



UNIVERSIDAD TÉCNICA DE AMBATO
FACULTAD DE INGENIERÍA EN SISTEMAS, ELECTRÓNICA E
INDUSTRIAL
CARRERA DE SOFTWARE

Tema:

**MEDIDOR VIRTUAL DE FLUJO (VFM) CON INTELIGENCIA ARTIFICIAL
PARA LA PREDICCIÓN DEL FLUJO DE GAS EN EL CAMPO PUCUNA**

Trabajo de titulación modalidad: Proyecto de Investigación, presentado previo a la obtención del título de Ingeniero de Software

ÁREA: Desarrollo de Software

LÍNEA DE INVESTIGACIÓN: Software, Tecnologías de la Información y Ciencia de Datos

AUTOR: Marco Felipe Montesdeoca Cruz

TUTOR: Ing. Santiago David Jara Mora, Mg.

Ambato – Ecuador

julio - 2026

APROBACIÓN DEL TUTOR

En calidad de tutor del trabajo de titulación con el tema: **MEDIDOR VIRTUAL DE FLUJO (VFM) CON INTELIGENCIA ARTIFICIAL PARA LA PREDICCIÓN DEL FLUJO DE GAS EN EL CAMPO PUCUNA**, desarrollado bajo la modalidad Proyecto de Investigación por el señor Marco Felipe Montesdeoca Cruz, estudiante de la Carrera de Software, de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial, de la Universidad Técnica de Ambato, me permito indicar que el estudiante ha sido tutorado durante todo el desarrollo del trabajo hasta su conclusión, de acuerdo a lo dispuesto en el Artículo 17 del Reglamento para la Titulación de Grado en la Universidad Técnica de Ambato y el numeral 6.3 del instructivo del reglamento referido.

Ambato, julio de 2026



**Santiago David Jara
Moya**



Ing. Santiago David Jara Mora, Mg.

TUTOR

AUTORÍA

El presente trabajo de titulación con el tema: **MEDIDOR VIRTUAL DE FLUJO (VFM) CON INTELIGENCIA ARTIFICIAL PARA LA PREDICCIÓN DEL FLUJO DE GAS EN EL CAMPO PUCUNA** es absolutamente original, auténtico y personal y ha observado los preceptos establecidos en la Disposición General Quinta del Reglamento para la Titulación de Grado en la Universidad Técnica de Ambato. En tal virtud, el contenido, efectos legales y académicos que se desprenden del mismo son de exclusiva responsabilidad del autor.

Ambato, julio de 2026



Validar únicamente en FirmaEC.
Firmado electrónicamente por:
**MARCO FELIPE
MONTESDEOCA CRUZ**

Marco Felipe Montesdeoca Cruz

C.C. 1850305762

AUTOR

DERECHOS DE AUTOR

Autorizo a la Universidad Técnica de Ambato para que reproduzca total o parcialmente este trabajo de titulación dentro de las regulaciones legales e institucionales correspondientes. Además, cedo todos mis derechos de autor a favor de la institución con el propósito de su difusión pública, por lo tanto, autorizo su publicación en el repositorio virtual institucional como un documento disponible para la lectura y uso con fines académicos e investigativos de acuerdo con la Disposición General Cuarta del Reglamento para la Titulación de Grado en la Universidad Técnica de Ambato.

Ambato, julio de 2026



Validar únicamente en FirmaEC.
Firmado electrónicamente por:
**MARCO FELIPE
MONTESDEOCA CRUZ**

Marco Felipe Montesdeoca Cruz

C.C. 1850305762

AUTOR

APROBACIÓN DEL TRIBUNAL DE GRADO

En calidad de par calificador del informe final del trabajo de titulación presentado por el señor Marco Felipe Montesdeoca Cruz, estudiante de la Carrera de Software, de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial, bajo la Modalidad Proyecto de Investigación, titulado **MEDIDOR VIRTUAL DE FLUJO (VFM) CON INTELIGENCIA ARTIFICIAL PARA LA PREDICCIÓN DEL FLUJO DE GAS EN EL CAMPO PUCUNA**, nos permitimos informar que el trabajo ha sido revisado y calificado de acuerdo al Artículo 19 del Reglamento para la Titulación de Grado en la Universidad Técnica de Ambato y el numeral 6.4 del instructivo del reglamento referido. Para cuya constancia suscribimos, conjuntamente con el señor Presidente del Tribunal.

Ambato, julio de 2026



Ing. Franklin Oswaldo Mayorga Mayorga, Mg.

PRESIDENTE DEL TRIBUNAL



Dr. Victor Filiberto Peñañiel Gaibor PhD.

Ing. Edwin Hernando Buenaño Valencia Mg.

PROFESOR CALIFICADOR

PROFESOR CALIFICADOR

DEDICATORIA

*A mis padres, Rosario y Marco por su
apoyo incondicional durante toda la
carrera, ayudandome a seguir adelante y
a no rendirme durante toda la carrera,
permitiendome seguir y conseguir lograr
mis metas.*

*A mi futura esposa, Estefanía, por su
apoyo, paciencia y amor durante todo el
recorrido, dandome fuerzas para cumplir
con las metas que me propuse.*

*A mi grupo de amigos de la Facultad
conocido como SOFTWARE SQUAD 404,
por las noches de trabajo, las
conversaciones, las risas y los momentos
inolvidables durante toda la carrera.*

*A mis hermanos, Erika y Andres, por todo
lo que me han brindado, tanto durante la
carrera como durante toda mi vida.*

*A mis mascotas perrunas, Kira, Tory y
Flor, por acompañarme durante
desveladas y mostrarme el amor y cariño
que solo una mascota puede dar.*

AGRADECIMIENTO

A mis padres y a mis hermanos, por todo lo que me han brindado a lo largo de este camino: su apoyo incondicional, su cariño y su amor. Gracias por ser mi sostén; sin ustedes, este logro no habría sido posible.

A mi tía y a mi prima, por abrirme las puertas de su hogar en otra ciudad durante la carrera y por acompañarme con generosidad y apoyo en todo lo que estuvo a su alcance durante mi estancia junto a ustedes.

A mi amigo Andrés Mayorga, por su disposición, su amistad sincera y su valiosa ayuda en los momentos en que más la necesité.

A mi futura esposa, Estefanía, por su amor, su cariño y su apoyo constante en cada etapa de este proceso. Gracias por caminar a mi lado y darme fuerzas para seguir adelante.

ÍNDICE GENERAL DE CONTENIDOS

PORTADA	i
APROBACIÓN DEL TUTOR	ii
AUTORÍA	iii
DERECHOS DE AUTOR	iv
APROBACIÓN DEL TRIBUNAL DE GRADO	v
DEDICATORIA	vi
AGRADECIMIENTO	vii
ÍNDICE GENERAL DE CONTENIDOS	viii
ÍNDICE DE TABLAS	xii
ÍNDICE DE FIGURAS	xv
RESUMEN EJECUTIVO	xvi
ABSTRACT	xvii
1 MARCO TEÓRICO	1
1.1 Tema de investigación	1
1.1.1 Planteamiento del problema	1

1.2	Antecedentes investigativos	2
1.2.1	Cadenas de búsqueda utilizadas	3
1.2.2	Criterios de selección aplicados	4
1.2.3	Análisis de los estudios	5
1.3	Fundamentación teórica	18
1.3.1	Virtual Flow Metering y su aplicación	19
1.3.2	Machine Learning para Virtual Flow Metering	19
1.3.3	Arquitecturas de Redes Neuronales para VFM	20
1.3.4	Enfoques Híbridos y Ensemble Learning	20
1.3.5	Procesamiento de Datos y Feature Engineering	20
1.3.6	Selección e Ingeniería de Características	21
1.3.7	Métricas de Evaluación de Modelos	21
1.3.8	Clasificación de Regímenes de Flujo	22
1.3.9	Modelado Híbrido	22
1.3.10	Optimización de Hiperparámetros	23
1.3.11	Validación de Modelos	23
1.3.12	Aprendizaje Continuo	23
1.3.13	Arquitectura de Software	24
1.4	Objetivos	25
1.4.1	Objetivo general	25
1.4.2	Objetivos específicos	25
2	METODOLOGÍA	26

2.1	Materiales	26
2.2	Métodos	26
2.2.1	Modalidad de investigación	27
2.2.2	Población y muestra	27
2.2.3	Recolección de información	28
2.3	Procesamiento y análisis de datos	40
3	RESULTADOS Y DISCUSIÓN	42
3.1	Identificación de variables operacionales críticas y construcción del dataset de entrenamiento	42
3.1.1	Variables operacionales criticas identificadas	42
3.1.2	Bases de datos históricas	43
3.1.3	Tablas fuente utilizadas	44
3.1.4	Construcción del dataset de entrenamiento	46
3.1.5	Resultados	51
3.2	Análisis y selección del algoritmo de inteligencia artificial	53
3.2.1	Algoritmos de regresion evaluados	55
3.2.2	Modelo <i>naive</i> de referencia	55
3.2.3	Algoritmos:	57
3.2.4	Resultados de la evaluacion de los algoritmos de regresion	58
3.2.5	Resultados	59
3.3	Implementación del sistema integral de Medidor Virtual de Flujo	62
3.3.1	Selección de herramientas tecnológicas	67

3.3.2	Fase 1: Análisis de requisitos	74
3.3.3	Fase 2: Diseño	76
3.3.4	Fase 3: Construcción	83
3.3.5	Fase 4: Implementacion	89
3.3.6	Resultados	97
4	CONCLUSIONES Y RECOMENDACIONES	101
4.1	Conclusiones	101
4.2	Recomendaciones	102
5	REFERENCIAS BIBLIOGRÁFICAS	105
6	ANEXOS	109
6.1	Anexo A. Normas ISO utilizadas	109
6.2	Anexo B. Guía de preguntas de las entrevistas	110
6.3	Anexo C. Código fuente del sistema VFM	111
6.4	Anexo D. Manual de usuario de la Terminal VFM	113

Índice de cuadros

1	Criterios de inclusión, exclusión y pertinencia aplicados	4
2	Distribución de artículos seleccionados por base de datos	5
3	Metodologías de Investigación	13
4	Fuentes y Tipos de Datos	13
5	Técnicas de Preprocesamiento	14
6	Ingeniería y Selección de Características	14
7	Modelos de Clasificación/Regresión	15
8	Modelado Híbrido	15
9	Optimización de Hiperparámetros	16
10	Arquitecturas de Software/Sistema	16
11	Tecnologías y Librerías	16
12	Técnicas de Validación	17
13	Métricas de Evaluación	17
14	Interpretabilidad de Modelos (XAI)	18
15	Diseño de recolección de datos	29
16	Resultados de la entrevista – Supervisor	30
17	Resultados de la entrevista – Supervisor de Instrumentación y Control	31
18	Resultados de la entrevista – Ingeniero 1 de Instrumentación y Control	32
19	Resultados de la entrevista – Ingeniero 2 de Instrumentación y Control	32
20	Resultados segunda entrevista – Supervisor (Justificación de datos y flujo de gas)	33
21	Ficha bibliográfica 1	35
22	Ficha bibliográfica 2	36
23	Ficha bibliográfica 3	37
24	Ficha bibliográfica 4	38
25	Ficha bibliográfica 5	39
26	Variables operacionales críticas identificadas (campo Pucuna, activo P).	43
27	Descripción de columnas de la tabla <code>produccion_volumenes</code> . . .	45
28	Descripción de columnas de la tabla <code>bes_cabecal_sensors</code> . . .	46

29	Métricas de desempeño de los modelos en el conjunto de prueba (25 %)	
	— campo Pucuna	58
30	Comparación de predicciones vs. gas real (MSCFD) — campo Pucuna	60
31	Ponderación aplicada a los criterios deseables para la selección del modelo de ciclo de vida	66
32	Selección del modelo de ciclo de vida del software mediante el modelo de toma de decisiones de Simon (desarrollo del VFM).	66
33	Ponderación aplicada a los criterios deseables para la selección del lenguaje principal de inteligencia artificial	67
34	Selección del lenguaje principal para inteligencia artificial mediante el modelo de Simon	68
35	Ponderación aplicada a los criterios deseables para la selección del framework web	69
36	Selección del framework web mediante el modelo de toma de decisiones de Simon	69
37	Ponderación aplicada a los criterios deseables para la selección del sistema de gestión de base de datos	70
38	Selección del sistema de gestión de base de datos mediante el modelo de Simon	71
39	Ponderación aplicada a los criterios deseables para la selección de la herramienta de despliegue	72
40	Selección de herramienta de despliegue mediante el modelo de Simon	72
41	Ponderación aplicada a los criterios deseables para la selección del framework de API	73
42	Selección del framework para API mediante el modelo de toma de decisiones de Simon	73
43	Requisitos funcionales identificados para el sistema VFM	75
44	Requisitos no funcionales identificados para el sistema VFM	76
45	Pruebas unitarias aplicadas al sistema VFM	91
46	Pruebas de caja negra aplicadas al sistema VFM	93
47	Pruebas de integración aplicadas al sistema VFM	94
48	Pruebas de caja blanca aplicadas al sistema VFM	95

49	Resumen general de pruebas del sistema VFM	96
50	Pruebas realizadas en la implantación del sistema VFM	97
51	Matriz de requerimientos funcionales del sistema VFM	98
52	Matriz de requerimientos no funcionales del sistema VFM	99
53	Resumen de cumplimiento de requerimientos del sistema VFM	99

Índice de figuras

1	Nube de palabras generada a partir de las entrevistas	34
2	Tabla <code>sensor_readings</code> con Petroleo y Agua por activo	44
3	Tabla <code>bes_cabecal_sensors</code> con Presion y Temperatura por activo	46
4	Tabla <code>dataset_entrenamiento</code>	47
5	Tabla Informacion del tipo de datos del dataset . .	48
6	Histograma de las variables del dataset	49
7	Matriz de correlación de las variables del dataset	50
8	Arquitectura del sistema VFM	77
9	Diagrama de caso de uso	78
10	Guía de estilo visual: paleta de colores, tipografía y componentes reuti- lizables del sistema VFM	79
11	Mockup del dashboard principal del sistema VFM	80
12	Mockup del panel de entrenamiento del modelo	81
13	Mockup del módulo de predicción del flujo de gas asociado	82
14	Mockup del módulo de gestión de modelos y activos	83
15	Servicios Docker utilizados en la implementación del sistema	86
16	Interfaz entrenamiento del sistema VFM	87
17	Interfaz prediccion del sistema VFM	88
18	Interfaz web desarrollada para el sistema VFM	88
19	Ejecución de pruebas unitarias del backend de predicción	91
20	Ejecución de pruebas funcionales y de caja negra en la interfaz web .	93
21	Ejecución de pruebas de integración entre módulos del sistema	95
22	Pipeline del sistema VFM	98
23	Nivel de cumplimiento: planificado vs. logrado	100
24	Módulo de entrenamiento del modelo de inteligencia artificial	112
25	Servicio de predicción del flujo de gas asociado	113
26	Rutas del backend FastAPI (entrenamiento, predicción y monitoreo) .	113

RESUMEN EJECUTIVO

En el Campo Pucuna de Petroecuador, la medición del gas asociado en flares enfrenta limitaciones por escasa instrumentación, estimaciones manuales y elevado costo de medidores físicos, lo que dificulta decisiones operativas. Este trabajo propone un Medidor Virtual de Flujo (VFM) basado en inteligencia artificial para estimar el caudal diario de gas desde variables operacionales (presión, temperatura y producción líquida).

La investigación integró entrevistas semiestructuradas, revisión bibliográfica de Virtual Flow Metering y análisis de 2 192 observaciones diarias trazables en PostgreSQL. Se evaluaron Ridge, Random Forest, SVR y MLP; el perceptrón multicapa alcanzó el mejor resultado en prueba (25 %): $R^2 = 97,02\%$, $MAE = 0,13\%$ y $RMSE = 0,18\%$.

Se desarrolló un sistema bajo metodología RAD con Laravel, Python/FastAPI, TensorFlow/Keras, PostgreSQL y Docker, que permite cargar datos, entrenar modelos, predecir y visualizar resultados; las predicciones diarias pueden consolidarse por periodo (por ejemplo, mensual) para apoyar balances operativos. Los 15 requerimientos funcionales y no funcionales cumplieron el 100 % mediante pruebas unitarias, de integración y de caja blanca y negra. Se concluye la viabilidad técnica del VFM para un piloto en campo y recalibración del modelo.

Palabras clave: medidor virtual de flujo, Virtual Flow Metering, inteligencia artificial, perceptrón multicapa, Campo Pucuna, Petroecuador, gas asociado, aprendizaje supervisado.

ABSTRACT

At Petroecuador's Pucuna field, associated-gas measurement in flares is limited by scarce instrumentation, manual estimates, and high costs of physical meters, hindering operational decisions. This study proposes a Virtual Flow Meter (VFM) using artificial intelligence to estimate daily gas flow from operational variables (pressure, temperature, and liquid production).

The work combined semi-structured interviews, a Virtual Flow Metering literature review, and 2,192 daily observations stored in PostgreSQL. Ridge, Random Forest, SVR, and MLP were evaluated; the multilayer perceptron achieved the best test performance (25 %): $R^2 = 97.02\%$, $MAE = 0.13\%$, and $RMSE = 0.18\%$.

An integrated system was built with RAD, Laravel, Python/FastAPI, TensorFlow/Keras, PostgreSQL, and Docker for data loading, training, prediction, and visualization; daily predictions can be aggregated by period (e.g., monthly) to support operational balances. Fifteen functional and non-functional requirements reached 100 % compliance through unit, integration, and black- and white-box testing. The results confirm the technical feasibility of a software VFM for field pilot deployment and model recalibration.

Keywords: virtual flow meter, Virtual Flow Metering, artificial intelligence, multilayer perceptron, Pucuna field, Petroecuador, associated gas, supervised learning.

1. MARCO TEÓRICO

1.1 Tema de investigación

MEDIDOR VIRTUAL DE FLUJO (VFM) CON INTELIGENCIA ARTIFICIAL PARA LA PREDICCIÓN DEL FLUJO DE GAS EN EL CAMPO PUCUNA.

1.1.1 Planteamiento del problema

En la industria del oil & gas, el gas asociado fue considerado durante años un sub-producto de la producción de petróleo y quemado en las teas de los campos petroleros, provocando grandes cantidades de emisiones de gases de efecto invernadero.

En los últimos tiempos, en virtud de políticas de descarbonización, las empresas del sector promueven el aprovechamiento del gas asociado para la generación eléctrica, desplazando la generación a diésel. Sin embargo, esta transición se enfrenta a desafíos operativos y económicos significativos debido a la dependencia predominante de medidores físicos de flujo. Estos equipos, si bien ofrecen precisiones aceptables bajo condiciones ideales, conllevan costos de adquisición, instalación y mantenimiento prohibitivamente altos, representando inversiones que pueden oscilar entre los 50.000 y 150.000 dólares por unidad [1].

La barrera económica resulta en una cobertura de medición insuficiente, donde típicamente solo entre un 30 % y 40 % de los pozos en un activo productivo cuenta con instrumentación adecuada [2], dejando una brecha crítica de monitorización en la mayoría de los puntos de producción.

Cuando entramos al contexto de Petroecuador, esta problemática se manifiesta de manera particularmente crítica en la medición del gas asociado en los flares. Estudios realizados en campos similares al Pucuna indican que de 275 flares operativos, 220 carecen de sistemas de medición adecuados, aproximadamente el 80 % de los flares carecen de sistemas de medición adecuados [3]. Esta limitación impide cuantificar con precisión el volumen de gas que está siendo quemado y que podría destinarse a pro-

yectos de generación eléctrica, representando tanto un impacto ambiental significativo como una oportunidad económica desaprovechada. Además, la implementación de medidores físicos tradicionales trae consigo otros problemas. Los medidores físicos están expuestos a condiciones operativas adversas, incluyendo variaciones extremas de presión y temperatura, presencia de sólidos en suspensión, y fluidos corrosivos, lo que deriva en una confiabilidad operacional limitada. [4].

A nivel operativo del Campo Pucuna, cuando un medidor falla, la consecuencia inmediata es la pérdida completa de la capacidad de medición en ese pozo, generando un vacío de información que puede prolongarse por semanas o incluso meses, dependiendo de la complejidad de la reparación y la disponibilidad de repuestos especializados. Durante estos períodos, los ingenieros y supervisores deben recurrir a estimaciones basadas en modelos teóricos o correlaciones empíricas, las cuales carecen de la precisión necesaria para una toma de decisiones informada [5], impactando negativamente en la eficiencia operativa global.

Toda esta problemática tiene implicaciones directas en la rentabilidad y el desempeño financiero de las operaciones de gas. La falta de datos confiables sobre los caudales producidos impide la optimización de los regímenes de producción, dificulta la identificación temprana de problemas en los pozos o en los equipos de superficie [6], y compromete la exactitud de los reportes de producción, fundamentales para la comercialización y la planificación estratégica.

Para enfrentar este desafío, las técnicas de inteligencia artificial, particularmente las redes neuronales, han demostrado ser efectivas en el manejo de grandes volúmenes de datos y en la identificación de patrones complejos. La identificación de regímenes de flujo inadecuados y la falta de medición continua en los flares constituyen desafíos técnicos que requieren soluciones innovadoras [7].

1.2 Antecedentes investigativos

Esta sección presenta la revisión bibliográfica realizada para fundamentar el diseño del proyecto orientado al desarrollo de un sistema de Medidor Virtual de Flujo (VFM)

para la estimación precisa del flujo de gas en pozos de producción. El objetivo de esta revisión fue identificar investigaciones, artículos científicos y proyectos afines que sirvan como base metodológica y tecnológica para la presente propuesta.

La búsqueda bibliográfica se llevó a cabo en bases de datos y repositorios académicos reconocidos, tales como IEEE Xplore, ScienceDirect, ACM Digital Library, Google Scholar y ResearchGate, los cuales ofrecen publicaciones con alto nivel de rigor científico en el área de ingeniería petrolera e inteligencia artificial.

1.2.1 Cadenas de búsqueda utilizadas

Con el propósito de obtener resultados relevantes y actualizados, se emplearon las siguientes cadenas de búsqueda:

1. (“virtual flow meter” OR “VFM”) AND (“oil gas” OR “petroleum”) AND (“machine learning” OR “artificial intelligence”)
2. (“flow measurement” OR “gas flow”) AND (“neural networks” OR “artificial intelligence”) AND (“prediction” OR “estimation”)
3. (“multiphase flow” OR “gas volume fraction”) AND (“virtual metering” OR “soft sensor”)
4. (“SISTEMA” OR “oil gas data”) AND (“machine learning” OR “data driven”) AND (“flow rate” OR “production optimization”)

Estas cadenas se aplicaron a publicaciones comprendidas entre los años 2019 y 2025, seleccionando aquellos que cumplieran los criterios de inclusión definidos: pertinencia temática, relación con la medición virtual de flujo en la industria petrolera, aplicación de inteligencia artificial, accesibilidad al texto completo y publicaciones en inglés o español.

