

Modelo Analítico basado en Algoritmos Genéticos Difusos (FGA) para optimizar el proceso de conminución de minerales

Tema: Optimización de procesos en minería con analítica de datos

Leonardo Almeyda Tejada¹, Daniel Florencio Lovera Davila² y Clayton Jordan Iliquin Zavaleta³

¹ UNMSM, Lima, Perú, leonardo.almeyda@unmsm.edu.pe, +51 944 921 725

² UNMSM, Lima, Perú, dloverad@unmsm.edu.pe, +51 999 275 349

³ UNMSM, Lima, Perú, clayton.iliquin@unmsm.edu.pe, +51 978 604 206

RESUMEN

La eficiencia del proceso de molienda en operaciones minero-metalúrgicas está estrechamente ligada al control del tonelaje de alimentación, el caudal de agua y la granulometría del producto (P80). En este estudio se comparan tres enfoques de control inteligente: lógica difusa (FIS), sistemas neurodifusos (ANFIS) y motores genéticos difusos (FGA), aplicados al circuito de molienda de una unidad minera en operación. Cada modelo fue implementado en lenguaje Python y validado con datos reales, evaluando su capacidad para sugerir mejoras operativas sin comprometer la estabilidad del proceso.

Los resultados muestran que tanto el FIS como el ANFIS permitieron incrementar el tonelaje de alimentación en aproximadamente 1% en promedio, mientras que el FGA logró una mejora de 1.5%, con picos de hasta 5% en meses específicos. En términos de consumo hídrico, los tres modelos propusieron una reducción sostenida del caudal de agua, logrando una disminución promedio del 6%. Respecto al P80, se evidenció que los modelos FIS y ANFIS tienden a engrosar la granulometría del producto en su búsqueda por maximizar el tonelaje, mientras que el FGA mantuvo un mejor equilibrio entre productividad y calidad de molienda.

La comparación integral demuestra que los modelos basados en inteligencia computacional representan herramientas valiosas para la optimización de procesos industriales. En particular, el enfoque FGA destaca por su capacidad de adaptación, estabilidad en las recomendaciones y potencial para identificar configuraciones operativas óptimas no evidentes, incluso en entornos de alta variabilidad.

1. Introducción

La industria minera enfrenta desafíos estructurales cada vez más complejos, derivados de la disminución en las leyes del mineral, el aumento

sostenido de los costos energéticos, la variabilidad mineralógica y el incremento en el consumo de agua (Bascur et al., 2011). En este contexto, diversos autores como Concha y Bascur (2024) coinciden en que el futuro de la minería dependerá en gran medida de la implementación de modelos analíticos con capacidad de adaptación frente a cambios en la composición del mineral y en las condiciones de operación.

Dentro de este escenario, la conminución se consolida como uno de los subprocesos más críticos del procesamiento de minerales, dado que las condiciones granulométricas y reológicas generadas en esta etapa impactan de forma directa en la eficiencia de los procesos posteriores, afectando la recuperación metalúrgica y, por ende, la rentabilidad global de la operación.

Si bien se han desarrollado modelos analíticos orientados al control de la conminución, la mayoría de ellos opera bajo supuestos simplificados, considerando un conjunto limitado de variables y sin capturar completamente la dinámica no lineal e incierta del sistema. Estas limitaciones han incentivado la búsqueda de enfoques alternativos que integren el conocimiento experto con mecanismos de aprendizaje y mayor capacidad de adaptación ante la variabilidad del proceso.

En respuesta a esta necesidad, se han explorado diferentes tecnologías de control aplicadas al procesamiento de minerales. Candiotti et al. (2020) destacan entre las más utilizadas el control PID, los sistemas expertos basados en lógica difusa, y los modelos predictivos como el Model Predictive Control (MPC), tal como se ilustra en la Figura 1.

Figura 1
Tipos de Controladores más Utilizados en Molienda



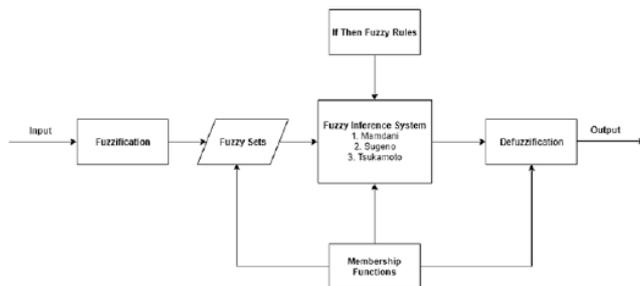
Nota. Extraída de Candiotti et al., 2020.

Dentro del ámbito de los sistemas expertos, Cipriano (2010) resalta que una de las técnicas más eficaces para manejar la incertidumbre inherente a los procesos es la lógica difusa.

Según Sivanandam et al. (2006), la lógica difusa fue introducida en 1965 por Lotfi Zadeh como una herramienta matemática para tratar la incertidumbre y la ambigüedad. En la misma línea, Mathworks (2020) la define como un método basado en reglas “si-entonces”, que transforma entradas numéricas en salidas lingüísticas a través de funciones de pertenencia y procesos de inferencia.

Adicionalmente, Singh y Ahmad (2020) explican que un Sistema de Inferencia Difusa (Fuzzy Inference System, FIS), como se muestra en la Figura 2, busca emular el razonamiento humano para resolver problemas complejos sin necesidad de una formulación matemática precisa.

Figura 2
Sistema de Inferencia Difusa (FIS)



Nota. Extraído de Singh y Ahmad (2020).

Un FIS se compone comúnmente de:

- **Fuzzificación:** convierte entradas numéricas en grados de pertenencia.
- **Base de reglas:** contiene el conocimiento experto expresado como reglas lingüísticas.
- **Motor de inferencia:** evalúa las reglas activas.
- **Defuzzificación:** transforma la salida difusa en un valor numérico.

En cuanto a los tipos de FIS, Mathworks (2020) distingue principalmente dos modelos: Mamdani y Takagi–Sugeno. La Tabla 1 resume sus principales ventajas y aplicaciones.

Tabla 1
Comparación entre Sistemas de Inferencia Difusa (FIS)

FIS	Ventajas
Mamdani	<ul style="list-style-type: none"> ▪ Intuitivo. ▪ Adaptado a la intervención humana. ▪ Base de reglas más interpretable. ▪ Tiene una amplia aceptación.
Sugeno	<ul style="list-style-type: none"> ▪ Computacionalmente eficiente. ▪ Funciona bien con técnicas lineales, como el control PID. ▪ Trabaja bien con técnicas de optimización y adaptación. ▪ Muy adecuado para el análisis matemático.

Nota. Adaptado de Mathworks (2020)

Para la implementación del FIS en este trabajo se utilizó FuzzyLite, una biblioteca desarrollada por Rada-Vilela (2015) para construir motores difusos de alto rendimiento tanto en C++ como en Python. Esta herramienta permite simular de forma eficiente sistemas tipo Mamdani o Takagi–Sugeno sin necesidad de desarrollar un motor desde cero. Sus funcionalidades están organizadas en componentes según se detalla en la Tabla 2.

