalúrgicos. Introducen el análisis espectral singular (SSA) como una técnica de descomposición que separa la señal original en componentes de tendencia, oscilatorias y ruido, basándose en la teoría de sistemas dinámicos y la descomposición en valores singulares. Teóricamente, cualquier serie temporal con estructura determinista puede reconstruirse a partir de sus componentes principales, lo que mejora la relación

señal/ruido antes de aplicar modelos predictivos. Si bien Shang et al. utilizan redes neuronales profundas (deep learning) como paso final, su fundamento teórico es aplicable a cualquier algoritmo de regresión: al filtrar el ruido, se facilita la identificación de relaciones causales.

2.3.8. Árboles de decisión

Zhu et al. (2026) desarrollan una base teórica para el diagnóstico de fallas en celdas de electroobtención de zinc basada en la medición de campos magnéticos generados por las corrientes de electrodo. Según la ley de Biot-Savart, una corriente eléctrica produce un campo magnético proporcional; la distorsión de ese campo indica la presencia de un cortocircuito. A partir de esta relación física, los autores construyen un clasificador (árboles de decisión y SVM) que asigna cada patrón de campo a un estado de operación normal o anormal. Teóricamente, la combinación de sensores económicos con algoritmos de machine learning permite sustituir métodos manuales costosos. Aunque nuestro objetivo es regresión (concentración de Mn) y no clasificación, esta base teórica refuerza la idea de que las celdas electrolíticas pueden ser monitoreadas inteligentemente utilizando variables indirectas. Además, sugiere que variables como la distribución de corriente o el campo magnético – no incluidas en nuestro estudio – podrían mejorar la predicción de impurezas.

2.4. Definición de términos

Los siguientes son términos importantes para esclarecer el marco conceptual de este trabajo de investigación.

2.4.1. Actividad iónica

Medida de la concentración efectiva de una especie en una solución real, que difiere de la concentración molar debido a las interacciones electrostáticas entre iones en electrolitos de alta fuerza iónica (Bockris & Reddy, 2000).

2.4.2. Ánodo

Electrodo positivo de la celda donde ocurre la oxidación. En la industria del zinc, se utilizan aleaciones de Pb-Ag que facilitan la evolución de oxígeno y durante la electroobtención de Zn, los iones de Mn^{2+} presentes en el electrolito se oxidan en la superficie del ánodo, depositando una capa de MnO_2. (Ye, W. et al., 2020)

2.4.3. Cátodo

Electrodo de la celda electrolítica con carga negativa donde se produce la reducción de los cationes metálicos; para el zinc, se emplean placas de aluminio de alta pureza para permitir una adhesión suficiente del zinc y que no se desprenda

durante la electroobtención de zinc, pero no tan fuerte que impida su remoción mecánica en las máquinas deslaminadoras al final del ciclo de deposición (Habashi, 1997). (Almeida et al. 2025)

2.4.4. Árboles de decisión

Es un modelo predictivo no lineal que particiona el espacio de las variables predictoras en regiones rectangulares mediante preguntas binarias sucesivas sobre los valores de las variables (Killi, M., & Tran, N., 2019).

2.4.5. Densidad de corriente

Relación entre la intensidad de corriente eléctrica aplicada y el área de la superficie del electrodo. Es el parámetro que rige la velocidad de deposición y la cinética anódica (Newman & Thomas-Alyea, 2012).

2.4.6. Eficiencia de corriente

Fracción de la corriente eléctrica total que se utiliza efectivamente para la reacción de interés, respecto a la corriente teórica calculada por las leyes de Faraday (Davenport et al., 2011).

2.4.7. Electroobtención

Proceso de extracción de metales a partir de una solución rica mediante la aplicación de una corriente eléctrica externa que provoca la deposición del metal en el cátodo (Habashi, 1997).

2.4.8. Electrolito agotado

Solución resultante de la electrólisis que presenta baja concentración de zinc y alta acidez residual debido a la regeneración del ácido sulfúrico (Biswas & Davenport, 1994).

2.4.9. Entrenamiento del modelo

Proceso en el cual un algoritmo de Machine Learning ajusta sus parámetros internos mediante la exposición a datos históricos para minimizar el error predictivo (Hastie et al., 2009).

2.4.10. Error Absoluto Medio (Mean Absolute Error, MAE)

Es una métrica de evaluación para problemas de regresión que calcula el promedio de las diferencias absolutas entre los valores predichos y los valores reales. Esta métrica expresa el error en las mismas unidades que la variable objetivo. (Breiman, 2001).

2.4.11. Impureza crítica

Elemento presente en el electrolito que afecta la cristalización del zinc o disminuye el sobrevoltaje del hidrógeno, causando pérdida de eficiencia (Fleming, 2010).

2.4.12. Machine Learning

Subcampo de la IA que desarrolla algoritmos capaces de identificar patrones y realizar predicciones basadas en datos sin programación explícita (Hastie et al., 2009).

2.4.13. Manganeso

Metal de transición presente en el electrolito que, bajo condiciones de oxidación anódica, forma dióxido de manganeso sólido MnO_2 (Gamburg & Zangari, 2011).

2.4.14. Predictor

Variable de entrada independiente (como temperatura o acidez) utilizada por un modelo predictivo para estimar el valor de una variable objetivo (Hastie et al., 2009).

III. HIPÓTESIS Y VARIABLES

3.1. Hipótesis

A continuación, se presentan las respuestas tentativas a los problemas de investigación planteados. Estas buscan establecer la relación esperada entre las variables operativas del proceso de electroobtención y la precisión en la concentración de manganeso en electrolito agotado.

3.1.1. Hipótesis general

Un modelo de Machine Learning, basado en datos de variables operativas del proceso de electroobtención de zinc, permitirá predecir la reducción de concentración de manganeso en el electrolito agotado y analizar la influencia de dichas variables sobre el desempeño electroquímico de las celdas electrolíticas.

3.1.2. Hipótesis específicas

- Mediante un análisis deductivo se logrará un arreglo sistemático y depurado de los datos que permita la aplicación efectiva de la herramienta de Machine Learning para el control de concentración de manganeso en el electrolito agotado de celdas electrolíticas de la etapa de electroobtención de zinc.
- La identificación de patrones mediante Machine Learning permitirá reconocer las variables que afectan directamente al control de concentración de manganeso en el electrolito agotado de celdas electrolíticas de la etapa de electroobtención de zinc.
- La optimización de los parámetros del modelo de Machine Learning mejorará la exactitud en el control de concentración de manganeso en el electrolito agotado de celdas electrolíticas de la etapa de electroobtención de zinc.

3.2. Variables

Estas variables se identifican para el cumplimiento de los objetivos, clasificadas según su función en el modelo de Machine Learning y su naturaleza operativa.

3.2.1. Variable Dependiente

Es el parámetro que el modelo de Machine Learning busca predecir la concentración de manganeso en el electrolito agotado: Se define como la diferencia cuantitativa de la

presencia de manganeso tras el proceso de electroobtención. Es una variable cuantitativa continua, medida en gramos por litro (g/L). Su control es crítico para evitar los efectos despolarizantes y la formación de lodos anódicos descritos por Zhang (2009).

3.2.2. Variable Independientes

Son las variables operativas que alimentan el algoritmo y cuyas fluctuaciones influyen en el comportamiento del manganeso. Concentración de manganeso en solución pura: Nivel de ingreso de la impureza proveniente de la etapa de purificación (g/L).

3.3. Operacionalización de variables

Para mayor claridad, se presenta el cuadro de operacionalización que vincula los conceptos teóricos con los indicadores medibles:

3.3.1. Variables operativas independientes

En la tabla 2 se presentan la variable enmarcada en la operacionalización del proceso.

Tabla 2. Variable operativa independiente

Variable	Definición Conceptual	Definición Operacional	Unidad
Concentración Mn en solución pura (Mn_{SP}^{2+})	Cantidad de manganeso que ingresa a la celda.	Análisis químico de la solución pura	g/L

3.3.2. Variables dependientes

Estas se describen en la tabla 3.

Tabla 3. Variable operativa dependiente

Variable	Definición Conceptual	Definición Operacional	Unidad
Concentración de Mn en EA (Mn_{EA}^{2+})	Variación de la concentración de manganeso a la salida de la celda.	Análisis químico del electrolito agotado	g/L

IV. MATERIALES Y MÉTODOS

4.1. Área de estudio

El presente estudio se delimita en datos recopilados en el periodo diciembre 2024 a diciembre 2025 del proceso de electroobtención de Zinc en el área de Electrólisis de la Refinería Cajamarquilla, concesionada por la empresa brasileña Nexa Resources S.A. La duración del estudio es de 5 meses.

4.2. Diseño de investigación

El enfoque de esta investigación es cuantitativo del tipo aplicada y de nivel explicativo. Como etapa preliminar se crearon datos teóricos para entrenamiento de los modelos de Machine Learning (ver Anexo 2 - A7), definidas las etapas (ver Anexo 2 - A8), se procedió al modelamiento con datos reales (Anexos 3 y 4). El tamaño muestral inicial para la exploración del modelo predictivo es de 5040 datos reales, recuperados de la base de datos del área de Electrólisis en Nexa Resources Refinería Cajamarquilla, antes Votorantim Metais, que permite cubrir el espacio de variables operativas con suficiente densidad estadística para identificar patrones relevantes y evaluar el desempeño del modelo bajo condiciones extremas dentro del rango operativo definido. Los datos que se usará el programa computacional se basan en especificación de proceso de electroobtención de zinc según lo establece Votorantim Metais (como se citó en Leguía Huamaccto, Vilma; 2018) y se presentan en la tabla 4.

Tabla 4. Información para el desarrollo del programa computacional

Variable	Rangos permitidos
Temperatura de electrolito	36°C – 41°C
Zn en solución pura	145 – 155 g/L
Fe en solución pura	≤ 5,0 mg/L
Sb en solución pura	≤ 0,03 mg/L
Cu en solución pura	≤ 0,2 mg/L
Mn en solución pura	5 – 8 g/L
Densidad	1290 – 1300 g/L
Zn en EA	40 – 45 g/L
Mn en EA	12 – 15 g/L
Acidez	170 – 180%
Horas de depósito	48 h

Peso del depósito	80 – 85 kg

4.3. Procedimientos, Técnicas e instrumentos de recolección de información

En esta sección se detallan las etapas sistemáticas diseñadas para la gestión de los datos, las cuales garantizan la integridad y utilidad de la información recolectada para el estudio que abarcan desde el tratamiento técnico inicial hasta la ejecución del modelo en Machine Learning. Se empieza con una etapa preliminar de entrenamiento de los modelos de Machine Learning, se procedió al modelamiento con XGBoost y Random Forest haciendo uso de los datos reales recopilados de la Refinería Cajamarquilla. A continuación, se presenta el diagrama de flujo de desarrollo definido de los modelos usados tanto para XGBoost como para Random Forest.

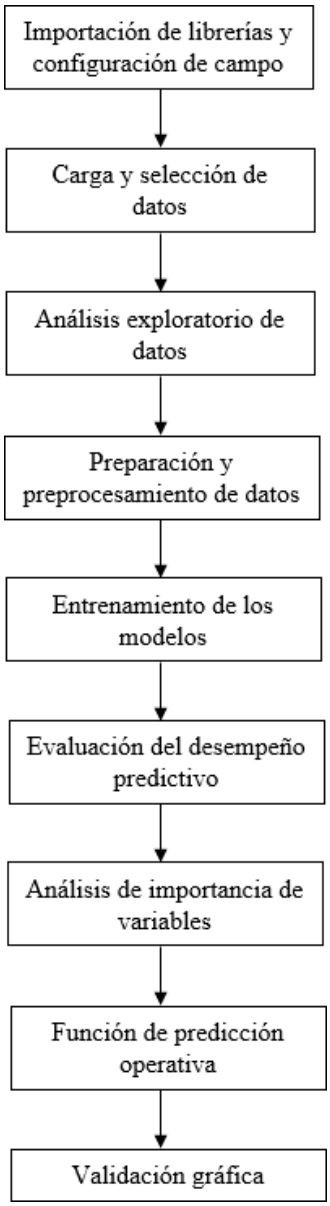

Figura 3. Diagrama de flujo de etapas sistemáticas del modelamiento en Machine

Learning con datos reales.

4.3.1. Importación de librerías y configuración del entorno de trabajo

En esta etapa se emplearon la biblioteca Pandas para la manipulación y análisis de datos tabulares, NumPy para operaciones numéricas, Matplotlib y Seaborn para la generación de gráficos estadísticos, Scikit-Learn para el preprocesamiento de datos y evaluación del modelo, y XGBoost y Random Forest para la construcción del algoritmo de aprendizaje automático. Además, se configuró el entorno de trabajo para evitar la aparición de mensajes de advertencia innecesarios durante la ejecución del código, garantizando una ejecución más limpia y ordenada.

4.3.2. Carga y selección de datos

En esta etapa se realizó la lectura del archivo de datos almacenado en formato Excel. El conjunto de datos estuvo conformado por 5040 registros y diez variables del proceso del área de Electrólisis de la Refinería Cajamarquilla. Se definió la variable objetivo del estudio que el modelo debe predecir: concentración de Mn^{2+} (g/L). Se definieron rangos operativos de referencia y se evaluó qué porcentaje de los datos reales coincidía dentro de los rangos establecidos.

4.3.3. Análisis exploratorio de datos

Este análisis tuvo como propósito comprender la distribución de las variables, identificar posibles anomalías y detectar las relaciones existentes entre ellas, especialmente con la variable objetivo.

Se calcularon estadísticas descriptivas completas, matriz de correlación y mapa de calor. Se crearon además histogramas para visualizar la distribución de Mn sol pura y Mn EA respecto a los rangos operativos establecidos.

Este análisis reveló una correlación directamente proporcional y perfecta entre Mn sol pura y Mn EA.

4.3.4. Preparación y preprocesamiento de datos

Una vez completado el análisis exploratorio, se procedió al preprocesamiento de los datos para optimizar su uso en ambos algoritmos de Machine Learning.

El conjunto de datos fue dividido utilizando la técnica train test split en una proporción 80% para entrenamiento (4032 registros) y 20% para prueba (1008 registros). Esta división permite evaluar la capacidad de generalización del modelo con

datos que no participaron en su entrenamiento.

Posteriormente, se aplicó la técnica de estandarización mediante StandardScaler de Scikit-Learn. Este paso transforma todas las variables a una escala con media cero y desviación estándar uno, evitando que variables con diferentes magnitudes (como densidad en g/L y concentraciones en mg/L como sucede con las impurezas) afecten negativamente el rendimiento de los modelos.

4.3.5. Entrenamiento de los modelos

XGBoost: Se entrenó un modelo con 400 estimadores, tasa de aprendizaje 0,05 y profundidad máxima 6.

Random Forest: Se entrenó un modelo con 400 árboles y profundidad máxima controlada.

4.3.6. Importancia de variables

El análisis de importancia de variables confirmó que el Mn^{2+}_{SP} es, con gran diferencia, la variable más influyente en ambos modelos. El resto de las variables operativas presentan una contribución marginal, lo cual es consistente con la alta correlación observada entre Mn^{2+}_{SP} y Mn^{2+}_{EA}.

4.3.7. Función de predicción operativa

Se desarrolló una función que permite ingresar parámetros operativos y obtener la predicción de Mn^{2+}_{EA} junto con una validación automática contra los rangos objetivo de entrada y salida.

V. RESULTADOS

5.1. Presentación y análisis de resultados

En esta sección se muestran los resultados obtenidos del modelo XGBoost con datos teóricos la comparación de XGBoost y Random Forest con datos reales y regresión lineal.

5.1.1. Resultados de modelo XGBoost entrenado correctamente con datos teóricos

En la tabla 5 se presentan los resultados por aspecto MAE, RMSE y R^2 del modelo XGBoost.

Tabla 5. Resultados por aspecto en XGBoost con datos teóricos.

Error Absoluto Medio (MAE)	2.844
Raíz del Error Cuadrático Medio (RMSE)	3.348
Coeficiente de Determinación R^2	-0.1285 (-12.85%)

Aunque los valores obtenidos (MAE = 2.844, RMSE = 3.348 y R^2 = -0.1285) muestran un desempeño bajo del modelo, esta etapa con datos teóricos tuvo un propósito principalmente de validación del pipeline. Sirvió como línea de base para comprobar que todo el flujo de preprocesamiento, entrenamiento y evaluación del modelo XGBoost funcionaba correctamente antes de aplicarlo a los datos reales. El mal ajuste era esperable, ya que los datos teóricos no reproducían fielmente las relaciones que existen en el proceso real de electroobtención. Esta prueba permitió identificar y corregir posibles errores en el código y en la configuración del modelo antes de trabajar con la información real de la planta.

Una vez validado el funcionamiento del pipeline con datos teóricos, se procedió a entrenar los modelos con la información real recolectada de las celdas de electroobtención de la Refinería Cajamarquilla. Este paso era fundamental, ya que los datos teóricos solo sirvieron como prueba de concepto, mientras que los datos reales permiten evaluar el comportamiento real del proceso y la utilidad práctica de los modelos.

5.1.2. Validación gráfica de predicciones XGBoost con datos teóricos

A continuación se presentan las gráficas estadísticas resultantes del uso de datos teóricos con XGBoost sobre las predicciones vs datos de variables operativas teóricas y la importancia de las variables operativas en la predicción de la variable objetivo.

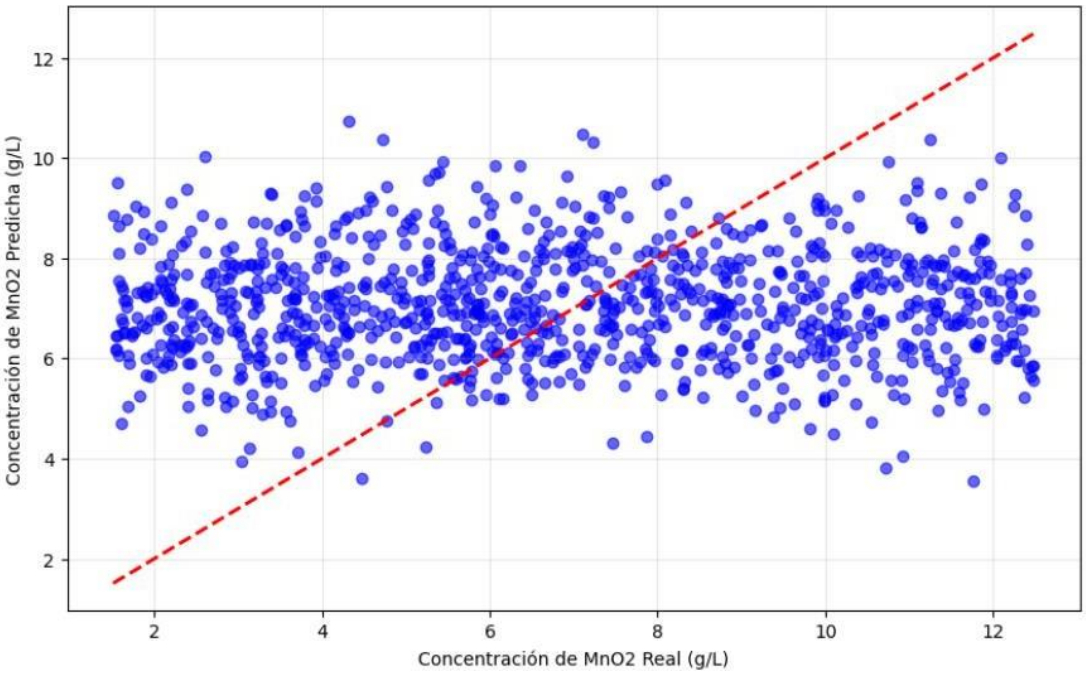

Figura 4. *Predicciones vs variables operativas – XGBoost*. Elaboración propia en Python.