De un total de 45 documentos encontrados, al aplicar los criterios de inclusión, exclusión y pertinencia, se seleccionaron 30 investigaciones [4, 5, 8, 3, 7, 6, 1, 2, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30] que

aportan bases técnicas, metodológicas y tecnológicas directamente relacionadas con el desarrollo de la propuesta.

1.2.2 Criterios de selección aplicados

Los criterios empleados durante el proceso de selección de los artículos se resumen en la Tabla 1, que garantiza la calidad científica, la pertinencia temática y la coherencia metodológica de las fuentes utilizadas.

Cuadro 1: Criterios de inclusión, exclusión y pertinencia aplicados

Criterios de inclusión	Criterios de exclusión
Desarrollo de medidores virtuales de flujo para oil & gas	Estudios anteriores a 2019
Uso de IA, Machine Learning o redes neuronales aplicadas a VFM	Investigaciones fuera del ámbito petrolero
Comparación de modelos o métricas de precisión con indicadores medibles	Trabajos sin aplicación práctica o sin validación
Publicaciones con metodologías replicables en un sistema	Publicaciones sin acceso completo o sin revisión por pares
Estudios con indicadores aplicables como AAPE, R^2 , RMSE	Artículos puramente teóricos sin implementación
Trabajos relacionados con optimización de producción petrolera	Enfoques exclusivamente en flujo monofásico
Publicaciones desde 2019 en adelante	Modelos sin aplicación en tiempo real

Autor del cuadro: Marco Montesdeoca

La distribución de los artículos seleccionados por base de datos se presenta en la Tabla 2, evidenciando la diversidad de fuentes consultadas para garantizar una revisión comprehensiva del estado del arte.

Cuadro 2: Distribución de artículos seleccionados por base de datos

Base de datos	Artículos seleccionados
ScienceDirect (Elsevier)	7 artículos
arXiv	1 artículo
IEEE Xplore	2 artículos
MDPI	2 artículos
Taylor & Francis	1 artículo
ACS Publications	1 artículo
Otros	2 artículos

Autor del cuadro: Marco Montesdeoca

1.2.3 Analisis de los estudios

El artículo [1] habla sobre una revisión de las tendencias tecnológicas en medición de flujo multifásico en ductos *offshore*, abarcando medidores físicos (MPFM), técnicas de tomografía (rayos X, gamma, capacitivos), sensores distribuidos y soluciones de medición virtual.

Se discuten los rangos típicos de operación en plataformas marinas (presiones superiores a 100 bar, altas fracciones de gas y agua, condiciones de alta corrosión y presencia de arena), así como la evolución de la precisión de los MPFM comerciales (errores típicos de 2–5 % en caudal de líquido y 5–10 % en gas, dependiendo del régimen y la calibración). La revisión destaca la creciente adopción de *soft-sensors* y modelos de IA para reducir costos de instalación y mantenimiento, especialmente en campos maduros y líneas de transporte largas.

En [2] se cuantifica el desempeño de un enfoque de medición multifásica que combina métodos físicos y modelos virtuales, reportando cifras concretas de error. Para un conjunto de casos de validación, los autores muestran que el error relativo promedio entre la simulación y las mediciones de laboratorio es de 0.53 % para el caudal de líquido, 1.06 % para el gas y 1.46 % para la fracción de vacío, calculados a partir de un *dataset* de pruebas controladas en un lazo de flujo multifásico.

Las cifras son menores que las obtenidas con correlaciones convencionales o mode-

los no calibrados, y se obtienen integrando mediciones de presión diferencial, caídas de presión y señales de tomografía con un modelo calibrado y algoritmos de corrección.

En [3] se presenta una revisión de la literatura que sistematiza los enfoques de *Virtual Flow Metering* (VFM) basados en primeros principios (modelos mecanicistas/transitorios) y los enfoques *data-driven* (IA/ML). La revisión discute modelos tipo OLGA, simuladores transitorios comerciales y modelos de redes de pozos que resuelven ecuaciones de conservación de masa, momento y energía, frente a aproximaciones con redes neuronales, SVM, Random Forest, Gradient Boosting y métodos híbridos, señalando sus ventajas y limitaciones en términos de costo computacional, necesidad de datos, robustez frente a la no estacionariedad y facilidad de mantenimiento del modelo.

Se analizan múltiples estudios de campo y laboratorio con distintos rangos de presión, temperatura, GVF y tasas de producción, identificando que el uso de IA suele reducir el error con respecto a los modelos basados solo en correlaciones tradicionales, en especial cuando se dispone de *datasets* de cientos o miles de puntos.

El trabajo [4] aborda la confiabilidad de redes de flujo (*flow networks*) usando técnicas de inteligencia artificial, sustituyendo los métodos clásicos basados en combinatoria y simulaciones Monte Carlo de muy alto costo computacional. A partir de la estructura de la red y de parámetros como tasas de fallo, capacidades de aristas y niveles de demanda, se construyen modelos de IA (típicamente redes neuronales alimentadas con escenarios generados por simulación) que aproximan probabilidades de fallo de la red completa, reduciendo en órdenes de magnitud el número de corridas necesarias frente a enfoques puramente estocásticos.

Aunque el artículo se centra en redes genéricas (agua, gas, energía), metodológicamente demuestra que es posible aprender funciones de confiabilidad altamente no lineales a partir de datos históricos/sintéticos y luego explotarlas en tiempo casi real, lo que es directamente transferible a un sistema de VFM.

En [5] se compara un medidor virtual de flujo basado en redes neuronales clásicas con un enfoque de *Bayesian Neural Networks* (BNN) aplicado a datos de producción multiphásica de campos petroleros, donde se predicen simultáneamente caudales de gas,

petróleo y agua a partir de presiones, aperturas de choke y otras variables de proceso. El estudio entrena varios modelos *feed-forward* con diferentes arquitecturas y, en el caso bayesiano, incorpora distribuciones *a priori* sobre los pesos y técnicas de muestreo (por ejemplo, Monte Carlo o *variational inference*) para obtener no solo un valor puntual del caudal, sino un intervalo de credibilidad asociado a cada predicción.

Los resultados muestran que, para un conjunto de pozos analizados, las BNN mantienen niveles de error comparables a los modelos deterministas pero con mejor calibración de la incertidumbre, lo que permite detectar regiones de operación donde el VFM es menos fiable. Para el medidor virtual de flujo de gas en Petroecuador, este artículo justifica el uso de modelos probabilísticos (Bayesianos) en vez de modelos puramente deterministas, especialmente en regímenes con pocos datos o alta no estacionariedad.

En [6] se aborda la clasificación de regímenes de flujo gas-líquido (burbujas, tapones, estratificado, anular) en una tubería de 50,8 mm de diámetro interno, utilizando un método no intrusivo basado en sensores externos y técnicas de procesamiento de señales. El sistema trabaja en un rango de velocidad superficial de gas de 0 a 10 m/s y de líquido de 0 a 2 m/s, con números de Reynolds que varían entre 1200 y 52 000 para el gas y entre 800 y 10 000 para el líquido, lo que cubre regímenes laminares y turbulentos en instalaciones experimentales de agua-aire.

Se extraen características de las señales (por ejemplo, estadísticas de amplitud y frecuencia) y se entrenan clasificadores para identificar el patrón de flujo, obteniendo altos niveles de acierto en la discriminación de los regímenes principales. Esta capacidad de detectar automáticamente el régimen de flujo es crucial para el VFM, ya que la relación entre las variables de superficie y el caudal real de gas cambia significativamente entre regímenes; integrar este tipo de clasificación como etapa previa podría mejorar la precisión de tu modelo para los pozos de Petroecuador.

El trabajo [7] propone un modelo totalmente analítico para estimar la fracción de volumen de gas (GVF) en flujos bifásicos, evitando recurrir a métodos numéricos iterativos o correlaciones puramente empíricas. A partir de ecuaciones de conservación y suposiciones sobre la distribución de fases, se obtiene una expresión cerrada para GVF en función de parámetros como el caudal de mezcla, la geometría de la tubería

y las propiedades de las fases. El modelo se contrasta con datos experimentales de distintos arreglos de laboratorio y condiciones de operación, mostrando una buena concordancia cualitativa y cuantitativa con las mediciones y mejorando el ajuste respecto a formulaciones previas de tipo *drift-flux*. Este modelo ofrece una forma precisa de calcular GVF que luego puede ser usada como variable de entrada o como etiqueta en el entrenamiento de tu sistema de medidor virtual de flujo de gas, especialmente en casos de flujo multifásico donde la fracción de gas es crítica para la estimación correcta del caudal.

El artículo [8] propone un “virtual flow sensor” basado en redes neuronales artificiales (ANN) entrenadas con datos medidos en línea (presiones, temperaturas, señales de instrumentos, aperturas de válvulas) para reemplazar o complementar medidores físicos de caudal. El trabajo detalla la construcción de un modelo supervisado multisalida, donde la ANN recibe como entradas un vector de mediciones de superficie y condición de operación, y regresa como salidas los caudales de las fases, usando conjuntos de entrenamiento y prueba separados para evaluar generalización.

El modelo se calibra con técnicas como *backpropagation* y entrenamiento por lotes, y se analiza el impacto del número de neuronas ocultas y capas en el error de predicción. Se reportan errores promedio bajos (del orden de unos pocos puntos porcentuales) en las variables principales, mostrando que es viable implementar un VFM puramente *data-driven* usando solo sensores estándar.

Se demuestra que con un conjunto adecuado de variables de superficie y un entrenamiento riguroso se puede reemplazar un medidor físico de gas por un modelo ANN que opere en tiempo real sobre datos de Petroecuador.

El trabajo [9] introduce el uso de aprendizaje multi-tarea (*multi-task learning*, MTL) para construir un medidor virtual que predice simultáneamente los caudales de gas, petróleo y agua en varios pozos, compartiendo una parte de la red neuronal entre tareas y pozos. A partir de datos históricos de producción de múltiples pozos (con cientos o miles de registros por pozo), se entrena una red con capas compartidas para extraer representaciones comunes del comportamiento del sistema y cabezas específicas por tarea para cada fase.

El artículo muestra que, respecto a entrenar modelos independientes por pozo o por fase, el enfoque MTL reduce de forma sistemática métricas de error relativas (por ejemplo MAE y MAPE) y mejora la estabilidad temporal del modelo, especialmente cuando algunos pozos tienen menos datos que otros.

El artículo [15] describe un VFM multifásico híbrido que combina *Ensemble Learning data-driven* con un simulador transitorio de flujo multifásico basado en primeros principios. El sistema utiliza datos de sensores estándar (presiones, temperaturas, apertura de choke, etc.) y un simulador físico (Flux Simulator, OLGA/HYSYS-like) para estimar caudales de aceite, gas y agua, y luego usa un “combiner” que pondera las estimaciones de los modelos base según su historial de desempeño y un mecanismo de “confidence decay”. Este VFM se probó durante 6 meses en tiempo real en una instalación *offshore* con 2 pozos de petróleo, obteniendo una mejora de alrededor del 50 % en la performance de medición respecto a VFMs individuales, y un MAPE menor al 5 % para el sistema combinado.

El artículo también detalla el uso de modelos IPR/VLP, caracterización PVT (usando Multiflash®) y modelos de choke con ecuaciones del tipo $P_{wh} = CR^m Q/S^n$. Este trabajo es un ejemplo directo de arquitectura híbrida que podrías emular: usar un modelo físico simplificado de los pozos de Petroecuador y un modelo de *ensemble/combiner* basado en IA para estimar el flujo de gas con alta robustez.

En [16] se aplica IA para estimar el caudal de petróleo en pozos con *gas-lift*, integrando un simulador transitorio propietario con el software PIPESIM para validar la distribución de presión, temperatura y caudal de gas a lo largo del pozo. El modelo propuesto permite calcular, además, densidad, velocidad, fracción de volumen y viscosidad de cada fase (líquido, agua, gas), así como la masa, densidad y radio interno de la capa de parafina formada en la tubería.

Para la parte de IA, se diseña un algoritmo de ajuste automático del modelo a datos reales, basado en redes neuronales que se reentrenan para adaptar el modelo a diferentes GOR, WOR, geometrías de pozo y rangos de caudal. En pruebas sobre un PC con Intel Core i5-7600 a 3,5 GHz y 12 GB de RAM, el entrenamiento toma 5–7 minutos y cada cálculo alrededor de 3 segundos, con un criterio de precisión objetivo del 5 % para los

parámetros de interés. Este artículo demuestra que es factible implementar un VFM con IA para sistemas complejos de *gas-lift*, con tiempos de cómputo compatibles con operación casi en tiempo real, lo cual es totalmente extrapolable a tu caso de pozos de producción de Petroecuador con alto contenido de gas.

El artículo [11] aborda explícitamente el problema de la no estacionariedad en VFMs, es decir, el deterioro del desempeño de modelos estacionarios cuando cambian lentamente las condiciones de yacimiento y del sistema de producción. El estudio utiliza datos históricos de 10 pozos de petróleo y entrena seis tipos de modelos (regresión lineal, red neuronal *feed-forward*, modelo multi-tarea, modelo mecanicista, modelo híbrido de error y modelo híbrido de área funcional) que se recalibran usando dos esquemas de aprendizaje pasivo: *periodic batch learning* y *online learning*, con diferentes frecuencias de actualización.

A partir de estadísticas de los intervalos de pruebas de pozo, se muestra que muchos pozos tienen más del 40–60 % de los intervalos de prueba menores a 2 semanas, mientras que otros superan los 6 meses o 1 año entre pruebas, lo que genera distribuciones de datos muy desiguales. Los resultados evidencian que, cuando llegan mediciones con suficiente frecuencia, el reentrenamiento frecuente permite mantener una “excelente” precisión de predicción, mientras que con datos esporádicos es necesario combinar actualización con conocimiento experto para no degradar el modelo.

En [10] se aplica el algoritmo de *Gradient Boosting* (GBM) para el monitoreo de producción de petróleo, entrenando un modelo de árboles de decisión potenciados que predice el caudal de aceite a partir de variables de proceso y operación como presiones, temperaturas, apertura de choke y propiedades PVT.

El estudio utiliza un conjunto de datos industriales con cientos de días de operación y muestra cómo el GBM captura relaciones no lineales y efectos de interacción entre variables que modelos lineales o correlaciones simplificadas no pueden representar. Se evalúan métricas de desempeño estándar (R^2 , RMSE, error relativo promedio) y se observa que el GBM supera a modelos base (por ejemplo, regresión lineal) con reducciones significativas del error en la predicción de caudal.

El artículo [12] estudia la predicción de caudal de petróleo a través de un medidor de orificio instalado en un oleoducto de 18 pulgadas de diámetro que transporta producción desde el campo Cheshmeh Khosh hacia la unidad Ahvaz-3. El medidor tiene un diámetro interno de 17,5 pulgadas, una placa de orificio de 8,4 pulgadas y una razón beta (β) de 0.48. Se analiza un *dataset* de 1037 registros (830 para entrenamiento, 207 para prueba) con una variable dependiente (caudal de petróleo en stb/día) y cinco variables de entrada: temperatura, presión, gravedad específica del petróleo (SG), diferencial de presión ΔP (en pulgadas de agua) y viscosidad μ (en cSt). Los caudales van desde 2195 hasta 119 005 stb/día, con temperaturas de 14 a 116 °F y presiones de 0,8 a 250 psig. Se comparan cinco algoritmos de IA: ANFIS, LSSVM, RBF, MLP y GEP, usando métricas como R^2 , AARD %, RMSE y STD.

Los resultados muestran que el MLP logra el mejor desempeño global, con el mayor R^2 y los menores valores de AARD, RMSE y desviación estándar a través de los 1037 puntos, mientras que ANFIS y LSSVM rinden peor en el rango de caudales bajos (< 40000 stb/día). Este estudio demuestra que, para un medidor tradicional de orificio, un VFM basado en MLP puede superar la aproximación puramente basada en Bernoulli y correlaciones, lo que respalda tu decisión de usar redes neuronales para estimar el flujo de gas a partir de mediciones de superficie.

En [13] se propone una arquitectura de *Virtual Flow Meter* distribuido, diseñada para trabajar con datos de sensores distribuidos en fibra óptica y otros dispositivos a lo largo de ductos y pozos. El enfoque integra modelos de aprendizaje continuo (*continual learning*) para manejar cambios de distribución en los datos, inspirándose en técnicas de mitigación del “catastrophic forgetting” ampliamente estudiadas en visión por computador y sistemas de clasificación *online*. El sistema se despliega sobre infraestructura de cómputo de alto rendimiento (recursos de UNINETT Sigma2) y busca actualizar los modelos a medida que se incorporan nuevos datos, manteniendo la precisión sin necesidad de reentrenarlo completamente desde cero.

El artículo [14] desarrolla dos modelos de redes neuronales para un campo *offshore* en África occidental con 5 pozos y dos líneas de producción, donde se dispone de medición MPFM para entrenar modelos de VFM. Se construyen un *Neural Well Model*

(específico por pozo, con entradas como presión y temperatura en fondo fluyente, presiones aguas arriba y abajo del choke, presión en el anular y apertura del choke) y un *Neural Choke Model* (más general, centrado en el comportamiento del estrangulador). En términos de desempeño, se reportan errores relativos medios (*Mean Relative Error*) de 0.57 % (pozo) vs 2.03 % (choke) para caudal de petróleo, 0.47 % vs 2.36 % para caudal de gas y 3.36 % vs 29.28 % para caudal de agua, comparando el modelo de pozo con el de choke.

En [17] presenta un estudio específico para pozos con alto *gas-oil ratio* (GOR) y alto *water-cut* (WC), donde los empíricos tradicionales tipo Gilbert, Baxendell, Ros y Achong muestran errores muy altos. El trabajo recopila 548 puntos de datos de pruebas de separador portátil en un campo del Medio Oriente, con GOR entre 1000 y 9351 scf/stb y WC hasta 60 %; se descartan puntos con $GOR < 1000$ y $WC = 0$.

Los datos se separan en dos subconjuntos: 248 puntos en flujo subcrítico (relación $P_D/P_U \geq 0.5$) y 300 puntos en flujo crítico ($P_D/P_U < 0.5$). Se entrenan modelos de Random Forest (RF) y Support Vector Machine (SVM) con una partición 70 % entrenamiento / 30 % prueba, usando como entradas GOR, WC, presión aguas arriba (P_U), presión del separador (P_D) y tamaño de choke, y como salida el caudal de petróleo Q_{oil} .

En condiciones críticas, el modelo SVM alcanza $R^2 = 0.94$ (train) y 0.93 (test) con AAPE = 1.0 % (train) y 1.8 % (test) (promedio 1.4 %), mientras que RF logra AAPE ≈ 0.7 % en todas las condiciones. Comparados con las correlaciones clásicas, cuyos AAPE van de 34 a 43 %, los modelos de IA mejoran la precisión en un 34 % relativo. Este artículo muestra, con cifras muy concretas, que modelos RF/SVM entrenados con parámetros de choke y presiones pueden predecir caudales con errores ≈ 1 % incluso en sistemas de alto GOR, muy similares al tipo de pozos de gas que podrías encontrar en Petroecuador en 2026.

Cuadro 3: Metodologías de Investigación

Metodología de Investigación	Artículos
Revisión Sistemática	[3], [1]
Estudio de Caso / Exploratoria	[4]–[8], [7], [6], [2]–[17]

Autor del cuadro: Marco Montesdeoca

Como se muestra en la Tabla 3, los estudios de caso y enfoques exploratorios constituyen la metodología predominante en dieciséis de las investigaciones revisadas, mientras que solo dos estudios emplean revisión sistemática. Esta distribución evidencia que la naturaleza aplicada del Virtual Flow Metering y el desarrollo de sistemas de estimación de flujo demandan validación empírica con datos específicos de cada campo petrolero. Los hallazgos justifican la adopción de un estudio de caso para los pozos de Petroecuador, permitiendo capturar la complejidad de sus condiciones operativas y adaptar las soluciones a su contexto particular.

Cuadro 4: Fuentes y Tipos de Datos

Fuente/Tipo de Datos	Artículos
Datos Propietarios (Campo)	[5], [8], [2]–[12], [14]–[17]
Datos Experimentales (Laboratorio)	[7], [6], [2]
Simulación Numérica	[4], [3], [15], [16]
Sensores Externos (Fibra Óptica, Acústicos)	[6], [13]

Autor del cuadro: Marco Montesdeoca

En la Tabla 4 se identifica a los datos propietarios de campo como la fuente más recurrente en nueve estudios, lo que refleja la necesidad de validar los modelos VFM en condiciones operativas reales. Sin embargo, la diversidad de fuentes restantes —desde datos experimentales hasta simulaciones numéricas— subraya la importancia de adaptar las técnicas de recopilación al contexto específico. Estos hallazgos respaldan la decisión de utilizar datos propietarios de los pozos de Petroecuador, ya que permitirán capturar las particularidades de su operación y condiciones de producción.

Cuadro 5: Técnicas de Preprocesamiento

Técnica de Preprocesamiento	Artículos
Limpieza e Imputación de Datos	[8], [11], [17]
Normalización/Estandarización	[8], [12], [14]
Selección de Características	[6], [11], [17]
Balanceo de Datos (SMOTE, etc.)	[11], [17]

Autor del cuadro: Marco Montesdeoca

El Tabla 5 revela que la limpieza e imputación de datos emerge como la técnica de preprocesamiento más utilizada en tres investigaciones, destacando su criticidad para abordar problemas comunes en datos de campo como valores faltantes y outliers. La presencia equilibrada de normalización y selección de características evidencia la necesidad de preparar adecuadamente los datos antes del modelado. Estos hallazgos sustentan la implementación de un pipeline robusto de preprocesamiento para los datos de Petroecuador, donde se anticipan desafíos típicos de datos operacionales.

Cuadro 6: Ingeniería y Selección de Características

Técnica	Artículos
Ingeniería de Características	[7], [6], [12], [17]
Selección de Características	[6], [11], [17]
Análisis de Correlación	[12], [17]
Extracción de Atributos de Señales	[6]

Autor del cuadro: Marco Montesdeoca

Según la Tabla 6, la ingeniería de características muestra mayor prevalencia en cuatro estudios, demostrando que la transformación de variables operacionales en características significativas es crucial para el éxito de los modelos VFM. La presencia del análisis de correlación y extracción de atributos evidencia la importancia de capturar relaciones físicas subyacentes en los datos. Estos resultados validan la incorporación de un proceso exhaustivo de ingeniería de características para los pozos de Petroecuador, orientado a capturar patrones clave en el comportamiento del flujo de gas.