Tabla 2
Componentes principales de la librería FuzzyLite

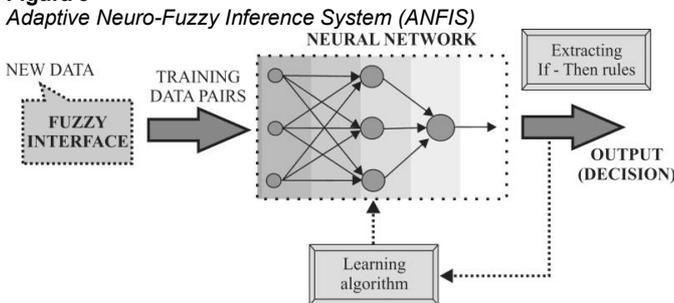
Componente	Descripción
<i>Engine</i>	Es el núcleo del sistema difuso. Define entradas, salidas, reglas y método de inferencia.
<i>InputVariable</i>	Representa una variable de entrada con un rango definido y funciones de membresía.
<i>OutputVariable</i>	Variable de salida que puede operar en modo Mamdani (con superficie de salida) o Takagi-Sugeno (valor numérico).
<i>Term</i>	Define funciones de membresía como <i>Triangle</i> , <i>Trapezoid</i> , <i>Gaussian</i> , <i>Bell</i> , etc.
<i>RuleBlock</i>	Conjunto de reglas difusas definidas de forma lingüística (<i>if-then</i>).

<i>Rule</i>	Expresión lógica que conecta términos de entrada con una conclusión de salida.
<i>Defuzzifier</i>	Método para convertir una salida difusa en una salida numérica (ej. <i>Centroid, Bisector</i>).

No obstante, una limitación fundamental de los FIS tradicionales radica en que tanto las reglas como las funciones de membresía deben ser definidas manualmente por expertos. Este proceso, además de ser intensivo en conocimiento, restringe la capacidad de adaptación del sistema frente a variaciones en los datos o cambios operacionales. Ante este desafío, surgen modelos híbridos que integran capacidades de aprendizaje automático, destacando entre ellos el sistema de inferencia neurodifuso ANFIS.

El Adaptive Neuro-Fuzzy Inference System (ANFIS) es un modelo que combina la representación flexible e interpretable de los sistemas de lógica difusa con la capacidad de aprendizaje de las redes neuronales artificiales. Propuesto por Jang (1993), ANFIS ha sido ampliamente aplicado en problemas de predicción, clasificación y control, especialmente en entornos donde se requiere manejar relaciones no lineales y variables inciertas de manera eficiente (ver Figura 3).

Figura 3



Nota. Extraído de Sremac et al. (2018).

La arquitectura clásica de ANFIS se compone de cinco capas conectadas en red, las cuales simulan el comportamiento de un sistema difuso tipo Takagi–Sugeno. Estas capas incluyen:

1. *Capa de entrada difusa*: calcula los grados de pertenencia de las variables de entrada mediante funciones de membresía (gaussianas o triangulares).
2. *Capa de reglas*: construye combinaciones de entradas para formar reglas difusas.
3. *Capa de normalización*: normaliza los niveles de activación de cada regla.
4. *Capa de consecuencia*: aplica funciones lineales que definen la salida de cada regla.

5. *Capa de agregación/salida*: integra las salidas de todas las reglas activas para obtener una única predicción continua.

La principal ventaja de ANFIS frente a los FIS clásicos es su capacidad para aprender automáticamente los parámetros de las funciones de membresía y los coeficientes de salida a partir de datos históricos. Esto lo convierte en una herramienta valiosa cuando el conocimiento experto es limitado o cuando el proceso presenta una alta variabilidad operativa.

Para la implementación práctica, este trabajo adoptó PyTorch como plataforma de desarrollo, en lugar de bibliotecas especializadas como *anfis* o *simpful*, debido a su mayor estabilidad, flexibilidad y compatibilidad con entornos modernos. PyTorch, desarrollado por Facebook AI Research, ofrece una estructura modular que permite construir redes neuronales personalizadas, facilitando la creación de arquitecturas híbridas como ANFIS desde cero.

En una implementación personalizada, ANFIS se construye utilizando los bloques fundamentales de PyTorch, dado que no existe un módulo nativo para este tipo de sistema. La Tabla 3 resume los componentes principales utilizados en este trabajo para modelar ANFIS en PyTorch.

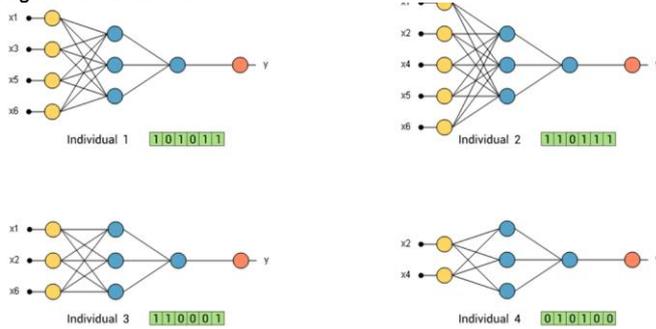
Tabla 3
Componentes de PyTorch en la construcción de ANFIS

Componente	Rol en ANFIS
<i>nn.Module</i> (AnfisLayer)	Clase que define el modelo ANFIS. Utiliza tensores para representar parámetros de funciones de membresía (μ y σ) y coeficientes de salida.
Funciones de membresía (Gaussiana)	Implementadas mediante operaciones tensoriales como <i>torch.exp</i> para activaciones suaves.
Normalización difusa	Uso de <i>torch.prod</i> y <i>torch.sum</i> para calcular la fuerza de activación y su normalización entre reglas.
Aprendizaje supervisado	Entrenamiento mediante <i>MSELoss</i> y optimización con <i>Adam</i> , permitiendo ajustar los parámetros del sistema neurodifuso.

A pesar de las ventajas que ofrece ANFIS, su desempeño puede verse condicionado por la calidad de los valores iniciales asignados a las

funciones de membresía y las reglas de inferencia. Aunque estos parámetros pueden ser ajustados mediante entrenamiento supervisado, la convergencia hacia soluciones óptimas no siempre está garantizada, especialmente en espacios de búsqueda complejos y altamente no lineales. En este contexto, surge una alternativa robusta y flexible: los algoritmos genéticos difusos (Fuzzy Genetic Algorithms, FGA).

Figura 4
Algoritmos Genéticos



Nota. Extraído de Gómez (2023).

Los FGA, mostrados en la Figura 4, integran dos paradigmas fundamentales de la inteligencia computacional: los sistemas de lógica difusa (Fuzzy Logic Systems, FLS), que modelan incertidumbre y conocimiento experto, y los algoritmos genéticos (Genetic Algorithms, GA), que permiten realizar procesos de búsqueda y optimización inspirados en la evolución natural. Esta combinación resulta particularmente útil para ajustar de forma automática los parámetros clave de un sistema difuso, tales como las funciones de pertenencia, las reglas tipo Takagi–Sugeno o los coeficientes de salida.

En un enfoque FGA típico, los elementos del sistema difuso son codificados como individuos dentro de una población evolutiva, los cuales se someten a un proceso iterativo de mejora utilizando operadores genéticos. Los principales conceptos que estructuran un algoritmo FGA son:

- **Individuos:** representaciones codificadas de una solución candidata (por ejemplo, los parámetros de funciones de membresía o reglas difusas).
- **Población:** conjunto de individuos evaluados y evolucionados en cada generación.
- **Función de aptitud (fitness):** métrica que evalúa el rendimiento de cada individuo, generalmente a partir del error cuadrático medio (MSE), precisión o alguna otra métrica objetivo.

El proceso de un FGA puede resumirse en los siguientes pasos:

1. Inicializar una población aleatoria de individuos.
2. Evaluar el desempeño de cada individuo mediante la función de aptitud.
3. Seleccionar a los individuos con mejor desempeño.
4. Aplicar operadores genéticos (cruce, mutación) para generar nuevas soluciones.
5. Repetir el ciclo hasta alcanzar un criterio de parada, como una generación máxima o una mejora mínima.

Para la implementación práctica de este enfoque, se utilizó la librería DEAP (Distributed Evolutionary Algorithms in Python), una herramienta ampliamente reconocida por su versatilidad para desarrollar algoritmos evolutivos personalizados. DEAP fue introducida por Fortin et al. (2012) y se caracteriza por su arquitectura modular, lo que facilita la integración de mecanismos evolutivos en distintos tipos de problemas, incluyendo la optimización de sistemas difusos.