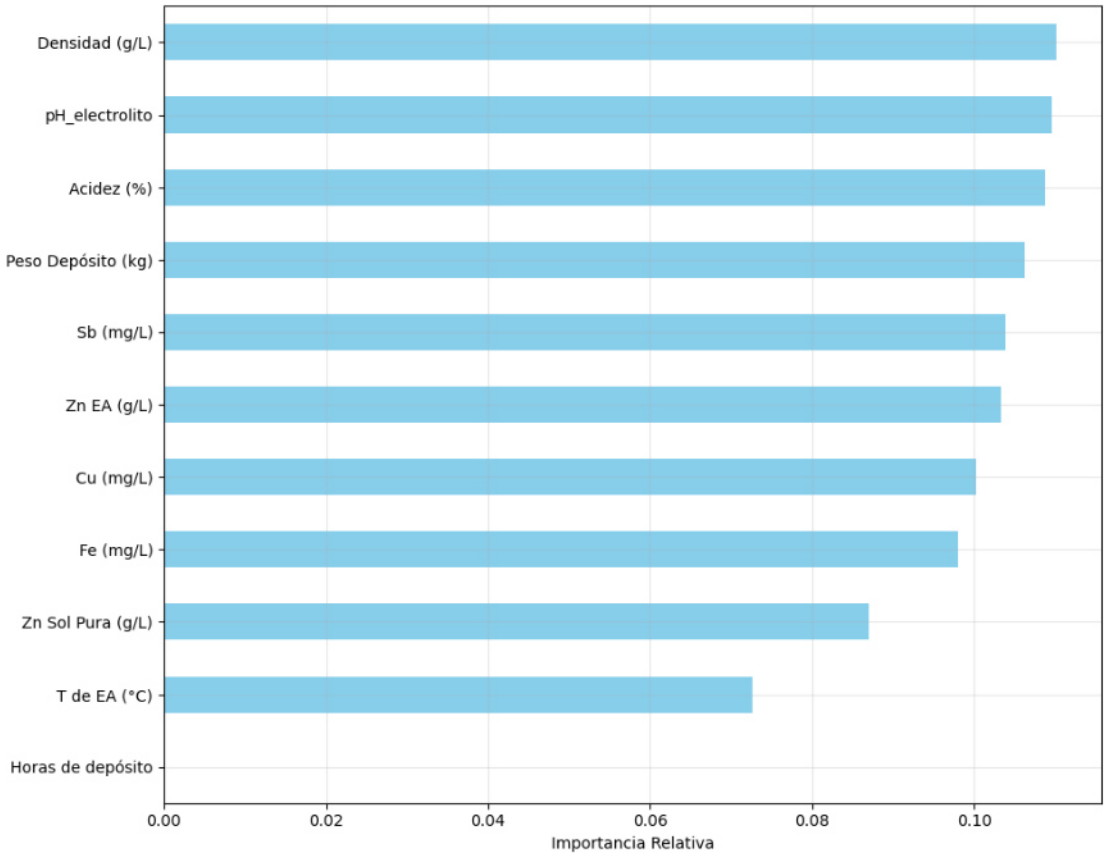

Figura 5. *Importancia de las variables operativas en la predicción de MnO_2.* Elaboración propia en Python

De lo expuesto en la figura 5, las variables más importantes según el modelo se presentan en a tabla 6.

Tabla 6. Variables más importantes según modelo XGBoost con datos teóricos

Variable operativa	Importancia relativa
Densidad (g/L)	0,110
pH electrolito	0,109
Acidez (%)	0,109
Peso depósito (kg)	0,106
Sb (mg/L)	0,104
Zn EA (g/L)	0,103
Cu (mg/L)	0,100
Fe (mg/L)	0,098

5.1.3. Validación gráfica de los modelos XGBoost y Random Forest con datos reales

Con el pipeline debidamente validado, se procedió al entrenamiento y evaluación de los modelos XGBoost y Random Forest utilizando el conjunto de datos reales obtenidos del proceso industrial. A diferencia de los datos teóricos, la información real refleja las condiciones operativas reales de las celdas electrolíticas, incluyendo las variaciones propias del proceso de electroobtención de zinc. Los resultados obtenidos en esta etapa fueron superiores, como se detalla a continuación.

Se generaron gráficos de predicciones vs valores reales. Los rangos operativos de referencia de Mn^{2+}_{SP} (entrada) es 5 – 8 g/L y de Mn^{2+}_{EA} que es la variable objetivo es 12 – 15 g/L. Ambos modelos muestran una alineación perfecta con la línea ideal.

Tanto el modelo XGBoost como el Random Forest fueron entrenados y evaluados bajo las mismas condiciones para poder comparar su desempeño de manera justa. Los rangos operativos típicos del proceso indican que la concentración de Mn^{2+} (entrada) suele estar entre 5 y 8 g/L, mientras que la concentración de Mn^{2+} se encuentra normalmente entre 12 y 15 g/L. A continuación, se presentan los gráficos de predicción versus valores reales para ambos modelos.

XGBoost **Random Forest**

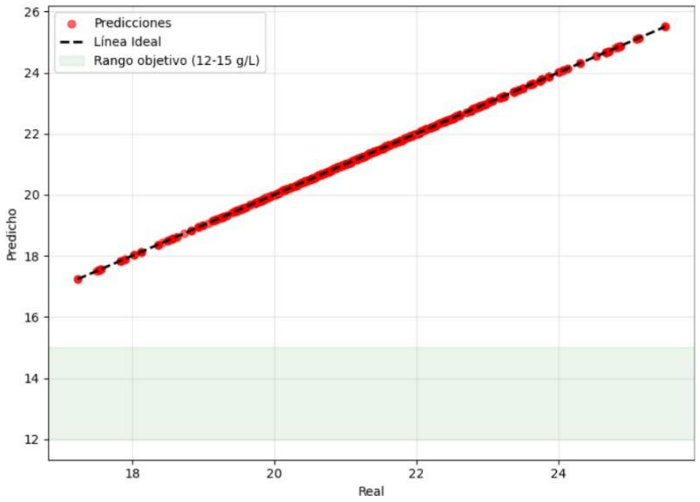

Figura 6. *Predicción vs Real – Mn EA (XGBoost con datos reales)*. Generación propia en Python.

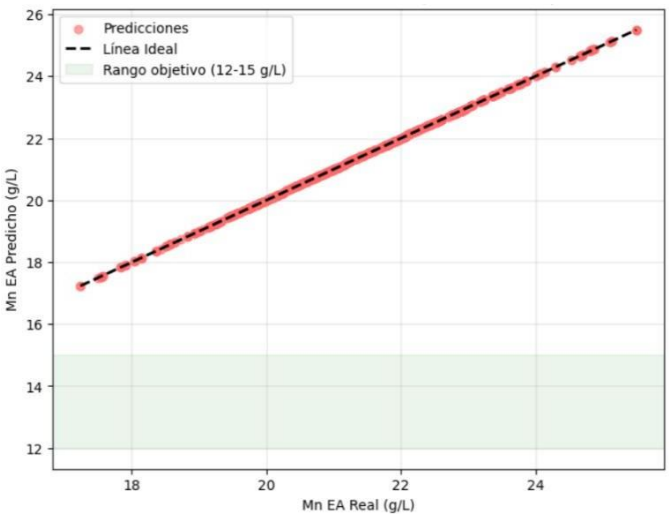

Figura 7. *Predicción vs Real – Mn EA (Random Forest con datos reales)*. Generación propia en Python.

Una vez evaluado el desempeño predictivo de los modelos, se consideró importante realizar un análisis exploratorio de datos (EDA) con el conjunto real para entender mejor las relaciones entre las variables. Este análisis ayuda a interpretar por qué los modelos lograron un ajuste tan elevado y permite identificar qué factores influyen (o no influyen) en la concentración de Mn^{2+}.

5.1.4. Resultados de la etapa de análisis exploratorio de datos en XGBoost y Random Forest con datos reales

En esta etapa se reveló que la variable Mn^{2+} presenta una correlación lineal. El resto de las variables operativas (temperatura, acidez, densidad, Fe, Cu, Sb, horas de depósito, peso de depósito) mostraron correlaciones cercanas a cero. El resumen de estas correlaciones se muestra en la siguiente tabla.

Tabla 7. Resumen de correlación y promedios de casos extremos bajo y alto con Mn^{2+}_{EA}

Variable / Parámetro	Correlación (r)	Tipo de Relación	Promedio en Extremo BAJO (Mn EA ↓)	Promedio en Extremo ALTO (Mn EA ↑)
Mn Sol Pura	1	Directamente proporcional	6.73	8.554
Densidad (g/L)	-0.023	Inversamente proporcional	1302.111	1301.443
Zn Sol Pura	-0.014	Inversamente proporcional	161.729	161.313
Acidez (%)	0.009	Inversamente proporcional	188.921	188.664
T en EA (°C)	0.006	Inversamente proporcional	36.667	36.625
Horas de depósito	-0.051	Inversamente proporcional	43.95	43.536
Peso Depósito (kg)	-0.051	Inversamente proporcional	73.25	72.559
Fe (mg/L)	-0.006	Inversamente proporcional	2.373	2.446
Sb (mg/L)	0.013	Inversamente proporcional	0.012	0.012
Cu (mg/L)	-0.059	Inversamente proporcional	0.149	0.128

Se generó un gráfico de mapa de calor (heatmap) como matriz de correlación entre las 11 variables, que se presenta a continuación.

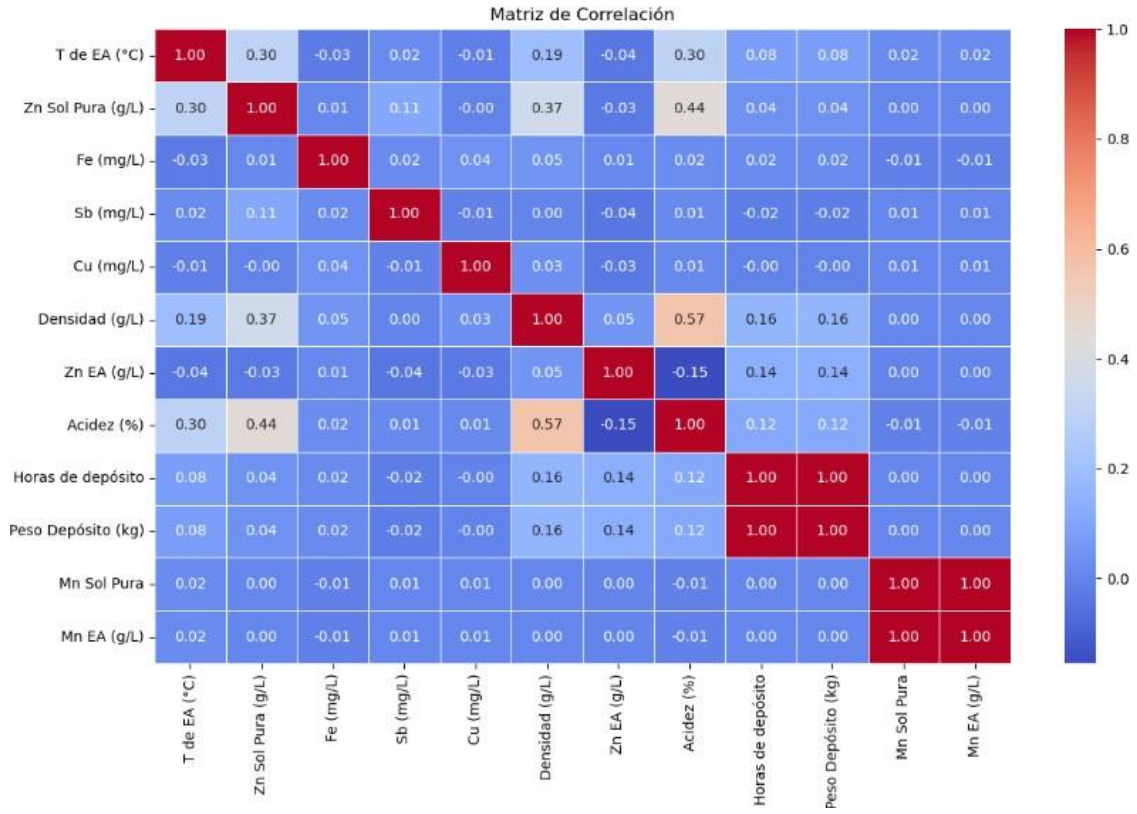

Figura 8. *Matriz de correlación de variables operativas*. Generación propia en Python.

De esta matriz de correlación de variables operativas, resalta lo siguiente presentado en las tablas 8 y 9:

Tabla 8. Correlaciones con Mn_{EA}^{2+} (g/L)

Parámetros de entrada	r
Mn_{EA}^{2+} (g/L)	1,000
Mn_{SP}^{2+} (g/L)	1,000
T de EA (°C)	0,021
Cu (mg/L)	0,007
Sb (mg/L)	0,005
Zn Sol Pura (g/L)	0,004
Densidad (g/L)	0,004
Horas de depósito	0,003
Peso depósito (kg)	0,003
Zn EA (g/L)	0,000
Acidez (%)	-0,009
Fe (mg/L)	-0,013

Tabla 9. Correlaciones con Mn_{SP}^{2+} (g/L)

Parámetros de entrada	r
Mn_{EA}^{2+} (g/L)	1,000
Mn Sol Pura (g/L)	1,000
T de EA (°C)	0,021
Cu (mg/L)	0,007
Sb (mg/L)	0,005
Zn Sol Pura (g/L)	0,004
Densidad (g/L)	0,004
Horas de depósito	0,003
Peso depósito (kg)	0,003
Zn EA (g/L)	0,000
Acidez (%)	-0,009

Fe (mg/L)	-0,013

De la figura 8 también se observan otras correlaciones de moderada a alta y extremadamente bajas, independiente de Mn sol pura y Mn EA.

Tabla 10. Correlaciones de moderada a alta y extremadamente baja

Correlaciones	r
Zn Sol Pura (g/L) vs Acidez (%)	0,44
Densidad (g/L) vs Acidez (%)	0,57
Zn EA (g/L) vs Acidez (%)	-0,15

Las siguientes figuras, 10 y 11, muestran las correlaciones de variables vs Mn_{EA}^{2+} predicha. Así se tiene que brindan resultados similares tanto por XGBoost y Random Forest; observándose que hay una fuerte correlación positiva entre el Mn de la solución electrolito agotado predicho con el Mn de la solución pura. Por último, se muestra en la tabla 11 el análisis comparativo entre ambos modelos de Machine Learning.

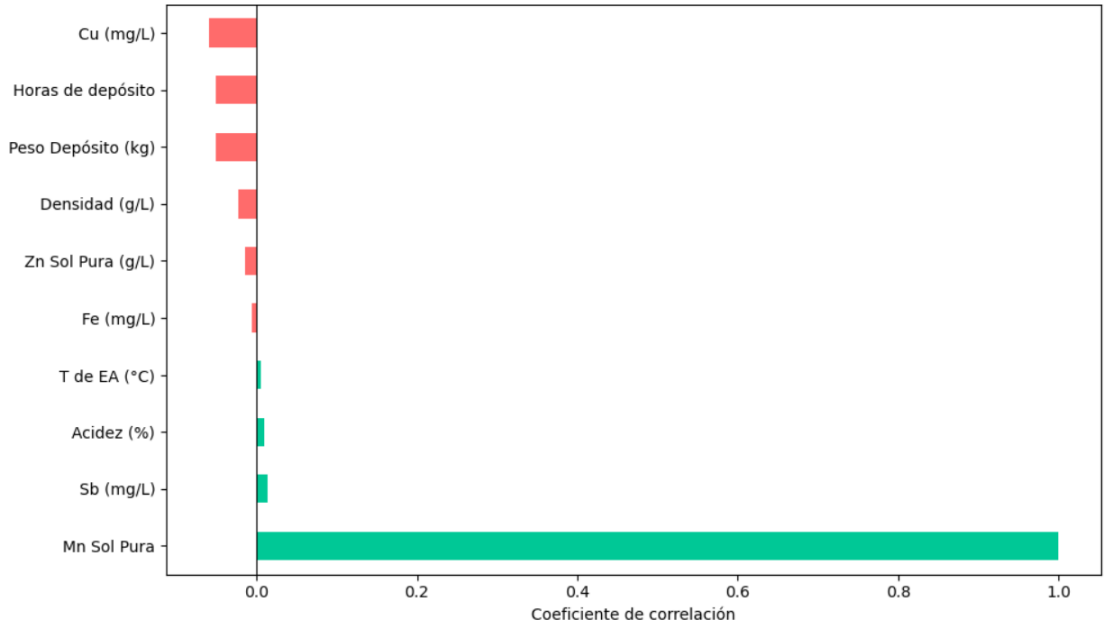

Figura 9. *Correlación de variables de entrada vs Mn EA predicho (XGBoost).* Generación propia en Python.

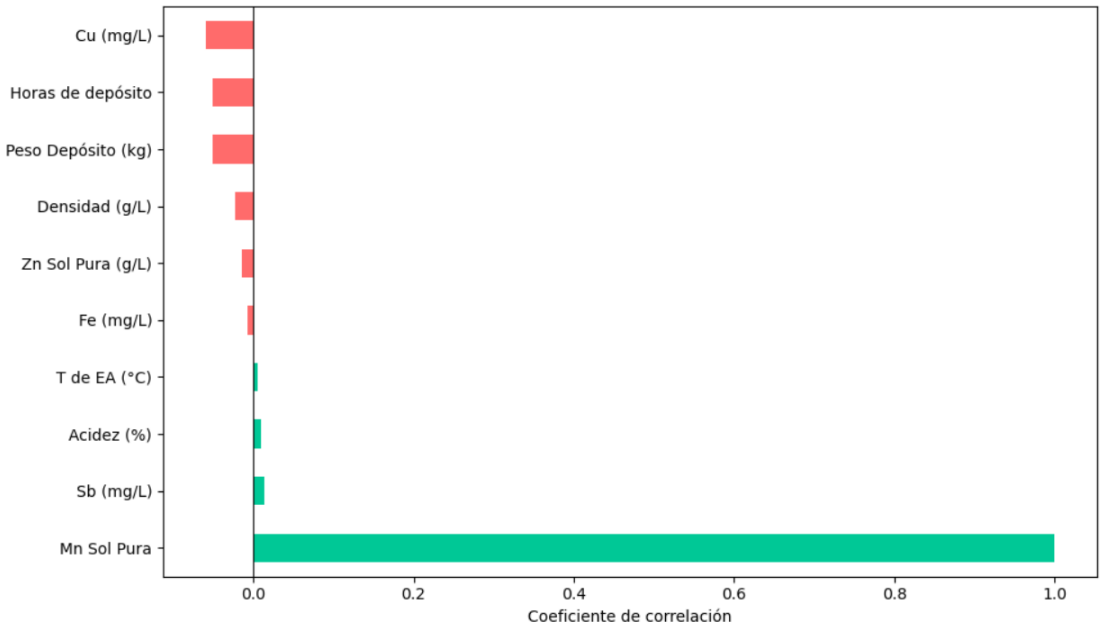

Figura 10. *Correlación de variables de entrada vs Mn EA predicho (Random Forest).* Generación propia en Python.