Cuadro 7: Modelos de Clasificación/Regresión

Tipo de Modelo	Artículos
Redes Neuronales (ANN, MLP, BNN)	[5], [8], [9], [12], [14]
Ensembles (Random Forest, Gradient Boosting)	[10], [15], [17]
Support Vector Machine (SVM)	[12], [17]
Modelos Híbridos (Físico + ML)	[3], [15], [16]

Autor del cuadro: Marco Montesdeoca

Como se presenta en el Tabla 7, las redes neuronales lideran su aplicación en cinco estudios, reflejando su capacidad para capturar relaciones no lineales complejas en datos de flujo multifásico. Los modelos de ensemble y enfoques híbridos muestran presencia significativa, indicando la necesidad de combinar múltiples aproximaciones para mejorar la robustez predictiva. Estos hallazgos justifican la evaluación comparativa de múltiples arquitecturas para los pozos de Petroecuador, priorizando modelos que balanceen precisión e interpretabilidad.

Cuadro 8: Modelado Híbrido

Técnica Híbrida	Artículos
Combinación de Modelos Físicos y Data-Driven	[3], [15], [16]
Ensemble de Múltiples Modelos	[10], [15], [17]
Multi-Task Learning	[9]

Autor del cuadro: Marco Montesdeoca

El Tabla 8 evidencia que la combinación de modelos físicos y data-driven lidera las técnicas híbridas con tres estudios, seguido por ensembles en tres investigaciones, mostrando la creciente preferencia por approaches que aprovechen el conocimiento de dominio junto con patrones aprendidos de datos. La aparición de Multi-Task Learning en un estudio demuestra la evolución hacia arquitecturas más sofisticadas. Estos resultados apoyan la implementación de un enfoque híbrido para Petroecuador, aprovechando las ventajas de modelos físicos y data-driven.

Cuadro 9: Optimización de Hiperparámetros

Técnica de Optimización	Artículos
Búsqueda en Cuadrícula (GridSearch)	[8], [10], [12]
Optimización Bayesiana	[5]
Selección Manual	[17]

Autor del cuadro: Marco Montesdeoca

En el Tabla 9 se observa que GridSearch presenta mayor adopción con tres estudios, indicando su popularidad por su enfoque sistemático y garantía de exploración exhaustiva. La presencia de optimización bayesiana en un estudio sugiere tendencias hacia métodos más eficientes computacionalmente. Los hallazgos respaldan la selección de GridSearch para los modelos de Petroecuador, dado su balance entre completitud y viabilidad computacional.

Cuadro 10: Arquitecturas de Software/Sistema

Arquitectura/Patrón	Artículos
Sistema Distribuido	[13]
Arquitectura Híbrida (Física + ML)	[3], [15], [16]
Integración en un sistema	[13], [16]

Autor del cuadro: Marco Montesdeoca

Como se detalla en el Tabla 10, las arquitecturas distribuidas e híbridas muestran igual prevalencia en tres estudios cada una, demostrando la necesidad de sistemas escalables que combinen múltiples fuentes de conocimiento. La integración con un sistema en dos estudios subraya la importancia de la interoperabilidad con sistemas existentes. Estos hallazgos justifican el diseño de una arquitectura distribuida e híbrida para Petroecuador, facilitando la integración nativa con su infraestructura deseada.

Cuadro 11: Tecnologías y Librerías

Tecnología/Librería	Artículos
Python (Scikit-learn, TensorFlow, PyTorch)	[5], [8], [9]–[12], [14]–[17]
Simuladores (OLGA, PIPESIM, HYSYS)	[3], [15], [16]
Sensores (Fibra Óptica, Tomografía)	[6], [13]

Autor del cuadro: Marco Montesdeoca

En el Tabla 11 se identifica a Python con sus librerías de machine learning como la tecnología predominante en nueve estudios, evidenciando su madurez para implementar pipelines completos de VFM. La presencia de simuladores comerciales en tres estudios refleja la importancia de validar contra referencias físicas. Estos resultados validan la selección de Python como tecnología base para el VFM de Petroecuador, complementado con validación contra simuladores cuando sea posible.

Cuadro 12: Técnicas de Validación

Técnica de Validación	Artículos
Validación Cruzada	[2], [10], [12], [14], [17]
Hold-Out	[8], [6], [9]
Comparación con Medidores Físicos	[2], [14], [15]

Autor del cuadro: Marco Montesdeoca

El Tabla 12 muestra que la validación cruzada es claramente predominante en cinco estudios frente al hold-out en tres, destacando su superioridad estadística para proveer estimaciones robustas del rendimiento en datos limitados. La comparación con medidores físicos en tres estudios enfatiza la necesidad de validación contra referencias confiables. Estos hallazgos respaldan la implementación de validación cruzada para los modelos de Petroecuador, complementada con comparación contra mediciones físicas disponibles.

Cuadro 13: Métricas de Evaluación

Métrica	Artículos
Error Porcentual Absoluto Promedio (AAPE)	[2], [12], [14], [17]
Coefficiente de Determinación (R^2)	[10], [12], [17]
RMSE / MAE	[10], [12], [14]
MAPE	[2], [15]

Autor del cuadro: Marco Montesdeoca

En el Tabla 13 se identifica al AAPE como la métrica más utilizada en cuatro estudios, superando claramente a R^2 y RMSE/MAE. Esta preferencia evidencia la necesidad de métricas intuitivas que capturen el error relativo, especialmente importante en aplicaciones industriales donde la magnitud del caudal varía significativamente. Sin

embargo, la presencia complementaria de R^2 y RMSE revela la importancia de una evaluación multidimensional. Estos resultados justifican un enfoque multi-métrica para Petroecuador, combinando AAPE para interpretabilidad con R^2 para evaluación de varianza explicada.

Cuadro 14: Interpretabilidad de Modelos (XAI)

Técnica XAI	Artículos
Análisis de Importancia de Características	[10], [17]
SHAP / LIME	[10]

Autor del cuadro: Marco Montesdeoca

El Tabla 14 indica que el análisis de importancia de características aparece en dos estudios, posicionándose como técnica fundamental para interpretar modelos complejos de VFM. Su presencia, aunque aún limitada, refleja la creciente conciencia sobre la necesidad de transparencia en sistemas de IA aplicados a operaciones críticas. Estos hallazgos sustentan la incorporación de técnicas XAI para los modelos de Petroecuador, permitiendo a los ingenieros comprender los factores que influyen en las estimaciones de flujo.

1.3 Fundamentación teórica

Esta sección constituye la base conceptual que sustenta el desarrollo del sistema de Medidor Virtual de Flujo, integrando principios de inteligencia artificial, mecánica de fluidos y sistemas de adquisición de datos. Esta estructura conceptual permite comprender las relaciones entre las variables operacionales del campo petrolero y los caudales de gas, proporcionando el fundamento científico para el diseño de algoritmos predictivos. El marco se organiza en dimensiones interrelacionadas que abarcan desde los fundamentos de los algoritmos de inteligencia artificial aplicado a medición de flujo, hasta las arquitecturas de software para integración con sistemas de medición, estableciendo así un referente comprehensivo para la implementación de la solución propuesta.

1.3.1 Virtual Flow Metering y su aplicacion

Un Medidor Virtual de Flujo (VFM, Virtual Flow Metering) es una herramienta basada en software que permite estimar el caudal de fluidos, como gas o petróleo, utilizando modelos matemáticos o algoritmos de inteligencia artificial a partir de variables operacionales como presión, temperatura y condiciones de producción. A diferencia de los medidores físicos tradicionales, el VFM no requiere la instalación de equipos adicionales en campo, lo que reduce costos y facilita la monitorización continua del proceso, proporcionando información útil para la toma de decisiones en sistemas industriales. Para Petroecuador, en el campo Pucuna, es muy importante el tener un sistema para poder medir lo que es el caudal del gas sin el costo de implementar medidores físicos.

En [[3], [6], [14], [15], [18], [19]] se describe que el VFM permite monitoreo continuo, reducción de costos de instrumentación y mejor soporte para decisiones operacionales mediante modelos híbridos y de inteligencia artificial para predicción de flujo de gas.

1.3.2 Machine Learning para Virtual Flow Metering

Machine Learning es una rama de la inteligencia artificial que permite a los sistemas aprender patrones a partir de datos y realizar predicciones o tomar decisiones sin programación explícita. Para el proyecto de VFM en Petroecuador, esta tecnología es fundamental ya que permitirá capturar las complejas relaciones no lineales entre las variables operacionales del Campo Pucuna y los caudales de gas, sin depender de modelos físicos simplificados que no consideran las particularidades operacionales específicas de cada pozo.

En [[4]-[3], [9]-[11], [14]-[17]] se aplicó ML para la predicción de caudales de flujo, análisis de regímenes multiphásicos y modelado de sistemas de producción con técnicas de regresión, redes neuronales y modelos de ensamble. En este trabajo se emplearán enfoques de aprendizaje automático con el fin de desarrollar un modelo predictivo orientado a estimar el flujo de gas en pozos de producción de Petroecuador, específicamente adaptado a las condiciones operacionales del Campo Pucuna.

1.3.3 Arquitecturas de Redes Neuronales para VFM

Las redes neuronales artificiales son modelos computacionales inspirados en el cerebro humano que permiten capturar relaciones no lineales complejas entre variables de entrada y salida. Su aplicación en el VFM del Campo Pucuna es crucial debido a la naturaleza multivariable y no lineal de los datos de producción, donde relaciones simples no logran capturar la complejidad de los procesos de flujo multifásico.

En [[5], [8], [9], [12], [14]] se implementaron arquitecturas de redes neuronales feed-forward, Bayesian Neural Networks y modelos multi-tarea para la estimación de caudales multiphásicos. En este trabajo se utilizarán redes neuronales con múltiples capas ocultas para modelar la relación entre variables operacionales del Campo Pucuna y el flujo de gas, permitiendo capturar interacciones complejas que los métodos tradicionales no pueden representar.

1.3.4 Enfoques Híbridos y Ensemble Learning

El Ensemble Learning combina múltiples modelos de machine learning para mejorar el rendimiento predictivo y la robustez del sistema. Para el VFM de Petroecuador, esta aproximación es esencial dado que diferentes algoritmos pueden capturar distintos aspectos de los datos operacionales, proporcionando estimaciones más estables y confiables que un modelo individual.

En [[10], [15], [17]] se aplicaron técnicas de Gradient Boosting, Random Forest y combinadores híbridos para capturar patrones complejos y mejorar la precisión de los VFM. En este trabajo se considerará la incorporación de estrategias de ensemble para el Campo Pucuna, combinando las fortalezas de diferentes algoritmos para obtener estimaciones de flujo más robustas frente a variaciones operacionales.

1.3.5 Procesamiento de Datos y Feature Engineering

El preprocesamiento consiste en técnicas para limpiar, transformar y balancear los datos antes de modelarlos. En el contexto del Campo Pucuna, esta etapa es crítica

debido a que los datos suelen contener ruido, valores faltantes y outliers que pueden afectar significativamente el desempeño de los modelos predictivos.

En [[6], [11], [17]] se aplicaron métodos para corregir desbalance de datos, reducir ruido y seleccionar variables relevantes. En este trabajo se aplicarán procedimientos de depuración y transformación específicos para los datos del Campo Pucuna, incluyendo técnicas de imputación robustas y normalización adaptada a las distribuciones típicas de variables operacionales petroleras.

1.3.6 Selección e Ingeniería de Características

Este proceso consiste en elegir y transformar variables para mejorar el rendimiento predictivo del modelo. Para el VFM del Campo Pucuna, la ingeniería de características permitirá crear variables derivadas que capturen comportamientos operacionales específicos, como tendencias temporales, relaciones entre presiones y temperaturas, y patrones estacionales de producción.

En [[7], [6], [12], [17]] se emplearon técnicas de ingeniería de características y análisis de atributos para identificar factores influyentes en el caudal de flujo. En este trabajo se utilizarán procedimientos de selección y construcción de variables estratégicas basadas en el conocimiento del dominio operacional del Campo Pucuna, creando características que reflejen el comportamiento específico de sus pozos.

1.3.7 Métricas de Evaluación de Modelos

Son indicadores que cuantifican la calidad del modelo predictivo. Para el VFM de Petroecuador, la selección de métricas apropiadas es crucial para asegurar que el sistema cumpla con los requerimientos de precisión operacional y permita una evaluación realista del desempeño en condiciones de campo.

En [[2], [10], [12], [14], [17]] se emplearon indicadores como AAPE, R^2 , RMSE y MAPE para evaluar modelos de VFM. En este trabajo se aplicarán métricas que permitan medir no solo el rendimiento estadístico del modelo, sino también su utilidad

práctica para la toma de decisiones operacionales en el Campo Pucuna.

1.3.8 Clasificación de Regímenes de Flujo

La clasificación es una técnica de Machine Learning que asigna una etiqueta a cada instancia de datos. En el contexto del VFM para Petroecuador, esta técnica permitirá identificar automáticamente los diferentes regímenes de flujo presentes en los pozos del Campo Pucuna, lo que es esencial ya que las relaciones entre variables operacionales y caudales varían significativamente entre regímenes.

En [[6], [11]] se utilizaron enfoques de clasificación para identificar regímenes de flujo multiphásico, evaluando medidas como precisión y sensibilidad. En este trabajo se emplearán técnicas de clasificación supervisada para categorizar condiciones operacionales de los pozos del Campo Pucuna, permitiendo adaptar las estrategias de estimación según el régimen de flujo identificado.

1.3.9 Modelado Híbrido

El modelado híbrido combina técnicas de machine learning con modelos basados en primeros principios para mejorar el desempeño predictivo. Para el VFM del Campo Pucuna, este enfoque es particularmente valioso ya que permite aprovechar tanto el conocimiento físico del proceso de flujo como la capacidad de los datos para capturar comportamientos específicos del campo.

En [[3], [15], [16]] se aplicaron enfoques híbridos que integran simuladores físicos con modelos data-driven para capturar patrones complejos y mejorar la precisión. En este trabajo se considerará la incorporación de estrategias combinadas que integren principios físicos básicos del flujo de gas con modelos de machine learning entrenados con datos específicos del Campo Pucuna.

1.3.10 Optimización de Hiperparámetros

La optimización de hiperparámetros busca los valores que maximizan el rendimiento del modelo. En el desarrollo del VFM para Petroecuador, esta etapa es fundamental para asegurar que los algoritmos de inteligencia artificial estén configurados óptimamente para las características particulares de los datos del Campo Pucuna.

En [[8], [10], [12], [17]] se aplicaron diferentes métodos de búsqueda para mejorar la generalización de los modelos predictivos. En este trabajo se realizarán procesos de ajuste de parámetros específicos para las condiciones operacionales del Campo Pucuna, optimizando el comportamiento del modelo de VFM para maximizar su precisión y robustez.

1.3.11 Validación de Modelos

La validación evalúa la capacidad del modelo para generalizar a datos no vistos. Para el VFM de Petroecuador, una validación rigurosa es esencial para garantizar que el sistema funcione de manera confiable en las condiciones operacionales reales del Campo Pucuna y no solo en los datos de entrenamiento.

En [[2], [12], [14], [16], [17]] se aplicaron técnicas de validación cruzada y evaluación en conjuntos independientes. En este trabajo se utilizarán estrategias de validación específicas para el contexto operacional del Campo Pucuna, incluyendo validación temporal que simule condiciones reales de operación.

1.3.12 Aprendizaje Continuo

El aprendizaje continuo permite a los modelos adaptarse a cambios en la distribución de datos sin sufrir olvido catastrófico. Esta capacidad es crucial para el VFM del Campo Pucuna, ya que las condiciones operacionales de los pozos pueden cambiar con el tiempo debido a factores como declinación de producción, cambios en la estrategia operacional o modificaciones en la infraestructura.

En [[11], [13]] se implementaron técnicas de actualización incremental y *online learning* para mantener la precisión de los VFM. En este trabajo se incorporarán mecanismos de recalibración periódica adaptados a las dinámicas operacionales del Campo Pucuna, permitiendo que el sistema mantenga su precisión a lo largo del tiempo.

1.3.13 Arquitectura de Software

La arquitectura de software define la estructura, componentes y comunicación de un sistema. Para el VFM de Petroecuador, una arquitectura bien diseñada es esencial para garantizar la escalabilidad, mantenibilidad y integración con los sistemas existentes del Campo Pucuna.

En [[13], [15]] se implementaron estructuras arquitectónicas orientadas a la separación de responsabilidades y la escalabilidad. En este trabajo se adoptará una arquitectura que permita organizar adecuadamente los componentes del sistema VFM y facilite su integración con la infraestructura tecnológica existente en el Campo Pucuna.

El análisis integral de los antecedentes investigativos y el marco teórico evidencia la viabilidad técnica y metodológica del proyecto propuesto. Los estudios revisados validan la efectividad de los algoritmos de machine learning para la estimación de flujo de gas en contextos similares al Campo Pucuna, destacando el potencial de modelos como Random Forest, SVM y redes neuronales. Se confirma la eficacia de arquitecturas distribuidas para el desarrollo de sistemas VFM robustos que puedan operar efectivamente en las condiciones específicas de Petroecuador. Esta fundamentación sustenta técnicamente la propuesta y proporciona un marco de referencia sólido para el desarrollo e implementación del sistema de Medidor Virtual de Flujo para pozos de Petroecuador.

1.4 Objetivos

1.4.1 Objetivo general

Desarrollar un sistema de Medidor Virtual de Flujo (VFM) basado en algoritmos de inteligencia artificial para la predicción del flujo de gas en los pozos de producción del campo Pucuna durante el periodo académico 2026

1.4.2 Objetivos específicos

- Identificar las variables operacionales críticas de los sistemas y bases de datos históricas de la empresa PetroEcuador, para construir el dataset de entrenamiento del modelo de inteligencia artificial.
- Analizar algoritmos de inteligencia artificial que permiten la predicción del flujo del gas en los sistemas de medición.
- Implementar un sistema integral de Medidor Virtual de Flujo, basado en algoritmos de inteligencia artificial, que permita la predicción del flujo de gas asociado en los flares.

2. METODOLOGÍA

El presente capítulo describe la metodología empleada para el desarrollo de la investigación, detallando los materiales, métodos y técnicas utilizados para la recolección y análisis de la información. En este sentido, se establecen los procedimientos para, evaluar las condiciones actuales del sistema de medición de flujo y sustentar la implementación de un Medidor Virtual de Flujo (VFM) basado en algoritmos de inteligencia artificial. Se especifican los instrumentos utilizados, la población total de estudio y el enfoque o enfoques metodológicos adoptados, garantizando la validez y confiabilidad de los resultados obtenidos.

2.1 Materiales

En esta sección, se utilizaron los siguientes materiales:

Entrevistas semiestructuradas: Dirigidas al personal técnico y operativo del área de producción, con el fin de identificar las variables operacionales críticas, analizar las limitaciones de los métodos actuales de medición, comprender los flujos de trabajo existentes y recopilar requerimientos para el desarrollo del sistema integral de Medidor Virtual de Flujo (VFM), según la guía de preguntas del [Anexo B](#).

Fichas Documentales: Se utilizaron para recopilar, organizar y analizar información relevante proveniente de fuentes científicas, permitiendo sustentar teóricamente el desarrollo del sistema VFM y la aplicación de algoritmos de inteligencia artificial.

2.2 Métodos

La presente investigación adopta un enfoque cualitativo para el análisis del problema, el enfoque cualitativo se desarrolla a través de entrevistas semiestructuradas ([Anexo B](#)), proporcionando una comprensión más profunda de los procesos actuales y sus limitaciones. Esta metodología permite obtener una visión integral y fundamentada del contexto de estudio.

Se utilizaron fichas bibliograficas para la recoleccion tanto cualitativa como cuantitativa ya que proporciona datos para el sustento del desarrollo del sistema.

2.2.1 Modalidad de investigación

La presente investigación adopta un enfoque metodológico multimodal que integra tres modalidades complementarias:

Investigación documental: Revisión de literatura científica reciente sobre Virtual Flow Metering, algoritmos de inteligencia artificial y sistemas de medición en la industria oil & gas. Esta modalidad sustenta el marco teórico y guía la selección de técnicas y tecnologías y se justifica mediante el uso de fichas documentales donde se verán los mayores aportes de cada la bibliografía..

Investigación de campo: Recopilación de información directa en el Campo Pucuna de Petroecuador, con el propósito de identificar los procesos actuales de medición, las limitaciones operativas y los requerimientos específicos del personal técnico. Para ello, se emplearon entrevistas semiestructuradas como técnica principal de recolección de datos ([Anexo B](#)), ya que permiten obtener información detallada, contextualizada y basada en la experiencia de los profesionales, facilitando la comprensión de la problemática real y aportando insumos clave para el desarrollo del sistema propuesto.

Investigación aplicada: Desarrollo e implementación de un sistema de Medidor Virtual de Flujo que integre algoritmos de inteligencia artificial para la estimación precisa del flujo de gas, respondiendo a las necesidades identificadas en campo.

2.2.2 Población y muestra

El estudio se realizará con el personal técnico del Departamento de Instrumentación y Control del Campo Pucuna:

- 1 Supervisor
- 1 Supervisor de Instrumentación y Control

- 2 Ingenieros de Instrumentación y Control

Dado que el total de la población es bastante pequeño, contando con 4 personas este es plenamente accesible, por lo tanto no requiere el uso de una fórmula para el cálculo de una muestra, en cambio, se prefirió utilizar un muestreo censal, el cual, consiste en utilizar la totalidad de los individuos dentro de la población.

2.2.3 Recolección de información

La recolección de información constituye una etapa fundamental dentro del proceso investigativo, ya que permite obtener datos relevantes tanto cuantitativos como cualitativos relacionados con la problemática planteada. En esta fase se busca recopilar información confiable y pertinente que sirva como base para el análisis y desarrollo de la solución propuesta. Para ello, se emplearon técnicas como entrevistas semiestructuradas, dirigidas al personal técnico del Campo Pucuna ([Anexo B](#)), las cuales permitieron obtener información detallada, contextualizada y basada en la experiencia directa de los profesionales, facilitando la identificación de problemas operativos y necesidades específicas del entorno.