Tabla 4
Componentes principales de DEAP

Componente	Descripción
Creator	Permite definir la estructura de los individuos y su función objetivo (fitness).
base.Toolbox	Es el núcleo de DEAP. Aquí se registran las funciones que definen: <ul style="list-style-type: none"> ▪ Generación de atributos. ▪ Inicialización de individuos y poblaciones. ▪ Función de evaluación. ▪ Operadores de cruce, mutación y selección. ▪ Estrategias evolutivas.
tools	Contiene los operadores: <ul style="list-style-type: none"> ▪ cxBlend: cruce por mezcla ▪ mutGaussian: mutación gaussiana ▪ selTournament: selección por torneo
algorithms	Implementa algoritmos clásicos como eaSimple, eaMuPlusLambda, etc., que controlan el ciclo evolutivo (evaluación, selección, reproducción).

El uso de FGA permite una optimización evolutiva sin necesidad de conocimiento experto detallado. A diferencia de los FIS, que requieren la definición manual de reglas y funciones, y de ANFIS, que depende del gradiente para el ajuste de parámetros, los FGA pueden explorar el espacio de

soluciones de manera más amplia y con menor sensibilidad a las condiciones iniciales.

A lo largo del desarrollo del presente trabajo se ha analizado una evolución progresiva en la sofisticación de los sistemas inteligentes aplicados al control y optimización de procesos metalúrgicos. Para facilitar la comparación entre estos tres enfoques —FIS, ANFIS y FGA— se presenta a continuación una tabla resumen con los criterios más relevantes desde el punto de vista práctico y computacional.

Tabla 5
Comparación entre modelos FIS, ANFIS y FGA.

Criterio	Motor Difuso (FuzzyLite)	Motor Neurodifuso (ANFIS)	Motor Genético Difuso (FGA)
Modelo base	Takagi-Sugeno (definido manualmente)	Takagi-Sugeno (aprendido automáticamente)	FIS optimizado mediante GA
Fuente de información	Experto humano (reglas definidas manualmente)	Datos históricos (entrenamiento supervisado)	Datos históricos (optimización evolutiva)
Entrenamiento	No requiere (solo definición de reglas y membresías)	Requiere entrenamiento (gradiente descendente)	Requiere evolución de poblaciones (DEAP)
Adaptabilidad	Limitada (reglas fijas)	Alta (ajuste de parámetros de funciones de membresía)	Alta (optimiza funciones y relaciones mediante evolución)
Dependencia de parámetros iniciales	Alta (definición experta es crítica)	Media (requiere valores iniciales, pero aprende)	Baja (evoluciona desde poblaciones aleatorias)
Tipo de salida	Determinística (funciones lineales en Takagi-Sugeno)	Aproximación funcional (regresión difusa)	Aproximación funcional (por coeficientes optimizados)
Librerías utilizadas	Fuzzylite	PyTorch, Simpful	DEAP

2. Objetivos

- Analizar los principales factores operacionales y que influyen en la eficiencia del proceso de conminución, con énfasis en su relación con la granulometría.
- Comparar el desempeño del modelo propuesto frente a enfoques tradicionales como lógica difusa pura y redes neuronales - ANFIS, evaluando métricas de precisión, interpretabilidad y robustez.
- Validar el modelo con datos reales obtenidos de operaciones minero-metalúrgicas, evaluando su capacidad para predecir y optimizar el tonelaje, tamaño de partícula (P80/F80) y cantidad de agua consumida.

3. Diagrama de Flujo e Información Disponible

En una encuesta realizada por Wei y Craig (2009, como se citó en Candiotti et al., 2020), se identificaron las principales variables consideradas en estrategias de mejora para circuitos de molienda cerrada. Tal como se observa en la Figura 5, el molino es abordado como un sistema de múltiples entradas y salidas, el cual puede descomponerse en subprocesos

Figura 5
Variables Utilizadas en una Estrategia de Control para Molienda

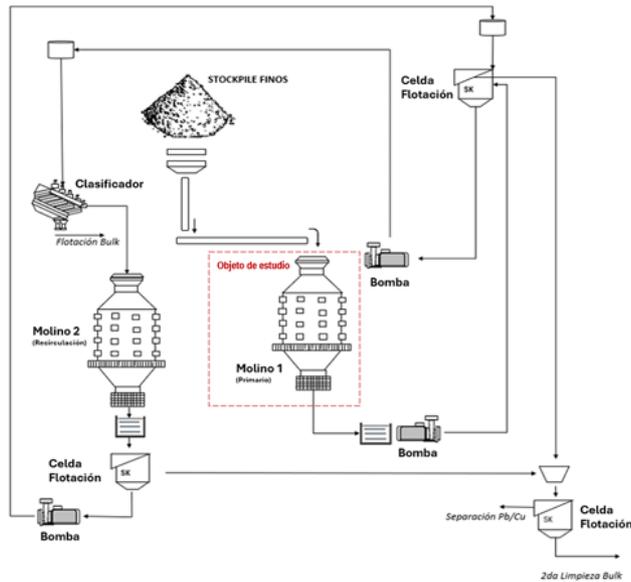


Nota. Extraída de Candiotti et al. (2020)

Dentro de este enfoque, los autores introdujeron el concepto de variables manipulables del proceso, es decir, aquellas variables que pueden ser ajustadas para influir en el comportamiento global del circuito. Entre las más relevantes se encuentran:

- Potencia del molino (kW).
- Ingreso de mineral (t/h).
- Tamaño de partícula de entrada.
- Porcentaje y volumen de bolas (% , m³).
- Velocidad de operación (% de la velocidad crítica).
- Caudal de agua de alimentación (m³/h).
- Carga circulante (m³/h).
- Nivel de llenado del sumidero (m³).
- Densidad de pulpa al ingreso del ciclón (kg/m³).
- Densidad de pulpa en la carga circulante (kg/m³).

Figura 6
Diagrama de Flujo de la UM



Nota. Extraído de Almeida (2025).

Tomando como referencia dicho marco conceptual, en este trabajo se empleó información operativa proveniente de una unidad minera (UM), cuya configuración del circuito de molienda se muestra en la Figura 6. Este circuito está compuesto por dos molinos: un molino primario (Molino 1) y un molino secundario (Molino 2).

A la entrada del Molino 1 se encuentra instalado el analizador de granulometría F80Meter, una cámara que emplea inteligencia artificial para estimar en tiempo real la distribución granulométrica del material de alimentación. Además, el molino cuenta con una bomba regulable que permite ajustar el caudal de agua (m^3/h), modificando así las condiciones operativas según los requerimientos del proceso.

La pulpa generada por el Molino 1 se descarga en un cajón que alimenta a dos bombas, las cuales impulsan el flujo hacia un conjunto de celdas de flotación Flash. Las colas generadas por estas celdas son dirigidas a las zarandas de alta frecuencia (ZAF). El producto pasante de las ZAF es enviado al circuito de flotación Bulk, mientras que el material retenido, en forma de pulpa, es redirigido al Molino 2 para su remolienda.

Posteriormente, la pulpa del Molino 2 es descargada en un segundo cajón, desde donde es bombeada nuevamente hacia un sistema de flotación Flash. En esta etapa, tanto las espumas obtenidas de esta flotación como las espumas generadas previamente en el Molino 2, se combinan en un cajón común, el cual alimenta una tercera etapa de flotación Flash. Finalmente, las espumas de esta última flotación son dirigidas al

proceso de separación Pb/Cu, mientras que las colas alimentan una etapa de limpieza Bulk secundaria.

A partir del circuito descrito, se estructuró un conjunto de variables operacionales que sirvieron como entrada para los modelos de inferencia diseñados. Estas variables se detallan en la Tabla 6, y representan los factores clave que influyen en el comportamiento del proceso de molienda.