Tabla 11. Análisis comparativo entre XGBoost y Random Forest con datos reales

Aspecto	XGBoost	Random Forest	Observación
MAE (Mn EA)	0,000	0,000	Empate perfecto
R^2	1,000	1,000	Empate perfecto
Variable más importante	Mn Sol Pura	Mn Sol Pura	Igual
Dispersión en gráficos	Muy ajustado	Muy ajustado	Similar
Velocidad de entrenamiento	Más rápido	Ligeramente más lento	XGBoost ventajoso
Interpretabilidad	Alta	Alta	Similar

Esta comparación permite evaluar si la complejidad de los modelos XGBoost y Random Forest aporta ventajas reales frente a un enfoque más sencillo, especialmente en términos de precisión y utilidad práctica para la operación de la planta.

5.1.5. Resultados comparativos de regresión lineal, XGBoost y Random Forest con datos reales

Ajuste a una correlación lineal:

$$Mn_{EA}^{2+} = a + bMn_{SP}^{2+} \quad (1)$$

Donde:

a: intercepto

b: pendiente

Tabla 12. Análisis de varianza

	Grados de libertad	**Suma de cuadrados**	**Promedio de los cuadrados**	**F**	**Valor crítico de F**
Regresión	1	2096,01732	2096,01732	$1{,}4318 \square 10^{32}$	0
Residuos	1006	1,4727E-26	1,4639E-29		
Total	1007	2096,01732			

Tabla 13. Intercepto y pendiente de regresión

	Coeficientes
Intercepción	$-2{,}131610^{-14}$
Pendiente	2,8

Por lo tanto:

a: $-2{,}13 \square 10^{-14} \approx 0$

b: 2,8

En consecuencia

$$Mn_{EA}^{2+} = 2{,}8 Mn_{SP}^{2+} \tag{2}$$

Entonces el coeficiente de determinación R^2:

$$R^2 = \frac{2096{,}01732}{2096{,}01732} = 1 \tag{3}$$

Se calcula el MAE para hacer el comparativo con los resultados de XGBoost y Random Forest mostrados en la anterior tabla (Tabla 12).

$$MAE = \frac{1}{1008} \sum_{i=1}^{1008} \left| Mn_{EA}^{2+} - (a + b(Mn^{2+}))_{SP} \right| \tag{4}$$

Resultando $MAE = 1{,}11x10^{-14} \approx 0$

Tabla 14. Comparativo de regresión lineal y modelos de Machine Learning

Modelo	$\mathbf{R^2}$	**MAE**
Regresión lineal simple ($Mn_{EA}^{2+} = 2{,}8Mn_{SP}^{2+}$)	1	$1{,}11\text{x}10^{-14} \approx 0$
XGBoost	1	0
Random Forest	1	0

Los resultados de aspectos estadísticos obtenidos tanto por regresión lineal simple, XGBoost y Random Forest, resultan lo mismo en su coeficiente de determinación y MAE.

De la relación expresad en la ecuación (1):

$$Mn_{EA}^{2+} = 2{,}8Mn_{SP}^{2+} \tag{5}$$

En una celda de electroobtención de zinc, de los resultados anteriores de los modelos de Machine Learning, el manganeso no se oxida significativamente en el ánodo. Por lo tanto, el flujo másico de manganeso que entra es igual al que sale:

$$Q_{SP}\, Mn_{SP} = Q_{EA}\, Mn_{EA} \tag{6}$$

Donde:

Q_{SP} = Causal volumétrico del electrolito de entrada

Q_{EA} = Causal volumétrico del electrolito de salida

Sustituyendo la relación empírica $Mn_{EA}^{2+} = 2{,}8Mn_{SP}^{2+}$:

$$Q_{SP}\, Mn_{SP} = Q_{EA}\,(2{,}8\, Mn_{SP}) \tag{7}$$

Cancelando Mn_{SP} (que es >0) y despejando:

$$\frac{Q_{SP}}{Q_{EA}} = 2{,}8 \tag{8}$$

Esta igualdad demuestra que:

$$Q_{EA} = \frac{Q_{SP}}{2{,}8} \tag{9}$$

Es decir, el caudal de electrolito que sale de la celda es 2,8 veces menor que el

que entró. Dado que la densidad del electrolito varía muy poco, la reducción de caudal es equivalente a una reducción de volumen:

$$V_{EA} = \frac{V_{SP}}{2{,}8} \tag{10}$$

Esta reducción de volumen se debe a la pérdida de agua en el circuito que es resultado de la regeneración del ácido sulfúrico por consumo de agua, reacción que ocurre en el ánodo para generar los protones que mantienen la acidez del electrolito.

$$2H_2O \rightarrow O_2 + 4H^+ + 4e^- \tag{11}$$

El factor de contracción volumétrica (V_C) se define como:

$$V_C = \frac{V_{SP}}{V_{EA}} = \frac{Q_{SP}}{Q_{EA}} = 2{,}8 \tag{12}$$

Esto significa que el volumen de electrolito se contrae en un 64,3% al pasar por la celda:

$$\text{Contracción} = \left(1 - \frac{1}{2{,}8}\right) \times 100 = 64{,}3\% \tag{13}$$

VI. DISCUSIÓN DE RESULTADOS

El modelo XGBoost entrenado con datos teóricos (estimado o simulados) fue evaluado mediante métricas de error y análisis gráfico, con el fin de determinar su capacidad predictiva para la concentración de Mn^{2+} en función de variables operativas. Como se presenta en la Tabla 5, el modelo presentó un MAE de 2,844 g/L y RMSE de 3,348 g/L. Estos valores indican que, en promedio, la predicción del modelo se desvía en aproximadamente 2,844 g/L respecto al valor real. La diferencia entre RMSE y MAE indica que existen valores atípicos o predicciones muy desviadas en puntos específicos. El valor más crítico es el R^2 de
-0,1285 (-12,85%), esto implica que el modelo explica menos varianza que una simple media de los datos, es decir un mal ajuste general que puede deberse a que los datos teóricos son poco representativos respecto a la realidad, falta de variables relevantes, hiperparámetros no optimizados en la configuración del XGBoost o relaciones no lineales que el modelo no logra capturar adecuadamente con los datos proporcionados. Este resultado sugiere que, aunque XGBoost es un algoritmo potente, su rendimiento depende de la calidad y cantidad de datos. Con datos teóricos, el modelo no logra generalizar patrones útiles. En la figura 4 (predicciones vs variables operativas con datos teóricos) la dispersión entre los valores teóricos y predichos confirma la baja capacidad predictiva observada en el R^2. Esto refuerza la necesidad de usar datos reales en vez de teóricos. En la figura 5 y tabla 6 (importancia de variables), se indica que según el modelo XGBoost, ninguna variable domina claramente la predicción, lo que puede reflejar que los datos teóricos fueron generados sin relaciones causales fuertes, lo que dificulta que el modelo encuentre patrones discriminantes.

Es importante mencionar que, aunque el modelo XGBoost no logró un buen ajuste con los datos teóricos, estos resultados no se consideran un fracaso del algoritmo. Esta etapa se realizó principalmente como prueba de concepto y línea de base de la metodología desarrollada, esto es, validar todo el proceso de entrenamiento, escalado de variables y evaluación de métricas antes de introducir los datos reales de la Refinería Cajamarquilla. El bajo R^2 obtenido confirma que los datos teóricos no capturaban adecuadamente la complejidad y las relaciones presentes en el proceso industrial, lo cual refuerza la importancia de trabajar con información real para obtener modelos útiles desde el punto de vista operativo.

Los modelos de XGBoost y Random Forest haciendo uso de datos reales, alcanzaron demostrar que la concentración de Mn_{EA}^{2+} y la concentración de Mn_{SP}^{2+} tienen una relación

directamente proporcional, con un coeficiente de determinación de 1 y MAE = 0 en ambos casos, expresado en resumen en la tabla 11. Esto indica que los modelos son capaces de explicar el 100% de la varianza de la variable objetivo con los datos reales proporcionados.

De las tablas 8 y 9 se interpreta que Mn_{SP}^{2+} y Mn_{EA}^{2+} están prácticamente determinadas la una por la otra, lo que sugiere que el proceso de electrodeposición mantiene una relación estequiométrica casi constante. Las correlaciones con otras variables operativas son despreciables ($r<0{,}02$ en la mayoría de los casos), lo que indica que factores como temperatura de electrolito (°C), acidez (%), densidad (g/L), impurezas (Fe, Cu, Sb mg/L), horas de depósito y peso de depósito (kg) no presentan una relación lineal significativa con Mn EA dentro del conjunto de datos analizado. Este hallazgo es sostenido por Zhang & Cheng (2007), quienes señalan que la remoción de manganeso en la etapa de electroobtención es típicamente baja y depende críticamente de condiciones anódicas, pero que esta remoción representa solo una fracción menor del inventario total. Asimismo, Mahon & Alfantazi (2014) también reportaron que en muchas refinerías se requiere de etapas adicionales de precipitación o control en la purificación porque la electrolisis no es un sumidero eficiente de Mn. Este comportamiento tiene explicación en la termodinámica y la cinética de los procesos electroquímicos involucrados en el ánodo de las celdas. El Mn^{2+} tiene un potencial estándar de reducción más negativo que Zn^{+2}, lo que implica que en condiciones normales de electrolisis, la deposición catódica de manganeso es termodinámicamente desfavorable y cinéticamente inhibida. En consecuencia el manganeso permanece en la solución y se direcciona hacia el ánodo (Zhang & Cheng, 2007). En el ánodo la formación de capas de MnO_2 incrementa la resistencia óhmica, reduce la actividad electrocatalítica y favorece la evolución de oxígeno en vez de la oxidación de manganeso. Este efecto es debido a las temperaturas moderadas entre 35 – 45°C y densidades de corriente, propias de la operación de electroobtención de zinc. Como resultado, la eficiencia de remoción de manganeso en el ánodo es deficiente (Mahon & Alfantazi, 2014).

Desde la perspectiva electroquímica, se podría asumir que en la relación expresada por la ecuación $Mn_{EA}^{2+} = 2{,}8Mn_{SP}^{2+}$ determinada a partir de regresión lineal simple, el factor 2,8 responde a una estequiometría de oxidación anódica. Sin embargo, las correlaciones

nulas encontradas entre el Mn^{2+} y las variables operativas a excepción de la concentración de

Mn^{2+}_{SP}, descartan que las condiciones de la celda influyan en la remoción de esta impureza. Dado que el manganeso no se deposita en el cátodo (Anexo A3) y su oxidación anódica es despreciable bajo las condiciones estudiadas, se comporta como un elemento conservativo.

El aumento de su concentración es un factor de 2,8 lo que indica una contracción volumétrica del electrolito. Esta es resultado de la electrolisis del agua (Anexo 2), que consume agua de manera continua para generar protones que regeneran el ácido sulfúrico; y la evaporación térmica inherente de la electroobtención en las celdas que operan a temperaturas entre 36 y 41°C. Por lo tanto, 2,8 es un valor que representa el factor de concentración. Los algoritmos de Machine Learning al aislar y descartar las variables operativas relacionadas al Mn^{2+} demuestra que la celda de electroobtención actúa como un concentrador de manganeso y no un reactor de purificación. En consecuencia, invalida que las variables operativas al ser ajustadas permitirán controlar los picos de manganeso. Cualquier estrategia para mitigar la acumulación de Mn^{2+} no debe basarse en la electroobtención, sino en etapas previas de lixiviación y purificación de la solución pura (Mahon & Alfantazi, 2014).

En síntesis, el factor 2,8 se debe interpretar como un parámetro diagnóstico que refleja la ineficiencia intrínseca de la electroobtención para reducción de la concentración de manganeso en electrolito agotado. Su valor está condicionado por las condiciones estudiadas dentro de la celda. Cualquier estrategia que busque reducir la concentración de manganeso en el electrolito agotado debe pasar necesariamente por reducir la concentración de manganeso en la solución de alimentación.

A diferencia del manganeso y sus correlaciones con las demás variables operativas, en la tabla 10 a partir de la figura 8, se observan correlaciones moderadas entre variables operativas, lo que sugiere relaciones secundarias que podrían ser relevantes para otros objetivos o para entender la química del proceso, aunque no influyan en la predicción de Mn^{2+}_{EA}. La ausencia de correlación entre Mn^{2+}_{EA} y las variables operativas de entrada (temperatura de electrolito, densidad, acidez, impurezas, horas de depósito y peso de depósito) sugiere que la operación en las celdas electrolíticas de la Refinería Cajamarquilla no está optimizada para la oxidación de Mn^{2+} a MnO_2. Este resultado es consistente con la literatura que señala que la remoción de manganeso en la electrólisis suele ser baja y fuertemente dependiente de las condiciones anódicas, las cuales podrían no estar siendo capturadas por las variables operativas monitorizadas. Se infiere, por tanto, que la etapa de electroobtención actúa como un *concentrador pasivo* para el Mn^{2+}y no como un reactor de purificación,

El hallazgo sobre la correlación entre densidad vs acidez (r = 0,57) es positiva moderada a fuerte. Es consistente con la química fundamental del proceso de electroobtención de zinc y constituye una validación de que el sistema está operando

correctamente desde la perspectiva del zinc. Durante la electrólisis, ocurren reacciones en ambos electrodos cátodo y ánodo (ver Anexo A1, A2 y A3), la electroobtención de zinc regenera ácido sulfúrico, por lo que su acumulación en el electrolito agotado contribuye directamente a aumentar la densidad de la solución. La correlación positiva (r = 0,57) entre densidad y acidez indica que: a medida que el zinc se deposita en el cátodo, el ácido sulfúrico se regenera. El aumento de ácido sulfúrico incrementa la densidad del electrolito, por consiguiente el proceso está operando de manera consistente con lo esperado termodinámicamente. Desde el punto de vista termodinámico y estequiométrico, la reacción global de la electroobtención de zinc (ver Anexo 1) indica que por cada mol de zinc metálico depositado en el cátodo, se generan 2 moles de protones en el ánodo, los cuales se combinan con el sulfato para regenerar ácido sulfúrico. El incremento de acidez y la presencia de sulfato aumentan la densidad del electrolito porque el ácido sulfúrico tiene una densidad significativamente mayor que el agua y su solubilización en el electrolito incrementa la masa total por unidad de volumen. Dado que la densidad es una propiedad directamente proporcional a la concentración de sulfato y acidez, el coeficiente de correlación 0,57 confirma que estadísticamente, el 57% de la variación en la densidad es explicada por la variación en la acidez. Desde la perspectiva de la termodinámica, el potencial de electrodo está siendo controlado por la actividad de los iones Zn^{2+} y la concentración de H^{+}, favoreciendo la deposición selectiva del zinc sobre la reducción de protones. Este hallazgo es importante porque confirma que, a diferencia del manganeso, el zinc sí está respondiendo a las condiciones operativas de la celda electrolítica. La correlación de densidad y acidez es un indicador indirecto de que la eficiencia de corriente $_{EA}$es adecuada, el ácido se regenera correctamente, el balance de masa del zinc en el sistema es consistente. Es particularmente relevante contrastar este resultado con el hallazgo anterior de la correlación entre Mn^{2+} y
Mn_{SP}^{2+} (r = 1,00). Mientras que el zinc muestra un comportamiento dinámico y responde a las condiciones de la celda (evidenciado por la correlación densidad y acidez), el manganeso se comporta como un elemento inerte y conservativo. Esta diferencia de comportamiento confirma que el zinc y el manganeso siguen caminos electroquímicos completamente diferentes en el sistema.

Se realizó también una comparación de los modelos XGBoost y Random Forest

con regresión lineal para discutir el valor de predicción con modelamiento en Machine Learning. Mediante regresión lineal se identifica que la relación entre Mn sol pura y Mn EA es perfectamente lineal. Entonces en términos de precisión matemática (R2 y MAE) resultan lo

mismo. De la tabla 11, XGBoost frente a Random Forest presenta una ventaja en velocidad de entrenamiento, lo que lo hace más adecuado para aplicaciones en tiempo real o reentrenamientos frecuentes. En operación real, dada la robustez de XGBoost, ante eventos de falla de sensores o falsas alarmas, por su naturaleza de ensamble reduce estas fallas en el sistema de control. Además, los modelos de Machine Learning generan registros de importancia de variables que pueden ser utilizados para auditorías de calidad, que no puede ofrecer la regresión lineal simple por sí misma. En términos de despliegue operativo, automatización y manejo de datos en tiempo real, los modelos de Machine Learning sí tiene tienen ventajas significativas sobre la regresión lineal, en especial XGBoost que demostró tener mayor velocidad de entrenamiento. Tienen capacidad de reentrenamiento continuo, actualizarse automáticamente con nuevos datos, adaptándose a cambios durante el proceso. La importancia de las variables es coincidente en ambos modelos: Mn^{2+} es la variable dominante, con una contribución cercana al 100%. Esto refuerza la conclusión de que las demás variables son irrelevantes para este objetivo predictivo. Y desde el punto de vista operativo, el modelo XGBoost puede implementarse como sistema de monitoreo en línea. La implementación de este modelo XGBoost tendría efecto positivo en mayor estabilidad del proceso al detectar desviaciones de Mn^{2+} más rápidamente, evitando la formación excesiva de lodos de MnO_2 que ensucian los ánodos. También brinda la oportunidad de optimizar la dosificación de Mn en la etapa de purificación en caso se requiera un Mn^{2+} objetivo que en condiciones actuales es el rango de 12 -15 g/L. Si la planta decide cambiar el rango operacional, el modelo actual podría fallar, variables que ahora quedan demostradas no son relevantes en la correlación para predecir la concentración de Mn^{2+}, podrían llegar a serlo. Por tanto, se recomienda implementar el modelo junto con revisiones periódicas y alertas por cambio de distribución para mantener su naturaleza adaptativa.