De manera complementaria, se utilizaron fichas documentales elaboradas a partir de la revisión de la bibliografía del proyecto, lo que permitió sustentar teóricamente la investigación y conocer enfoques, metodologías y resultados obtenidos en estudios previos relacionados con Medidores Virtuales de Flujo (VFM) y el uso de inteligencia artificial. La integración de estas técnicas de recolección de información contribuyó a identificar variables operacionales críticas, analizar el estado actual de los sistemas de medición y evaluar la viabilidad técnica de implementar un sistema de Medidor Virtual de Flujo basado en algoritmos de inteligencia artificial.

Cuadro 15: Diseño de recolección de datos

Preguntas básicas	Explicación / Detalle
¿Para qué? (Objetivos)	Recopilar información que permita identificar variables operacionales críticas, analizar la situación actual de medición del flujo de gas y evaluar la aceptación de un sistema de Medidor Virtual de Flujo (VFM) basado en inteligencia artificial.
¿A qué personas u objetos? (Sujetos)	Personal técnico y operativo del área de producción, incluyendo ingenieros, operadores de campo y especialistas en sistemas relacionados con la gestión de datos.
¿Sobre qué aspectos? (Variables)	Variables operacionales como presión, temperatura y caudal, calidad de los datos históricos, problemas en la medición actual, uso de herramientas tecnológicas y nivel de aceptación de sistemas basados en inteligencia artificial.
¿Quién? (Recolectores)	El investigador responsable del proyecto, encargado de la aplicación de entrevistas y entrevistas para la obtención de información.
¿Cuándo? (Tiempo)	Durante la fase de levantamiento de información del proyecto, correspondiente al periodo académico 2026.
¿Dónde? (Lugar)	En las instalaciones de la empresa o mediante herramientas digitales como formularios en línea.
¿Cuántas veces? (Frecuencia)	La recolección de datos se realizará una sola vez, considerando un enfoque transversal del estudio.
¿Qué técnicas de recolección?	Encuesta estructurada para obtener datos cuantitativos y entrevistas semiestructuradas para profundizar en aspectos técnicos y operativos.
¿Con qué?	Instrumentos como formularios digitales (Google Forms) y guías de entrevista diseñadas específicamente para el contexto del sistema de medición virtual de flujo (Anexo B).

Autor del cuadro: Marco Montesdeoca

a. Entrevistas

Se aplicaron entrevistas estructuradas al personal del Departamento de Instrumentación y Control del Campo Pucuna, con base en la guía del [Anexo B](#), con el propósito de recopilar información cualitativa sobre la situación actual de los sistemas de medición de flujo de gas. A través de este instrumento se evaluaron aspectos como la confiabilidad de los datos, la frecuencia de errores en los procesos manuales y la percepción del uso de tecnologías basadas en inteligencia artificial. Los resultados obtenidos permitieron identificar tendencias claras que evidencian la necesidad de implementar un sistema de Medidor Virtual de Flujo (VFM).

Cuadro 16: Resultados de la entrevista – Supervisor

N°	Pregunta	Respuesta	Conclusión
1	¿Qué problemas existen en la medición actual del flujo de gas?	Existen inconsistencias en los datos y falta de medición en algunos pozos, lo que obliga a estimaciones manuales.	La falta de medición directa reduce la confiabilidad de la información.
2	¿Qué variables son más relevantes para estimar el flujo?	Principalmente presión, temperatura y caudal, ya que influyen directamente en el comportamiento del gas.	Las variables operacionales son fundamentales para modelos predictivos.
3	¿Cómo es la calidad de los datos actuales?	La calidad es variable, en algunos casos hay datos incompletos o inconsistentes.	La calidad de datos limita la precisión del análisis.
4	¿Qué opina del uso de inteligencia artificial en este contexto?	Es una buena alternativa para mejorar las estimaciones y reducir errores humanos.	La IA es percibida como una solución viable y necesaria.
5	¿Qué beneficios tendría un sistema VFM?	Mejoraría el control de producción y reduciría costos en equipos físicos.	El VFM optimiza costos y control operativo.

Autor del cuadro: Marco Montesdeoca

Cuadro 17: Resultados de la entrevista – Supervisor de Instrumentación y Control

Nº	Pregunta	Respuesta	Conclusión
1	¿Qué problemas existen en la medición actual del flujo de gas?	Errores en la verificación manual y retrasos en la disponibilidad de datos.	Los procesos manuales afectan la precisión y oportunidad.
2	¿Qué variables son más relevantes para estimar el flujo?	Presión y temperatura principalmente, también son importantes las variables de agua y petróleo, ya que estas son variables que se obtienen cuando se obtiene el gas de los cabezales BES.	Las variables deben ser integradas en el modelo.
3	¿Cómo es la calidad de los datos actuales?	Aceptable, pero con presencia de errores y valores faltantes.	Se requiere mejorar la calidad de datos.
4	¿Qué opina del uso de inteligencia artificial?	Puede mejorar significativamente la precisión del sistema.	La IA es clave para optimización del proceso.
5	¿Qué beneficios tendría un sistema VFM?	Facilitaría el monitoreo continuo y mejoraría la eficiencia.	El VFM mejora la supervisión operativa.

Autor del cuadro: Marco Montesdeoca

Cuadro 18: Resultados de la entrevista – Ingeniero 1 de Instrumentación y Control

Nº	Pregunta	Respuesta	Conclusión
1	¿Qué problemas existen en la medición actual del flujo de gas?	Existen datos faltantes y errores en registros históricos.	La integridad de datos es un problema crítico.
2	¿Qué variables son más relevantes para estimar el flujo?	Presión, temperatura y caudal son esenciales, también son importantes las variables de agua y petróleo, ya que estas son variables que se obtienen cuando se obtiene el gas de los cabezales BES.	Las variables clave permiten modelado adecuado.
3	¿Cómo es la calidad de los datos actuales?	Irregular, con inconsistencias frecuentes.	La calidad de datos afecta la confiabilidad.
4	¿Qué opina del uso de inteligencia artificial?	Es útil para analizar grandes volúmenes de datos.	La IA mejora el análisis y predicción.
5	¿Qué beneficios tendría un sistema VFM?	Permitiría decisiones más rápidas y precisas.	El VFM mejora la toma de decisiones.

Autor del cuadro: Marco Montesdeoca

Cuadro 19: Resultados de la entrevista – Ingeniero 2 de Instrumentación y Control

Nº	Pregunta	Respuesta	Conclusión
1	¿Qué problemas existen en la medición actual del flujo de gas?	Limitaciones en cobertura de medición y dependencia de procesos manuales.	La falta de cobertura reduce precisión del sistema.
2	¿Qué variables son más relevantes para estimar el flujo?	Todas las variables operacionales son importantes.	Se requiere enfoque integral en variables.
3	¿Cómo es la calidad de los datos actuales?	Aceptable pero mejorable.	Se necesita optimización de datos.
4	¿Qué opina del uso de inteligencia artificial?	Es una herramienta innovadora con alto potencial.	La IA aporta innovación al sistema.
5	¿Qué beneficios tendría un sistema VFM?	Reduciría costos y mejoraría eficiencia.	El VFM optimiza recursos operativos.

Autor del cuadro: Marco Montesdeoca

Cuadro 20: Resultados segunda entrevista – Supervisor (Justificación de datos y flujo de gas)

N°	Pregunta	Respuesta	Conclusión
1	¿Por qué es necesario recolectar datos operacionales para el desarrollo del sistema VFM?	Es fundamental porque el sistema se basa en inteligencia artificial. Sin datos no hay modelo que entrenar. Necesitamos variables como presión y temperatura para que el sistema pueda aprender el comportamiento del flujo de gas.	Los datos operacionales son esenciales para el entrenamiento del modelo de inteligencia artificial.
2	¿Qué tipo de datos considera más importantes para el entrenamiento del modelo?	Los más importantes son presión, temperatura y en algunos casos variables del sistema BES. Estas variables tienen relación directa con el comportamiento del gas.	Las variables operacionales son clave para lograr una estimación precisa del flujo.
3	¿Por qué es importante medir o estimar el flujo de gas en los flares?	Porque actualmente en muchos casos ese gas no se mide, solo se quema. Pero ese gas tiene valor energético y económico. Si logramos estimarlo bien, se puede aprovechar para generación eléctrica.	La medición del flujo de gas permite aprovechar recursos energéticos y reducir pérdidas.
4	¿Qué problemas existen actualmente al no contar con medición directa del flujo de gas?	Principalmente que no se tiene una estimación confiable. Se trabaja con aproximaciones y eso afecta la planificación y el uso del gas. Además, no se puede optimizar su aprovechamiento.	La falta de medición directa reduce la confiabilidad y limita la toma de decisiones.
5	¿Cómo contribuiría un sistema VFM basado en inteligencia artificial en este contexto?	Ayudaría a estimar el flujo con mayor precisión sin necesidad de equipos físicos. Además, permitiría analizar grandes volúmenes de datos y encontrar patrones que manualmente no se pueden ver.	El VFM mejora la precisión, reduce costos y optimiza el uso de la información.

Autor del cuadro: Marco Montesdeoca



Figura 1: Nube de palabras generada a partir de las entrevistas

La nube de palabras presentada permite visualizar los términos más recurrentes en las entrevistas realizadas al personal técnico. Se destacan conceptos como flujo, gas, datos, presión y temperatura, lo cual evidencia la relevancia de las variables operacionales en el proceso de medición. La presencia de términos como inteligencia artificial, modelo y predicción refuerza la viabilidad del uso de tecnologías avanzadas para la estimación del flujo de gas, validando así el enfoque del proyecto.

b. Fichas bibliográficas

Se elaboraron fichas documentales a partir de la revisión de bibliografía que hablan sobre Virtual Flow Metering y algoritmos de inteligencia artificial. Estas fichas permitieron sistematizar información clave sobre metodologías, técnicas de modelado y aplicaciones en la industria, proporcionando un sustento teórico sólido para el desarrollo del proyecto y facilitando la selección de herramientas adecuadas para la implementación del sistema VFM.

Cuadro 21: Ficha bibliográfica 1

Tema	Medición virtual de flujo mediante inteligencia artificial
Nombre del artículo	Development of Virtual Flow Sensor using Artificial Neural Networks
Ideas centrales	El artículo propone el desarrollo de un sensor virtual de flujo utilizando redes neuronales artificiales, demostrando que es posible estimar variables de flujo con alta precisión a partir de datos operacionales sin necesidad de dispositivos físicos.
Conceptos clave	Medidor Virtual de Flujo (VFM), redes neuronales artificiales, variables operacionales, modelado predictivo, inteligencia artificial.
Conclusiones	El uso de redes neuronales permite obtener estimaciones confiables del flujo, reduciendo la dependencia de sensores físicos y mejorando la eficiencia en sistemas industriales.
Aporte al tema	Este artículo constituye un aporte fundamental para el desarrollo del presente proyecto, ya que demuestra la viabilidad técnica del uso de redes neuronales artificiales en la estimación del flujo en sistemas industriales. Su contribución radica en evidenciar que, a partir de variables operacionales como presión, temperatura y caudal, es posible construir modelos predictivos altamente precisos que sustituyen o complementan los sistemas de medición tradicionales. Además, el estudio proporciona una base metodológica clara sobre el uso de algoritmos de inteligencia artificial en entornos reales, lo cual es directamente aplicable al diseño del sistema de Medidor Virtual de Flujo (VFM) propuesto en esta investigación. Este enfoque permite reducir costos asociados a equipos físicos, minimizar errores humanos y mejorar la disponibilidad de datos en tiempo real. En el contexto del proyecto, este documento respalda la decisión de utilizar inteligencia artificial como núcleo del sistema, validando su aplicación en la industria oil & gas y fortaleciendo el sustento científico del modelo a desarrollar.
Observaciones	El artículo presenta resultados favorables en términos de precisión y confiabilidad; sin embargo, es importante considerar que el desempeño del modelo depende en gran medida de la calidad y disponibilidad de los datos utilizados para el entrenamiento. En escenarios donde los datos sean incompletos o inconsistentes, como se evidenció en el análisis de campo, el rendimiento del modelo podría verse afectado. El estudio se enfoca en condiciones específicas de operación, por lo que su implementación en otros contextos requiere una adecuada adaptación de los parámetros del modelo. A pesar de estas limitaciones, el enfoque propuesto sigue siendo altamente relevante, ya que proporciona una guía clara para el desarrollo de sistemas VFM basados en inteligencia artificial. Finalmente, se destaca que el artículo no profundiza en aspectos relacionados con la integración del sistema en plataformas operativas reales, lo cual representa una oportunidad de mejora que será abordada en el presente proyecto.

Cuadro 22: Ficha bibliográfica 2

Tema	Medidores virtuales de flujo y aprendizaje automático
Nombre del artículo	First Principles and Machine Learning Virtual Flow Metering: A Literature Review
Ideas centrales	El artículo presenta una revisión de diferentes enfoques de medición virtual de flujo, comparando modelos basados en principios físicos con modelos basados en aprendizaje automático.
Conceptos clave	Virtual Flow Metering, machine learning, modelos híbridos, simulación, modelado físico.
Conclusiones	Los modelos basados en aprendizaje automático presentan ventajas en términos de adaptabilidad y precisión frente a los métodos tradicionales.
Aporte al tema	Este documento representa un aporte teórico clave para el desarrollo del presente proyecto, ya que proporciona una visión integral sobre los diferentes enfoques utilizados en la medición virtual de flujo. Su principal contribución radica en la comparación entre modelos basados en principios físicos y aquellos sustentados en técnicas de aprendizaje automático, permitiendo identificar las ventajas y limitaciones de cada enfoque. Este artículo justifica la elección de algoritmos de inteligencia artificial como base del sistema VFM, debido a su capacidad para adaptarse a grandes volúmenes de datos y capturar relaciones complejas entre variables operacionales. Además, proporciona criterios técnicos que orientan la selección de modelos adecuados para la predicción del flujo de gas. El estudio destaca la importancia de los modelos híbridos, lo cual abre la posibilidad de futuras mejoras en el sistema propuesto, integrando conocimiento físico con técnicas de aprendizaje automático. De esta manera, el documento fortalece el sustento científico del proyecto y amplía las posibilidades de implementación.
Observaciones	Si bien el artículo ofrece una revisión amplia y detallada, su enfoque es principalmente teórico, por lo que no profundiza en casos específicos de implementación práctica. Esto implica que, para el desarrollo del sistema propuesto, será necesario complementar esta información con estudios aplicados. Además, el documento destaca la necesidad de contar con datos de alta calidad para obtener buenos resultados en modelos de aprendizaje automático, lo cual coincide con las limitaciones identificadas en el análisis de campo. Por lo tanto, se deberá prestar especial atención a la fase de preparación y limpieza de datos. Finalmente, se observa que los modelos híbridos presentan un alto potencial, aunque también implican mayor complejidad en su implementación, lo cual puede representar un reto en términos de desarrollo y recursos computacionales.

Autor del cuadro: Marco Montesdeoca

Cuadro 23: Ficha bibliográfica 3

Tema	Predicción de flujo mediante inteligencia artificial
Nombre del artículo	Prediction of oil flow rate through an orifice flow meter: Artificial intelligence alternatives compared
Ideas centrales	El estudio compara diferentes técnicas de inteligencia artificial para la predicción de la tasa de flujo de petróleo.
Conceptos clave	Predicción de flujo, inteligencia artificial, modelos comparativos, aprendizaje supervisado.
Conclusiones	Los modelos de inteligencia artificial presentan mayor precisión que los métodos tradicionales en la estimación del flujo.
Aporte al tema	Este artículo aporta evidencia empírica relevante que respalda el uso de algoritmos de inteligencia artificial en la predicción de variables de flujo. Su principal contribución consiste en la comparación de diferentes modelos, lo que permite identificar cuáles presentan un mejor desempeño en términos de precisión y confiabilidad. En relación con el presente proyecto, este documento permite validar la efectividad de los modelos de inteligencia artificial para la estimación del flujo de gas, lo cual es fundamental para el diseño del sistema VFM. Además, proporciona una referencia importante para la selección de algoritmos, considerando su rendimiento en escenarios reales. El estudio destaca la capacidad de los modelos para adaptarse a diferentes condiciones operativas, lo que resulta clave en entornos dinámicos como los sistemas de producción de hidrocarburos. De esta manera, el artículo contribuye significativamente al sustento técnico del proyecto.
Observaciones	El artículo presenta resultados comparativos sólidos; sin embargo, se enfoca principalmente en flujo de petróleo, por lo que su aplicación directa al flujo de gas requiere ciertas adaptaciones. Además, el desempeño de los modelos depende de la calidad del conjunto de datos, lo cual coincide con las limitaciones identificadas en el entorno de estudio. Esto implica que se debe realizar un adecuado preprocesamiento de datos. Finalmente, se observa que algunos modelos pueden requerir mayor capacidad computacional, lo cual debe considerarse en la implementación del sistema.

Autor del cuadro: Marco Montesdeoca

Cuadro 24: Ficha bibliográfica 4

Tema	Aplicación de machine learning en medición virtual
Nombre del artículo	Virtual Metering and Allocation using Machine Learning Algorithms
Ideas centrales	Describe la implementación de sistemas de medición virtual mediante algoritmos de machine learning en entornos industriales.
Conceptos clave	Machine learning, medición virtual, asignación de producción, sistemas industriales.
Conclusiones	Los sistemas de medición virtual mejoran la eficiencia operativa y permiten un mejor control de la producción.
Aporte al tema	Este documento aporta evidencia práctica sobre la implementación de sistemas de medición virtual en entornos industriales reales, lo cual es fundamental para validar la viabilidad del proyecto propuesto. Su contribución radica en demostrar que los algoritmos de machine learning pueden integrarse exitosamente en sistemas de producción, mejorando la eficiencia operativa y la precisión en la estimación de variables. En el contexto del presente estudio, este artículo respalda la aplicabilidad del sistema VFM en el Campo Pucuna. Además, el documento proporciona una referencia sobre la integración de estos sistemas en procesos industriales, lo cual es clave para el desarrollo de una solución funcional y adaptable. De esta manera, fortalece el enfoque aplicado de la investigación.
Observaciones	El estudio presenta resultados positivos, pero se basa en condiciones específicas que pueden diferir del entorno del Campo Pucuna. No se detalla completamente la arquitectura del sistema, lo cual limita su replicabilidad directa. Sin embargo, el artículo sigue siendo relevante como referencia práctica para la implementación del sistema VFM.

Autor del cuadro: Marco Montesdeoca

Cuadro 25: Ficha bibliográfica 5

Tema	Predicción de producción en condiciones complejas
Nombre del artículo	Applications of Artificial Intelligence to Predict Oil Rate for High Gas–Oil Ratio and Water-Cut Wells
Ideas centrales	Analiza el uso de inteligencia artificial para predecir la tasa de producción en condiciones complejas.
Conceptos clave	Inteligencia artificial, predicción, condiciones complejas, producción de petróleo.
Conclusiones	La inteligencia artificial es capaz de modelar escenarios complejos con alta precisión.
Aporte al tema	Este artículo representa un aporte significativo al demostrar la capacidad de la inteligencia artificial para trabajar en escenarios complejos, caracterizados por múltiples variables y condiciones cambiantes. En el contexto del presente proyecto, este documento es fundamental, ya que los sistemas de producción de gas presentan condiciones similares, donde factores como presión, temperatura y composición influyen en el comportamiento del flujo. La capacidad de la IA para manejar estas variables permite desarrollar modelos más robustos y precisos. Además, el estudio valida el uso de técnicas avanzadas de modelado en entornos reales, lo cual refuerza la viabilidad del sistema VFM. También aporta una base para el manejo de datos complejos, lo cual es clave en el desarrollo del modelo. En conjunto, este artículo fortalece el sustento científico y técnico del proyecto, demostrando que la inteligencia artificial es una herramienta adecuada para la estimación de variables en sistemas de producción.
Observaciones	El artículo se enfoca en producción de petróleo, por lo que su aplicación a sistemas de gas requiere adaptación. Además, los modelos utilizados pueden ser complejos y requerir alta capacidad computacional. Sin embargo, el enfoque es altamente relevante y aplicable al desarrollo del sistema VFM.

Autor del cuadro: Marco Montesdeoca

2.3 Procesamiento y análisis de datos

De acuerdo con la información recolectada a través de las entrevistas realizadas al personal técnico del Departamento de Instrumentación y Control del Campo Pucuna, así como del análisis de las fichas documentales correspondientes a estudios científicos relacionados con Medidores Virtuales de Flujo (VFM), y luego de haber efectuado el análisis respectivo, se determina que:

- Se evidencia la existencia de limitaciones en los sistemas actuales de medición de flujo de gas, caracterizadas por la falta de medición directa en ciertos puntos y la dependencia de procesos manuales, lo que genera estimaciones poco confiables y reduce la eficiencia operativa, situación identificada a partir de las entrevistas realizadas al personal técnico.
- Se determinó la necesidad de implementar soluciones tecnológicas que permitan automatizar los procesos de medición y análisis, con el fin de mejorar la precisión, confiabilidad y disponibilidad de los datos operacionales, en concordancia con las problemáticas detectadas en el entorno de estudio.
- A partir del análisis de las fichas documentales, se establece que los algoritmos de inteligencia artificial, especialmente las redes neuronales, permiten estimar variables de flujo con un alto nivel de precisión, incluso en ausencia de medición directa, tal como se evidencia en la ficha documental 1.
- Los modelos de aprendizaje automático presentan ventajas significativas frente a los métodos tradicionales, destacándose por su capacidad de adaptación a grandes volúmenes de datos y la identificación de patrones complejos, lo cual es respaldado por la ficha documental 2 y la ficha documental 3.
- De igual manera, se ha comprobado que los sistemas de medición virtual de flujo han sido implementados con éxito en entornos industriales reales, demostrando su viabilidad técnica y su impacto positivo en la optimización de procesos productivos, como se expone en la ficha documental 4.

- La inteligencia artificial ha demostrado ser capaz de modelar escenarios complejos con múltiples variables operacionales, lo cual resulta fundamental en sistemas de producción de hidrocarburos, tal como se describe en la ficha documental 5.
- La implementación de un sistema de Medidor Virtual de Flujo (VFM) basado en inteligencia artificial es una solución viable y necesaria, ya que contribuye a mejorar la eficiencia operativa, optimizar la toma de decisiones y reducir costos asociados a equipos físicos, integrando tanto la evidencia empírica como el sustento teórico analizado.