Tabla 6
Variables Utilizadas.

Variable	Unidad	Descripción
Tonelaje	t/h	Mineral alimentado
Corr_M1	A	Corriente bomba descarga M1
Corr_M2	A	Corriente bomba descarga M2
Corr_ZAF	A	Corriente bomba ZAF
Agua	m^3/h	Agua alimentada
F80	mm	Granulometría alimento
Potencia	kWh	Potencia M1

Asimismo, con base en las salidas generadas por los distintos motores desarrollados (FIS, ANFIS y FGA), se calcularon variables adicionales —como recomendaciones de tonelaje y agua sugerida, o predicciones de granulometría— las cuales se resumen en la Tabla 7.

Tabla 7
Variables Calculadas.

Variable	Unidad	Descripción
Delta_ton	t/h	Variación tonelaje
Tonelaje_Sugerido	t/h	Tonelaje sugerido por el modelo
Agua_Sugerida	m^3/h	Caudal de agua sugerido
P80	μm	Granulometría del producto real
P80_predicho	μm	Granulometría estimada por el modelo

4. Metodología y desarrollo de la Investigación

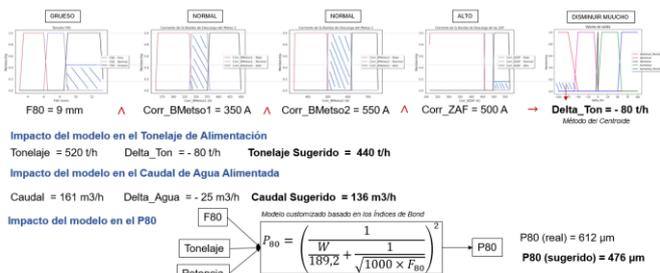
La presente investigación adoptó un enfoque cuantitativo y experimental, sustentado en técnicas de modelado matemático, simulación computacional y validación con datos reales provenientes de entornos industriales. Su objetivo fue desarrollar y comparar modelos inteligentes capaces de predecir y optimizar variables críticas del proceso de molienda, con énfasis en el tonelaje de alimentación, la granulometría del producto (P80) y el caudal de agua alimentado.

Para tal fin, se implementaron cuatro modelos principales en lenguaje Python, cada uno representando un enfoque distinto dentro del espectro de la inteligencia computacional aplicada:

- **Motor de Inferencia Difusa (FIS)** – basado en reglas expertas y funciones de pertenencia definidas manualmente.
- **Red Neuronal Predictiva (RRN)** – orientada a la predicción supervisada del P80 a partir de variables operativas.
- **Sistema Neurodifuso Adaptativo (ANFIS)** – arquitectura híbrida que integra lógica difusa y redes neuronales con ajuste automático de parámetros.
- **Motor Genético Difuso (FGA)** – modelo difuso optimizado evolutivamente mediante algoritmos genéticos.

El FIS tiene como propósito recomendar ajustes dinámicos en el tonelaje de alimentación, en función de condiciones operativas observadas. Su implementación se realizó con la librería FuzzyLite, empleando un esquema tipo Mamdani, según se observa en la Figura 5, con reglas y funciones de pertenencia definidas manualmente.

Figura 5
FIS tipo Mamdani

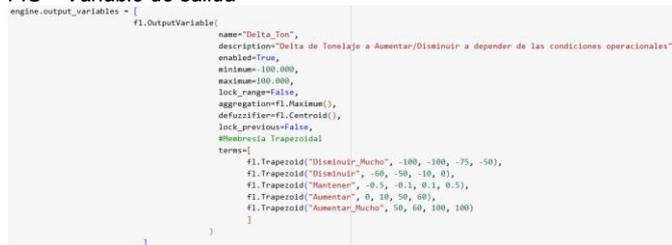


Como se observa en la Figura 6 y 7, se consideraron cuatro variables de entrada: F80 (mm), Corriente del Molino 1, Corriente del Molino 2 y Corriente de Zarandas de Alta Frecuencia (ZAF). Cada variable fue modelada mediante funciones trapezoidales, clasificadas como Bajo, Normal o Alto. La variable de salida, denominada Delta_Ton, representa la recomendación de ajuste de tonelaje, con etiquetas como Disminuir_Mucho, Mantener o Aumentar_Mucho.

Figura 6
FIS – Variables de entrada y salida

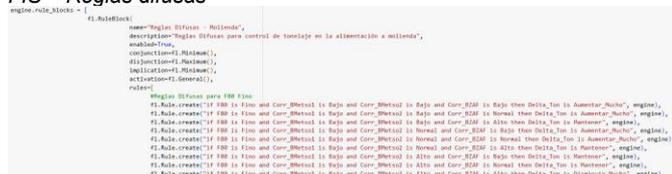


Figura 7
FIS – Variable de salida



Se definieron 81 reglas difusas, Figura 8, que combinan todas las posibles interacciones entre entradas y salida. A partir de los datos promediados por mes, el FIS emite una recomendación horaria. Posteriormente, se estiman el valor de P80 mediante la RRN (modelo 2) y el caudal de agua requerido para mantener una proporción adecuada de pulpa.

Figura 8
FIS – Reglas difusas



Este modelo constituye la línea base de comparación, emulando el comportamiento de un sistema de control experto estático y explícito.

El segundo modelo corresponde a una red neuronal artificial diseñada para predecir el P80 en función de tres variables de entrada: F80, tonelaje de alimentación y potencia del molino. Estas fueron seleccionadas por su alta correlación con la eficiencia del proceso de conminución.

Los datos fueron normalizados mediante MinMaxScaler, y divididos en subconjuntos de entrenamiento (80%) y prueba (20%). La red fue construida con PyTorch e incluyó: una capa de entrada con 3 neuronas, dos capas ocultas con 16 y 8 neuronas (ReLU), y una capa de salida lineal para estimar el P80.

El entrenamiento se realizó durante 500 épocas con el optimizador Adam y la función de pérdida MSE. Finalmente, se evaluó el modelo utilizando las métricas MAE y RMSE, mostrando una buena capacidad predictiva.

Este enfoque permite capturar relaciones no lineales complejas, aunque con menor interpretabilidad en comparación con los sistemas basados en reglas.

El tercer modelo desarrollado fue un sistema ANFIS (Adaptive Neuro-Fuzzy Inference System), que

combina la interpretabilidad de la lógica difusa con la capacidad de aprendizaje de las redes neuronales. A diferencia del FIS, según se observa en la Figura 9, este modelo ajusta automáticamente los parámetros de las funciones de pertenencia y reglas de inferencia a partir de los datos.

Figura 9

Definición del modelo ANFIS simplificado

```
class AnfisLayer(nn.Module):
    def __init__(self, n_inputs, n_rules):
        super(AnfisLayer, self).__init__()
        self.n_inputs = n_inputs
        self.n_rules = n_rules
        self.mu = nn.Parameter(torch.rand(n_inputs, n_rules))
        self.sigmas = nn.Parameter(torch.rand(n_inputs, n_rules))
        self.linear = nn.Linear(n_rules, 1)

    def forward(self, x):
        x_expanded = x.unsqueeze(2).expand(-1, -1, self.n_rules)
        mu = torch.exp(-(x_expanded - self.mu) ** 2) / (2 * self.sigmas ** 2)
        firing_strengths = torch.prod(mu, dim=1)
        norm_firing = firing_strengths / torch.sum(firing_strengths, dim=1, keepdim=True)
        return self.linear(norm_firing)
```

Fue implementado en PyTorch, simulando una arquitectura de cinco capas funcionales. Se utilizaron las mismas variables de entrada del FIS (F80, Corr_M1, Corr_M2, Corr_ZAF), transformadas mediante funciones de membresía gaussianas cuyos parámetros (μ y σ) fueron aprendidos durante el entrenamiento.