SP

EA

EA

En las condiciones estudiadas en este trabajo de investigación, la relación de proporcionalidad directa entre la concentración de manganeso en solución pura y electrolito agotado implica que en la etapa de electroobtención cualquier incremento de la concentración de manganeso en solución pura, se traduce en un incremento equivalente en el electrolito agotado. Esto tiene una implicancia operativa directa: cualquier estrategia de control que pretenda regular el manganeso mediante ajustes en

los parámetros de electroobtención pueden provocar efecto en otras variables operativas no relacionadas a la concentración de manganeso en electrolito agotado. Además, redirige el foco de atención hacia la etapa de purificación, donde se debe implementar un sistema robusto de

precipitación de manganeso, dado que el control del manganeso en el proceso de electroobtención es poco o nula en términos de eliminación. Se puede construir un sensor virtual o software de sensor que estime Mn^{2+}_{EA} en tiempo real a partir de la medición de Mn sol pura (medible con sonda o análisis rápido). Esto elimina la necesidad de esperar análisis químico del electrolito agotado, reduciendo el tiempo de respuesta de horas a segundos. Al demostrar que impurezas como Fe, Cu y Sb no influyen en Mn^{2+}_{EA} dentro del rango actual, se puede reducir la frecuencia de monitoreo de esas impurezas respecto al Mn^{2+}. Es decir, si antes se median estas impurezas en un intervalo de cuatro horas, con esta implementación, medir Mn^{2+}_{SP} cada hora y Mn^{2+}_{EA} solo una vez al día como verificación. Esto liberaría la capacidad analítica para otros parámetros críticos para la operación de electroobtención de zinc.

VII. CONCLUSIONES Y RECOMENDACIONES

7.1 Conclusiones

Se logró un arreglo sistemático y depurado de los datos mediante la limpieza de 5040 registros operativos reales, el preprocesamiento correspondiente, la división de los datos en conjuntos de entrenamiento y prueba (80/20) y la estandarización de variables mediante StandardScaler. Este procedimiento permitió la aplicación efectiva de los modelos de Machine Learning. Los resultados confirman que al usar datos teóricos o datos que no sean representativos, cercanos a la realidad, por más que se usen modelos sofisticados en Machine Learning como son XGBoost y Random Forest, no se alcanzará una correcta predicción, no lo logra generar patrones útiles, como se vio un coeficiente de determinación de -0,1285 (-12,85%) cuando se usaron datos teóricos; por lo que usar datos reales es fundamental para desarrollar modelos predictivos confiables en procesos hidrometalúrgicos.

La identificación de patrones mediante los modelos de Machine Learning, apoyada en la matriz de correlación y el análisis de importancia de variables, permitió reconocer claramente las variables que influyen en la concentración de manganeso en el electrolito agotado. Se determinó que la concentración de Mn en solución pura presenta una correlación prácticamente perfecta con la concentración de Mn en el electrolito agotado (r = 1,00), constituyéndose como la variable dominante. En contraste, las variables operativas tales como temperatura del electrolito, acidez, densidad, concentraciones de Fe, Cu y Sb, horas de depósito y peso de depósito mostraron correlaciones cercanas a cero y contribuciones marginales en los modelos. Este hallazgo indica que, dentro de los rangos operativos estudiados, dichas variables no ejercen una influencia significativa sobre la concentración final de manganeso.

La optimización de los parámetros de los modelos permitió alcanzar una exactitud predictiva prácticamente perfecta. El modelo XGBoost se configuró con 400 estimadores, con la tasa de aprendizaje de 0,05 y profundidad máxima de 6, mientras que el Random Forest utilizó 400 árboles. Ambos modelos demostraron una excelente capacidad para representar la relación entre las variables. Si bien esta alta exactitud se debe principalmente a la relación lineal determinista entre Mn en solución pura y Mn

en electrolito agotado, la optimización confirmó la robustez de los modelos. Entre ambos, se recomienda XGBoost

por su mayor velocidad de entrenamiento, lo que lo hace más adecuado para aplicaciones de monitoreo en tiempo real.

La implementación de técnicas de Machine Learning en este estudio ha demostrado ser una herramienta de diagnóstico extraordinariamente útil para desentrañar la complejidad de los procesos hidrometalúrgicos, donde múltiples variables operativas interactúan simultáneamente de manera no lineal. A diferencia de los enfoques estadísticos tradicionales como la regresión lineal simple, los modelos de Machine Learning son capaces de capturar relaciones complejas y no paramétricas entre las variables de entrada y la variable respuesta, sin imponer supuestos previos sobre la forma funcional de dichas relaciones. Esta capacidad los convierte en una plataforma ideal para el monitoreo predictivo y la toma de decisiones en tiempo real en plantas de electroobtención de zinc. La aplicación de estos modelos al circuito de la Refinería Cajamarquilla permitió determinar, con un alto nivel de precisión y robustez estadística, la influencia relativa de cada variable operativa sobre la concentración final de manganeso en electrolito agotado. El hallazgo principal y más relevante de este análisis es concluyente: la concentración final de manganeso en electrolito agotado depende fundamental y casi exclusivamente de la concentración de manganeso en solución pura, mientras que las variables operativas de las celdas electrolíticas no mostraron influencia significativa dentro de los rangos operativos estudiados. Desde una perspectiva de ingeniería de procesos, la recomendación más útil es que el control efectivo de la impureza de manganeso en el circuito de electroobtención debe centrarse en la etapa de purificación de la solución, aguas arriba de las celdas electrolíticas. En síntesis, el Machine Learning no solo ha permitido construir un modelo predictivo de alta exactitud para la concentración de manganeso, sino que ha proporcionado una evidencia cuantitativa para redirigir la estrategia de control de esta impureza hacia la etapa de purificación. Este enfoque predictivo y basado en datos se consolida como una herramienta robusta y valiosa para el monitoreo predictivo y la toma de decisiones en procesos de electroobtención de zinc.

7.2 Recomendaciones

Las siguientes recomendaciones se hacen para mejorar el proceso en planta.

7.2.1 Operativas:

Fortalecer el control del manganeso en la etapa de purificación de la solución pura, manteniendo su concentración preferentemente entre 2 y 5 g/L, con el fin de reducir la
acumulación de lodos anódicos y mejorar la eficiencia general del proceso.

7.2.2 Tecnológicas:

Implementar el sensor virtual basado en los modelos de Machine Learning desarrollados para el monitoreo en tiempo real de la concentración de manganeso en el electrolito agotado, reduciendo la dependencia de análisis químicos frecuentes.

7.2.3 De mejora continua:

Realizar ensayos piloto orientados a optimizar las condiciones anódicas (densidad de corriente, material de ánodos o aditivos) que favorezcan la oxidación de Mn^{2+} a MnO_2 y mejoren la remoción electroquímica de manganeso.

7.2.4 Investigativas

- Incorporar variables electroquímicas adicionales (potencial anódico, densidad de corriente real, estado de los ánodos) en futuros modelos.
- Aplicar validación cruzada temporal para evaluar la robustez de los modelos con nuevos datos operativos.
- Extender el estudio a otras impurezas críticas en el proceso de electroobtención de zinc.

VIII. BIBLIOGRAFÍA Y ANEXOS

Leguía Huamaccto, V. A. (2018). *Estudio técnico de ocurrencia de cortocircuitos en la electrodeposición de zinc para mejorar su eficiencia - Votorantim Metais S.A – Cajamarquilla*. [Tesis de pregrado, Universidad Nacional de San Cristóbal de Huamanga]. Citando a Votorantim Metais S.A.

Bautista, R. G. (2010). The Solution Chemistry of Manganese in the Zinc Electrowinning Process. En A. Siegmund, L. Centomarcelli, C. Ferron, C. J. Lim, & J. Ramachandran (Eds.), *Lead-Zinc 2010: Proceedings of the 6th International Symposium on Lead and Zinc Processing* (pp. 455-470). Springer.

Chen, A., Chen, X., & Wang, Z. (2019). Technological upgrade and sustainability of zinc hydrometallurgy in China: A review. *Journal of Cleaner Production*, *210*, 1480-1491.

González Bonilla, B. R., Dávila Pulido, G. I., Garza García, M., & Charles, D. A. (2023) *La Hidrometalurgia del zinc: una solución sostenible para concentrados minerales y residuos mineros. EPISTEMUS, e3806320.* https://doi.org/10.36790/epistemus.v19i38.320.

Instituto Geológico Minero y Metalúrgico. (2000). *Estudio de los recursos minerales del Perú: Franja n.º 1.* (Boletín n.º 8, Serie B: Geología Económica). https://www.gob.pe/ingemmet

Mahon, M., & Alfantazi, A. (2014). Manganese consumption during zinc electrowinning using a dynamic process simulation. *Hydrometallurgy, 146*, 94–103. https://doi.org/10.1016/j.hydromet.2014.03.004

Mahon, M., Wasik, L., & Alfantazi, A. (2012). Development and Implementation of a Zinc Electrowinning Process Simulation. *Journal of the Electrochemical Society*, *159 (8)*, 486-492. DOI: 10.1149/2.084208jes

Shang, K., Li, Z., Zhang, W., & Li, Y. (2025). A study on measures to preserve chlorine and ammonia oxygen removal. *Materials*, *18(6)*, 1347. DOI: 10.3390/ma18061347

Wark, I. W. (1979). The electrodeposition of zinc from acidified zinc sulphate solution. *Journal of Applied Electrochemistry, 9*(6), 721–730. https://doi.org/10.1007/BF00616002

Zhang, Q. B., & Hua, Y. (2014). Effect of Mn^{2+} ions on the electrodeposition of zinc from acidic sulphate solutions. *Hydrometallurgy, 144–145*, 110–117.

Yaghoubi, M. (2021). *Mn control in zinc electrowinning process by electrochemical means using Pb–Ag anode* (Tesis de maestría). Université du Québec, Canadá.

Biswas, A. K., & Davenport, W. G. (2013). *Extractive metallurgy of copper, gold and zinc* (2nd ed.). Elsevier.

Fouad, O. A., & El-Sayed, A. (2015). Hydrometallurgical processing of zinc. *Journal of Chemical and Pharmaceutical Research, 7*(3), 394–401.

Habashi, F. (1997). *Handbook of extractive metallurgy* (Vol. 2). Wiley-VCH.

Rosenqvist, T. (2004). *Principles of extractive metallurgy* (2nd ed.). Tapir Academic Press.

MacKinnon, D. J., & Brannen, J. M. (1981). The electrowinning of zinc. *Journal of Applied Electrochemistry, 11*(3), 285–291. https://doi.org/10.1007/BF00617112

Oh, S. J., & Yoo, K. (2016). Electrochemical behavior of zinc electrowinning in sulfate electrolytes. *Electrochimica Acta, 188*, 825–834. https://doi.org/10.1016/j.electacta.2015.12.074

Davey, T. W., & MacKinnon, D. J. (1987). The effect of impurities on zinc electrowinning. *Hydrometallurgy, 17*(2), 165–180. https://doi.org/10.1016/0304-386X(87)90030-1

Habashi, F. (2005). *Textbook of hydrometallurgy* (2nd ed.). Métallurgie Extractive Québec.

Pourbaix, M. (1974). *Atlas of electrochemical equilibria in aqueous solutions*. NACE International.

Morgan, J. J. (2005). Kinetics of reaction between O_2 and Mn(II) species in aqueous solutions. *Geochimica et Cosmochimica Acta, 69*(1), 35–48. https://doi.org/10.1016/j.gca.2004.06.010

Kelsall, G. H., Thompson, I., & Barnes, A. (2011). Manganese dioxide deposition during oxygen evolution in acidic electrolytes. *Electrochimica Acta, 56*(7), 2530–2539. https://doi.org/10.1016/j.electacta.2010.12.026

Hu, J., Li, J., & Li, Y. (2014). Anodic deposition of manganese dioxide in sulfate

electrolytes. *Journal of Electroanalytical Chemistry, 727*, 16–23. https://doi.org/10.1016/j.jelechem.2014.05.014

Bard, A. J., & Faulkner, L. R. (2001). *Electrochemical methods: Fundamentals and applications* (2nd ed.). Wiley.

De Oliveira, L. M. L., da Silva, F. L., Janissek, P. R., & Toniolo, J. C. (2025). Technological advances in electroplating: Artificial intelligence to predict zinc coating thickness on SAE 1008 low carbon steels. *Química Nova*, *48*(4), e-20250103, 1-10. https://doi.org/10.21577/0100-4042.20250103

Shang, X., Tan, Y., Yin, W., Pan, J., Wang, M., Wang, H., Yang, K., & Xiao, Q. (2026). Prediction of cadmium ion concentration in purification process of zinc hydrometallurgy via deep learning and singular spectrum analysis. *Chemical Engineering Science*, *331*, 124031. https://doi.org/10.1016/j.ces.2025.124031

Wang, W., Wu, Z., Peters, D., Citmaci, B., Morales-Guio, C. G., & Christofides, P. D. (2025). Machine learning in modeling, analysis and control of electrochemical reactors: A tutorial review. *Digital Chemical Engineering*, *15*, 100237. https://doi.org/10.1016/j.dche.2025.100237

Zhang, F., Li, J., Wei, Y., Dong, L., Wang, J., & Liu, J. (2024). Sustainable reduction of heavy metal pollution in zinc electrowinning: Insights from a cleaner production perspective. *Journal of Cleaner Production*, *470*, 143268. https://doi.org/10.1016/j.jclepro.2024.143268

Zhu, Y., Li, C., Tie, J., Meng, Y., Zhao, R., Wu, W., & Pen, X. (2026). Rapid online intelligent diagnostic method for inter-electrode short circuits in zinc electrowinning cells. *Measurement*, *258*, 119356. https://doi.org/10.1016/j.measurement.2025.119356

Schlesinger, M. E., King, M. J., Sole, K. C., & Davenport, W. G. (2011). *Extractive metallurgy of copper* (5th ed.). Elsevier. *(Capítulos aplicables a control de impurezas y electrólisis).*

Zhang, Y., & Cheng, C. Y. (2007). Review of impurity control in zinc electrowinning. *Hydrometallurgy, 89*(3–4), 193–202. https://doi.org/10.1016/j.hydromet.2007.07.002

González-Bonilla, B. R., Dávila-Pulido, G. I., Garza-García, M., & Charles, D.

A. (2025). Hidrometalurgia del zinc: una solución sostenible para concentrados minerales y residuos mineros. *Epistemus*, *19*(38). https://doi.org/10.36790/epistemus.v19i38.320

Newman, J., & Thomas-Alyea, K. E. (2012). *Electrochemical Systems (3rd ed.)* John Wiley & Sons.

Zhang, W., & Cheng, C. Y. (2007). Manganese metallurgy review. Part III: Manganese control in zinc and copper electrolytes. *Hydrometallurgy*, *89*(3–4), 178-188. https://doi.org/10.1016/j.hidromet.2007.08.011

Killi, M., & Tran, N. (2019). *Classification and Regression with Breiman Random Forests (Paper: Random Forests - Breiman 2001).*

Bockris, J. O'M., Reddy, A. K. N., & Gamboa-Aldeco, M. E. (2000). *Modern Electrochemistry* (2nd ed.). Plenum Press/Springer US.

Almeida, N.G. S., Sala, B. M., De Wet, J. R., Souza, A. D., Ciminelli, V. S. T., & Sett, D. (2025). Eliminating fluoride-induced corrosion of aluminum cathodes in zinc electrowinning by aluminum ion complexation. *Journal of Materials Research and Technology, 34*, 2116-2126. *https://doi.org/10.1016/j.jmrt.2024.12.179*

Ye, W., Xu, F., Jiang, L., Duan, N., Li, J., Zhang, F., & Chen, L. (2020). A novel functional lead-based anode for efficient lead dissolution inhibition and slime generation reduction in zinc electrowinning. *Journal of Cleaner Production*, *284*, 124767. https://doi.org/10.1016/j.jclepro.2020.124767

Anexo 1: Proceso resumido de las reacciones de Zn y Mn en el ánodo

1. Reacción principal de electroobtención de zinc

Cátodo

$$Zn^{2+} + 2e^- \rightarrow Zn(s) \quad \text{(A1)}$$

Ánodo

$$2H_2O \rightarrow O_2 + 4H^+ + 4e^- \quad \text{(A2)}$$

$$2Mn^{+2} + O^+ + 3H_2O \rightarrow 2MnO_2 + 6H^{+1} + 2e^- \quad \text{(A3)}$$

Reacción global de electroobtención de zinc

$$ZnSO4 + H_2O \xrightarrow[\text{electrólisis}]{} Zn(s) + H_2SO4 + \frac{1}{2}O_2 \quad \text{(A4)}$$

Estas son las reacciones fundamentales del proceso industrial de electroobtención de zinc en medio sulfato ácido

2. Oxidación anódica de Mn^{2+} y formación de MnO_2

La reacción generalmente aceptada para la oxidación anódica es:

$$Mn^{2+} + 2H_2O \rightarrow MnO_2(s) + 4H^+ + 2e^- \quad \text{(A5)}$$

Esta reacción ocurre simultáneamente con la evolución de oxígeno debido a que los potenciales de ambas reacciones son cercanos

3. Participación de Mn^{3+} como intermediario

La literatura electroquímica propone que la oxidación $Mn^{2+} \rightarrow MnO_2$ no ocurre en una sola etapa, sino mediante especies intermedias Mn^{3+} adsorbidas o disueltas:

$$Mn^{2+} \rightarrow Mn^{3+} \rightarrow MnO_2 \quad \text{(A6)}$$

Este mecanismo ha sido estudiado mediante espectroscopía UV-Vis in situ y técnicas electroquímicas.

es termodinámicamente posible a potenciales muy elevados, pero no constituye la reacción predominante en la electroobtención industrial de zinc.