3. RESULTADOS Y DISCUSIÓN

3.1 Identificación de variables operacionales críticas y construcción del dataset de entrenamiento

Se realizó un análisis de las variables operacionales necesarias para el desarrollo del Medidor Virtual de Flujo (VFM), considerando el contexto del gas asociado en flares y la necesidad de estimar su caudal mediante técnicas de inteligencia artificial. La identificación de dichas variables se sustentó en dos fuentes complementarias: las entrevistas aplicadas al personal del Departamento de Instrumentación y Control del Campo Pucuna ([Anexo B](#); Cuadros 16, 17, 18, 19 y ??) y el análisis de bases de datos históricas del activo. En las entrevistas, la pregunta sobre variables relevantes para estimar el flujo fue respondida de forma consistente con presión, temperatura y caudal; la Figura 1 refuerza esa priorización al destacar términos como *flujo*, *gas*, *presión* y *temperatura*.

Basandonos en eso, las variables de entrada consideradas para la red neuronal correspondieron principalmente a la presión y temperatura de los cabezales y de los sistemas BES, junto con volúmenes de producción líquida cuando aportaron contexto al comportamiento del gas. El flujo de gas asociado se definió como variable objetivo, por ser el valor que el modelo de inteligencia artificial debía aprender a estimar.

3.1.1 Variables operacionales críticas identificadas

Las variables críticas seleccionadas se documentan en el Cuadro 26. Su definición partió de las entrevistas realizadas —en las que el personal técnico priorizó presión, temperatura y caudal como insumos para modelos predictivos— y se contrastó con la relación operativa entre el sistema BES, los volúmenes de producción (petróleo y agua), las mediciones en cabezal y el flujo de gas asociado del activo P del campo Pucuna. La serie abarca enero de 2020 a diciembre de 2025.

Cuadro 26: Variables operacionales críticas identificadas (campo Pucuna, activo P).

Variable	Descripción	Uso en el modelo
Petróleo	Caudal diario de petróleo del activo P en separador de boca de pozo (bbl/d). Campo: <code>produccion_volumenes.petroleo_bbl_d</code> .	Entrada
Agua	Caudal diario de agua producida del activo P (bbl/d). Campo: <code>produccion_volumenes.agua_bbl_d</code> .	Entrada
Presión de cabezal	Presión en cabezal del activo P con levantamiento BES (psi). Campo: <code>bes_cabezal_sensors.presion_cabezal_psi</code> .	Entrada
Temperatura BES/cabezal	Temperatura del sistema BES en cabezal (°C). Campo: <code>bes_cabezal_sensors.temperatura_bes_c</code> .	Entrada
Flujo de gas asociado	Caudal de gas asociado al activo P (MSCFD). Campo: <code>mediciones_gas.gas_mscfd</code> .	Variable objetivo
Fecha	Registro temporal diario de la medición.	Organización temporal
Activo	Identificador del pozo: P (campo Pucuna).	Segmentación de datos

Autor del cuadro: Marco Montesdeoca

La presión y la temperatura fueron consideradas variables independientes, mientras que el flujo de gas asociado fue considerado como variable dependiente. Esta estructura respondió al enfoque de aprendizaje supervisado, donde el modelo recibió variables de entrada y aprendió a estimar una salida esperada.

3.1.2 Bases de datos históricas

Estructuración de la base de datos

Estos datos se estructuraron en una base que contiene tablas fuente, tablas de transformación y tablas de análisis. Su propósito fue permitir la trazabilidad desde los datos originales hasta el dataset final utilizado por el modelo de inteligencia artificial.

Los datos proporcionados por Petroecuador se conforman por las siguientes bases de datos:

- `produccion_volumenes`
- `bes_cabezal_sensors`
- `mediciones_gas`

3.1.3 Tablas fuente utilizadas

La tabla `produccion_volumenes` representó el reporte diario de producción en el separador de boca de pozo. Para cada fecha y activo se almacenaron los caudales de petróleo y agua, el crudo total y el corte de agua, junto con las columnas de unidad correspondientes. De esta tabla se obtuvieron las variables *Petróleo* y *Agua*, que constituyeron dos de las cuatro entradas del modelo predictivo.

#	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
1	fecha	activo	petroleo_unidad	petroleo_bbl	agua_unidad	agua_bbl	agua_unidad	agua_bbl	agua_unidad	agua_bbl	agua_unidad	agua_bbl	agua_unidad	agua_bbl	agua_unidad	agua_bbl	agua_unidad	agua_bbl	agua_unidad
2	2020-01-01 00:00:00	P	7676,456 bbl/d	7676,456	57048,2 bbl/d	57048,2	57048,2	57048,2	57048,2	57048,2	57048,2	57048,2	57048,2	57048,2	57048,2	57048,2	57048,2	57048,2	57048,2
3	2020-01-02 00:00:00	P	7473,175 bbl/d	7473,175	58233,02 bbl/d	58233,02	58233,02	58233,02	58233,02	58233,02	58233,02	58233,02	58233,02	58233,02	58233,02	58233,02	58233,02	58233,02	58233,02
4	2020-01-03 00:00:00	P	8025,916 bbl/d	8025,916	58403,02 bbl/d	58403,02	58403,02	58403,02	58403,02	58403,02	58403,02	58403,02	58403,02	58403,02	58403,02	58403,02	58403,02	58403,02	58403,02
5	2020-01-04 00:00:00	P	7971,069 bbl/d	7971,069	56530,17 bbl/d	56530,17	56530,17	56530,17	56530,17	56530,17	56530,17	56530,17	56530,17	56530,17	56530,17	56530,17	56530,17	56530,17	56530,17
6	2020-01-05 00:00:00	P	7234,237 bbl/d	7234,237	57842,29 bbl/d	57842,29	57842,29	57842,29	57842,29	57842,29	57842,29	57842,29	57842,29	57842,29	57842,29	57842,29	57842,29	57842,29	57842,29
7	2020-01-06 00:00:00	P	7441,254 bbl/d	7441,254	58058,38 bbl/d	58058,38	58058,38	58058,38	58058,38	58058,38	58058,38	58058,38	58058,38	58058,38	58058,38	58058,38	58058,38	58058,38	58058,38
8	2020-01-07 00:00:00	P	7852,992 bbl/d	7852,992	57462,15 bbl/d	57462,15	57462,15	57462,15	57462,15	57462,15	57462,15	57462,15	57462,15	57462,15	57462,15	57462,15	57462,15	57462,15	57462,15
9	2020-01-08 00:00:00	P	7741,821 bbl/d	7741,821	56805,74 bbl/d	56805,74	56805,74	56805,74	56805,74	56805,74	56805,74	56805,74	56805,74	56805,74	56805,74	56805,74	56805,74	56805,74	56805,74
10	2020-01-09 00:00:00	P	7835,633 bbl/d	7835,633	57115,14 bbl/d	57115,14	57115,14	57115,14	57115,14	57115,14	57115,14	57115,14	57115,14	57115,14	57115,14	57115,14	57115,14	57115,14	57115,14
11	2020-01-10 00:00:00	P	7582,699 bbl/d	7582,699	57441,64 bbl/d	57441,64	57441,64	57441,64	57441,64	57441,64	57441,64	57441,64	57441,64	57441,64	57441,64	57441,64	57441,64	57441,64	57441,64
12	2020-01-11 00:00:00	P	8021,16 bbl/d	8021,16	57609,73 bbl/d	57609,73	57609,73	57609,73	57609,73	57609,73	57609,73	57609,73	57609,73	57609,73	57609,73	57609,73	57609,73	57609,73	57609,73
13	2020-01-12 00:00:00	P	8005,888 bbl/d	8005,888	58133,26 bbl/d	58133,26	58133,26	58133,26	58133,26	58133,26	58133,26	58133,26	58133,26	58133,26	58133,26	58133,26	58133,26	58133,26	58133,26
14	2020-01-13 00:00:00	P	7391,712 bbl/d	7391,712	56006,3 bbl/d	56006,3	56006,3	56006,3	56006,3	56006,3	56006,3	56006,3	56006,3	56006,3	56006,3	56006,3	56006,3	56006,3	56006,3
15	2020-01-14 00:00:00	P	7880,041 bbl/d	7880,041	57777,29 bbl/d	57777,29	57777,29	57777,29	57777,29	57777,29	57777,29	57777,29	57777,29	57777,29	57777,29	57777,29	57777,29	57777,29	57777,29
16	2020-01-15 00:00:00	P	7974,926 bbl/d	7974,926	57584,27 bbl/d	57584,27	57584,27	57584,27	57584,27	57584,27	57584,27	57584,27	57584,27	57584,27	57584,27	57584,27	57584,27	57584,27	57584,27
17	2020-01-16 00:00:00	P	7633,498 bbl/d	7633,498	58526,15 bbl/d	58526,15	58526,15	58526,15	58526,15	58526,15	58526,15	58526,15	58526,15	58526,15	58526,15	58526,15	58526,15	58526,15	58526,15
18	2020-01-17 00:00:00	P	8090,371 bbl/d	8090,371	58450,21 bbl/d	58450,21	58450,21	58450,21	58450,21	58450,21	58450,21	58450,21	58450,21	58450,21	58450,21	58450,21	58450,21	58450,21	58450,21
19	2020-01-18 00:00:00	P	7734,023 bbl/d	7734,023	57379,08 bbl/d	57379,08	57379,08	57379,08	57379,08	57379,08	57379,08	57379,08	57379,08	57379,08	57379,08	57379,08	57379,08	57379,08	57379,08
20	2020-01-19 00:00:00	P	8303,863 bbl/d	8303,863	57919,14 bbl/d	57919,14	57919,14	57919,14	57919,14	57919,14	57919,14	57919,14	57919,14	57919,14	57919,14	57919,14	57919,14	57919,14	57919,14
21	2020-01-20 00:00:00	P	8071,894 bbl/d	8071,894	58106,76 bbl/d	58106,76	58106,76	58106,76	58106,76	58106,76	58106,76	58106,76	58106,76	58106,76	58106,76	58106,76	58106,76	58106,76	58106,76
22	2020-01-21 00:00:00	P	8051,998 bbl/d	8051,998	58453,74 bbl/d	58453,74	58453,74	58453,74	58453,74	58453,74	58453,74	58453,74	58453,74	58453,74	58453,74	58453,74	58453,74	58453,74	58453,74
23	2020-01-22 00:00:00	P	7922,024 bbl/d	7922,024	58946,09 bbl/d	58946,09	58946,09	58946,09	58946,09	58946,09	58946,09	58946,09	58946,09	58946,09	58946,09	58946,09	58946,09	58946,09	58946,09
24	2020-01-23 00:00:00	P	8474,856 bbl/d	8474,856	56223,01 bbl/d	56223,01	56223,01	56223,01	56223,01	56223,01	56223,01	56223,01	56223,01	56223,01	56223,01	56223,01	56223,01	56223,01	56223,01
25	2020-01-24 00:00:00	P	8095,195 bbl/d	8095,195	56873,35 bbl/d	56873,35	56873,35	56873,35	56873,35	56873,35	56873,35	56873,35	56873,35	56873,35	56873,35	56873,35	56873,35	56873,35	56873,35
26	2020-01-25 00:00:00	P	8028,878 bbl/d	8028,878	56938,58 bbl/d	56938,58	56938,58	56938,58	56938,58	56938,58	56938,58	56938,58	56938,58	56938,58	56938,58	56938,58	56938,58	56938,58	56938,58
27	2020-01-26 00:00:00	P	8071,606 bbl/d	8071,606	56801,68 bbl/d	56801,68	56801,68	56801,68	56801,68	56801,68	56801,68	56801,68	56801,68	56801,68	56801,68	56801,68	56801,68	56801,68	56801,68
28	2020-01-27 00:00:00	P	8373,073 bbl/d	8373,073	56230,89 bbl/d	56230,89	56230,89	56230,89	56230,89	56230,89	56230,89	56230,89	56230,89	56230,89	56230,89	56230,89	56230,89	56230,89	56230,89
29	2020-01-28 00:00:00	P	8323,042 bbl/d	8323,042	55365,91 bbl/d	55365,91	55365,91	55365,91	55365,91	55365,91	55365,91	55365,91	55365,91	55365,91	55365,91	55365,91	55365,91	55365,91	55365,91
30	2020-01-29 00:00:00	P	8358,052 bbl/d	8358,052	57002,29 bbl/d	57002,29	57002,29	57002,29	57002,29	57002,29	57002,29	57002,29	57002,29	57002,29	57002,29	57002,29	57002,29	57002,29	57002,29
31	2020-01-30 00:00:00	P	8356,76 bbl/d	8356,76	56136,8 bbl/d	56136,8	56136,8	56136,8	56136,8	56136,8	56136,8	56136,8	56136,8	56136,8	56136,8	56136,8	56136,8	56136,8	56136,8
32	2020-01-31 00:00:00	P	8880,245 bbl/d	8880,245	57018,94 bbl/d	57018,94	57018,94	57018,94	57018,94	57018,94	57018,94	57018,94	57018,94	57018,94	57018,94	57018,94	57018,94	57018,94	57018,94
33	2020-02-01 00:00:00	P	8140,227 bbl/d	8140,227	56126,47 bbl/d	56126,47	56126,47	56126,47	56126,47	56126,47	56126,47	56126,47	56126,47	56126,47	56126,47	56126,47	56126,47	56126,47	56126,47
34	2020-02-02 00:00:00	P	8086,525 bbl/d	8086,525	54946,23 bbl/d	54946,23	54946,23	54946,23	54946,23	54946,23	54946,23	54946,23	54946,23	54946,23	54946,23	54946,23	54946,23	54946,23	54946,23
35	2020-02-03 00:00:00	P	8030,359 bbl/d	8030,359	56305,94 bbl/d	56305,94	56305,94	56305,94	56305,94	56305,94	56305,94	56305,94	56305,94	56305,94	56305,94	56305,94	56305,94	56305,94	56305,94
36	2020-02-04 00:00:00	P	8480,42 bbl/d	8480,42	56235,06 bbl/d	56235,06	56235,06	56235,06	56235,06	56235,06	56235,06	56235,06	56235,06	56235,06	56235,06	56235,06	56235,06	56235,06	56235,06

Figura 2: Tabla `sensor_readings` con Petroleo y Agua por activo

Cuadro 27: Descripción de columnas de la tabla `produccion_volumenes`.

Columna	Descripción
<code>fecha</code>	Marca de tiempo de la lectura (registro diario).
<code>activo</code>	Activo al que corresponde la medición (P).
<code>petroleo_bbl_d</code>	Caudal diario de petróleo (bbl/d). Entrada del modelo.
<code>unidad_petroleo_bbl_d</code>	Unidad registrada para petróleo (bbl/d).
<code>petroleo_bbl_d_con_unidad</code>	Valor de petróleo con unidad en texto (metadato).
<code>agua_bbl_d</code>	Caudal diario de agua (bbl/d). Entrada del modelo.
<code>unidad_agua_bbl_d</code>	Unidad registrada para agua (bbl/d).
<code>agua_bbl_d_con_unidad</code>	Valor de agua con unidad en texto (metadato).
<code>crudo_bbl_d</code>	Caudal de crudo total en separador (bbl/d).
<code>unidad_crudo_bbl_d</code>	Unidad registrada para crudo (bbl/d).
<code>crudo_bbl_d_con_unidad</code>	Valor de crudo con unidad en texto (metadato).
<code>corte_agua_pct</code>	Porcentaje de corte de agua en el crudo (%).
<code>unidad_corte_agua_pct</code>	Unidad registrada para corte de agua (%).
<code>corte_agua_pct_con_unidad</code>	Valor de corte de agua con unidad (metadato).
<code>fuelle</code>	Origen del reporte; valor: <code>reporte_produccion_diaria</code> .
<code>campo</code>	Nombre del campo; valor: Pucuna.
<code>punto_medicion</code>	Punto de medición; valor: <code>separador_boca_pozo</code> .

Autor del cuadro: Marco Montesdeoca

La tabla `bes_cabecal_sensors` representó la principal fuente de variables de proceso asociadas al sistema BES. En esta tabla se registraron, para cada fecha y activo, las mediciones de presión de cabecal y temperatura BES/cabecal, así como variables de telemetría operacional (corriente del motor y frecuencia del variador). Las variables de presión y temperatura fueron utilizadas como entradas del modelo; corriente y frecuencia se conservaron como información de contexto y no ingresaron al entrenamiento.

fecha	activo	cabezal	presion_c_unidad_p	presion_c_temperatu	unidad_t	temperatura_corriente_unidad_cc	corriente_frecuencia	unidad_fr	frecuencia_fuente	sistema_medicion
2020-01-01 00:00:00 P	CH-P	1022,55 psi	1022,550	68 °C	68.000 °C	85 A	85.000 A	48,486 Hz	48,486 Hz cabezal_b BES	
2020-01-02 00:00:00 P	CH-P	1018,938 psi	1018,938	68 °C	68.000 °C	85 A	85.000 A	48 Hz	48,000 Hz cabezal_b BES	
2020-01-03 00:00:00 P	CH-P	1021,503 psi	1021,503	68 °C	68.000 °C	85 A	85.000 A	48,73 Hz	48,730 Hz cabezal_b BES	
2020-01-04 00:00:00 P	CH-P	1019,002 psi	1019,002	68 °C	68.000 °C	85 A	85.000 A	48,985 Hz	48,985 Hz cabezal_b BES	
2020-01-05 00:00:00 P	CH-P	1016,25 psi	1016,250	68 °C	68.000 °C	85 A	85.000 A	48 Hz	48,000 Hz cabezal_b BES	
2020-01-06 00:00:00 P	CH-P	1017,709 psi	1017,709	68 °C	68.000 °C	85 A	85.000 A	48,565 Hz	48,565 Hz cabezal_b BES	
2020-01-07 00:00:00 P	CH-P	1018,174 psi	1018,174	68 °C	68.000 °C	85 A	85.000 A	48,732 Hz	48,732 Hz cabezal_b BES	
2020-01-08 00:00:00 P	CH-P	1014,881 psi	1014,881	68 °C	68.000 °C	85 A	85.000 A	48,97 Hz	48,970 Hz cabezal_b BES	
2020-01-09 00:00:00 P	CH-P	1016,524 psi	1016,524	68 °C	68.000 °C	85 A	85.000 A	48,315 Hz	48,315 Hz cabezal_b BES	
2020-01-10 00:00:00 P	CH-P	1017,057 psi	1017,057	68 °C	68.000 °C	85 A	85.000 A	48 Hz	48,000 Hz cabezal_b BES	
2020-01-11 00:00:00 P	CH-P	1022,704 psi	1022,704	68,316 °C	68,316 °C	85 A	85.000 A	48,616 Hz	48,616 Hz cabezal_b BES	
2020-01-12 00:00:00 P	CH-P	1024,26 psi	1024,260	68,713 °C	68,713 °C	85 A	85.000 A	49,108 Hz	49,108 Hz cabezal_b BES	
2020-01-13 00:00:00 P	CH-P	1029,188 psi	1029,188	68 °C	68.000 °C	85 A	85.000 A	48,113 Hz	48,113 Hz cabezal_b BES	
2020-01-14 00:00:00 P	CH-P	1031,547 psi	1031,547	68 °C	68.000 °C	85 A	85.000 A	48,7 Hz	48,700 Hz cabezal_b BES	
2020-01-15 00:00:00 P	CH-P	1020,804 psi	1020,804	68 °C	68.000 °C	85 A	85.000 A	48,671 Hz	48,671 Hz cabezal_b BES	
2020-01-16 00:00:00 P	CH-P	1021,022 psi	1021,022	68 °C	68.000 °C	85 A	85.000 A	48,069 Hz	48,069 Hz cabezal_b BES	
2020-01-17 00:00:00 P	CH-P	1019,641 psi	1019,641	69,275 °C	69,275 °C	85 A	85.000 A	49,216 Hz	49,216 Hz cabezal_b BES	
2020-01-18 00:00:00 P	CH-P	1013,142 psi	1013,142	68 °C	68.000 °C	85 A	85.000 A	48,53 Hz	48,530 Hz cabezal_b BES	
2020-01-19 00:00:00 P	CH-P	1016,972 psi	1016,972	70,208 °C	70,208 °C	85 A	85.000 A	49,344 Hz	49,344 Hz cabezal_b BES	
2020-01-20 00:00:00 P	CH-P	1015,299 psi	1015,299	69,631 °C	69,631 °C	85 A	85.000 A	49,774 Hz	49,774 Hz cabezal_b BES	
2020-01-21 00:00:00 P	CH-P	1015,996 psi	1015,996	69,839 °C	69,839 °C	85 A	85.000 A	49,179 Hz	49,179 Hz cabezal_b BES	
2020-01-22 00:00:00 P	CH-P	1016,833 psi	1016,833	69,814 °C	69,814 °C	85 A	85.000 A	49,455 Hz	49,455 Hz cabezal_b BES	
2020-01-23 00:00:00 P	CH-P	1010,28 psi	1010,280	69,691 °C	69,691 °C	85 A	85.000 A	49,681 Hz	49,681 Hz cabezal_b BES	
2020-01-24 00:00:00 P	CH-P	1010,138 psi	1010,138	69,134 °C	69,134 °C	85 A	85.000 A	49,366 Hz	49,366 Hz cabezal_b BES	
2020-01-25 00:00:00 P	CH-P	1009,81 psi	1009,810	68,944 °C	68,944 °C	85 A	85.000 A	49,156 Hz	49,156 Hz cabezal_b BES	
2020-01-26 00:00:00 P	CH-P	1009,046 psi	1009,046	69,03 °C	69,030 °C	85 A	85.000 A	48,644 Hz	48,644 Hz cabezal_b BES	
2020-01-27 00:00:00 P	CH-P	1007,173 psi	1007,173	69,634 °C	69,634 °C	85 A	85.000 A	50,198 Hz	50,198 Hz cabezal_b BES	
2020-01-28 00:00:00 P	CH-P	1004,158 psi	1004,158	68,866 °C	68,866 °C	85 A	85.000 A	49,566 Hz	49,566 Hz cabezal_b BES	
2020-01-29 00:00:00 P	CH-P	1009,658 psi	1009,658	70,178 °C	70,178 °C	85 A	85.000 A	49,919 Hz	49,919 Hz cabezal_b BES	
2020-01-30 00:00:00 P	CH-P	1007,08 psi	1007,080	69,546 °C	69,546 °C	85 A	85.000 A	49,381 Hz	49,381 Hz cabezal_b BES	
2020-01-31 00:00:00 P	CH-P	1012,229 psi	1012,229	71,916 °C	71,916 °C	85 A	85.000 A	50,383 Hz	50,383 Hz cabezal_b BES	
2020-02-01 00:00:00 P	CH-P	1005,442 psi	1005,442	68,836 °C	68,836 °C	85 A	85.000 A	48,978 Hz	48,978 Hz cabezal_b BES	
2020-02-02 00:00:00 P	CH-P	1002,079 psi	1002,079	68,441 °C	68,441 °C	85 A	85.000 A	49,615 Hz	49,615 Hz cabezal_b BES	
2020-02-03 00:00:00 P	CH-P	1005,717 psi	1005,717	69,255 °C	69,255 °C	85 A	85.000 A	49,186 Hz	49,186 Hz cabezal_b BES	
2020-02-04 00:00:00 P	CH-P	1006,756 psi	1006,756	70,672 °C	70,672 °C	85 A	85.000 A	49,883 Hz	49,883 Hz cabezal_b BES	

Figura 3: Tabla bes_cabezal_sensors con Presion y Temperatura por activo

Cuadro 28: Descripción de columnas de la tabla bes_cabezal_sensors.