El sistema calcula los niveles de activación de las reglas, los normaliza y combina mediante una capa de salida lineal para estimar Delta_Ton. Posteriormente, este valor es utilizado para calcular el tonelaje sugerido, el P80 predicho y el caudal de agua requerido.

Según se muestra en la Figura 10, el entrenamiento se realizó sobre el conjunto consolidado mensual con aprendizaje supervisado, usando MSE como función de pérdida, el optimizador Adam y escalamiento MinMax. Este enfoque permite capturar relaciones no lineales de forma interpretable, superando al FIS en adaptabilidad y a la RRN en comprensibilidad.

Figura 10

Función para entrenar modelo neurodifuso

```
def entrenar_motor_neurodifuso(dfs_promediados, n_rules=5, epochs=100, lr=0.01):
    df_all = pd.concat(dfs_promediados.values()).dropna().copy()

    if "Delta_Ton" not in df_all.columns and "Tonelaje" in df_all.columns and "Tonelaje_sugerido" in df_all.columns:
        df_all["Delta_Ton"] = df_all["Tonelaje_sugerido"] - df_all["Tonelaje"]
    elif "Delta_Ton" not in df_all.columns:
        df_all["Delta_Ton"] = 0.0

    X = df_all[["F80", "Corr_M1", "Corr_M2", "Corr_ZAF"]].values
    y = df_all["Delta_Ton"].values.reshape(-1, 1)

    scaler_X = MinMaxScaler()
    scaler_y = MinMaxScaler()
    X_scaled = scaler_X.fit_transform(X)
    y_scaled = scaler_y.fit_transform(y)

    X_tensor = torch.tensor(X_scaled, dtype=torch.float32)
    y_tensor = torch.tensor(y_scaled, dtype=torch.float32)

    model = AnfisLayer(n_inputs=4, n_rules=5)
    optimizer = torch.optim.Adam(model.parameters(), lr=lr)
    loss_fn = nn.MSELoss()

    for epoch in range(epochs):
        optimizer.zero_grad()
        output = model(X_tensor)
        loss = loss_fn(output, y_tensor)
        loss.backward()
        optimizer.step()
        if epoch % 10 == 0:
            print(f"Epoch {epoch}, Loss: {loss.item():.5f}")

    return model, scaler_X, scaler_y
```

El cuarto modelo corresponde a un Motor Genético Difuso (FGA), que combina lógica difusa y algoritmos genéticos (GA) para ajustar automáticamente las relaciones entre variables, sin requerir reglas explícitas.

Fue desarrollado con la librería DEAP, ampliamente usada para la implementación de algoritmos evolutivos. Las variables de entrada fueron nuevamente F80, Corr_M1, Corr_M2 y Corr_ZAF, escaladas previamente con MinMaxScaler.

Cada individuo en la población representa un conjunto de coeficientes (pesos) que multiplican las variables de entrada. El modelo evoluciona, según detalla la Figura 11, esta población usando selección por torneo, cruce blend, mutación gaussiana y reemplazo generacional, durante un número determinado de generaciones.

Figura 11

Función de evolución FGA

```
def evaluar_individuo(individuo, X, y_true):
    y_pred = np.dot(X, np.array(individuo).reshape(-1, 1))
    error = np.mean((y_pred - y_true) ** 2)
    return error,
```

Como se muestra en la Figura 12, la función objetivo minimiza el MSE entre el Delta_Ton predicho y el real. El mejor conjunto de coeficientes es luego aplicado por mes, permitiendo calcular el tonelaje sugerido, el P80 estimado y el caudal de agua requerido.

Figura 12

Función de entrenamiento de FGA

```
def entrenar_motor_genetico(dfs_promediados, n_rules=4, ngen=50, pop_size=50, cxbp=0.6, mutpb=0.2):
    df_all = pd.concat(dfs_promediados.values()).dropna().copy()

    # Validar existencia de columnas necesarias
    if "Delta_Ton" not in df_all.columns and "Tonelaje" in df_all.columns and "Tonelaje_sugerido" in df_all.columns:
        df_all["Delta_Ton"] = df_all["Tonelaje_sugerido"] - df_all["Tonelaje"]
    elif "Delta_Ton" not in df_all.columns:
        raise ValueError("Falta columna Delta_Ton o datos base para calcularla.")

    X = df_all[["F80", "Corr_M1", "Corr_M2", "Corr_ZAF"]].values
    y = df_all["Delta_Ton"].values.reshape(-1, 1)

    scaler_X = MinMaxScaler()
    scaler_y = MinMaxScaler()
    X_scaled = scaler_X.fit_transform(X)
    y_scaled = scaler_y.fit_transform(y)

    n_inputs = X_scaled.shape[1]

    creator.create("FitnessMin", base.Fitness, weights=(-1.0,))
    creator.create("Individual", list, fitness=creator.FitnessMin)

    toolbox = base.Toolbox()
    toolbox.register("attr_float", random.uniform, -2, 2)
    toolbox.register("individual", tools.initRepeat, creator.Individual, toolbox.attr_float, n=n_inputs)
    toolbox.register("population", tools.initRepeat, list, toolbox.individual)

    toolbox.register("evaluate", evaluar_individuo, X=X_scaled, y_true=y_scaled)
    toolbox.register("mate", tools.cxBlend, alpha=0.5)
    toolbox.register("mutate", tools.mutGaussian, mu=0, sigma=0.3, indpb=0.2)
    toolbox.register("select", tools.selTournament, tournsize=3)

    pop = toolbox.population(n=pop_size)
    algorithms.easimple(pop, toolbox, cxbp=cxbp, mutpb=mutpb, ngen=ngen, verbose=True)

    best_ind = tools.selBest(pop, k=1)[0]

    return np.array(best_ind), scaler_X, scaler_y
```

Este enfoque es altamente flexible y robusto ante entornos industriales variables, con la ventaja de escapar de óptimos locales y descubrir relaciones no evidentes entre las variables del proceso.

4. Presentación y discusión de resultados

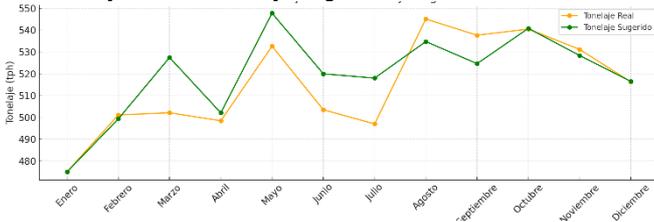
Para evaluar el desempeño del modelo basado en lógica difusa (FIS), se presenta la Tabla 8, la cual incluye los valores reales de Tonelaje, Caudal de Agua y P80, junto con los valores sugeridos por el modelo FIS y los porcentajes de mejora asociados.