Anexo 2: Etapa preliminar de entrenamiento de modelamiento en Machine Learning

1. Código de Excel Macro VBA para generación de datos teóricos

```
Sub
   GenerarHistorialElectrolisis_Corregido()
   Dim i As Integer
   Dim znPura As Double, densidad As Double, znEA As Double, acidez As
   Double Dim temp As Double, fe As Double, sb As Double, cu As Double,
   peso As Double

   ' Configuración de encabezados
   Cells(1, 1).Value = "ID"
   Cells(1, 2).Value = "Temp Electrolito
   (°C)" Cells(1, 3).Value = "Zn Sol Pura
   (g/L)" Cells(1, 4).Value = "Fe (mg/L)"
   Cells(1, 5).Value = "Sb (mg/L)"
   Cells(1, 6).Value = "Cu (mg/L)"
   Cells(1, 7).Value = "Densidad
   (g/L)" Cells(1, 8).Value = "Zn EA
   (g/L)" Cells(1, 9).Value =
   "Acidez (%)" Cells(1, 10).Value
   = "Horas"
   Cells(1, 11).Value = "Peso Depósito (kg)"

   ' Generar 1000
   registros For i = 2 To
   1001
      ' 1. Zn Solución Pura (145-155, con variaciones +/- 5)
      znPura = Round(140 + (Rnd * 20), 1)

      ' 2. Densidad (Directamente proporcional a Zn
      Pura) ' Rango objetivo: 1290 - 1300
      densidad = Round(1145 + znPura + (Rnd * 2 - 1), 0)

      ' 3. Zn en EA (Directamente proporcional a
      Densidad) ' Rango objetivo: 40 - 45
      znEA = Round((densidad - 1290) * 0.5 + 40 + (Rnd * 0.6 - 0.3), 1)

      ' 4. ACIDEZ (INVERSAMENTE PROPORCIONAL A LA DENSIDAD)
      ' Si densidad sube (ej. 1305), la acidez baja (ej.
      165) ' Si densidad baja (ej. 1285), la acidez
      sube (ej. 185)
      acidez = Round(175 - ((densidad - 1295) * 1.2) + (Rnd * 2 - 1), 1)

      ' 5. Otras variables
      temp = Round(35 + (Rnd * 7), 1) ' Rango
      36-41 fe = Round(Rnd * 6.5, 2)          '
      Límite 5.0
      sb = Round(Rnd * 0.045, 3) ' Límite 0.03
      cu = Round(Rnd * 0.28, 2)  ' Límite 0.2
```

```
peso = Round(78 + (Rnd * 9), 1) ' Rango 80-85

' Llenado de celdas
Cells(i, 1).Value = i - 1
Cells(i, 2).Value =
temp Cells(i, 3).Value =
znPura Cells(i,
4).Value = fe Cells(i,
5).Value = sb Cells(i,
6).Value = cu
Cells(i, 7).Value = densidad
```

```
    Cells(i, 8).Value =
    znEA Cells(i, 9).Value
    = acidez Cells(i,
    10).Value = 48 Cells(i,
    11).Value = peso
  Next i

  MsgBox "Historial generado con balance Densidad/Acidez corregido."
End Sub                                                              (A7)
```

2.Código de modelo XGBoost con datos teóricos:

```
#
==========================================
======= # MODELO DE MACHINE LEARNING -
CONTROL DE MANGANESO
# Tesis: Optimización del control de manganeso en electroobtención
de zinc #
==============================================
====

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import
train_test_split from sklearn.preprocessing
import StandardScaler from sklearn.ensemble
import IsolationForest import xgboost as xgb
from sklearn.metrics import mean_absolute_error, r2_score,
mean_squared_error import joblib
import warnings

warnings.filterwarnings('ignore')

print("Todas las librerías importadas correctamente")
print("Notebook listo para cargar los datos")
# ==========================================
# Cargar datos del Excel
# ==========================================

import os

# Verificar si el archivo existe en la carpeta actual
archivo = "Historial por turno electrólisis.xlsx.xlsx"

if os.path.exists(archivo):
  df = pd.read_excel(archivo, sheet_name="Hoja1")
  print(f"Archivo cargado correctamente: {df.shape[0]} filas y {df.shape[1]} columnas")

  # Mostrar columnas
  print("\nColumnas del
  dataset:")
```

```
print(df.columns.tolist())

# Verificar columnas clave
if 'ph Electrolito' in df.columns and 'Concen. Mn' in df.columns:
    print("\nLas columnas 'ph Electrolito' y 'Concen. Mn' fueron
encontradas.") else:
    print("\nNo se encontraron una o ambas columnas. Revisa los nombres exactos
en el Excel.")

# Estadísticas
print("\nEstadísticas de variables clave:")
print(df[['T de EA (°C)', 'Acidez (%)', 'ph Electrolito', 'Concen. Mn']].describe().round(3))
```

```
  print("\nPrimeras 5 filas:")
  print(df[['T de EA (°C)', 'Acidez (%)', 'ph Electrolito', 'Concen. Mn']].head())

else:
  print(f"ERROR: No se encontró el archivo '{archivo}' en esta carpeta.")
  print("Archivos que hay en esta carpeta:")
  print(os.listdir('.'))
# ============================================
# Análisis
# ============================================

# Renombrar
df = df.rename(columns={
  'ph Electrolito': 'pH_electrolito',
  'Concen. Mn':
  'Conc_MnO2_gL'
})

# Variables de entrada
features = ['T de EA (°C)', 'Zn Sol Pura (g/L)', 'Fe (mg/L)', 'Sb (mg/L)',
       'Cu (mg/L)', 'Densidad (g/L)', 'Zn EA (g/L)', 'Acidez
       (%)', 'Horas de depósito', 'Peso Depósito (kg)',
       'pH_electrolito']

target = 'Conc_MnO2_gL'

X =
df[features] y
= df[target]

print(f"Variables de entrada: {len(features)}
columnas") print(f"Variable objetivo: {target}\n")

# Estadísticas finales
print("Estadísticas de la variable objetivo
(Conc_Mn_EA_gL):") print(y.describe().round(3))

# Matriz de correlación
plt.figure(figsize=(10, 8))
corr_matrix = df[features + [target]].corr()
sns.heatmap(corr_matrix, annot=True, cmap='coolwarm', fmt='.2f', linewidths=0.5)
plt.title('Matriz de Correlación entre Variables')
plt.tight_layout()
plt.show()

print("\nCorrelaciones más fuertes con Conc_MnO2_gL:")
correlaciones = corr_matrix[target].abs().sort_values(ascending=False)
print(correlaciones.head(8))
# ============================================
# Preparación de datos
# ============================================
```

```
# Variables de entrada
features = ['T de EA (°C)', 'Zn Sol Pura (g/L)', 'Fe (mg/L)', 'Sb (mg/L)',
        'Cu (mg/L)', 'Densidad (g/L)', 'Zn EA (g/L)', 'Acidez
        (%)', 'Horas de depósito', 'Peso Depósito (kg)',
        'pH_electrolito']

target = 'Conc_MnO2_gL'

X =
df[features] y
= df[target]
```

```
print(f"Variables de entrada seleccionadas: {len(features)} variables")
print(f"Variable objetivo: {target}\n")

# Dividir los datos en entrenamiento (80%) y prueba (20%)
X_train, X_test, y_train, y_test = train_test_split(X, y,
                                   test_size=0.20,
                                   random_state=42)

print(f"Tamaño del conjunto de entrenamiento: {X_train.shape[0]}
muestras") print(f"Tamaño del conjunto de prueba: {X_test.shape[0]}
muestras\n")

# Escalar las variables (normalizar)
scaler = StandardScaler()
X_train_scaled =
scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)

print("Datos escalados correctamente")
print("  (Esto ayuda a que el modelo aprenda mejor)")

# Resumen final
print("\nResumen de la preparación:")
print(f"  → Total de datos: {len(df)}")
print(f"  → Datos para entrenar: {len(X_train)}
({len(X_train)/len(df)*100:.1f}%)") print(f"  → Datos para probar:
{len(X_test)} ({len(X_test)/len(df)*100:.1f}%)")
# ============================================
# Entrenamiento del modelo XGBoost
# ============================================

# Crear y entrenar el modelo XGBoost
modelo =
  xgb.XGBRegressor(
  n_estimators=500,
  learning_rate=0.05,
  max_depth=6,
  random_state=42,
  eval_metric='mae'
)

# Entrenar el modelo
modelo.fit(X_train_scaled, y_train)

print("Modelo XGBoost entrenado correctamente\n")

# Hacer predicciones en el conjunto de prueba
y_pred = modelo.predict(X_test_scaled)

# Evaluar el modelo
mae = mean_absolute_error(y_test,
y_pred)  rmse =
```

```
mean_squared_error(y_test, y_pred)**0.5
r2 = r2_score(y_test, y_pred)

print("RESULTADOS DEL MODELO:")
print(f"  Error Absoluto Medio (MAE)    : {mae:.3f} g/L")
print(f"  Raíz del Error Cuadrático Medio (RMSE):
{rmse:.3f} g/L") print(f"  Coeficiente de Determinación R² :
{r2:.4f} ({r2*100:.2f}%)")

# Interpretación
if r2 > 0.85:
   print("\nEl modelo tiene alta precisión.")
```

```
elif r2 > 0.70:
  print("\nEl modelo tiene buena
precisión.") else:
  print("\nEl R² es bajo. Podemos ajustar los parámetros del modelo.")
# ============================================
# Detector de Outliers (Verificar "realidad" de nuevos
datos) #
=======================================
======

# Entrenar el detector de outliers
detector_outliers = IsolationForest(
  contamination=0.05,  # 5% de los datos se consideran
  "raros" random_state=42
)

detector_outliers.fit(X_train_scaled)

print("Detector de outliers (Isolation Forest) entrenado
correctamente") print(" Este detector ayudará a identificar cuando
los parámetros nuevos") print(" se alejan demasiado de los datos
usados en el entrenamiento.\n")

# Ejemplo de cómo usarlo
ejemplo_outliers = detector_outliers.predict(X_test_scaled[:5])

print("Ejemplo de detección en datos de
prueba:") for i, pred in
enumerate(ejemplo_outliers):
  estado = "Dentro de la realidad" if pred == 1 else "Fuera de la realidad
  (outlier)" print(f"  Muestra {i+1}: {estado}")
# ============================================
# Función para hacer predicciones nuevas
# ============================================

def predecir_concentracion_mno2(nuevos_parametros):
  """
  Función para predecir la concentración de Mn usando nuevos
  parámetros. Incluye verificación de si los datos son similares a los
  del entrenamiento. """

  # Convertir el diccionario a DataFrame
  df_nuevo = pd.DataFrame([nuevos_parametros])

  # Asegurar que tenga exactamente las mismas columnas en el mismo orden
  df_nuevo = df_nuevo[features]

  # Escalar los nuevos datos
  nuevo_scaled = scaler.transform(df_nuevo)

  # Detectar si está dentro de la realidad
  es_outlier = detector_outliers.predict(nuevo_scaled)[0]

  # Hacer la predicción
```

```
mn_predicho = modelo.predict(nuevo_scaled)[0]

print("PREDICCIÓN DE CONCENTRACIÓN DE MANGANESO")
print("="*55)
print(f"• pH del electrolito: {nuevos_parametros.get('pH_electrolito', 'No
ingresado')}") print(f"• Temperatura: {nuevos_parametros.get('T de EA (°C)',
'No ingresado')} °C") print(f"• Acidez: {nuevos_parametros.get('Acidez (%)',
'No ingresado')} %")
print("-" * 55)
print(f"Concentración de Mn predicha: {mn_predicho:.2f} g/L")
```

```
  if es_outlier == 1:
    print("Los parámetros ingresados están dentro del rango de la base de
    datos.") print("  La predicción es **confiable**.")
  else:
    print(" ATENCIÓN: Los parámetros ingresados se alejan de los datos de
    entrenamiento.") print("  La predicción puede **no ser confiable**
    (extrapolación).")

  return mn_predicho


# ====================== EJEMPLO DE USO ======================
print("Ejemplo de predicción:\n")

ejemplo = {
  'T de EA (°C)': 40.0,
  'Zn Sol Pura (g/L)': 150.0,
  'Fe (mg/L)': 3.0,
  'Sb (mg/L)': 0.03,
  'Cu (mg/L)': 0.1,
  'Densidad (g/L)': 1295.0,
  'Zn EA (g/L)': 42.0,
  'Acidez (%)': 175.0,
  'Horas de depósito':
  48.0, 'Peso Depósito
  (kg)': 82.0,
  'pH_electrolito': 1.5
}

predecir_concentracion_mno2(ejemplo)
# ================================================
# Gráficos para la tesis
# ================================================

# 1. Gráfico: Predicciones vs Valores Reales (en conjunto de prueba)
y_pred = modelo.predict(X_test_scaled)

plt.figure(figsize=(10, 6))
plt.scatter(y_test, y_pred, alpha=0.6, color='blue')
plt.plot([y_test.min(), y_test.max()], [y_test.min(), y_test.max()], 'r--', lw=2)
plt.xlabel('Concentración de MnO2 Real (g/L)')
plt.ylabel('Concentración de MnO2 Predicha (g/L)')
plt.title('Predicciones vs Valores Reales\nModelo
XGBoost') plt.grid(True, alpha=0.3)
plt.show()

# 2. Importancia de las variables
importancia = pd.Series(modelo.feature_importances_, index=features)
importancia = importancia.sort_values(ascending=True)

plt.figure(figsize=(10, 8))
importancia.plot(kind='barh',
color='skyblue')
```

```
plt.title('Importancia de las Variables en la Predicción de MnO2')
plt.xlabel('Importancia Relativa')
plt.grid(True,
alpha=0.3)
plt.tight_layout()
plt.show()

print("Gráficos generados correctamente")
print("\nLas variables más importantes según el modelo son:")
```

```
print(importancia.sort_values(ascending=False).head(8))
# ============================================
# Resumen Final
# ============================================

# Guardar los modelos
joblib.dump(modelo, 'modelo_xgboost_mno2.pkl')
joblib.dump(scaler, 'scaler_mn.pkl')
joblib.dump(detector_outliers,
'detector_outliers_mno2.pkl') joblib.dump(features,
'features_mn.pkl')

# ================== RESUMEN FINAL ==================

print("="*60)
print("RESUMEN FINAL - MODELO DE MACHINE LEARNING")
print("="*60)
print(f"• Algoritmo utilizado     : XGBoost
Regressor") print(f"• Número de muestras totales
                                : {len(df)}") print(f"•
División Train/Test          : 80% / 20%") print(f"•
Error Absoluto Medio (MAE)   : {mae:.3f} g/L")
print(f"• RMSE                : {rmse:.3f} g/L")
print(f"• Coeficiente de Determinación R²  : {r2:.4f}
({r2*100:.2f}%)") print(f"• Variables de entrada utilizadas  :
{len(features)}")
print(f"• Variable objetivo      : Conc_Mn_EA_gL (Concentración de Mn en
electrolito agotado)")
print("-"*60)
print("Variables más importantes según el modelo:")
print(importancia.sort_values(ascending=False).head(6)
) print("="*60)

print("\nNotebook completado exitosamente.")
print("El modelo está listo para ser usado con nuevos parámetros.")
```
(A8)