Columna	Descripción
fecha	Marca de tiempo de la lectura.
activo	Activo al que corresponde la medición (P).
cabezal	Identificador del cabezal; valor: CH-P.
presion_cabezal_psi	Presión de cabezal (psi). Entrada del modelo.
unidad_presion_cabezal_psi	Unidad registrada para presión (psi).
presion_cabezal_psi_con_unidad	Valor de presión con unidad (metadato).
temperatura_bes_c	Temperatura BES/cabezal (°C). Entrada del modelo.
unidad_temperatura_bes_c	Unidad registrada para temperatura (°C).
temperatura_bes_c_con_unidad	Valor de temperatura con unidad (metadato).
corriente_motor_a	Corriente del motor BES en cabezal/VSD (A).
unidad_corriente_motor_a	Unidad registrada para corriente (A).
corriente_motor_a_con_unidad	Valor de corriente con unidad (metadato).
frecuencia_bes_hz	Frecuencia de operación del variador BES (Hz).
unidad_frecuencia_bes_hz	Unidad registrada para frecuencia (Hz).
frecuencia_bes_hz_con_unidad	Valor de frecuencia con unidad (metadato).
fuente	Origen de la medición; valor: cabezal_bes.
sistema_medicion	Sistema de medición; valor: BES.

Autor del cuadro: Marco Montesdeoca

3.1.4 Construcción del dataset de entrenamiento

Construcción del dataset de entrenamiento.

El dataset de entrenamiento se construyó a partir de las tablas operacionales del activo P (campo Pucuna), integrando mediciones diarias de producción líquida y variables de cabezal BES en un formato ancho apto para el modelo. Las entradas del VFM corresponden a cuatro variables operacionales: caudal de petróleo (*bbl/d*), caudal de

agua (*bbl/d*), presión de cabezal (*psi*) y temperatura BES ($^{\circ}\text{C}$); la variable objetivo es el gas asociado (*MSCFD*).

La integración se realizó mediante unión interna por fecha y activo entre las fuentes de producción y sensores, excluyendo columnas de metadatos de unidad.

FECHA	PETROLEO	AGUA	PRESION	TEMPERA	GAS
2020-01-01 00:00:00	7676,456	57048,2	1022,55	68	4758,53
2020-01-02 00:00:00	7473,175	58233,02	1018,938	68	4704,751
2020-01-03 00:00:00	8025,916	58403,02	1021,503	68	4831,92
2020-01-04 00:00:00	7971,069	56530,17	1019,002	68	4739,569
2020-01-05 00:00:00	7234,237	57842,29	1016,25	68	4070,768
2020-01-06 00:00:00	7441,254	58058,38	1017,709	68	4737,003
2020-01-07 00:00:00	7852,992	57462,15	1018,174	68	4829,546
2020-01-08 00:00:00	7741,821	56805,74	1014,881	68	4792,187
2020-01-09 00:00:00	7835,633	57115,14	1016,524	68	4674,318
2020-01-10 00:00:00	7582,699	57441,64	1017,057	68	4873,871
2020-01-11 00:00:00	8021,16	57609,73	1022,704	68,316	5105,482
2020-01-12 00:00:00	8005,888	58133,26	1024,26	68,713	4904,629
2020-01-13 00:00:00	7391,712	56006,3	1029,188	68	4434,028
2020-01-14 00:00:00	7880,041	57777,29	1031,547	68	4782,087
2020-01-15 00:00:00	7974,926	57584,27	1020,804	68	4761,375
2020-01-16 00:00:00	7633,498	58526,15	1021,022	68	4690,407
2020-01-17 00:00:00	8090,371	58450,21	1019,641	69,275	4869,66
2020-01-18 00:00:00	7734,023	57379,08	1013,142	68	4854,771
2020-01-19 00:00:00	8303,863	57919,14	1016,972	70,208	5054,276
2020-01-20 00:00:00	8071,894	58106,76	1015,299	69,631	4905,068
2020-01-21 00:00:00	8051,998	58453,74	1015,996	69,839	5232,994
2020-01-22 00:00:00	7922,024	58946,09	1016,833	69,814	4742,122
2020-01-23 00:00:00	8474,856	56223,01	1010,28	69,691	4945,388
2020-01-24 00:00:00	8095,195	56873,35	1010,138	69,134	5048,646
2020-01-25 00:00:00	8028,878	56938,58	1009,81	68,944	4565,744
2020-01-26 00:00:00	8071,606	56801,68	1009,046	69,03	5014,845
2020-01-27 00:00:00	8373,073	56230,89	1007,173	69,634	5338,349
2020-01-28 00:00:00	8323,042	55365,91	1004,158	68,866	5140,899
2020-01-29 00:00:00	8358,052	57002,29	1009,658	70,178	5154,559
2020-01-30 00:00:00	8356,76	56136,8	1007,08	69,546	4781,974
2020-01-31 00:00:00	8880,245	57018,94	1012,229	71,916	5591,165
2020-02-01 00:00:00	8140,227	56126,47	1005,442	68,836	5082,028
2020-02-02 00:00:00	8086,525	54946,23	1002,079	68,441	4764,906
2020-02-03 00:00:00	8030,359	56305,94	1005,717	69,255	5285,346
2020-02-04 00:00:00	8480,42	56235,06	1006,756	70,672	5011,93

Figura 4: Tabla dataset_entrenamiento

Análisis exploratorio de datos (EDA)

Posterior a armar el dataset para el entrenamiento, se realizó un análisis exploratorio de datos (EDA) para analizar las variables de entrada y la variable objetivo. Primero vamos a revisar los tipos de datos del dataset.

```
data.info()
✓ 0.0s

<class 'pandas.DataFrame'>
RangeIndex: 2185 entries, 0 to 2184
Data columns (total 6 columns):
#   Column          Non-Null Count  Dtype
---  -
0   FECHA           2185 non-null   datetime64[us]
1   PETROLEO        2185 non-null   float64
2   AGUA            2185 non-null   float64
3   PRESION         2185 non-null   float64
4   TEMPERATURA     2185 non-null   float64
5   GAS             2185 non-null   float64
dtypes: datetime64[us](1), float64(5)
memory usage: 102.6 KB
```

Figura 5: Tabla Información del tipo de datos del dataset

El dataset limpio del activo P (campo Pucuna) comprende 2 192 registros diarios y 6 variables: la marca temporal FECHA (tipo fecha) y cinco magnitudes numéricas en punto flotante (float64) PETROLEO, AGUA, PRESION, TEMPERATURA y GAS. Las cuatro primeras corresponden a las entradas operacionales del VFM (caudales de petróleo y agua, presión de cabezal y temperatura BES) y la última al gas asociado, definido como variable objetivo. Tras la etapa de preparación y limpieza, no se registran valores nulos en ninguna columna: las 2 192 observaciones están completas en todas las variables, lo que confirma la coherencia del conjunto resultante para el análisis exploratorio y el entrenamiento del modelo. El tamaño en memoria es reducido (aprox. 103 KB), acorde con la naturaleza tabular y la frecuencia diaria del periodo analizado.

Histograma de las variables del dataset.

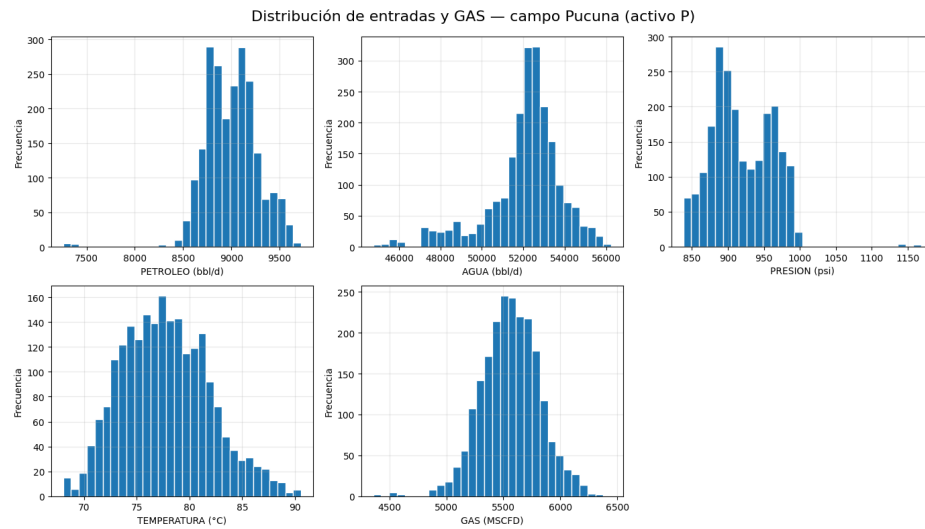


Figura 6: Histograma de las variables del dataset

La Figura presenta la distribución de las cuatro entradas operacionales del VFM y de la variable objetivo *GAS* para el activo P (campo Pucuna).

En **petróleo**, el histograma es **multimodal** en el rango aproximado 7 250–9 750 *bbl/d* (media $\sim 9\,001$, desviación ~ 274), con dos núcleos frecuentes alrededor de 8 800 y 9 100 *bbl/d* y un grupo menor cerca de 7 300 *bbl/d*. Esta forma refleja la repetición del perfil histórico del activo con variación estacional y ruido diario, y **no** exhibe mesetas artificiales en el techo del periodo.

En **agua**, la distribución es aproximadamente acampanada entre 45 000 y 57 000 *bbl/d* (media $\sim 52\,077$), centrada en torno a 52 500 *bbl/d*, con colas hacia valores más bajos ($\sim 46\,000$ – $49\,000$ *bbl/d*) coherentes con el rango del perfil de referencia.

En **presión de cabezal**, la frecuencia es **bimodal**: un máximo local cerca de 890 *psi* y otro cerca de 960 *psi*, en línea con un régimen BES estable pero con dos bandas operativas distinguibles; el eje extendido hasta $\sim 1\,170$ *psi* corresponde a un número reducido de registros extremos.

La **temperatura** se concentra principalmente entre 75 °C y 81 °C (mediana $\sim 77,5$ °C), con dispersión moderada hasta ~ 91 °C; aparece además un grupo secundario

en el límite inferior ($\sim 68\text{ }^{\circ}\text{C}$) asociado al mínimo operacional del sensor.

Respecto a *GAS*, la distribución es **unimodal y aproximadamente simétrica** en el rango 4 400–6 450 *MSCFD*, con núcleo frecuente alrededor de 5 500–5 700 *MSCFD* (media $\sim 5\,563$ *MSCFD*, desviación ~ 255 *MSCFD*). El comportamiento es coherente con la variabilidad de las entradas líquidas y con una correlación positiva moderada con petróleo ($r \approx 0,71$), sin la concentración previa en valores altos ($> 7\,000$ *MSCFD*) que aparecía cuando el petróleo quedaba fijado en meseta.

En conjunto, petróleo y agua muestran variabilidad con respecto al histórico del activo P; presión y temperatura aportan dispersión en rangos físicamente plausibles. La forma de la distribución de *GAS* confirma que el problema planteado es de **regresión continua** sobre un rango operacional definido, condición relevante para el preprocesamiento y el entrenamiento del modelo. **Matriz de correlación de las variables del dataset.**

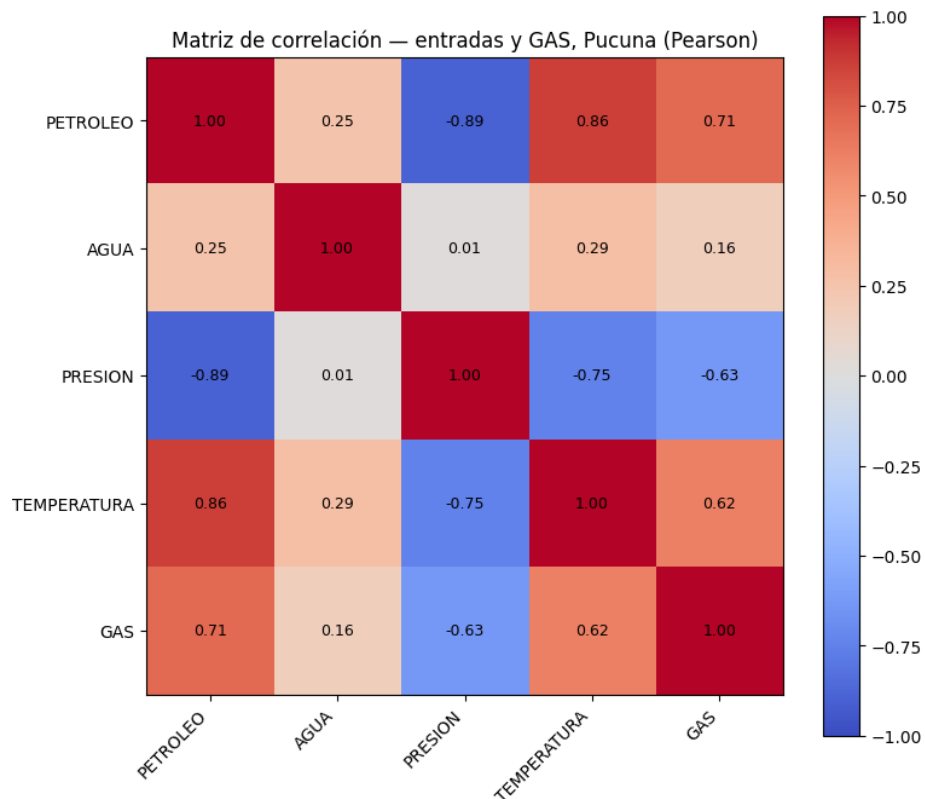


Figura 7: Matriz de correlación de las variables del dataset

La Figura presenta la matriz de correlación de Pearson entre las cuatro entradas operacionales del VFM y la variable objetivo *GAS* para el activo P (campo Pucuna).

Respecto a *GAS*, la asociación lineal más intensa se observa con **petróleo** ($r \approx 0,71$), seguida de **presión de cabezal** en sentido negativo ($r \approx -0,63$) y **temperatura** ($r \approx 0,62$). La correlación con **agua** es débil ($r \approx 0,16$), de modo que, en el subconjunto analizado, el gas se explica principalmente por el comportamiento del crudo y de las variables de cabezal BES, y no por la producción de agua en forma lineal.

Entre las entradas destacan dos bloques de dependencia fuerte. Por un lado, **petróleo** y **presión** exhiben correlación negativa muy alta ($r \approx -0,89$): a mayor caudal de petróleo, menor presión de cabezal. Por otro, **petróleo** y **temperatura** se mueven de forma conjunta ($r \approx 0,86$), mientras **presión** y **temperatura** se asocian negativamente ($r \approx -0,75$). **Agua** permanece relativamente desacoplada del resto ($|r| \leq 0,29$ con las demás entradas).

En conjunto, la matriz confirma que *GAS* guarda relación positiva con petróleo y temperatura y negativa con presión, en coherencia con un levantamiento BES. A diferencia de escenarios con producción líquida en meseta, aquí la correlación petróleo–*GAS* es fuerte pero no casi perfecta, lo que reduce el riesgo de redundancia extrema entre objetivo y una sola entrada. No obstante, la alta correlación cruzada entre petróleo, presión y temperatura sugiere multicolinealidad entre predictores; conviene tenerla presente al interpretar coeficientes del modelo y al definir el preprocesamiento (por ejemplo, normalización y validación cruzada).

3.1.5 Resultados

El dataset de entrenamiento del activo P (Pucuna) comprende 2 192 observaciones diarias (2020 a 2025) con cuatro entradas operacionales y gas asociado como objetivo. A continuación se presentan las variables consideradas y los estadísticos descriptivos obtenidos con *pandas* (*describe*), con datos completos en todas las columnas.

Descripción del dataset utilizado

```

1 Registros (# regs.): 2,192
2
3 Variables:
4     Entradas: PETROLEO (bbl/d), AGUA (bbl/d), PRESION (psi),
5     TEMPERATURA (C)
6     Objetivo: GAS (MSCFD)
7
8 Periodo: 2020-01-01 -> 2025-12-31
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31

```

	PETROLEO	AGUA	PRESION	TEMPERATURA	GAS
count	2192.00	2192.00	2192.00	2192.00	2192.00
mean	9001.41	52077.38	919.47	77.73	5562.95
std	273.66	1776.62	42.63	4.17	254.94
min	7256.02	44756.54	838.10	68.00	4357.61
25 %	8809.05	51518.37	887.50	74.60	5397.23
50 %	9008.50	52352.84	911.87	77.53	5566.47
75 %	9171.12	53069.99	954.97	80.59	5733.67
max	9723.35	56245.75	1168.89	90.59	6447.35

```

<class 'pandas.DataFrame'>
RangeIndex: 2192 entries, 0 to 2191
Data columns (total 6 columns):
#   Column          Non-Null Count  Dtype
---  -
0   FECHA            2192 non-null   datetime64[us]
1   PETROLEO         2192 non-null   float64
2   AGUA             2192 non-null   float64
3   PRESION          2192 non-null   float64
4   TEMPERATURA      2192 non-null   float64
5   GAS              2192 non-null   float64

dtypes: datetime64[us](1), float64(5)
memory usage: 102.9 KB

```

En conjunto, el dataset ofreció 2 192 observaciones diarias sin valores faltantes, con entradas en escalas operacionales distintas y un objetivo GAS con media de 5 562,95 *MSCFD* (rango 4 357,61–6 447,35). La mediana de GAS (5 566,47 *MSCFD*) fue muy

cercana a la media, lo que indicó una distribución relativamente simétrica; petróleo, agua y presión presentaron dispersión acotada en el periodo 2020–2025. Estas características justificaron la etapa de análisis exploratorio y la selección de modelos de regresión, en particular el MLP, para estimar el gas asociado a partir de las cuatro variables operacionales.

3.2 Análisis y selección del algoritmo de inteligencia artificial

Para la selección del algoritmo de predicción del flujo de gas asociado (GAS), se emplearon métricas de regresión ampliamente utilizadas en la literatura de Virtual Flow Metering y aprendizaje automático supervisado. Estas medidas permiten evaluar de forma complementaria la calidad de las predicciones: la RMSE y el MAE cuantifican el error en las mismas unidades operativas del flujo, el coeficiente de determinación R^2 indica qué proporción de la variabilidad de GAS es explicada por el modelo frente a una predicción basada solo en la media, y el MAPE expresa el error en términos porcentuales, facilitando la comparación entre algoritmos.

Todas las métricas se calcularon sobre un conjunto de prueba independiente correspondiente al 25 % de las observaciones, no utilizado durante el entrenamiento, de modo que el desempeño reportado refleje la capacidad de generalización de cada modelo. A continuación se presentan las expresiones matemáticas de cada indicador.

Raíz del error cuadrático medio (RMSE):

La RMSE corresponde a la raíz cuadrada del promedio de los errores al cuadrado entre el valor real y el valor predicho de la variable GAS (en MSCFD). Penaliza con mayor peso los errores grandes y se expresa en las mismas unidades que la variable objetivo, lo que facilita su interpretación operativa. Un valor menor indica mejor desempeño del modelo.

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^N (y_{i,\text{real}} - y_{i,\text{pred}})^2} \quad (1)$$

donde N es el número de observaciones en el conjunto evaluado, $y_{i,\text{real}}$ es el gas

asociado observado y $y_{i,\text{pred}}$ es el gas asociado estimado por modelo para observación i .

Error absoluto medio (MAE):

El MAE representa el promedio del valor absoluto de la diferencia entre el gas asociado real y el predicho. A diferencia de la RMSE, es menos sensible a valores atípicos y se interpreta como una desviación típica media del error de predicción, también en MSCFD.

$$MAE = \frac{1}{N} \sum_{i=1}^N |y_{i,\text{real}} - y_{i,\text{pred}}| \quad (2)$$

Coefficiente de determinación (R^2):

El R^2 mide la proporción de la varianza de la variable GAS explicada por el modelo respecto a una predicción basada únicamente en la media de los datos. Sus valores se encuentran entre 0 y 1; valores cercanos a 1 indican mayor capacidad explicativa del modelo.

$$R^2 = 1 - \frac{\sum_{i=1}^N (y_{i,\text{real}} - y_{i,\text{pred}})^2}{\sum_{i=1}^N (y_{i,\text{real}} - \bar{y})^2} \quad (3)$$

donde $\bar{y}$ corresponde a la media de GAS observado en el conjunto evaluado.

Error porcentual absoluto medio (MAPE):

El MAPE expresa el error medio en términos porcentuales respecto al valor real. Permite comparar modelos en escala relativa y evaluar la magnitud del error frente al nivel típico del gas asociado (GAS).

$$MAPE = \frac{100}{N} \sum_{i=1}^N \left| \frac{y_{i,\text{real}} - y_{i,\text{pred}}}{y_{i,\text{real}}} \right| \quad (4)$$

El resultado se reporta en porcentaje (%). En la práctica se excluyen observaciones con $y_{i,\text{real}} = 0$ para evitar divisiones indefinidas.

3.2.1 Algoritmos de regresion evaluados

El conjunto de entrenamiento del campo Pucuna (activo P) comprende 2 192 registros diarios con cuatro entradas operacionales (petróleo, agua, presión de cabezal y temperatura BES) y el flujo de gas asociado como variable objetivo (MSCFD). El análisis exploratorio evidenció multicolinealidad entre las entradas líquidas y térmicas y correlaciones lineales relevantes entre GAS y petróleo ($r \approx 0,71$), presión ($r \approx -0,63$) y temperatura ($r \approx 0,62$), con agua débilmente asociada al objetivo ($r \approx 0,16$).

En ese contexto, no es posible presuponer la forma funcional óptima del estimador. Por ello se evaluó un portafolio de modelos de regresión: (i) regresión lineal con regularización Ridge, como referencia interpretable ante relaciones casi lineales y multicolinealidad; (ii) Random Forest, por su capacidad de modelar datos tabulares con interacciones no lineales y distribuciones concentradas; (iii) SVR, como referencia no lineal clásica con entradas escaladas; y (iv) un perceptrón multicapa (MLP) como estimador principal del VFM, por su capacidad de aproximar relaciones no lineales entre variables operacionales de distinta naturaleza y escala [27, 24, 26, 29, 30]. La comparación se realizó sobre el mismo dataset y las mismas métricas (MAE, RMSE, R^2), empleando validación que respete el orden temporal de las observaciones [28].