Tabla 8
Resultados FIS

Mes	Tonelaje Real	Tonelaje Sugerido	Agua Real	Agua Sugerida	P80 Real	P80 Predicho	% Mejora Tonelaje	% Ahorro Agua	% Mejora P80
Enero	475.07	475.05	184.19	152.02	521.30	520.31	0.0%	21.16%	0.2%
Febrero	501.13	499.31	185.05	159.78	555.49	543.71	-0.4%	15.82%	2.1%
Marzo	502.15	527.46	168.95	168.79	559.93	660.88	5.0%	0.09%	-18.0%
Abril	496.47	502.09	188.67	160.67	574.12	584.99	0.7%	17.43%	-1.9%
Mayo	532.63	547.78	176.29	175.29	592.57	654.91	2.8%	0.57%	-10.5%
Junio	503.51	519.99	183.52	166.40	549.11	615.11	3.3%	10.29%	-12.0%
Julio	497.03	518.01	181.67	165.76	537.05	621.35	4.2%	9.59%	-15.7%
Agosto	545.09	534.77	175.25	171.13	620.09	572.16	-1.9%	2.41%	7.7%
Septiembre	537.66	524.64	171.43	167.88	635.13	572.87	-2.4%	2.11%	9.8%
Octubre	540.49	540.81	177.26	173.06	626.57	622.34	0.1%	2.43%	0.7%
Noviembre	531.18	528.38	166.74	169.08	628.36	614.20	-0.5%	-1.39%	2.8%
Diciembre	516.29	516.50	159.53	165.28	571.64	567.44	0.0%	-3.48%	0.7%
Promedios	515.06	519.57	176.55	166.26	580.95	595.86	0.9%	6.42%	-2.9%

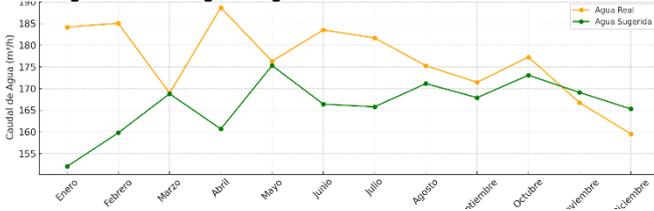
La Figura 13 muestra la evolución mensual del tonelaje real alimentado al molino en comparación con el tonelaje sugerido por el modelo FIS. En varios meses —como abril, mayo y agosto— el tonelaje real fue inferior al valor recomendado, lo que sugiere una oportunidad de incrementar la carga sin comprometer la eficiencia del proceso. En contraste, durante febrero y octubre se observa un ligero exceso respecto a la recomendación.

Figura 13
FIS -Tonelaje Real vs Tonelaje Sugerido



La Figura 14 compara el caudal de agua real con el sugerido por el FIS. En marzo, abril y diciembre, el caudal fue menor al recomendado, evidenciando oportunidades de mejora en la dilución de pulpa. En cambio, en enero y junio, se observa un exceso en el uso de agua respecto a lo sugerido por el modelo. Esta información es clave para optimizar el consumo hídrico en el proceso.

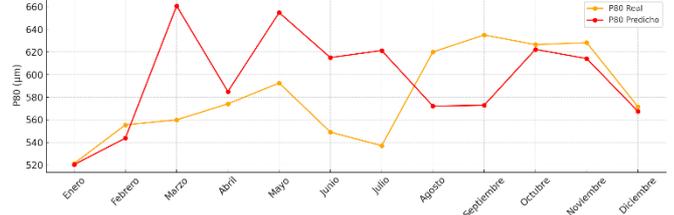
Figura 14
FIS - Agua Real vs Agua Sugerida



La Figura 15 muestra la comparación mensual entre los valores reales de la granulometría del producto (P80) y los valores predichos por el modelo FIS. En varios periodos —como marzo, mayo y julio— el P80 real superó el valor predicho, lo cual sugiere una molienda menos eficiente de lo

esperado. Esto evidencia oportunidades de mejora en la operación, dado que un P80 menor implica un producto más fino y mejor liberado para el proceso de flotación.

Figura 15
FIS - P80 real vs P80 Predicho



Para evaluar el desempeño del modelo neurodifuso adaptativo (ANFIS), se presenta la Tabla 9, que contiene los valores reales observados para Tonelaje, Agua y P80, así como los valores sugeridos o predichos por el modelo ANFIS. Se incluyen también los porcentajes de mejora estimados.

Tabla 9
Resultados ANFIS

Mes	Tonelaje Real	Tonelaje Sugerido	Agua Real	Agua Sugerida	P80 Real	P80 Predicho	% Mejora Tonelaje	% Ahorro Agua	% Mejora P80
Enero	475.07	450.34	184.19	157.87	521.30	620.60	3.8%	16.67%	-19.0%
Febrero	501.13	492.37	185.05	157.56	555.49	569.13	-1.7%	17.45%	-2.5%
Marzo	502.15	522.99	168.95	167.23	559.93	691.03	4.1%	1.03%	-7.3%
Abril	496.47	500.76	188.67	160.24	574.12	545.94	0.5%	17.74%	4.9%
Mayo	532.63	557.22	176.29	178.31	592.57	593.74	4.6%	-1.13%	-0.2%
Junio	503.51	528.21	183.52	169.03	549.11	577.43	4.9%	8.58%	-5.2%
Julio	497.03	510.38	181.67	163.32	537.05	551.93	2.7%	11.23%	-2.8%
Agosto	545.09	525.31	175.25	168.10	620.09	616.97	-3.6%	4.25%	0.5%
Septiembre	537.66	517.87	171.43	165.72	635.13	604.76	-3.7%	3.45%	4.8%
Octubre	540.49	546.42	177.26	174.85	626.57	645.96	1.1%	1.37%	-3.1%
Noviembre	531.18	524.87	166.74	167.96	628.36	609.57	-1.2%	-0.73%	3.0%
Diciembre	516.29	517.21	159.53	165.51	571.64	581.66	0.2%	-3.61%	-1.8%
Promedios	515.06	519.71	176.55	166.31	580.95	593.23	1.0%	6.36%	-2.4%

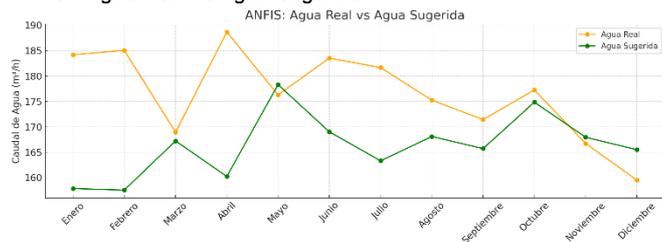
La Figura 16 muestra la evolución mensual del tonelaje real frente al tonelaje sugerido por el modelo ANFIS. Se aprecia que, en varios meses —como febrero, marzo y agosto— el tonelaje recomendado supera al valor real, lo que indica que el modelo identifica márgenes operativos no aprovechados completamente.

Figura 16
ANFIS -Tonelaje Real vs Tonelaje Sugerido



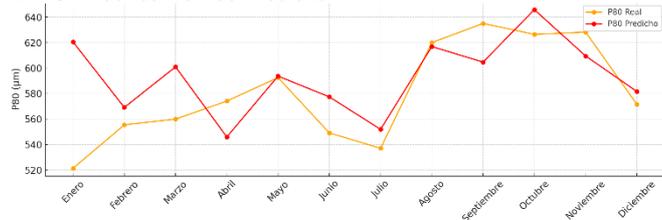
La Figura 17, muestra que el caudal de agua sugerido por el sistema ANFIS tiende a ser inferior al real en casi todos los meses, lo cual, al igual que en el FIS, sugiere que se podrían estar utilizando volúmenes mayores a los necesarios.

Figura 17
ANFIS - Agua Real vs Agua Sugerida



La Figura 18 presenta la comparación entre los valores reales de P80 y los valores predichos por el ANFIS. En general, el modelo se aproxima adecuadamente a los valores reales, mostrando buena capacidad de generalización. En meses como mayo, julio y octubre, donde el P80 real fue elevado (indicando molienda gruesa), el modelo permite anticipar esta condición, ayudando a tomar acciones correctivas a tiempo para evitar impactos negativos en la etapa de flotación.

Figura 18
ANFIS - P80 real vs P80 Predicho



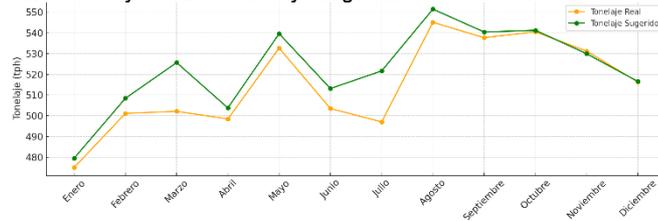
Para analizar el desempeño del modelo basado en lógica difusa optimizada por algoritmos genéticos (FGA), se presenta la Tabla 10, que resume los valores reales y sugeridos para Tonelaje, Agua y P80, así como los porcentajes de mejora asociados a cada variable.