Anexo 3: Código de modelo XGBoost con datos reales

```
# ============================================
# CELDA 1: Importación de librerías
# ============================================

import pandas as pd
import numpy as np
import matplotlib.pyplot as
plt import seaborn as sns

from sklearn.model_selection import
train_test_split from sklearn.preprocessing
import StandardScaler
from sklearn.metrics import mean_absolute_error, r2_score
```

```
import xgboost as xgb
import joblib
import warnings

warnings.filterwarnings('ignore')

print("Todas las librerías importadas correctamente")
```

```
# ============================================
# CELDA 2: Carga de Datos (Enfoque: Manganeso)
# ============================================

df = pd.read_excel(
    "Datos reales CJM ELE.xlsx",
    sheet_name="Hoja1"
)

print(
    f"Archivo cargado correctamente: "
    f"{df.shape[0]} filas y {df.shape[1]}
    columnas"
)

# Variable objetivo única: Mn EA
target = 'Mn EA (g/L)'

# Variables de entrada
features = [
    'T de EA (°C)',
    'Zn Sol Pura (g/L)',
    'Mn Sol Pura',
    'Fe (mg/L)',
    'Sb (mg/L)',
    'Cu (mg/L)',
    'Densidad (g/L)',
    'Acidez (%)',
    'Horas de
    depósito',
    'Peso Depósito (kg)'
]

X =
df[features] y
= df[target]

print(f"Variables de entrada: {len(features)}")
print(f"Variable objetivo: {target}")

# Rangos operativos de referencia
RANGO_MN_SOL_PURA = (5, 8)   #
g/L
RANGO_MN_EA = (12, 15)       # g/L

print("\n" + "=" * 60)
print("RANGOS OPERATIVOS DE REFERENCIA")
print("=" * 60)
```

```
print(
    f"Mn Sol Pura (entrada) : "
    f"{RANGO_MN_SOL_PURA[0]}  -  {RANGO_MN_SOL_PURA[1]}  g/L"
)
```

```
print(
   f"Mn EA (salida)      : "
   f"{RANGO_MN_EA[0]}  -  {RANGO_MN_EA[1]}  g/L"
)
# ============================================
# CELDA 3: Análisis Exploratorio de Datos (Mn)
# ============================================

print("Estadísticas descriptivas generales:")
print(df.describe().round(3))

corr_matrix = df.corr()

plt.figure(figsize=(12, 8))
sns.heatmap(corr_matrix,  annot=True,  cmap='coolwarm',  fmt='.2f',  linewidths=0.5)
plt.title('Matriz de Correlación - XGBoost (Mn)')
plt.show()

print("\n" + "="*60)
print("CORRELACIONES  CON  Mn  EA  (g/L)")
print("="*60)
print(corr_matrix['Mn  EA  (g/L)'].sort_values(ascending=False).round(4))

print("\n" + "="*60)
print("CORRELACIONES CON Mn Sol Pura")
print("="*60)
print(corr_matrix['Mn  Sol  Pura'].sort_values(ascending=False).round(4))

# Distribución de Mn Sol Pura y Mn EA respecto a sus rangos
fig, axes = plt.subplots(1, 2, figsize=(14, 5))

axes[0].hist(df['Mn Sol Pura'], bins=25, color='#4a9eff', edgecolor='white', alpha=0.8)
axes[0].axvline(RANGO_MN_SOL_PURA[0],  color='red',  linestyle='--',
label=f'Límite inf. ({RANGO_MN_SOL_PURA[0]})')
axes[0].axvline(RANGO_MN_SOL_PURA[1],  color='red',  linestyle='--',
label=f'Límite sup. ({RANGO_MN_SOL_PURA[1]})')
axes[0].set_title('Distribución Mn Sol Pura
(g/L)') axes[0].set_xlabel('Mn Sol Pura (g/L)')
axes[0].legend()

axes[1].hist(df['Mn EA (g/L)'], bins=25, color='#ff6b6b', edgecolor='white',
alpha=0.8) axes[1].axvline(RANGO_MN_EA[0],     color='red',      linestyle='--',
                                        label=f'Límite                  inf.
({RANGO_MN_EA[0]})')
axes[1].axvline(RANGO_MN_EA[1],  color='red',   linestyle='--',   label=f'Límite
                                        sup. ({RANGO_MN_EA[1]})')
axes[1].set_title('Distribución Mn EA
(g/L)') axes[1].set_xlabel('Mn EA (g/L)')
axes[1].legend()
```

```
plt.tight_layout()
```

```
plt.show()

# ============================================
# CELDA 4: Preparación de datos (Train + Test)
# ============================================

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,

random_state=42) scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)

print(f"Total de datos: {len(df)}")
print(f"Entrenamiento: {len(X_train)}
({len(X_train)/len(df)*100:.1f}%)")  print(f"Prueba: {len(X_test)}
({len(X_test)/len(df)*100:.1f}%)") print("Datos escalados
correctamente")

# ============================================
# CELDA 5: Entrenamiento XGBoost (Mn EA)
# ============================================

modelo_mn = xgb.XGBRegressor(n_estimators=400, learning_rate=0.05, max_depth=6,
random_state=42)
modelo_mn.fit(X_train_scaled, y_train)

# Predicciones
y_pred_mn = modelo_mn.predict(X_test_scaled)

# Evaluación
print("RESULTADOS DEL MODELO XGBoost - Mn EA")
print("="*50)
print(f"Mn EA → MAE: {mean_absolute_error(y_test, y_pred_mn):.3f} g/L | R²:
{r2_score(y_test, y_pred_mn):.4f}")

# ============================================
# CELDA 6: Importancia de Variables (Mn EA)
# ============================================

importancia_mn =
pd.Series(modelo_mn.feature_importances_,
index=features).sort_values(ascending=True)
plt.figure(figsize=(10, 6))
importancia_mn.plot(kind='barh', color='salmon')
plt.title('Importancia de Variables - Mn EA (XGBoost)')
plt.xlabel('Importancia')
plt.tight_layout()
plt.show()
```

```
# ============================================
# CELDA 7: Función de Predicción (Mn EA)
# ============================================
```

```
def predecir_mnea(nuevos_parametros):
    df_nuevo =
    pd.DataFrame([nuevos_parametros]) df_nuevo
    = df_nuevo[features]
    nuevo_scaled = scaler.transform(df_nuevo)

    mn_pred =

    modelo_mn.predict(nuevo_scaled)[0]

    print("PREDICCIÓN DEL PROCESO DE ELECTROOBTENCIÓN (XGBoost) -
Mn EA")
    print("="*70)
    print(f"Temperatura EA    : {nuevos_parametros.get('T de EA (°C)', 'N/A'):.1f}
    °C") print(f"Mn Solución Pura   : {nuevos_parametros.get('Mn Sol Pura', 0):.2f}
    g/L") print(f"Acidez   : {nuevos_parametros.get('Acidez (%)', 0):.2f} %")
    print("-"*70)
    print(f"Mn EA Predicho    : {mn_pred:.3f} g/L")

    # Verificación de rango Mn Sol Pura (entrada)
    mn_sol = nuevos_parametros.get('Mn Sol Pura', 0)
    if RANGO_MN_SOL_PURA[0] <= mn_sol <= RANGO_MN_SOL_PURA[1]:
       print(f"Mn Sol Pura está DENTRO del rango
                        esperado ({RANGO_MN_SOL_PURA[0]} -
{RANGO_MN_SOL_PURA[1]} g/L)")
    else:
     print(f"Mn Sol Pura FUERA del rango esperado ({RANGO_MN_SOL_PURA[0]} -
{RANGO_MN_SOL_PURA[1]}  g/L)")

    # Verificación de rango Mn EA (salida)
   if RANGO_MN_EA[0] <= mn_pred <= RANGO_MN_EA[1]:
     print(f"Mn EA está DENTRO del rango objetivo ({RANGO_MN_EA[0]} -
{RANGO_MN_EA[1]}  g/L)")
    else:
       print(f"Mn EA está FUERA del rango objetivo ({RANGO_MN_EA[0]} -
{RANGO_MN_EA[1]}  g/L)")
    print("="*70)

    return

    mn_pred

# Ejemplo de uso
ejemplo = {
    'T de EA (°C)': 37.5,
    'Zn Sol Pura (g/L)': 162.5,
    'Mn Sol Pura': 6.5,
    'Fe (mg/L)': 2.5,
```

```
    'Sb (mg/L)': 0.012,
    'Cu (mg/L)': 0.14,
    'Densidad (g/L)': 1302.0,
    'Acidez (%)': 188.0,
    'Horas de depósito':
    44.0, 'Peso Depósito
    (kg)': 73.0
}
```

```
predecir_mnea(ejemplo)

# ============================================
# CELDA 8: Gráfico de Evaluación (Mn EA)
# ============================================

y_pred_mn = modelo_mn.predict(X_test_scaled)

plt.figure(figsize=(8, 6))
plt.scatter(y_test, y_pred_mn, alpha=0.6, color='red', label='Predicciones')
plt.plot([y_test.min(), y_test.max()],
        [y_test.min(), y_test.max()], 'k--', lw=2, label='Línea Ideal')

# Banda del rango objetivo (12-15 g/L)
plt.axhspan(RANGO_MN_EA[0], RANGO_MN_EA[1], color='green', alpha=0.08,
          label=f'Rango objetivo ({RANGO_MN_EA[0]}-{RANGO_MN_EA[1]} g/L)')

plt.title('Predicciones vs Reales - Mn EA (XGBoost)')
plt.xlabel('Real')
plt.ylabel('Predicho')
plt.legend()
plt.grid(True,
alpha=0.3)
plt.tight_layout()
plt.show()

# ============================================
# CELDA 9: Exportación de Resultados (Mn EA)
# ============================================

y_pred_mn = modelo_mn.predict(X_test_scaled)

resultados = X_test.copy().reset_index(drop=True)

resultados['Mn EA Real (g/L)'] = y_test.values
resultados['Mn EA Predicho (g/L)'] = y_pred_mn
resultados['Error Mn EA'] = abs(resultados['Mn EA Real (g/L)'] - resultados['Mn
EA Predicho (g/L)'])

# Marcas de rango
resultados['Mn Sol Pura dentro rango (5-8)'] = resultados['Mn
                Sol Pura'].between(*RANGO_MN_SOL_PURA)
resultados['Mn EA Predicho dentro rango (12-15)'] = resultados['Mn EA Predicho
(g/L)'].between(*RANGO_MN_EA)

# Reordenar columnas
columnas = ['T de EA (°C)', 'Mn Sol Pura', 'Acidez (%)', 'Densidad (g/L)',
        'Horas de depósito', 'Peso Depósito (kg)', 'Zn Sol Pura (g/L)',
```

'Fe (mg/L)', 'Sb (mg/L)', 'Cu (mg/L)',
'Mn EA Real (g/L)', 'Mn EA Predicho (g/L)', 'Error Mn EA',

```
'Mn Sol Pura dentro rango (5-8)', 'Mn EA Predicho dentro rango (12-15)']

resultados = resultados[columnas]

resultados.to_excel("Predicciones_XGBoost_Mn_EA.xlsx", index=False)

print("Archivo 'Predicciones_XGBoost_Mn_EA.xlsx' exportado
correctamente")  print(f"  → {len(resultados)} registros guardados")

print("\nVista previa (primeras 8 filas):")
print(resultados.head(8).round(3))

# ================================================
# CELDA 10: Análisis de Casos Extremos (Mn EA Predicho)
# ================================================

df_analisis = resultados.copy()

percentil_bajo  = 10
percentil_alto  = 90

umbral_bajo = np.percentile(df_analisis['Mn EA Predicho (g/L)'], percentil_bajo)
umbral_alto = np.percentile(df_analisis['Mn EA Predicho (g/L)'], percentil_alto)

print("="*65)
print("DEFINICIÓN DE CASOS EXTREMOS - Mn EA Predicho (XGBoost)")
print("="*65)
print(f"Rango objetivo de referencia : {RANGO_MN_EA[0]}  -
{RANGO_MN_EA[1]} g/L")
print(f"Umbral extremo INFERIOR (percentil {percentil_bajo}%) :
{umbral_bajo:.3f} g/L")
print(f"Umbral extremo SUPERIOR (percentil {percentil_alto}%) : {umbral_alto:.3f}  g/L")

casos_bajos = df_analisis[df_analisis['Mn EA Predicho (g/L)'] <= umbral_bajo]
casos_altos = df_analisis[df_analisis['Mn EA Predicho (g/L)'] >= umbral_alto]
casos_normales = df_analisis[(df_analisis['Mn EA Predicho (g/L)'] > umbral_bajo) &
                             (df_analisis['Mn EA Predicho (g/L)'] < umbral_alto)]

print(f"\nRegistros en extremo INFERIOR :
{len(casos_bajos)}") print(f"Registros en extremo SUPERIOR
: {len(casos_altos)}") print(f'Registros en rango NORMAL
                                                :
{len(casos_normales)}")

variables_comparar = [c for c in features]

tabla_extremos = pd.DataFrame({
   'Promedio  General':  df_analisis[variables_comparar].mean(),
   'Promedio Caso BAJO (Mn EA ↓)':
```

```
    casos_bajos[variables_comparar].mean(), 'Promedio Caso ALTO (Mn EA
    ↑)': casos_altos[variables_comparar].mean(),
})
```

```
tabla_extremos['Δ% Caso BAJO'] = ((tabla_extremos['Promedio Caso BAJO (Mn EA
                        ↓)'] - tabla_extremos['Promedio General']) /
                        tabla_extremos['Promedio  General']  *  100)

tabla_extremos['Δ% Caso ALTO'] = ((tabla_extremos['Promedio Caso ALTO (Mn EA ↑)']
-
                        tabla_extremos['Promedio General']) /
                        tabla_extremos['Promedio General'] * 100)

tabla_extremos = tabla_extremos.round(3)

print("\n" + "="*65)
print("TABLA COMPARATIVA - VARIABLES DE ENTRADA EN CASOS
EXTREMOS DE Mn EA")
print("="*65)
print(tabla_extremos)

# ================================================
# CELDA 11: Variables Más Relevantes para Extremos de Mn EA
Predicho #
================================================

correlaciones_mn_ea = df_analisis[variables_comparar + ['Mn EA
                      Predicho (g/L)']].corr()['Mn EA Predicho (g/L)']
correlaciones_mn_ea = correlaciones_mn_ea.drop('Mn EA
                      Predicho (g/L)').sort_values(ascending=False)

print("="*65)
print("CORRELACIÓN DE VARIABLES DE ENTRADA CON Mn EA Predicho
(XGBoost)")
print("(valores cercanos a +1 = relación directamente
proporcional)") print("(valores cercanos a -1 = relación
inversamente proporcional)") print("="*65)
print(correlaciones_mn_ea.round(4))

def interpretar(corr):
    if corr >= 0.5:
        return "Fuerte directa ↑↑"
    elif corr >= 0.2:
        return "Moderada directa
    ↑" elif corr > -0.2:
        return "Débil / sin relación clara"
    elif corr > -0.5:
        return "Moderada inversa ↓"
    else:
        return "Fuerte inversa ↓↓"

tabla_interpretacion = pd.DataFrame({
    'Variable': correlaciones_mn_ea.index,
```

'Correlación con Mn EA Predicho': correlaciones_mn_ea.values.round(4),
'Relación': [interpretar(c) for c in correlaciones_mn_ea.values]

```
})

print("\n" + "="*65)
print("TABLA INTERPRETATIVA - RELACIÓN CON Mn EA Predicho")
print("="*65)
print(tabla_interpretacion.to_string(index=False))

plt.figure(figsize=(10, 6))
colores = ['#00c896' if c > 0 else '#ff6b6b' for c in correlaciones_mn_ea.values]
correlaciones_mn_ea.plot(kind='barh', color=colores)
plt.axvline(0, color='black', linewidth=0.8)
plt.title('Correlación de Variables de Entrada con Mn EA Predicho (XGBoost)')
plt.xlabel('Coeficiente de correlación')
plt.tight_layout()
plt.show()

top3 =
correlaciones_mn_ea.abs().sort_values(ascending=False).head(3)
print("\n" + "="*65)
print("TOP 3 VARIABLES MÁS RELEVANTES PARA LOS EXTREMOS DE Mn EA")
print("="*65)
for var in top3.index:
   valor_corr = correlaciones_mn_ea[var]
   direccion = "DIRECTAMENTE proporcional" if valor_corr > 0 else
"INVERSAMENTE  proporcional"
   print(f"• {var}: r = {valor_corr:.3f} → {direccion}")

# ================================================
# CELDA 12: Visualización de Variables Clave vs Mn EA Predicho
# ================================================

fig, axes = plt.subplots(1, 3, figsize=(18,

5)) for idx, var in enumerate(top3.index):
   ax = axes[idx]

   ax.scatter(casos_normales['Mn EA Predicho (g/L)'], casos_normales[var],
           alpha=0.4, color='gray', label='Normal', s=20)
   ax.scatter(casos_bajos['Mn EA Predicho (g/L)'], casos_bajos[var],
           alpha=0.8, color='#4a9eff', label='Extremo BAJO', s=30)
   ax.scatter(casos_altos['Mn EA Predicho (g/L)'], casos_altos[var],
           alpha=0.8, color='#ff6b6b', label='Extremo ALTO', s=30)

   ax.axvline(RANGO_MN_EA[0], color='green', linestyle='--', alpha=0.6, lw=1)
   ax.axvline(RANGO_MN_EA[1], color='green', linestyle='--', alpha=0.6, lw=1)

   ax.set_xlabel('Mn EA Predicho (g/L)')
   ax.set_ylabel(var)
   ax.set_title(f'{var} vs Mn EA Predicho\n(r = {correlaciones_mn_ea[var]:.3f})')
```

```
ax.legend(fontsize=8)
ax.grid(True,  alpha=0.3)
```

```
plt.tight_layout()
plt.show()

# ================================================
# CELDA 13: Tabla Resumen - Parámetros que Afectan Extremos de Mn EA (XGBoost) #
# ================================================

resumen_final = pd.DataFrame({
    'Variable / Parámetro':
    top3.index,
    'Correlación (r)': [round(correlaciones_mn_ea[v], 3) for v in top3.index],
    'Tipo de Relación': [
        "Directamente proporcional" if correlaciones_mn_ea[v] >
        0 else "Inversamente proporcional"
        for v in top3.index
    ],
    'Promedio en Extremo BAJO (Mn EA ↓)': [round(casos_bajos[v].mean(), 3) for v in top3.index],
    'Promedio en Extremo ALTO (Mn EA ↑)': [round(casos_altos[v].mean(), 3) for v in top3.index],
    'Interpretación':  [
        (f"Cuando {v} aumenta, Mn EA predicho tiende a
         aumentar" if correlaciones_mn_ea[v] > 0 else
         f"Cuando {v} aumenta, Mn EA predicho tiende a disminuir")
        for v in top3.index
    ]
})

print("="*80)
print("TABLA RESUMEN FINAL — VARIABLES QUE AFECTAN LOS EXTREMOS DE Mn EA PREDICHO (XGBoost)")
print("="*80)
print(resumen_final.to_string(index=False))

resumen_final.to_excel("Resumen_Extremos_Mn_EA_XGBoost.xlsx", index=False)
print("\nTabla exportada: Resumen_Extremos_Mn_EA_XGBoost.xlsx")

# ================================================
# CELDA 14: Análisis de Casos Extremos de Zn EA (variable real, sin predicción) #
# ================================================

# Incorporamos Zn EA real desde el dataset original
# (no se predice, se trabaja directamente con los valores reales del Excel)
df_zn = df.loc[X_test.index].copy().reset_index(drop=True)
df_zn['Mn EA Predicho (g/L)'] = resultados['Mn EA Predicho (g/L)'].values
```