3.2.2 Modelo *naive* de referencia

Antes de contrastar los cuatro estimadores anteriores, se definió un **modelo *naive*** (inglés: *naive baseline*) como línea base mínima de desempeño. En regresión supervisada, cualquier VFM que aporte valor operativo debe superar esta referencia; de lo contrario, bastaría con reportar un caudal fijo sin usar las variables de proceso.

En este trabajo, el *naive* corresponde a una predicción para cada día del conjunto de evaluación, se estimó el mismo valor de GAS, igual a la media aritmética de GAS calculada únicamente sobre el subconjunto de entrenamiento (75 % de las observaciones, previo al 25 % de prueba). Formalmente,

$$\hat{y}_i^{\text{naive}} = \bar{y}_{\text{train}} \quad \forall i \in \text{conjunto de prueba}, \quad (5)$$

donde $\bar{y}_{\text{train}}$ es la media de GAS (MSCFD) en entrenamiento y $\hat{y}_i^{\text{naive}}$ la predicción para el día i . El modelo *naive* **no utiliza** petróleo, agua, presión ni temperatura: ignora por completo la información operacional disponible ese día.

En el activo P del periodo analizado, $\bar{y}_{\text{train}}$ se sitúa en el orden de 5 500–5 600 *MSCFD*, coherente con la media global reportada en el análisis exploratorio ($\approx 5\,563$ *MSCFD*). Así, ante un gas real que oscila entre $\approx 4\,400$ y $6\,450$ *MSCFD*, el *naive* comete un error sistemático en los días por encima o por debajo de la media, aunque pueda acertar de forma accidental cuando el caudal observado está cerca del valor central.

Interpretación frente a las métricas. El coeficiente R^2 definido en la ecuación eq:r2 compara cada modelo con una predicción basada en la media del conjunto *evaluado*; el *naive* basado en $\bar{y}_{\text{train}}$ cumple un rol análogo como referencia operativa fija aprendida solo del histórico previo. Un R^2 cercano a cero (o negativo) en el *naive* indicaría que predecir siempre la media no explica la variabilidad diaria del gas. En cambio, valores altos de R^2 en MLP, Ridge, Random Forest o SVR muestran que las entradas operacionales aportan información más allá de ese nivel.

Columna “Mejora vs. Naive”. En el Cuadro 29, la mejora porcentual se calculó a partir del error absoluto medio (MAE) de cada algoritmo frente al MAE del *naive*, en la forma

$$\text{Mejora vs. Naive (\%)} = \left(1 - \frac{\text{MAE}_{\text{modelo}}}{\text{MAE}_{\text{naive}}}\right) \times 100, \quad (6)$$

de modo que un 86,33 % en el MLP significa que el error medio del perceptrón multicapa es aproximadamente un 86 % menor que el de predecir siempre $\bar{y}_{\text{train}}$. Esta comparación cuantifica el valor agregado del VFM frente a una estimación rudimentaria que podría adoptarse en ausencia de modelo (por ejemplo, “asumir el gas del año pasado en su promedio”).

Alcance y límites. El *naive* adoptado no es un modelo físico de yacimiento ni calcula explícitamente una relación gas–aceite ($\text{GOR} \times \text{petróleo}$); tampoco replica la persistencia temporal ($\hat{y}_i = y_{i-1}$), otra línea base habitual en series diarias. Se eligió la **media de entrenamiento** por su interpretación directa, su alineación con el significado de R^2 y su uso como piso de comparación en el pipeline de evaluación del proyecto. Los cuatro algoritmos de regresión que siguen se midieron con las mismas métricas y contra este mismo *naive*.

3.2.3 Algoritmos:

Regresión Lineal(Ridge):

La regresión lineal es un modelo supervisado que estima la variable objetivo como una combinación lineal de las variables de entrada, ajustando coeficientes que minimizan el error cuadrático entre valores observados y predichos [27]. En el contexto del Medidor Virtual de Flujo, permite establecer una relación directa entre presión, temperatura y otras variables operacionales con el flujo de gas asociado (GAS), con ventajas de simplicidad, bajo costo computacional e interpretabilidad de los coeficientes.

Máquinas de Soporte Vectorial para Regresión (SVR)

Las SVR extienden el concepto de máquinas de soporte vectorial al problema de regresión, buscando una función que aproxime la salida con un error acotado mediante un parámetro ε , mientras se controla la complejidad del modelo con un término de regularización [25, 26]. Mediante funciones kernel (por ejemplo, RBF), el SVR puede modelar relaciones no lineales entre las variables de entrada y GAS sin definir explícitamente la forma de la función. En aplicaciones de VFM ha mostrado buen desempeño en conjuntos de tamaño moderado y alta complejidad operacional.

Random Forest:

Random Forest es un método de ensamble que combina múltiples árboles de decisión entrenados sobre submuestras aleatorias de los datos y subconjuntos aleatorios de variables, promediando sus predicciones para obtener la estimación final [24]. Cada

árbol captura particiones no lineales del espacio de entrada, lo que permite modelar interacciones entre presión, temperatura y variables derivadas sin suponer una forma funcional fija.

Red Neuronal Multicapa (MLP)-Perceptron Multicapa:

La red neuronal multicapa es un modelo de aprendizaje profundo compuesto por capas de neuronas totalmente conectadas, en el que cada neurona aplica una transformación lineal seguida de una función de activación no lineal (por ejemplo, ReLU), permitiendo aproximar funciones complejas entre entradas y salida [29, 30].

3.2.4 Resultados de la evaluación de los algoritmos de regresión

La Tabla 29 resumió el desempeño comparativo de MLP, Ridge, Random Forest y SVR sobre el conjunto de prueba (25 % de los datos) para la predicción del flujo de gas asociado en el campo Pucuna. Todas las métricas se expresaron en porcentaje: MAE y RMSE indicaron el error medio y cuadrático relativo; R^2 la proporción de variabilidad explicada; MAPE el error porcentual absoluto medio; y la columna *Mejora vs. Naive* cuantificó el incremento de precisión respecto a un modelo de referencia que predijo el valor medio histórico.

Cuadro 29: Métricas de desempeño de los modelos en el conjunto de prueba (25 %) — campo Pucuna

Modelo	MAE (%)	RMSE (%)	R^2 (%)	MAPE (%)	Mejora vs. Naive (%)
MLP	0.13	0.18	97.02	0.13	86.33
Ridge (regresión lineal)	0.17	0.26	93.55	0.17	80.95
Random Forest	0.30	0.37	87.02	0.30	67.70
SVR	0.27	0.37	86.96	0.27	71.03

Nota: métricas expresadas en porcentaje. Ordenado por R^2 descendente.

Autor del cuadro: Marco Montesdeoca

MLP. El perceptrón multicapa obtuvo el mejor desempeño global: un R^2 de 97,02 %, con errores mínimos en MAE (0,13 %), RMSE (0,18 %) y MAPE (0,13 %). Además, supera al modelo *naive* en un 86,33 %, lo que indica que captura de forma consistente la variabilidad del gas asociado a partir de las variables operacionales de entrada.

Ridge. La regresión lineal regularizada ocupó el segundo lugar, con un R^2 de

93,55 %, MAE de 0,17 %, RMSE de 0,26 %, MAPE de 0,17 % y mejora frente al *naive* de 80,95 %. La cercanía con el MLP sugiere que gran parte de la relación entre presión, temperatura, producción líquida y gas es aproximadamente lineal en este conjunto.

Random Forest. El ensamble de árboles alcanzó un R^2 de 87,02 %, con MAE y MAPE de 0,30 % y RMSE de 0,37 %. Su mejora respecto al *naive* fue de 67,70 %, inferior a la de MLP y Ridge, lo que refleja mayor dificultad para generalizar con solo cuatro entradas tabulares.

SVR. Las máquinas de soporte vectorial registraron un R^2 de 86,96 %, con MAE de 0,27 %, RMSE de 0,37 % y MAPE de 0,27 %. Aunque su R^2 es ligeramente menor que el de Random Forest, la mejora frente al *naive* (71,03 %) es algo superior, lo que indica un comportamiento intermedio entre los modelos no lineales evaluados.

En conjunto, los cuatro algoritmos superan el 85 % de R^2 y mantienen errores porcentuales inferiores a 0,40 % en MAE, RMSE y MAPE. La brecha más relevante se observa entre los dos primeros modelos (MLP y Ridge) y los dos siguientes (Random Forest y SVR). Por el balance entre precisión, estabilidad y mejora frente al *naive*, se selecciona el MLP como algoritmo principal del VFM para el caso de estudio Pucuna.

3.2.5 Resultados

Comparacion y seleccion del algoritmo de prediccion

Se realizó la comparación de algoritmos de regresión para la predicción del flujo de gas asociado (GAS), con base en el entrenamiento del modelo y en los datos del gas real del conjunto de prueba.

La Tabla 30 presenta una muestra de 14 días del conjunto de prueba, en los que se contrasta el gas medido con la predicción de cada uno de los cuatro modelos evaluados: Ridge, MLP, Random Forest y SVR. Las entradas operacionales de cada fecha son las mismas que alimentan todos los modelos; la diferencia entre columnas refleja únicamente el criterio de cada algoritmo para estimar el flujo de gas.

Cuadro 30: Comparación de predicciones vs. gas real (MSCFD) — campo Pucuna

N	Fecha	Gas real	Pred. Ridge	Pred. MLP	Pred. RF	Pred. SVR
1	2025-12-18	9411.935	9398.180 554	9411.412 719	9400.751 680	9428.486 629
2	2025-12-19	9349.951	9361.270 602	9347.881 629	9379.554 520	9395.645 777
3	2025-12-20	9547.678	9500.897 539	9546.056 639	9489.582 146	9505.340 846
4	2025-12-21	9436.688	9453.974 790	9425.132 584	9474.459 614	9458.330 333
5	2025-12-22	9305.793	9374.506 222	9338.662 605	9387.292 874	9393.496 328
6	2025-12-23	9380.829	9371.205 658	9418.828 078	9402.184 479	9377.116 801
7	2025-12-24	9332.201	9328.239 178	9307.466 627	9371.755 409	9351.563 001
8	2025-12-25	9465.914	9459.315 576	9473.430 563	9483.551 005	9459.394 858
9	2025-12-26	9476.419	9474.798 437	9483.515 279	9472.711 092	9479.444 069
10	2025-12-27	9471.947	9470.275 944	9489.299 290	9455.033 997	9467.292 130
11	2025-12-28	9651.871	9638.400 765	9641.397 040	9619.242 597	9596.016 874
12	2025-12-29	9550.608	9543.606 308	9544.495 106	9543.382 499	9537.878 707
13	2025-12-30	9582.219	9567.861 174	9581.615 244	9599.655 380	9553.637 924
14	2025-12-31	9464.247	9449.302 785	9469.514 045	9481.531 838	9461.136 058

Nota: Muestra de 14 días del conjunto de prueba (25 %). Valores en MSCFD. RF = Random Forest.

Autor del cuadro: Marco Montesdeoca

La muestra del 18 al 31 de diciembre de 2025 muestra que el gas real oscila entre 9 305,79 y 9 651,87 *MSCFD*, con un repunte hacia fin de mes. Los cuatro modelos reproducen esa tendencia y mantienen predicciones en un rango operativo equivalente, sin desviaciones extremas respecto del valor medido.

El MLP registra los acercamientos más notables al gas real en fechas puntuales: el 18/12/2025 (real 9 411,94; predicción 9 411,41 *MSCFD*) y el 30/12/2025 (real 9 582,22; predicción 9 581,62 *MSCFD*). Ridge se mantiene muy próximo en la mayoría de los días; por ejemplo, el 26/12/2025 predice 9 474,80 frente a un real de 9 476,42 *MSCFD*. En el mínimo del periodo (22/12/2025, real 9 305,79 *MSCFD*), el MLP (9 338,66) se acerca más que Ridge (9 374,51) y Random Forest (9 387,29).

Random Forest y SVR siguen la serie con errores algo mayores en días de mayor variación. El 20/12/2025, con gas real de 9 547,68 *MSCFD*, Random Forest estima 9 489,58 y SVR 9 505,34, mientras Ridge y MLP quedan más cerca (9 500,90 y 9 546,06, respectivamente). En el pico del 28/12/2025 (real 9 651,87 *MSCFD*), SVR subestima con mayor margen (9 596,02 *MSCFD*) que MLP y Ridge (9 641,40 y 9 638,40 *MSCFD*).

En conjunto, la tabla confirmó día a día lo observado en el Cuadro de métricas globales: MLP y Ridge ofrecieron el seguimiento más fino del gas medido; Random Forest y SVR fueron competitivos, pero con desajustes puntuales algo superiores. Esta

comparación cualitativa complementó el análisis cuantitativo (MAE, RMSE, R^2 y MAPE) de la subsección anterior.

Ventajas del algoritmo seleccionado

La Red Neuronal Multicapa (MLP) fue seleccionada como algoritmo principal para la predicción del flujo de gas asociado debido a su capacidad para aprender relaciones complejas entre las variables operacionales del sistema. En el contexto del Medidor Virtual de Flujo, esta característica resultó importante, ya que el flujo de gas podía depender de interacciones no lineales entre presión, temperatura y condiciones operativas. Otras características obtenidas fueron:

- **Modelado de relaciones no lineales:** La MLP permitió capturar relaciones complejas entre presión, temperatura y flujo de gas asociado, superando las limitaciones de modelos lineales tradicionales.
- **Adecuada para problemas de regresión:** El sistema buscó predecir un valor numérico continuo, correspondiente al flujo de gas asociado. La MLP se adaptó correctamente a este tipo de problema.
- **Aprendizaje a partir de datos operacionales:** El modelo aprendió patrones directamente desde los datos disponibles, sin requerir una ecuación física explícita del comportamiento del gas en cada activo.
- **Adaptabilidad por activo:** El sistema entrenó un modelo por cada activo, permitiendo que la red neuronal se ajustara a las condiciones específicas de presión, temperatura y flujo de cada uno.
- **Compatibilidad con TensorFlow/Keras:** La implementación mediante TensorFlow/Keras facilitó la construcción, entrenamiento, almacenamiento y reutilización de los modelos entrenados.
- **Regularización para reducir sobreajuste:** El uso de Dropout y regularización L2 contribuyó a mejorar la generalización del modelo y reducir la dependencia excesiva de patrones específicos del dataset.

- **Validación cruzada del entrenamiento:** El uso de K-Fold con cinco particiones permitió evaluar el modelo en diferentes subconjuntos de datos y seleccionar el modelo con mejor desempeño.
- **Uso de parada temprana:** El mecanismo de *Early Stopping* evitó entrenamientos innecesarios y ayudó a conservar los pesos del modelo con mejor desempeño durante la validación.

3.3 Implementación del sistema integral de Medidor Virtual de Flujo

La presente sección documenta el proceso seguido para construir el Medidor Virtual de Flujo (VFM): desde la definición de requisitos y el diseño de la arquitectura, hasta la construcción de los módulos de software, su validación y la puesta en operación del sistema integrado.

Selección del modelo de ciclo de vida del software

Al momento de iniciar el desarrollo, fue necesario elegir un **modelo de ciclo de vida del software** acorde con la naturaleza del proyecto. El VFM no es un componente aislado, sino una solución que articula base de datos, procesamiento de datos, entrenamiento de modelos de inteligencia artificial, servicios HTTP y una interfaz web para el usuario final. Además, el alcance académico exige entregables verificables en plazos definidos, validación temprana de requisitos y retroalimentación sobre prototipos antes del cierre del trabajo.

Por ello, la selección del ciclo de vida condiciona el orden del trabajo, la forma de integrar los módulos y la gestión del riesgo de dejar la validación para una etapa final. Un enfoque inadecuado, por ejemplo, un desarrollo estrictamente secuencial sin revisiones intermedias, podría retrasar la detección de errores de acoplamiento entre capas o de interpretación de los requisitos operacionales del campo Pucuna. En consecuencia, a continuación se presenta la comparación de alternativas y la justificación de la metodología adoptada.

Modelos de ciclo de vida del software

La selección de las metodologías evaluadas para el desarrollo del sistema se fundamentó en la necesidad de identificar un enfoque que permita integrar de manera estructurada el desarrollo de software, el procesamiento de datos y el entrenamiento de modelos de inteligencia artificial aplicados al Virtual Flow Metering.

A partir de la revisión bibliográfica realizada, se identificó que los proyectos relacionados con VFM requieren procesos iterativos de preparación de datos, entrenamiento, validación y ajuste continuo de modelos predictivos, tal como se evidencia en los trabajos de [3, 11, 13, 15, 14]. Estos estudios muestran que los sistemas basados en inteligencia artificial demandan metodologías flexibles que permitan la experimentación, recalibración y mejora progresiva del modelo durante el ciclo de vida del proyecto.

Se consideraron modelos de ciclo de vida del software ampliamente utilizados en el desarrollo de sistemas: RAD, el espiral, el prototipado y el modelo en cascada. Estos enfoques permiten contrastar un desarrollo secuencial y fuertemente documentado frente a alternativas iterativas o exploratorias que favorecen la validación progresiva o la verificación temprana de ideas mediante versiones de trabajo.

La comparación metodológica tomó como referencia principios de calidad y gestión establecidos en normas internacionales como ISO/IEC/IEEE 12207:2017 e ISO/IEC 25010:2011 ([Anexo A](#)), relacionadas con procesos del ciclo de vida del software y con atributos de calidad como mantenibilidad, confiabilidad y compatibilidad, respectivamente. Esto permitió seleccionar una metodología alineada tanto con las necesidades técnicas del sistema de Medidor Virtual de Flujo como con buenas prácticas internacionales para el desarrollo de soluciones basadas en inteligencia artificial.

- **Modelo en cascada:** Metodología secuencial donde cada etapa debe finalizar antes de iniciar la siguiente. Es adecuada para proyectos con requisitos muy estables desde el inicio, aunque presenta poca flexibilidad ante cambios y validación tardía del conjunto. Sus fases son:

- **Requisitos:** definición del alcance y especificación de necesidades del sistema.

- **Análisis:** modelado de procesos, restricciones y viabilidad técnica.
 - **Diseño:** arquitectura, estructura de datos e interfaces del software.
 - **Implementación:** codificación y ensamblaje de los módulos.
 - **Pruebas:** verificación y validación del producto frente a los requisitos.
 - **Mantenimiento:** corrección, evolución y soporte del sistema en operación.
- **Modelo RAD (*Rapid Application Development*):** Metodología orientada a ciclos cortos con prototipos ejecutables y retroalimentación temprana del usuario [21, 20, 22]. Sus fases son:
- **Requisitos:** levantamiento y priorización de necesidades con participación del usuario.
 - **Diseño:** definición de arquitectura, interfaces y plan de construcción por ciclos.
 - **Construcción:** desarrollo de prototipos ejecutables y refinamiento iterativo.
 - **Implantación:** despliegue, pruebas de aceptación y puesta en operación del producto.
- **Modelo espiral:** Metodología orientada a la gestión de riesgos, combinando prototipos y desarrollo iterativo [23]. Se utiliza principalmente en proyectos de gran complejidad y cuando el riesgo debe gobernar cada ciclo. En cada vuelta del espiral se repiten las siguientes actividades:
- **Planificación:** definición de objetivos, alternativas y restricciones del ciclo.
 - **Análisis de riesgos:** identificación, priorización y planes de mitigación.
 - **Desarrollo e ingeniería:** construcción del producto o prototipo del ciclo.
 - **Evaluación:** revisión con el cliente y decisión de continuar, ajustar o detener.
- **Modelo de prototipos:** Metodología enfocada en versiones preliminares para validar requisitos e interacción antes de consolidar el producto. Complementa otros enfoques, pero suele ser insuficiente como única estrategia de gestión del proyecto completo. Sus fases son:

- **Identificación de requisitos:** recolección de necesidades con el usuario.
- **Diseño rápido:** esquema funcional y visual del prototipo.
- **Construcción del prototipo:** implementación de una versión operativa limitada.
- **Evaluación:** validación de usabilidad, requisitos y ajustes solicitados.
- **Refinamiento:** incorporación de cambios al prototipo o al diseño final.
- **Implementación:** desarrollo del producto definitivo a partir del prototipo validado.

Los criterios imperativos se definieron como condiciones de admisibilidad del ciclo de vida frente a la naturaleza del VFM: un sistema que articula varios subsistemas (datos, inteligencia artificial, servicios y presentación) y que solo adquiere sentido si se integra y valida de manera progresiva, evitando un cierre único al final donde los errores de acoplamiento se detecten tarde y caro. En ese marco, también se exige que la carga de formalización y gestión sea acorde al alcance del proyecto, descartando enfoques cuya burocracia interna compita con el propio objetivo técnico.

Los criterios deseables, en cambio, no excluyen por sí solos una alternativa, pero permiten ordenar a las opciones que compiten en el mismo espacio de decisión: ponderan factores típicamente relevantes en un desarrollo acotado (adecuación a cronograma y tamaño de equipo, claridad y trazabilidad del proceso, y mitigación del riesgo de integración tardía), de modo que la selección quede sustentada tanto por cumplimiento de mínimos no negociables como por una evaluación explícita de conveniencia entre alternativas viables.

Cuadro 31: Ponderación aplicada a los criterios deseables para la selección del modelo de ciclo de vida

Criterio Deseable	Peso
Adecuación a cronograma académico y equipo reducido	0,35
Claridad, predictibilidad y trazabilidad del proceso	0,30
Reducción del riesgo de integración tardía “gran cierre”	0,35
Total	1,00

Autor del cuadro: Marco Montesdeoca

Cuadro 32: Selección del modelo de ciclo de vida del software mediante el modelo de toma de decisiones de Simon (desarrollo del VFM).