Tabla 10
Resultados FGA

Mes	Tonelaje Real	Tonelaje Sugerido	Agua Real	Agua Sugerida	P80 Real	P80 Predicho	% Mejora Tonelaje	% Ahorro Agua	% Mejora P80
Enero	475.07	475.51	184.19	183.44	521.30	520.31	0.9%	20.00%	0.2%
Febrero	501.13	508.41	185.05	182.89	535.49	543.71	1.5%	13.74%	2.1%
Marzo	502.15	525.65	168.95	168.21	559.93	601.88	4.7%	0.44%	-7.5%
Abril	498.47	603.80	188.67	161.22	574.12	554.99	1.1%	17.03%	3.3%
Mayo	532.63	539.60	176.29	172.67	592.57	614.91	1.3%	2.10%	-3.8%
Junio	503.51	513.14	183.52	164.20	549.11	605.11	1.9%	11.77%	-10.2%
Julio	497.03	521.63	181.67	166.92	537.05	601.35	4.9%	8.84%	-12.0%
Agosto	545.09	551.43	175.25	176.46	620.09	572.16	1.2%	-0.68%	7.7%
Septiembre	537.66	540.41	171.43	172.93	635.13	572.87	0.5%	-0.87%	9.8%
Octubre	540.49	541.22	177.26	173.19	626.57	622.34	0.1%	2.35%	0.7%
Noviembre	531.18	529.91	166.74	169.57	628.36	614.20	-0.2%	-1.67%	2.3%
Diciembre	516.29	516.54	159.53	165.29	571.64	567.44	0.0%	-3.49%	0.7%
Promedios	515.06	522.60	176.55	167.23	580.95	582.61	1.5%	5.80%	-0.5%

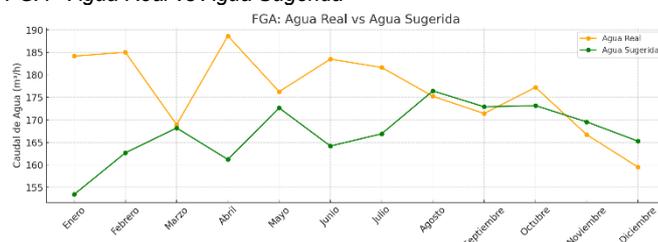
La Figura 19 muestra la comparación mensual entre el tonelaje real operado y el tonelaje sugerido por el modelo FGA. En varios periodos —como abril, agosto y octubre— el modelo propone valores superiores a los reales, lo que sugiere que existieron márgenes operativos desaprovechados para incrementar la carga del molino sin comprometer la estabilidad del proceso. Esto evidencia el potencial del modelo evolutivo para identificar condiciones óptimas no evidentes, mejorando la eficiencia global del circuito de molienda.

Figura 19
FGA - Tonelaje Real vs Tonelaje Sugerido



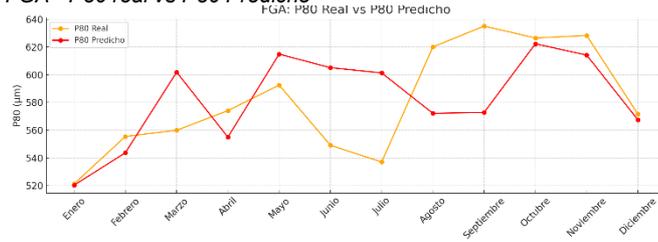
La Figura 20 presenta la evolución del caudal de agua real frente al valor sugerido por el modelo. Se observa que, en meses como marzo, junio y septiembre, el modelo propone caudales significativamente menores, lo cual podría corregir diluciones excesivas y mejorar la eficiencia energética del molino.

Figura 20
FGA - Agua Real vs Agua Sugerida



Finalmente, la Figura 21 compara los valores reales de P80 con los valores predichos por el modelo FGA. En meses como mayo, julio y noviembre, el P80 real fue superior al estimado, lo que indica una molienda más gruesa de la prevista. Esta discrepancia sugiere oportunidades para mejorar la liberación del mineral, reduciendo el tamaño del producto final.

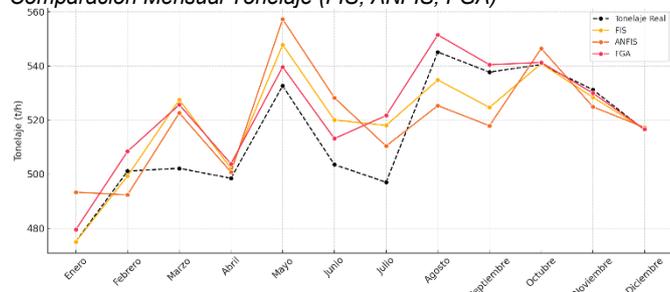
Figura 21
FGA - P80 real vs P80 Predicho



La Figura 22 muestra la evolución mensual del caudal de agua real frente a los valores sugeridos por los modelos FIS, ANFIS y FGA. En varios meses —como marzo, abril y octubre— el modelo FIS recomienda un caudal superior al utilizado en la operación, lo que sugiere que se pudo haber mejorado la dilución de pulpa para optimizar la molienda. El modelo ANFIS, en cambio, presenta un comportamiento más conservador, sugiriendo caudales inferiores al real en casi todos los meses, lo cual refleja una estrategia de uso hídrico más eficiente. El modelo FGA, por su parte, identifica meses específicos —como agosto y noviembre— en los que el agua utilizada fue mayor a la requerida, lo que evidencia oportunidades de

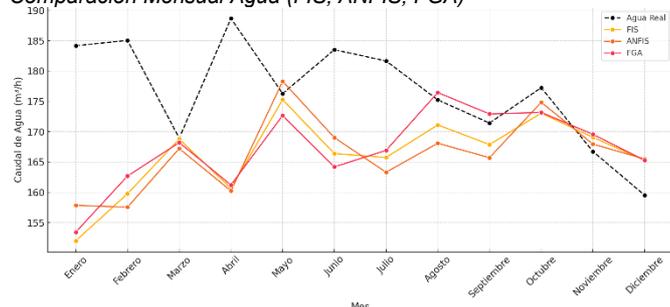
ahorro de recursos. La comparación conjunta resalta cómo cada enfoque modela de forma distinta la relación entre condiciones operativas y consumo de agua, permitiendo seleccionar la estrategia más adecuada según las restricciones del entorno.

Figura 22
Comparación Mensual Tonelaje (FIS, ANFIS, FGA)



La Figura 23 ilustra la comparación entre el tonelaje real alimentado al molino y el sugerido por los modelos FIS, ANFIS y FGA. El modelo FIS muestra recomendaciones conservadoras en varios meses, con ajustes modestos que buscan mantener la estabilidad del proceso. En contraste, el modelo ANFIS adapta sus sugerencias dinámicamente, proponiendo incrementos significativos en meses como mayo y agosto, lo que refleja su capacidad de aprendizaje y adaptación a patrones históricos. Por su parte, el modelo FGA identifica oportunidades de mejora en periodos como marzo, junio y septiembre, sugiriendo cargas superiores que podrían haber optimizado el rendimiento del molino. Esta visión comparativa evidencia que los modelos inteligentes pueden actuar como herramientas de soporte para maximizar la alimentación sin comprometer la eficiencia del proceso.