 # Definimos extremos de Zn EA real por percentiles (mismo criterio que Mn EA)
percentil_bajo = 10
percentil_alto = 90

```
umbral_zn_bajo = np.percentile(df_zn['Zn EA (g/L)'], percentil_bajo)
umbral_zn_alto = np.percentile(df_zn['Zn EA (g/L)'], percentil_alto)

print("="*65)
print("DEFINICIÓN DE CASOS EXTREMOS - Zn EA Real")
print("="*65)
print(f"Umbral extremo INFERIOR (percentil {percentil_bajo}%) :
{umbral_zn_bajo:.3f} g/L")
print(f"Umbral extremo SUPERIOR (percentil {percentil_alto}%) :
{umbral_zn_alto:.3f} g/L")

casos_zn_bajos  = df_zn[df_zn['Zn EA (g/L)'] <= umbral_zn_bajo]
casos_zn_altos  = df_zn[df_zn['Zn EA (g/L)'] >= umbral_zn_alto]
casos_zn_normales = df_zn[(df_zn['Zn EA (g/L)'] > umbral_zn_bajo) &
                    (df_zn['Zn EA (g/L)'] < umbral_zn_alto)]

print(f"\nRegistros en extremo INFERIOR :
{len(casos_zn_bajos)}") print(f"Registros en extremo SUPERIOR
: {len(casos_zn_altos)}") print(f"Registros en rango NORMAL  :
{len(casos_zn_normales)}")

# Tabla comparativa de variables de entrada en extremos de Zn EA
tabla_zn = pd.DataFrame({
   'Promedio General':              df_zn[features].mean(),
   'Promedio Caso BAJO (Zn EA ↓)':  casos_zn_bajos[features].mean(),
   'Promedio Caso ALTO (Zn EA ↑)':  casos_zn_altos[features].mean(),
})

tabla_zn['Δ% Caso BAJO'] = ((tabla_zn['Promedio Caso BAJO (Zn EA
                      ↓)'] - tabla_zn['Promedio General']) /
                      tabla_zn['Promedio General']  * 100)

tabla_zn['Δ% Caso ALTO'] = ((tabla_zn['Promedio Caso ALTO (Zn EA
                      ↑)'] - tabla_zn['Promedio General']) /
                      tabla_zn['Promedio General'] * 100)

tabla_zn = tabla_zn.round(3)

print("\n" + "="*65)
print("TABLA COMPARATIVA - VARIABLES DE ENTRADA EN
                      CASOS EXTREMOS DE Zn EA")
print("="*65)
print(tabla_zn)

# ================================================
# CELDA 15: Correlación de variables con Zn EA Real
# ================================================
```

```
correlaciones_zn = df_zn[features + ['Zn EA (g/L)']].corr()['Zn EA (g/L)']
correlaciones_zn = correlaciones_zn.drop('Zn EA (g/L)').sort_values(ascending=False)
```

```
print("="*65)
print("CORRELACIÓN DE VARIABLES DE ENTRADA CON Zn EA Real")
print("(valores cercanos a +1 = relación directamente
proporcional)") print("(valores cercanos a -1 = relación
inversamente proporcional)") print("="*65)
print(correlaciones_zn.round(4))

def interpretar(corr):
    if corr >= 0.5:
        return "Fuerte directa ↑↑"
    elif corr >= 0.2:
        return "Moderada directa
    ↑" elif corr > -0.2:
        return "Débil / sin relación clara"
    elif corr > -0.5:
        return "Moderada inversa ↓"
    else:
        return "Fuerte inversa ↓↓"

tabla_zn_interp = pd.DataFrame({
    'Variable': correlaciones_zn.index,
    'Correlación con Zn EA':
    correlaciones_zn.values.round(4), 'Relación':
    [interpretar(c) for c in correlaciones_zn.values]
})

print("\n" + "="*65)
print("TABLA INTERPRETATIVA - RELACIÓN CON Zn EA Real")
print("="*65)
print(tabla_zn_interp.to_string(index=False)
)

plt.figure(figsize=(10, 6))
colores_zn = ['#00c896' if c > 0 else '#ff6b6b' for c in correlaciones_zn.values]
correlaciones_zn.plot(kind='barh', color=colores_zn)
plt.axvline(0, color='black', linewidth=0.8)
plt.title('Correlación de Variables de Entrada con Zn EA
Real') plt.xlabel('Coeficiente de correlación')
plt.tight_layout()
plt.show()

top3_zn =
correlaciones_zn.abs().sort_values(ascending=False).head(3)
print("\n" + "="*65)
print("TOP 3 VARIABLES MÁS RELEVANTES PARA LOS EXTREMOS DE Zn EA")
print("="*65)
for var in top3_zn.index:
    valor_corr = correlaciones_zn[var]
```

```
    direccion = "DIRECTAMENTE proporcional" if valor_corr > 0 else
"INVERSAMENTE  proporcional"
    print(f"• {var}: r = {valor_corr:.3f} → {direccion}")

# ================================================
```

```
# CELDA 16: Visualización Variables Clave vs Zn EA Real
# ============================================

fig, axes = plt.subplots(1, 3, figsize=(18,

5)) for idx, var in

enumerate(top3_zn.index):
   ax = axes[idx]

   ax.scatter(casos_zn_normales['Zn EA (g/L)'], casos_zn_normales[var],
            alpha=0.4, color='gray', label='Normal', s=20)
   ax.scatter(casos_zn_bajos['Zn EA (g/L)'], casos_zn_bajos[var],
            alpha=0.8, color='#4a9eff', label='Extremo BAJO', s=30)
   ax.scatter(casos_zn_altos['Zn EA (g/L)'], casos_zn_altos[var],
            alpha=0.8, color='#ff6b6b', label='Extremo ALTO', s=30)

   ax.set_xlabel('Zn EA Real (g/L)')
   ax.set_ylabel(var)
   ax.set_title(f'{var} vs Zn EA Real\n(r = {correlaciones_zn[var]:.3f})')
   ax.legend(fontsize=8)
   ax.grid(True,  alpha=0.3)

plt.tight_layout()
plt.show()

# ================================================
# CELDA 17: Comparación Cruzada Mn EA vs Zn EA
# ================================================

# Resumen Zn EA
resumen_zn = pd.DataFrame({
   'Variable / Parámetro':
   top3_zn.index,
   'Correlación con Zn EA (r)': [round(correlaciones_zn[v], 3) for v in top3_zn.index],
   'Tipo de Relación (Zn EA)': [
      "Directamente proporcional" if correlaciones_zn[v] >
      0 else "Inversamente proporcional"
      for v in top3_zn.index
   ],
   'Prom. Extremo BAJO (Zn EA ↓)': [round(casos_zn_bajos[v].mean(), 3) for v in
top3_zn.index],
   'Prom. Extremo ALTO (Zn EA ↑)': [round(casos_zn_altos[v].mean(), 3) for v in
top3_zn.index],
})

print("="*80)
print("TABLA RESUMEN — VARIABLES QUE AFECTAN LOS EXTREMOS DE
Zn EA")
```

```
print("="*80)
print(resumen_zn.to_string(index=False))
```

```
print("\n\n" + "="*80)
print("COMPARACIÓN CRUZADA: Variables relevantes para Mn EA vs Zn EA")
print("="*80)

# Unimos correlaciones de ambas variables en una sola tabla comparativa
todas_variables = list(set(list(top3.index) + list(top3_zn.index)))

comparacion = pd.DataFrame({
    'Variable': todas_variables,
    'Corr. con Mn EA': [round(correlaciones_mn_ea.get(v, float('nan')), 3) for v in
todas_variables],
    'Relación Mn EA': [interpretar(correlaciones_mn_ea.get(v, 0)) for v in
    todas_variables], 'Corr. con Zn EA':
        [round(correlaciones_zn.get(v, float('nan')), 3) for v in
todas_variables],
    'Relación Zn EA': [interpretar(correlaciones_zn.get(v, 0)) for v in todas_variables],
})

# Columna de coincidencia
def misma_direccion(r_mn,
    r_zn): if pd.isna(r_mn) or
    pd.isna(r_zn):
        return "—"
    if r_mn > 0.1 and r_zn > 0.1:
        return "Ambas suben juntas"
    elif r_mn < -0.1 and r_zn < -0.1:
        return "Ambas bajan juntas"
    elif abs(r_mn) < 0.1 and abs(r_zn) < 0.1:
        return "Sin relación en ninguna"
    elif abs(r_mn) > 0.1 and abs(r_zn) < 0.1:
        return "⚡ Solo afecta a Mn EA"
    elif abs(r_mn) < 0.1 and abs(r_zn) > 0.1:
        return "⚡ Solo afecta a Zn EA"
    else:
        return "Direcciones opuestas"

comparacion['Comportamiento conjunto'] = comparacion.apply(
    lambda row: misma_direccion(
        correlaciones_mn_ea.get(row['Variable'], 0),
        correlaciones_zn.get(row['Variable'], 0)
    ), axis=1
)

comparacion = comparacion.sort_values('Corr. con Mn EA', ascending=False)

print(comparacion.to_string(index=False))
```

```
# ── Gráfico comparativo de correlaciones ──────────────────────────────
all_vars = list(correlaciones_mn_ea.index)
```

```
r_mn_vals = [correlaciones_mn_ea[v] for v in all_vars]
r_zn_vals = [correlaciones_zn.get(v, 0) for v in all_vars]

x = range(len(all_vars))
width = 0.35

fig, ax = plt.subplots(figsize=(14, 6))
bars1 = ax.bar([i - width/2 for i in x], r_mn_vals, width,
        label='Correlación con Mn EA', color='#ff6b6b',
        alpha=0.8)
bars2 = ax.bar([i + width/2 for i in x], r_zn_vals, width,
        label='Correlación con Zn EA', color='#4a9eff',
        alpha=0.8)

ax.axhline(0, color='black', linewidth=0.8)
ax.set_xticks(list(x))
ax.set_xticklabels(all_vars, rotation=30, ha='right', fontsize=9)
ax.set_ylabel('Coeficiente de correlación')
ax.set_title('Comparación de Correlaciones: Variables de Entrada → Mn EA vs Zn
EA') ax.legend()
ax.grid(True, alpha=0.3, axis='y')
plt.tight_layout()
plt.show()

# Exportar tabla comparativa
comparacion.to_excel("Comparacion_Mn_EA_vs_Zn_EA.xlsx", index=False)
print("\nTabla comparativa exportada: Comparacion_Mn_EA_vs_Zn_EA.xlsx")
```

Anexo 4: Código de modelo Random Forest con datos reales

```
# ============================================
# CELDA 1: Importaciones
# ============================================

import pandas as pd
import numpy as np
import matplotlib.pyplot as
plt import seaborn as sns
from sklearn.model_selection import
train_test_split from sklearn.preprocessing import
StandardScaler from sklearn.ensemble import
RandomForestRegressor
from sklearn.metrics import mean_absolute_error,
r2_score import joblib

print("Librerías importadas correctamente")

# ============================================
```

```
# CELDA 2: Carga de Datos (Enfoque: Manganeso)
# ============================================

df = pd.read_excel("Datos reales CJM ELE.xlsx", sheet_name="Hoja1")
```

```
print(f"Archivo cargado: {df.shape[0]} filas y {df.shape[1]} columnas")

# Variable objetivo única: Mn EA
target = 'Mn EA (g/L)'

# Variables de entrada (todas excepto las de Zn EA y Mn EA)
features = ['T de EA (°C)', 'Zn Sol Pura (g/L)', 'Mn Sol Pura',
        'Fe (mg/L)', 'Sb (mg/L)', 'Cu (mg/L)',
        'Densidad (g/L)', 'Acidez (%)',
        'Horas de depósito', 'Peso Depósito (kg)']

X =
df[features] y
= df[target]

print(f"\nVariables de entrada ({len(features)}):
{features}") print(f"Variable objetivo: {target}")

# ── Rangos operativos de referencia
──────────────────────────────────────────
RANGO_MN_SOL_PURA = (5, 8)  # g/L -> entrada al proceso
RANGO_MN_EA     = (12, 15)  # g/L -> salida esperada del
proceso

print("\n" + "="*60)
print("RANGOS OPERATIVOS DE REFERENCIA")
print("="*60)
print(f"Mn Sol Pura (entrada) : {RANGO_MN_SOL_PURA[0]} -
{RANGO_MN_SOL_PURA[1]} g/L")
print(f"Mn EA (salida)      : {RANGO_MN_EA[0]} - {RANGO_MN_EA[1]} g/L")

# Estadísticas de los datos reales frente a los rangos
esperados print("\nEstadísticas de Mn Sol Pura en los
datos:") print(df['Mn Sol Pura'].describe().round(3))

print("\nEstadísticas de Mn EA (g/L) en los datos:")
print(df['Mn EA (g/L)'].describe().round(3))

# Porcentaje de registros dentro de cada rango
pct_sol_pura_ok = df['Mn Sol Pura'].between(*RANGO_MN_SOL_PURA).mean() *
100 pct_mn_ea_ok  = df['Mn EA (g/L)'].between(*RANGO_MN_EA).mean() * 100

print(f"\n% de registros con Mn Sol Pura dentro de {RANGO_MN_SOL_PURA}:
{pct_sol_pura_ok:.1f}%")
print(f"% de registros con Mn EA dentro de {RANGO_MN_EA}: {pct_mn_ea_ok:.1f}%")

# ============================================
# CELDA 3: Análisis Exploratorio (Mn)
```

```
# ================================================

print("Estadísticas descriptivas generales:")
print(df.describe().round(3))
```

```
# Correlaciones
corr_matrix = df.corr()

print("\n" + "="*60)
print("CORRELACIONES CON Mn EA (g/L)")
print("="*60)
print(corr_matrix['Mn EA (g/L)'].sort_values(ascending=False).round(4))

print("\n" + "="*60)
print("CORRELACIONES CON Mn Sol
Pura") print("="*60)
print(corr_matrix['Mn Sol Pura'].sort_values(ascending=False).round(4))

# Gráfico de correlación
plt.figure(figsize=(12, 8))
sns.heatmap(corr_matrix, annot=True, cmap='coolwarm', fmt='.2f', linewidths=0.5)
plt.title('Matriz de Correlación')
plt.tight_layout()
plt.show()

# Distribución de Mn Sol Pura y Mn EA respecto a sus
rangos fig, axes = plt.subplots(1, 2, figsize=(14, 5))

axes[0].hist(df['Mn Sol Pura'], bins=25, color='#4a9eff', edgecolor='white', alpha=0.8)
axes[0].axvline(RANGO_MN_SOL_PURA[0], color='red', linestyle='--',
label=f'Límite inf. ({RANGO_MN_SOL_PURA[0]})')
axes[0].axvline(RANGO_MN_SOL_PURA[1], color='red', linestyle='--',
label=f'Límite sup. ({RANGO_MN_SOL_PURA[1]})')
axes[0].set_title('Distribución Mn Sol Pura
(g/L)') axes[0].set_xlabel('Mn Sol Pura (g/L)')
axes[0].legend()

axes[1].hist(df['Mn EA (g/L)'], bins=25, color='#ff6b6b', edgecolor='white',
alpha=0.8) axes[1].axvline(RANGO_MN_EA[0], color='red', linestyle='--',
                                    label=f'Límite inf.
({RANGO_MN_EA[0]})')
axes[1].axvline(RANGO_MN_EA[1], color='red', linestyle='--', label=f'Límite
                                    sup. ({RANGO_MN_EA[1]})')
axes[1].set_title('Distribución Mn EA
(g/L)') axes[1].set_xlabel('Mn EA (g/L)')
axes[1].legend()

plt.tight_layout()
plt.show()

# ============================================
# CELDA 4: Preparación de Datos
```

```
# ================================================

X_train_val, X_test, y_train_val, y_test = train_test_split(X, y,
```

```
                                        test_size=0.20,
                                        random_state=42)

scaler = StandardScaler()
X_train_val_scaled = scaler.fit_transform(X_train_val)
X_test_scaled = scaler.transform(X_test)

print(f"Total datos: {len(df)}")
print(f"Entrenamiento + Validación: {len(X_train_val)}")
print(f"Prueba: {len(X_test)}")
print("Datos preparados y escalados")

# ================================================
# CELDA 5: Entrenamiento con Random Forest (Mn EA)
# ================================================

modelo_rf = RandomForestRegressor(
    n_estimators=400,
    max_depth=8,
    min_samples_split=2,
    random_state=42,
    n_jobs=-1
)

print("Entrenando Random Forest para predecir Mn EA...\n")
modelo_rf.fit(X_train_val_scaled, y_train_val)

# Predicciones
y_pred = modelo_rf.predict(X_test_scaled)

# Evaluación
mae = mean_absolute_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)

print(f"Mn EA (g/L):")
print(f"   MAE = {mae:.3f}")
print(f"   R²  = {r2:.4f}
({r2*100:.2f}%)") print("-" * 50)

# ================================================
# CELDA 6: Importancia de Variables (Mn EA)
# ================================================

importancia = pd.Series(modelo_rf.feature_importances_, index=features)
importancia = importancia.sort_values(ascending=False)

print("IMPORTANCIA DE VARIABLES (Random Forest - Mn EA)")
print(importancia.round(4))
```