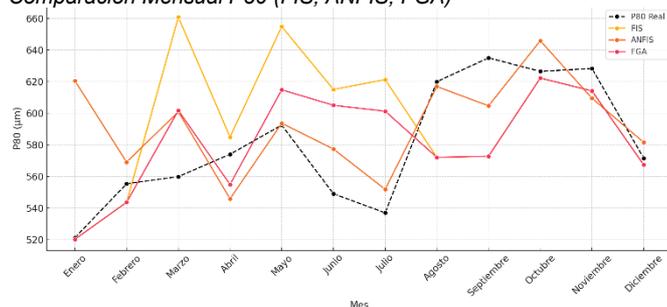
Figura 23
Comparación Mensual Agua (FIS, ANFIS, FGA)



La Figura 24 presenta la comparación del P80 real con el valor predicho por los modelos FIS, ANFIS y FGA. En varios meses —como mayo, julio y noviembre— se observa que el P80 real es más alto que el valor estimado por los modelos, lo cual indica una molienda más gruesa de lo esperado. Esta discrepancia sugiere que existen oportunidades para mejorar la eficiencia del proceso mediante ajustes operativos. El modelo ANFIS muestra una mayor capacidad de adaptación, con predicciones

cercanas a los valores reales en la mayoría de los meses. El FGA, en tanto, ofrece estimaciones consistentes, capturando tendencias generales, mientras que el FIS mantiene un comportamiento más lineal. Esta comparación permite evaluar el grado de precisión de cada modelo en la predicción del producto final, lo cual es clave para garantizar una buena liberación mineral en etapas posteriores como la flotación.

Figura 24
Comparación Mensual P80 (FIS, ANFIS, FGA)



5. Conclusiones

Los modelos inteligentes evaluados —FIS, ANFIS y FGA— demostraron un impacto positivo en la optimización del proceso de molienda. En promedio, los modelos FIS y ANFIS lograron incrementar el tonelaje de alimentación en aproximadamente un 1%, mientras que el modelo FGA alcanzó una mejora promedio del 1.5%. Cabe destacar que, en ciertos meses, se registraron incrementos puntuales de hasta un 5%, evidenciando el potencial de estos algoritmos para detectar oportunidades operativas no aprovechadas. Además, el modelo FGA se caracterizó por generar recomendaciones más estables, evitando variaciones abruptas que pudieran comprometer la continuidad del proceso.

En cuanto al consumo de agua, los tres modelos propusieron reducciones sostenidas del caudal alimentado, alcanzando en promedio una disminución del 6% sin afectar negativamente el desempeño global del circuito. Esto resalta la capacidad de los algoritmos para contribuir a una operación más eficiente y sostenible, especialmente en contextos industriales con restricciones hídricas.

Respecto a la variable granulométrica del producto (P80), se observó que los modelos FIS y ANFIS tienden a privilegiar el aumento del tonelaje aun cuando ello implica un ligero engrosamiento del mineral procesado, con un incremento promedio del P80 del orden del 2%. En contraste, el modelo FGA logra mantener el P80 en niveles similares a los reales, optimizando simultáneamente la carga del

molino. Esta característica lo posiciona como una alternativa robusta que equilibra producción y calidad del producto final.

6. Referencias bibliográficas

- Almeyda, L., Bravo, L. (2025). Implementación de una estrategia de control difusa para optimizar la molienda utilizando un analizador de tamaño de roca y sensores virtuales. *Revista Minería* (570).
- Almeyda, L. (2025). Optimización de la molienda primaria de una unidad minera ubicada al sur de Perú empleando un Sistema de Inferencia Difusa (FIS). Repositorio Institucional UNMSM.
- Bascur, O. y Concha, F. (2024). *The Engineering Science of Mineral Processing. A Fundamental and Practical Approach*. Taylor & Francis Group.
- Candiotti, S., Montaña, L., Barja, F., Charca, S., Quineche, L. y Flores, C. (2020). Estudio del arte de sistemas de molienda polimetálica. Universidad de Ingeniería y Tecnología.
- Cipriano, A. (2010). *Industrial Products for Advanced Control of Mineral Processing Plants*. En Sbárbaro, D. y Del Villar, R. (Ed.), *Advanced Control and Supervision of Mineral Processing Plants* (pp. 287 – 306). Springer.
- Mathworks. (2020). *Fuzzy Logic Toolbox. User's Guide*.
- Sremac, S., Tanackov, I., Kopic, M. y Radovic, D. (2018). ANFIS model for determining the economic order quantity. *Decision Making Applications in Management and Engineering*
- Singh, H. y Ahmad, Y. (2020). *Deep Neuro – Fuzzy Systems with Python. With caase studies and applications from the industry*. Apress.
- Sivnandam, S. N., Sumathi, S. y Deepa, S.N. (2007). *Introduction to Fuzzy Logic using MATLAB*. Springer.

7. Información de los autores

Leonardo Almeyda¹

Se desempeña como Ingeniero de Innovación en Nexa Resources. Experiencia en gestión de proyectos de innovación tecnológica, transformación digital y analítica avanzada en operaciones mineras y refinerías de Perú, Chile y Brasil. Titulado en Ingeniería Metalúrgica – UNMSM, con Maestría en Gerencia de Proyectos – UNI, certificación PMP® y especialización en Gestión del Desarrollo Tecnológico – MIT.

Daniel Lovera²

Profesional con formación científica, tecnológica y humanística en el campo de la ingeniería, especialista en formular, ejecutar y monitorear proyectos de investigación, desarrollo, innovación y de inversión. Maestría en Ingeniería Química – UNMSM, Posgrado en Ingeniería Metalúrgica – Universidad de Concepción – Chile, Doctorado en Ingeniería Industrial – UNMSM, Posdoctorado en Ciencias de la Tierra y Ambiente – Universidad de Manizales – Colombia.

Clayton Iliquin³

Se desempeña como Científico de Datos en Vertex Technologies, liderando proyectos de inteligencia artificial aplicada para incrementar throughput y recuperación principalmente en plantas de Cu y Mo en operaciones de Chile y Perú. Bachiller en Ingeniería Metalúrgica – UNMSM, certificado como Scrum Master, AWS Machine Learning Specialist y Seeq Analytics Engineer.

UNIVERSIDAD NACIONAL MAYOR DE SAN MARCOS

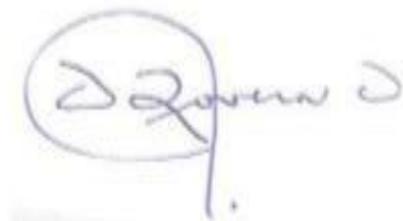
FACULTAD DE INGENIERIA GEOLOGICA, MINERA, METALURGICA y GEOGRAFICA INSTITUTO DE

INVESTIGACION IIGEO

GRUPO DE INVESTIGACION TECNOLOGIAS METALURGICAS SOSTENIBLES

AUTORIZACIÓN DE PARTICIPACIÓN

Yo Daniel Florencio Lovera Davila, Profesor Principal de la Escuela Profesional de Ingeniería Metalúrgica de la Universidad Nacional Mayor de San Marcos (UNMSM) autorizo que el trabajo titulado **“Modelo Analítico basado en Algoritmos Genéticos Difusos (FGA) para optimizar el proceso de conminución de minerales”** presentado por el autor Leonardo Almeyda Tejada y coautores Daniel Florencio Lovera Davila y Clayton Jordán Iliquin Zavaleta sea presentado en el concurso del **Premio Nacional de Minería del evento PERUMIN 37 Convención Minera** en las fechas del 22 al 26 de setiembre del 2025 en la ciudad de Arequipa.



Firma

DNI:06450640

Fecha: 18/07/2025

Nota: Esta autorización se entrega solo en el caso de que el participante se presente de manera independiente y el trabajo implique el desarrollo en el marco de una empresa o institución. La indicada autorización deberá ser entregada en hoja membretada.