 # Gráfico

```
plt.figure(figsize=(10, 6))
importancia.plot(kind='barh', color='#4a9eff')
plt.title('Importancia de Variables - Mn EA (Random Forest)')
plt.xlabel('Importancia')
plt.gca().invert_yaxis()
plt.tight_layout()
plt.show()

# ================================================
# CELDA 7: Función de Predicción (Mn EA)
# ================================================

def predecir_mnea(nuevos_parametros):
    df_nuevo =
    pd.DataFrame([nuevos_parametros]) df_nuevo
    = df_nuevo[features]

    nuevo_scaled = scaler.transform(df_nuevo)
    mn_ea_pred = modelo_rf.predict(nuevo_scaled)[0]

    print("PREDICCIÓN RANDOM FOREST - Mn EA")
    print("=" * 65)
    print(f"Temperatura EA            : {nuevos_parametros.get('T de EA (°C)',
    'N/A')} °C") print(f"Mn Sol Pura                    : {nuevos_parametros.get('Mn
    Sol Pura', 0):.2f} g/L") print(f"Acidez  : {nuevos_parametros.get('Acidez
    (%)', 0):.2f} %")
    print("-" * 65)
    print(f"Mn EA Predicho    : {mn_ea_pred:.3f}  g/L")

    # Verificación de rango Mn Sol Pura (entrada)
    mn_sol = nuevos_parametros.get('Mn Sol Pura', 0)
    if RANGO_MN_SOL_PURA[0] <= mn_sol <= RANGO_MN_SOL_PURA[1]:
        print(f"Mn Sol Pura está DENTRO del rango esperado
({RANGO_MN_SOL_PURA[0]} - {RANGO_MN_SOL_PURA[1]}  g/L)")
    else:
        print(f"Mn Sol Pura FUERA del rango esperado ({RANGO_MN_SOL_PURA[0]} -
{RANGO_MN_SOL_PURA[1]}  g/L)")

    # Verificación de rango Mn EA (salida)
    if RANGO_MN_EA[0] <= mn_ea_pred <= RANGO_MN_EA[1]:
        print(f"Mn EA está DENTRO del rango objetivo ({RANGO_MN_EA[0]} -
{RANGO_MN_EA[1]}  g/L)")
    else:
        print(f"Mn EA está FUERA del rango objetivo ({RANGO_MN_EA[0]} -
{RANGO_MN_EA[1]}  g/L)")
    print("=" * 65)

    return mn_ea_pred
```

```
# ============================================
# CELDA 8: Gráfico de Evaluación (Mn EA)
# ============================================
```

```
y_pred = modelo_rf.predict(X_test_scaled)

plt.figure(figsize=(8, 6))
plt.scatter(y_test, y_pred, alpha=0.6, color='#ff6b6b', label='Predicciones')
plt.plot([y_test.min(), y_test.max()],
        [y_test.min(),  y_test.max()],
        'k--', lw=2, label='Línea Ideal')

# Bandas del rango objetivo (12-15 g/L)
plt.axhspan(RANGO_MN_EA[0], RANGO_MN_EA[1], color='green', alpha=0.08,
          label=f'Rango objetivo ({RANGO_MN_EA[0]}-{RANGO_MN_EA[1]} g/L)')

plt.xlabel('Mn EA Real (g/L)')
plt.ylabel('Mn EA Predicho (g/L)')
plt.title('Predicciones vs Reales - Mn EA (Random Forest)')
plt.legend()
plt.grid(True, alpha=0.3)
plt.show()

print("Gráfico de evaluación generado para Mn EA")

# ================================================
# CELDA 9: Exportación de Resultados a Excel (Mn EA)
# ================================================

y_pred = modelo_rf.predict(X_test_scaled)

resultados =
X_test.copy().reset_index(drop=True)
resultados['Mn EA Real (g/L)']  = y_test.values
resultados['Mn EA Predicho (g/L)'] = y_pred
resultados['Error Mn EA (g/L)'] = abs(resultados['Mn EA Real (g/L)'] -
resultados['Mn EA Predicho (g/L)'])

# Marcas de rango
resultados['Mn Sol Pura dentro rango (5-8)']  = resultados['Mn Sol
Pura'].between(*RANGO_MN_SOL_PURA)
resultados['Mn EA Predicho dentro rango (12-15)'] = resultados['Mn EA Predicho
(g/L)'].between(*RANGO_MN_EA)

columnas_orden = [
   'T de EA (°C)', 'Mn Sol Pura', 'Acidez (%)',
   'Densidad (g/L)', 'Horas de depósito', 'Peso Depósito
   (kg)', 'Zn Sol Pura (g/L)', 'Fe (mg/L)', 'Sb (mg/L)', 'Cu
   (mg/L)',
   'Mn EA Real (g/L)', 'Mn EA Predicho (g/L)', 'Error Mn EA (g/L)',
   'Mn Sol Pura dentro rango (5-8)', 'Mn EA Predicho dentro rango (12-15)'
]
```

```
resultados = resultados[columnas_orden]
```

```
nombre_archivo = "Predicciones_Mn_EA_RandomForest.xlsx"
resultados.to_excel(nombre_archivo, index=False)

print(f"Archivo exportado correctamente:
**{nombre_archivo}**") print(f"  → {len(resultados)}
registros guardados")

print("\nVista previa (primeras 8 filas):")
print(resultados.head(8).round(3))

# ============================================
# CELDA 10: Análisis de Casos Extremos (Mn EA Predicho)
# ============================================

# Trabajamos sobre el conjunto de prueba con sus predicciones (celda 9)
df_analisis = resultados.copy()

# Definimos "extremo inferior" y "extremo superior" por
percentiles # sobre los valores PREDICHOS de Mn EA
percentil_bajo  = 10   # % inferior -> casos extremo bajo
percentil_alto  = 90   # % superior -> casos extremo alto

umbral_bajo = np.percentile(df_analisis['Mn EA Predicho (g/L)'], percentil_bajo)
umbral_alto = np.percentile(df_analisis['Mn EA Predicho (g/L)'], percentil_alto)

print("="*65)
print("DEFINICIÓN DE CASOS EXTREMOS - Mn EA Predicho")
print("="*65)
print(f"Rango objetivo de referencia : {RANGO_MN_EA[0]} -
{RANGO_MN_EA[1]} g/L")
print(f"Umbral extremo INFERIOR (percentil {percentil_bajo}%) : {umbral_bajo:.3f}
g/L")
print(f"Umbral extremo SUPERIOR (percentil {percentil_alto}%)  : {umbral_alto:.3f}
g/L")

# Subconjuntos de casos extremos
casos_bajos = df_analisis[df_analisis['Mn EA Predicho (g/L)'] <= umbral_bajo]
casos_altos = df_analisis[df_analisis['Mn EA Predicho (g/L)'] >= umbral_alto]
casos_normales = df_analisis[(df_analisis['Mn EA Predicho (g/L)'] > umbral_bajo) &
                    (df_analisis['Mn EA Predicho (g/L)'] < umbral_alto)]

print(f"\nRegistros en extremo INFERIOR :
{len(casos_bajos)}") print(f"Registros en extremo SUPERIOR
: {len(casos_altos)}") print(f"Registros en rango NORMAL
                                        :
{len(casos_normales)}")

# Variables de entrada a comparar (excluyendo las columnas de resultado/error)
```

```
variables_comparar = [c for c in features]  # las 10 variables de entrada originales

# Promedios por grupo
tabla_extremos = pd.DataFrame({
    'Promedio  General':  df_analisis[variables_comparar].mean(),
```

```
    'Promedio Caso BAJO (Mn EA ↓)':
    casos_bajos[variables_comparar].mean(), 'Promedio Caso ALTO (Mn EA
    ↑)': casos_altos[variables_comparar].mean(),
})

# Diferencia porcentual respecto al promedio general
tabla_extremos['Δ% Caso BAJO'] = ((tabla_extremos['Promedio Caso BAJO (Mn EA ↓)'] -
                                   tabla_extremos['Promedio General']) /
                                   tabla_extremos['Promedio General'] * 100)

tabla_extremos['Δ% Caso ALTO'] = ((tabla_extremos['Promedio Caso ALTO (Mn EA ↑)']
-
                                   tabla_extremos['Promedio General']) /
                                   tabla_extremos['Promedio General'] * 100)

tabla_extremos = tabla_extremos.round(3)

print("\n" + "="*65)
print("TABLA COMPARATIVA - VARIABLES DE ENTRADA EN CASOS
EXTREMOS DE Mn EA")
print("="*65)
print(tabla_extremos)

================================================
# CELDA 11: Variables Más Relevantes para Extremos de Mn EA
Predicho #
================================================

# Correlación de cada variable de entrada con Mn EA Predicho
correlaciones_mn_ea = df_analisis[variables_comparar + ['Mn EA Predicho
(g/L)']].corr()['Mn EA Predicho (g/L)']
correlaciones_mn_ea = correlaciones_mn_ea.drop('Mn EA Predicho
(g/L)').sort_values(ascending=False)

print("="*65)
print("CORRELACIÓN DE VARIABLES DE ENTRADA CON Mn EA Predicho")
print("(valores cercanos a +1 = relación directamente
proporcional)") print("(valores cercanos a -1 = relación
inversamente proporcional)") print("="*65)
print(correlaciones_mn_ea.round(4))

# Tabla resumen interpretativa
def interpretar(corr):
    if corr >= 0.5:
        return "Fuerte directa ↑↑"
    elif corr >= 0.2:
        return "Moderada directa ↑"
    elif corr > -0.2:
        return "Débil / sin relación
```

```
clara" elif corr > -0.5:
    return "Moderada inversa ↓"
else:
```

```
        return "Fuerte inversa ↓↓"

tabla_interpretacion = pd.DataFrame({
    'Variable': correlaciones_mn_ea.index,
    'Correlación con Mn EA Predicho': correlaciones_mn_ea.values.round(4),
    'Relación': [interpretar(c) for c in correlaciones_mn_ea.values]
})

print("\n" + "="*65)
print("TABLA INTERPRETATIVA - RELACIÓN CON Mn EA Predicho")
print("="*65)
print(tabla_interpretacion.to_string(index=False))

# Gráfico de barras de correlaciones
plt.figure(figsize=(10, 6))
colores = ['#00c896' if c > 0 else '#ff6b6b' for c in correlaciones_mn_ea.values]
correlaciones_mn_ea.plot(kind='barh', color=colores)
plt.axvline(0, color='black', linewidth=0.8)
plt.title('Correlación de Variables de Entrada con Mn EA Predicho')
plt.xlabel('Coeficiente de correlación')
plt.tight_layout()
plt.show()

# ── Top 3 variables más relevantes (por valor absoluto de correlación)
── top3 =
correlaciones_mn_ea.abs().sort_values(ascending=False).head(3)
print("\n" + "="*65)
print("TOP 3 VARIABLES MÁS RELEVANTES PARA LOS EXTREMOS DE Mn EA")
print("="*65)
for var in top3.index:
    valor_corr = correlaciones_mn_ea[var]
    direccion = "DIRECTAMENTE proporcional" if valor_corr > 0 else
"INVERSAMENTE proporcional"
    print(f"• {var}: r = {valor_corr:.3f} → {direccion}")

# ================================================
# CELDA 12: Visualización de Variables Clave vs Mn EA Predicho
# ================================================

fig, axes = plt.subplots(1, 3, figsize=(18, 5))

for idx, var in enumerate(top3.index):
    ax = axes[idx]

    # Puntos normales
    ax.scatter(casos_normales['Mn EA Predicho (g/L)'], casos_normales[var],
               alpha=0.4, color='gray', label='Normal', s=20)
    # Casos extremo bajo
```

```
ax.scatter(casos_bajos['Mn EA Predicho (g/L)'], casos_bajos[var],
        alpha=0.8, color='#4a9eff', label='Extremo BAJO', s=30)
# Casos extremo alto
```

```
    ax.scatter(casos_altos['Mn EA Predicho (g/L)'], casos_altos[var],
               alpha=0.8, color='#ff6b6b', label='Extremo ALTO', s=30)

    # Líneas de referencia del rango objetivo Mn EA (12-15)
    ax.axvline(RANGO_MN_EA[0], color='green', linestyle='--', alpha=0.6, lw=1)
    ax.axvline(RANGO_MN_EA[1], color='green', linestyle='--', alpha=0.6, lw=1)

    ax.set_xlabel('Mn EA Predicho (g/L)')
    ax.set_ylabel(var)
    ax.set_title(f'{var} vs Mn EA Predicho\n(r = {correlaciones_mn_ea[var]:.3f})')
    ax.legend(fontsize=8)
    ax.grid(True, alpha=0.3)

plt.tight_layout()
plt.show()

# ================================================
# CELDA 13: Tabla Resumen - Parámetros que Afectan Extremos de Mn EA #
# ================================================

resumen_final = pd.DataFrame({
    'Variable / Parámetro': top3.index,
    'Correlación (r)': [round(correlaciones_mn_ea[v], 3) for v in top3.index],
    'Tipo de Relación': [
        "Directamente proporcional" if correlaciones_mn_ea[v] > 0 else "Inversamente proporcional"
        for v in top3.index
    ],
    'Promedio en Extremo BAJO (Mn EA ↓)': [round(casos_bajos[v].mean(), 3) for v in top3.index],
    'Promedio en Extremo ALTO (Mn EA ↑)': [round(casos_altos[v].mean(), 3) for v in top3.index],
    'Interpretación': [
        (f"Cuando {v} aumenta, Mn EA predicho tiende a aumentar" if correlaciones_mn_ea[v] > 0 else
         f"Cuando {v} aumenta, Mn EA predicho tiende a disminuir") for v in top3.index
    ]
})

print("="*80)
print("TABLA RESUMEN FINAL — VARIABLES QUE AFECTAN LOS EXTREMOS DE Mn EA PREDICHO")
print("="*80)
print(resumen_final.to_string(index=False))

# Exportar tabla a Excel
```

```
resumen_final.to_excel("Resumen_Extremos_Mn_EA.xlsx", index=False)
print("\nTabla exportada: Resumen_Extremos_Mn_EA.xlsx")
```

```
# ============================================
# CELDA 14: Análisis de Casos Extremos de Zn EA (variable real, sin predicción) #
# ============================================

# Incorporamos Zn EA real desde el dataset original
# (no se predice, se trabaja directamente con los valores reales del Excel)
df_zn = df.loc[X_test.index].copy().reset_index(drop=True)
df_zn['Mn EA Predicho (g/L)'] = resultados['Mn EA Predicho (g/L)'].values

# Definimos extremos de Zn EA real por percentiles (mismo criterio que Mn EA)
percentil_bajo = 10
percentil_alto = 90

umbral_zn_bajo = np.percentile(df_zn['Zn EA (g/L)'], percentil_bajo)
umbral_zn_alto = np.percentile(df_zn['Zn EA (g/L)'], percentil_alto)

print("="*65)
print("DEFINICIÓN DE CASOS EXTREMOS - Zn EA Real")
print("="*65)
print(f"Umbral extremo INFERIOR (percentil {percentil_bajo}%) :
{umbral_zn_bajo:.3f} g/L")
print(f"Umbral extremo SUPERIOR (percentil {percentil_alto}%) :
{umbral_zn_alto:.3f} g/L")

casos_zn_bajos  = df_zn[df_zn['Zn EA (g/L)'] <= umbral_zn_bajo]
casos_zn_altos  = df_zn[df_zn['Zn EA (g/L)'] >= umbral_zn_alto]
casos_zn_normales = df_zn[(df_zn['Zn EA (g/L)'] > umbral_zn_bajo) &
                    (df_zn['Zn EA (g/L)'] < umbral_zn_alto)]

print(f"\nRegistros en extremo INFERIOR :
{len(casos_zn_bajos)}") print(f"Registros en extremo SUPERIOR
: {len(casos_zn_altos)}") print(f"Registros en rango NORMAL :
{len(casos_zn_normales)}")

# Tabla comparativa de variables de entrada en extremos de Zn EA
tabla_zn = pd.DataFrame({
   'Promedio General':          df_zn[features].mean(),
   'Promedio Caso BAJO (Zn EA ↓)': casos_zn_bajos[features].mean(),
   'Promedio Caso ALTO (Zn EA ↑)': casos_zn_altos[features].mean(),
})

tabla_zn['Δ% Caso BAJO'] = ((tabla_zn['Promedio Caso BAJO (Zn EA ↓)'] -
                    tabla_zn['Promedio General']) /
                    tabla_zn['Promedio General'] * 100)

tabla_zn['Δ% Caso ALTO'] = ((tabla_zn['Promedio Caso ALTO (Zn EA ↑)'] -
                    tabla_zn['Promedio General']) /
```

```
                        tabla_zn['Promedio General'] * 100)

tabla_zn = tabla_zn.round(3)
```

```
print("\n" + "="*65)
print("TABLA COMPARATIVA - VARIABLES DE ENTRADA EN CASOS
EXTREMOS DE Zn EA")
print("="*65)
print(tabla_zn)

# ================================================
# CELDA 15: Correlación de variables con Zn EA Real
# ================================================

correlaciones_zn = df_zn[features + ['Zn EA (g/L)']].corr()['Zn EA (g/L)']
correlaciones_zn = correlaciones_zn.drop('Zn EA (g/L)').sort_values(ascending=False)

print("="*65)
print("CORRELACIÓN DE VARIABLES DE ENTRADA CON Zn EA Real")
print("(valores cercanos a +1 = relación directamente
proporcional)") print("(valores cercanos a -1 = relación
inversamente proporcional)") print("="*65)
print(correlaciones_zn.round(4))

def interpretar(corr):
   if corr >= 0.5:
      return "Fuerte directa ↑↑"
   elif corr >= 0.2:
      return "Moderada directa ↑"
   elif corr > -0.2:
      return "Débil / sin relación
   clara" elif corr > -0.5:
      return "Moderada inversa ↓"
   else:
      return "Fuerte inversa ↓↓"

tabla_zn_interp = pd.DataFrame({
   'Variable': correlaciones_zn.index,
   'Correlación con Zn EA': correlaciones_zn.values.round(4),
   'Relación': [interpretar(c) for c in correlaciones_zn.values]
})

print("\n" + "="*65)
print("TABLA INTERPRETATIVA - RELACIÓN CON Zn EA Real")
print("="*65)
print(tabla_zn_interp.to_string(index=False
))

plt.figure(figsize=(10, 6))
colores_zn = ['#00c896' if c > 0 else '#ff6b6b' for c in correlaciones_zn.values]
correlaciones_zn.plot(kind='barh', color=colores_zn)
plt.axvline(0, color='black', linewidth=0.8)
```

```
plt.title('Correlación de Variables de Entrada con Zn EA
Real') plt.xlabel('Coeficiente de correlación')
plt.tight_layout()
```

```
plt.show()

top3_zn =
correlaciones_zn.abs().sort_values(ascending=False).head(3)
print("\n" + "="*65)
print("TOP 3 VARIABLES MÁS RELEVANTES PARA LOS EXTREMOS DE Zn EA")
print("="*65)
for var in top3_zn.index:
    valor_corr = correlaciones_zn[var]
    direccion = "DIRECTAMENTE proporcional" if valor_corr > 0 else
"INVERSAMENTE proporcional"
    print(f"• {var}: r = {valor_corr:.3f} → {direccion}")

# ================================================
# CELDA 16: Visualización Variables Clave vs Zn EA Real
# ================================================

fig, axes = plt.subplots(1, 3, figsize=(18, 5))

for idx, var in
    enumerate(top3_zn.index): ax =
    axes[idx]

    ax.scatter(casos_zn_normales['Zn EA (g/L)'], casos_zn_normales[var],
            alpha=0.4, color='gray', label='Normal', s=20)
    ax.scatter(casos_zn_bajos['Zn EA (g/L)'], casos_zn_bajos[var],
            alpha=0.8, color='#4a9eff', label='Extremo BAJO', s=30)
    ax.scatter(casos_zn_altos['Zn EA (g/L)'], casos_zn_altos[var],
            alpha=0.8, color='#ff6b6b', label='Extremo ALTO', s=30)

    ax.set_xlabel('Zn EA Real (g/L)')
    ax.set_ylabel(var)
    ax.set_title(f'{var} vs Zn EA Real\n(r = {correlaciones_zn[var]:.3f})')
    ax.legend(fontsize=8)
    ax.grid(True, alpha=0.3)

plt.tight_layout()
plt.show()

# ================================================
# CELDA 17: Comparación Cruzada Mn EA vs Zn EA
# ================================================

# Resumen Zn EA
resumen_zn = pd.DataFrame({
    'Variable / Parámetro':
    top3_zn.index,
    'Correlación con Zn EA (r)': [round(correlaciones_zn[v], 3) for v in
```

```
top3_zn.index], 'Tipo de Relación (Zn EA)': [
    "Directamente proporcional" if correlaciones_zn[v] >
    0 else "Inversamente proporcional"
    for v in top3_zn.index
],
```

```
    'Prom. Extremo BAJO (Zn EA ↓)': [round(casos_zn_bajos[v].mean(), 3) for v in
top3_zn.index],
    'Prom. Extremo ALTO (Zn EA ↑)': [round(casos_zn_altos[v].mean(), 3) for v in
top3_zn.index],
})

print("="*80)
print("TABLA RESUMEN — VARIABLES QUE AFECTAN LOS EXTREMOS DE
Zn EA")
print("="*80)
print(resumen_zn.to_string(index=False))

# ── Comparación cruzada
─────────────────────────────────────────────────────────

print("\n\n" + "="*80)
print("COMPARACIÓN CRUZADA: Variables relevantes para Mn EA vs Zn EA")
print("="*80)

# Unimos correlaciones de ambas variables en una sola tabla comparativa
todas_variables = list(set(list(top3.index) + list(top3_zn.index)))

comparacion = pd.DataFrame({
    'Variable': todas_variables,
    'Corr. con Mn EA': [round(correlaciones_mn_ea.get(v, float('nan')), 3) for v in
todas_variables],
    'Relación Mn EA':  [interpretar(correlaciones_mn_ea.get(v, 0)) for v in
    todas_variables], 'Corr. con Zn EA': [round(correlaciones_zn.get(v, float('nan')), 3)
    for v in
todas_variables],
    'Relación Zn EA':  [interpretar(correlaciones_zn.get(v, 0)) for v in todas_variables],
})

# Columna de coincidencia
def  misma_direccion(r_mn,  r_zn):
    if pd.isna(r_mn) or pd.isna(r_zn):
        return  "—"
    if r_mn > 0.1 and r_zn > 0.1:
        return "Ambas suben juntas"
    elif r_mn < -0.1 and r_zn < -0.1:
        return "Ambas bajan juntas"
    elif abs(r_mn) < 0.1 and abs(r_zn) < 0.1:
        return "Sin relación en ninguna"
    elif abs(r_mn) > 0.1 and abs(r_zn) < 0.1:
        return "⚡ Solo afecta a Mn EA"
    elif abs(r_mn) < 0.1 and abs(r_zn) > 0.1:
        return "⚡ Solo afecta a Zn EA"
    else:
        return  "Direcciones opuestas"
```

```
comparacion['Comportamiento conjunto'] = comparacion.apply(
    lambda row: misma_direccion(
```

```
        correlaciones_mn_ea.get(row['Variable'],   0),
        correlaciones_zn.get(row['Variable'],  0)
    ), axis=1
)

comparacion = comparacion.sort_values('Corr. con Mn EA', ascending=False)

print(comparacion.to_string(index=False))

# ── Gráfico comparativo de correlaciones
──────────────────────────────────────────

all_vars = list(correlaciones_mn_ea.index)
r_mn_vals = [correlaciones_mn_ea[v] for v in all_vars]
r_zn_vals = [correlaciones_zn.get(v, 0) for v in all_vars]

x = range(len(all_vars))
width = 0.35

fig, ax = plt.subplots(figsize=(14, 6))
bars1 = ax.bar([i - width/2 for i in x], r_mn_vals, width,
            label='Correlación con Mn EA', color='#ff6b6b',
            alpha=0.8)
bars2 = ax.bar([i + width/2 for i in x], r_zn_vals, width,
            label='Correlación con Zn EA', color='#4a9eff',
            alpha=0.8)

ax.axhline(0, color='black', linewidth=0.8)
ax.set_xticks(list(x))
ax.set_xticklabels(all_vars, rotation=30, ha='right', fontsize=9)
ax.set_ylabel('Coeficiente de correlación')
ax.set_title('Comparación de Correlaciones: Variables de Entrada → Mn EA vs Zn
EA') ax.legend()
ax.grid(True, alpha=0.3, axis='y')
plt.tight_layout()
plt.show()

# Exportar tabla comparativa
comparacion.to_excel("Comparacion_Mn_EA_vs_Zn_EA.xlsx", index=False)
print("\nTabla comparativa exportada: Comparacion_Mn_EA_vs_Zn_EA.xlsx")
```