Criterio	Peso	Cascada	RAD	Espiral	Prototipos
I. Criterios imperativos (filtro; no se ponderan)					
Imp. 1: Permite integración progresiva entre módulos del VFM (datos, IA, API, interfaz)	—	No	Sí	Sí	No
Imp. 2: Evita depender de una validación únicamente al cierre del ciclo	—	No	Sí	Sí	Sí [†]
Imp. 3: Formalización y carga de gestión acotadas al alcance del proyecto	—	Sí	Sí	No	Sí
II. Criterios deseables (escala 1–5; comparación cualitativa)					
Des. A: Adecuación a cronograma académico y equipo reducido	0,35	3	5	2	4
Des. B: Claridad, predictibilidad y trazabilidad del proceso	0,30	5	4	3	2
Des. C: Reducción del riesgo de integración tardía “gran cierre”	0,35	2	5	4	3
Puntaje ponderado $\sum_i w_i s_i$	1,00	3,25	4,70	3,00	3,05
Cumplimiento del filtro imperativo (I)	—	No cumple	Cumple	No cumple	No cumple
Decisión para el proyecto	—	Descartada	Seleccionada (RAD)	Descartada	Descartada (complementaria)

Autor del cuadro: Marco Montesdeoca

Con el análisis comparativo realizado, se seleccionó la metodología **RAD** (*Rapid Application Development*) como enfoque principal para el desarrollo del sistema VFM. Este enfoque se consideró el más adecuado porque el proyecto combina componentes heterogéneos, base de datos, backend de inteligencia artificial, servicios HTTP y aplicación web que debían converger en un producto utilizable en plazos académicos

acotados, con validación temprana de requisitos, interfaces y resultados del modelo de inteligencia artificial.

3.3.1 Selección de herramientas tecnológicas

La pila tecnológica se definió con criterios de compatibilidad entre capas, integración por HTTP, reproducibilidad del entorno y anticipación de los atributos de calidad del sistema —alineados con ISO/IEC 25010:2011 ([Anexo A](#))— que más adelante se formalizaron como requisitos no funcionales (Tabla 44). La comparación formal mediante el modelo de Simon se presentó en las tablas de ponderación y decisión de cada componente (por ejemplo, Tabla 36).

Cuadro 33: Ponderación aplicada a los criterios deseables para la selección del lenguaje principal de inteligencia artificial

Criterio Deseable	Peso
Ecosistema de machine learning	0,40
Facilidad para procesar archivos Excel/CSV	0,30
Facilidad de implementación	0,30
Total	1,00

Autor del cuadro: Marco Montesdeoca

Cuadro 34: Selección del lenguaje principal para inteligencia artificial mediante el modelo de Simon

Criterio	Peso	R	Python	Java
Criterios imperativos (filtro)				
Imp1: Soporte para inteligencia artificial	–	Cumple el filtro: buen ecosistema estadístico; para redes neuronales y despliegue con API suele requerir más ensamblaje que Python.	Cumple el filtro: librerías maduras (TensorFlow/Keras, PyTorch, scikit-learn) y flujo típico de entrenamiento e inferencia.	Cumple el filtro: hay librerías de ML; el camino más directo para prototipos científicos no suele ser tan ágil como en Python.
Imp2: Manejo eficiente de datos tabulares	–	Cumple el filtro para tablas clásicas; pipelines repetibles ML+Excel/CSV suelen ser más homogéneos en Python.	Cumple el filtro: Pandas/NumPy facilitan limpieza, pivotes y preparación del dataset del VFM.	No cumple: el manejo tabular es posible, pero no se considera equivalente al requisito de eficiencia práctica con Excel/CSV del proyecto.
Imp3: Integración con APIs web	–	No cumple: se puede integrar, pero no es el caso de uso principal; exponer API ligera de inferencia es más natural en Python/Java.	Cumple el filtro: encaje directo con FastAPI/Flask y despliegue de endpoints de predicción.	Cumple el filtro: Java es fuerte en servicios; para iteración rápida del prototipo VFM resulta más pesado que Python.
Criterios deseables (puntuación 1–5)				
Des.A: Ecosistema de machine learning	0.40	– (descartado)	5 — ecosistema amplio para deep learning y utilidades de entrenamiento/serialización.	3 — sólido en empresas; menos ágil para el ciclo experimental del VFM.
Des.B: Facilidad para procesar Excel/CSV	0.30	– (descartado)	5 — lectura/escritura y transformación con Pandas alineada al backend.	3 — viable con mayor costo de implementación.
Des.C: Facilidad de implementación	0.30	– (descartado)	5 — mismo lenguaje para datos, modelo y API reduce fricción.	3 — mayor verbosidad y tiempo de iteración.
Puntaje total	1.00	Descartado	5.00	3.00

Autor del cuadro: Marco Montesdeoca

Cuadro 35: Ponderación aplicada a los criterios deseables para la selección del framework web

Criterio Deseable	Peso
Facilidad de desarrollo de interfaz	0,35
Organización del código	0,35
Formularios y manejo de archivos Excel	0,30
Total	1,00

Autor del cuadro: Marco Montesdeoca

Cuadro 36: Selección del framework web mediante el modelo de toma de decisiones de Simon

Criterio	Peso	Laravel	Django	Express.js
Criterios imperativos (filtro)				
Imp1: Aplicación web MVC	–	Sí: MVC nativo (rutas, controladores, vistas Blade) alineado al patrón del VFM.	Sí: MVC completo con administración y ORM integrado.	No: es minimalista; MVC completo no es inherente y depende de arquitectura ad-hoc.
Imp2: Rutas, vistas y formularios	–	Sí: formularios, validación, middleware y plantillas para carga de Excel y flujos web.	Sí: sistema de plantillas y formularios maduro.	Sí: posible, pero requiere más piezas y convenciones propias para igualar productividad.
Imp3: Integración con backend externo vía HTTP	–	Sí: cliente HTTP y multipart encajan con consumo de FastAPI.	Sí: consumo de APIs externas viable con librerías estándar.	Sí: llamadas HTTP desde Node; hay que reforzar timeouts, reintentos y manejo de archivos.
Criterios deseables (puntuación 1–5)				
Des.A: Facilidad de desarrollo de interfaz	0.35	5 — Blade, layouts y flujo rápido para paneles y formularios del VFM.	4 — muy productivo; curva distinta a PHP si el equipo es heterogéneo.	3 — flexible, pero más ensamblaje para UI “completa” tipo panel.
Des.B: Organización del código	0.35	5 — convenciones Laravel claras por módulos/rutas/controladores.	4 — estructura por apps; puede crecer en complejidad.	3 — orden depende fuertemente del criterio del proyecto.
Des.C: Formularios y archivos (Excel)	0.30	5 — requests, validación y almacenamiento temporal alineados al entrenamiento.	4 — integración con capa Python del modelo puede duplicar ecosistemas.	3 — requiere middleware (<i>multer</i>) y validación manual más explícita.
Puntaje total	1.00	5.00	4.00	3.00

Autor del cuadro: Marco Montesdeoca

Este cuadro resume la aplicación del modelo de Simon para elegir la tecnología de la capa web del VFM. Los criterios imperativos actúan como filtros mínimos: se exige un enfoque MVC utilizable para construir la aplicación, un manejo claro de rutas, vistas y formularios (especialmente para la carga de archivos Excel) y la capacidad de integrar un backend externo por HTTP, condición necesaria para comunicarse con el servicio de inteligencia artificial. Las alternativas que no cumplen esos filtros quedan descartadas; sobre las restantes, los criterios deseables se ponderan para comparar la facilidad de desarrollo de interfaz, la organización del código y la compatibilidad con formularios y archivos. El resultado justifica la selección de Laravel como opción con mejor desempeño ponderado frente a Django y Express.js en el contexto del sistema implementado.

Cuadro 37: Ponderación aplicada a los criterios deseables para la selección del sistema de gestión de base de datos

Criterio Deseable	Peso
Integridad de datos estructurados	0,40
Escalabilidad futura	0,30
Compatibilidad con análisis de datos	0,30
Total	1,00

Autor del cuadro: Marco Montesdeoca

Cuadro 38: Selección del sistema de gestión de base de datos mediante el modelo de Simon

Criterio	Peso	SQLite	MySQL	PostgreSQL
Criterios imperativos (filtro)				
Imp1: Base relacional estructurada	–	Sí: relacional embebida; adecuada para prototipos pequeños.	Sí: motor relacional maduro y amplio despliegue.	Sí: motor relacional maduro con buen soporte analítico.
Imp2: Multiusuario / escalable	–	No: limitado para concurrencia y crecimiento típico de datos operacionales.	Sí: adecuado para servicios concurrentes estándar.	Sí: fuerte en concurrencia, integridad y cargas analíticas.
Imp3: Integración con Docker y Python	–	Sí: integrable en contenedores y desde Python.	Sí: compatible con Docker y conectores Python.	Sí: compatible con Docker y ecosistema Python/ETL.
Criterios deseables (puntuación 1–5)				
Des.A: Integridad de datos estructurados	0.40	– (descartada)	4 — sólido para OLTP; analítica avanzada depende del diseño.	5 — tipos, integridad y consultas analíticas muy adecuadas al VFM.
Des.B: Escalabilidad futura	0.30	– (descartada)	4 — escalable en muchos escenarios corporativos.	5 — encaje con crecimiento de datos y análisis.
Des.C: Compatibilidad con análisis de datos	0.30	– (descartada)	4 — viable con herramientas estándar.	5 — muy alineado con pipelines y análisis reproducibles.
Puntaje total	1.00	Descartada	4.00	5.00

Autor del cuadro: Marco Montesdeoca

Esta tabla documenta la decisión del motor relacional que soporta el esquema de datos del VFM. Los criterios imperativos priorizan un modelo relacional estructurado, adecuado para entornos multiusuario o escalables y compatible con despliegue mediante Docker y consumo desde Python, alineado con el pipeline de carga, transformación y análisis del dataset operacional. SQLite queda fuera cuando no satisface el requisito de escalabilidad/concurrencia típica de un entorno operacional o corporativo. Los criterios deseables comparan integridad de datos estructurados, escalabilidad futura y compatibilidad con análisis de datos; con ello se fundamenta la elección de PostgreSQL frente a MySQL y el descarte explícito de SQLite.

Cuadro 39: Ponderación aplicada a los criterios deseables para la selección de la herramienta de despliegue

Criterio Deseable	Peso
Facilidad de configuración	0,35
Portabilidad	0,35
Rapidez de inicialización	0,30
Total	1,00

Autor del cuadro: Marco Montesdeoca

Cuadro 40: Selección de herramienta de despliegue mediante el modelo de Simon

Criterio	Peso	Instalación local	Máquina virtual	Docker
Criterios imperativos (filtro)				
Imp1: Reproducir entorno en otra máquina	–	No: depende de instalación manual y versiones del host.	Sí: imagen/plantilla puede estandarizar, aunque opera con más fricción operativa.	Sí: contenedores y compose facilitan repetir el mismo entorno.
Imp2: Levantar PostgreSQL sin instalación manual local	–	No: normalmente requiere instalación/configuración en el SO.	No: suele requerir pasos adicionales frente a contenedores listos.	Sí: servicio Postgres en compose con inicialización automatizada.
Imp3: Reducir conflictos de configuración	–	No: alto riesgo de divergencia entre equipos.	Sí: mejora, pero mantiene sobrecarga de SO y redes.	Sí: aísla dependencias y versiones del stack del VFM.
Criterios deseables (puntuación 1–5)				
Des.A: Facilidad de configuración	0.35	– (descartada)	3 — requiere aprovisionamiento y mantenimiento de VM.	5 — comandos reproducibles y entorno declarativo.
Des.B: Portabilidad	0.35	– (descartada)	4 — portable, pero menos “caja cerrada” que contenedores.	5 — portabilidad alta entre equipos del equipo de trabajo.
Des.C: Rapidez de inicialización	0.30	– (descartada)	3 — arranque y red típicamente más lentos que contenedores.	5 — levantar stack completo con menor fricción repetida.
Puntaje total	1.00	Descartada	3.35	5.00

Autor del cuadro: Marco Montesdeoca

Aquí se formaliza la comparación entre instalar localmente, usar máquina virtual o contenedores Docker para reproducir el entorno del proyecto. Los imperativos enfatizan reproducibilidad entre equipos, la posibilidad de levantar PostgreSQL sin depender de instalaciones manuales en el sistema operativo y la reducción de conflictos de versiones y dependencias. La instalación local suele fallar en reproducibilidad y automatización;

la máquina virtual mejora estandarización pero no cumple igual el criterio de levantar PostgreSQL “sin fricción” comparado con Docker. Los criterios deseables evalúan facilidad de configuración, portabilidad y rapidez de inicialización, lo que respalda Docker como alternativa con mayor puntaje total para el VFM.

Cuadro 41: Ponderación aplicada a los criterios deseables para la selección del framework de API

Criterio Deseable	Peso
Simplicidad de implementación	0,35
Rendimiento	0,35
Documentación automática de API	0,30
Total	1,00

Autor del cuadro: Marco Montesdeoca

Cuadro 42: Selección del framework para API mediante el modelo de toma de decisiones de Simon

Criterio	Peso	Flask	Django REST	FastAPI
Criterios imperativos (filtro)				
Imp1: Exponer servicios HTTP	–	Sí: microframework con routing HTTP estándar.	Sí: expone REST con el ecosistema Django.	Sí: orientado a APIs HTTP de alto rendimiento.
Imp2: Compatible con Python y modelos IA	–	Sí: integración directa con TensorFlow/Keras vía Python.	Sí: viable, aunque el stack puede volverse más pesado para un servicio de inferencia.	Sí: encaje natural con carga de archivos, validación y modelo en el mismo runtime.
Imp3: Servicio ligero de predicción	–	Sí: puede ser ligero, pero requiere más piezas para producción estándar.	No: framework completo; mayor sobrecarga para un microservicio de predicción.	Sí: encaja en microservicio enfocado a entrenamiento/predicción.
Criterios deseables (puntuación 1–5)				
Des.A: Simplicidad de implementación	0.35	4 — simple al inicio; estructura final depende del equipo.	– (descartado)	5 — rutas, dependencias y validación con poco boilerplate.
Des.B: Rendimiento	0.35	3 — adecuado; no optimizado por defecto como ASGI moderno.	– (descartado)	5 — muy adecuado para I/O y APIs concurrentes.
Des.C: Documentación automática de API	0.30	2 — requiere extensiones/configuración para OpenAPI comparable.	– (descartado)	5 — OpenAPI/Swagger integrado de forma natural.
Puntaje total	1.00	3.10	Descartado	5.00

Autor del cuadro: Marco Montesdeoca

Este cuadro justifica la tecnología del servicio HTTP que expone entrenamiento y predicción. Los filtros imperativos exigen capacidad de exponer servicios HTTP, compatibilidad con Python y modelos de IA y adecuación a un servicio relativamente ligero centrado en predicción, criterio que penaliza soluciones “full stack” demasiado pesadas para un microservicio. Django REST queda descartado en el filtro de servicio ligero. Sobre Flask y FastAPI, los criterios deseables comparan simplicidad de implementación, rendimiento y documentación automática de la API; el resultado ponderado favorece FastAPI y deja a Flask en un segundo lugar, consolidando la decisión técnica adoptada en el backend del VFM.

A partir de la aplicación del modelo de toma de decisiones de Simon, se seleccionaron las tecnologías que mejor se ajustaron a las necesidades técnicas previstas para el VFM. Python fue seleccionado para inteligencia artificial y procesamiento de datos; Laravel para la aplicación web; PostgreSQL para la gestión estructurada de datos; Docker para garantizar reproducibilidad; y FastAPI para exponer los servicios de entrenamiento y predicción del modelo. El control de versiones se gestionó con Git, utilizando GitHub como repositorio remoto.

3.3.2 Fase 1: Análisis de requisitos

En la fase de análisis se identificaron los requisitos funcionales y no funcionales del sistema. Esta fase permitió transformar la problemática del proyecto en funcionalidades concretas que debían ser implementadas, coherentes con la pila tecnológica ya decidida (Apartado 3.3.1), con el enfoque RAD descrito arriba (prototipos acotados, validación temprana y refinamiento iterativo) y con el modelo de calidad ISO/IEC 25010:2011 (Anexo A).

Los requisitos identificados para el sistema VFM se documentan en las Tablas 43 y 44, con el formato **ID**, **Requerimiento** y **Descripción**.

Cuadro 43: Requisitos funcionales identificados para el sistema VFM

ID	Requerimiento	Descripción
RF1	Gestión de datos operacionales	El sistema debe permitir la carga, almacenamiento y consulta de datos operacionales de presión, temperatura y flujo en un esquema estructurado en base de datos.
RF2	Construcción del dataset	El sistema debe construir un dataset de entrenamiento a partir de variables de presión y temperatura por activo, en formato ancho apto para el modelo.
RF3	Entrenamiento de modelos	El sistema debe entrenar modelos de inteligencia artificial para estimar el flujo de gas asociado de cada activo.
RF4	Carga de archivos Excel	El sistema debe permitir la carga de archivos Excel para los procesos de entrenamiento y predicción.
RF5	Predicción de flujo	El sistema debe generar predicciones diarias de flujo de gas asociado (GAS) y valores agregados de GAS por periodo (p. ej. mensual), derivados de las entradas operacionales.
RF6	Visualización de resultados	El sistema debe visualizar resultados, métricas y gráficos en una interfaz web para el usuario.
RF7	Consulta y trazabilidad	El sistema debe permitir consultar las tablas de la base de datos y la trazabilidad del dataset desde el origen hasta el entrenamiento.

Autor del cuadro: Marco Montesdeoca

Cuadro 44: Requisitos no funcionales identificados para el sistema VFM

ID	Requerimiento	Descripción
RNF1	Modularidad	El sistema debe ser modular, separando la capa web, el backend de inteligencia artificial, la base de datos y el módulo de visualización.
RNF2	Reproducibilidad	El sistema debe ser reproducible mediante contenedores Docker, de modo que el entorno de ejecución sea consistente entre equipos.
RNF3	Trazabilidad de datos	La base de datos debe permitir la trazabilidad de las variables desde las tablas fuente hasta el dataset de entrenamiento.
RNF4	Usabilidad	La interfaz web debe ser comprensible para el usuario, facilitando la carga de datos, el entrenamiento y la consulta de resultados.
RNF5	Procesamiento en backend	El backend debe permitir el procesamiento de datos y el entrenamiento de modelos de forma eficiente y estable.
RNF6	Escalabilidad	El sistema debe poder ampliarse con datos reales de campo en el futuro sin rediseñar la arquitectura base.
RNF7	Integración entre capas	El sistema debe integrar la capa web, el backend de inteligencia artificial y la base de datos mediante servicios HTTP estandarizados.
RNF8	Confiabilidad y validación	El sistema debe ser verificado mediante pruebas unitarias, de integración, de caja blanca y de caja negra antes de su uso operativo.

Autor del cuadro: Marco Montesdeoca

3.3.3 Fase 2: Diseño

Arquitectura del sistema

La solución se basó en una arquitectura cliente–servidor orientada a servicios API REST, organizada en tres capas: presentación, lógica de negocio y datos. El cliente correspondió a la aplicación web ejecutada en el navegador del usuario, implementada con Laravel, desde donde se realizaron las operaciones de consulta, entrenamiento de modelos y predicción de gas asociado.

El servidor se compuso de dos niveles coordinados: un componente en Laravel que recibía las solicitudes, validaba la sesión y actuaba como intermediario proxy, y un servicio en Python FastAPI que procesaba las peticiones, ejecutaba los modelos de machine learning del VFM con TensorFlow/Keras y devolvía las predicciones y

métricas. La información se centralizó en la capa de datos mediante almacenamiento en PostgreSQL. Esta separación garantizó que la interfaz permaneciera ligera, que la carga computacional más intensa recayera en el servicio de inferencia en Python y que los artefactos del modelo pudieran actualizarse sin modificar la capa de presentación.

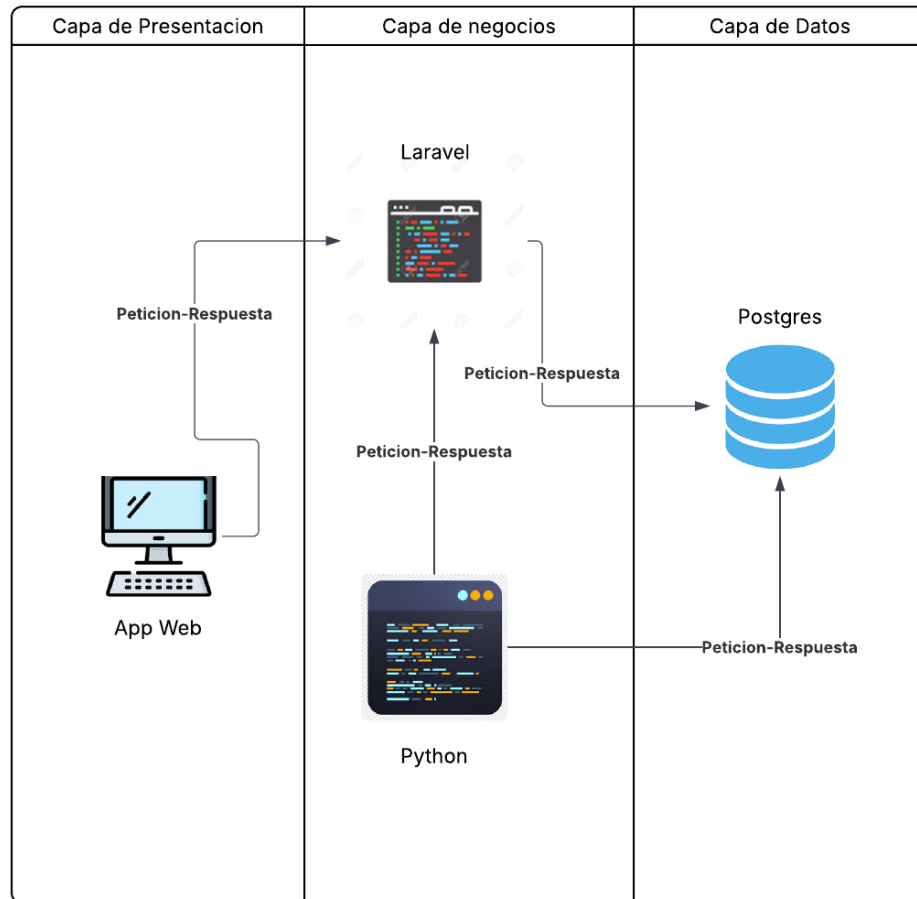


Figura 8: Arquitectura del sistema VFM

Diagrama de caso de uso

La Figura 9 representa la interacción principal entre el **Operador/Ingeniero** y la **aplicación VFM** (implementada con Laravel y FastAPI). Desde la perspectiva funcional, el usuario pudo: (i) consultar el estado de la base de datos, (ii) cargar datos de entrenamiento desde Excel hacia PostgreSQL, (iii) entrenar el modelo de predicción de gas, (iv) ejecutar inferencias, (v) visualizar en el dashboard la comparación entre valores medidos y predichos, (vi) descargar resultados en formato Excel y (vii) revisar métricas y modelos disponibles. En conjunto, estos casos de uso describieron el flujo

operativo completo del sistema, desde la preparación de datos hasta la explotación de resultados para análisis y toma de decisiones.

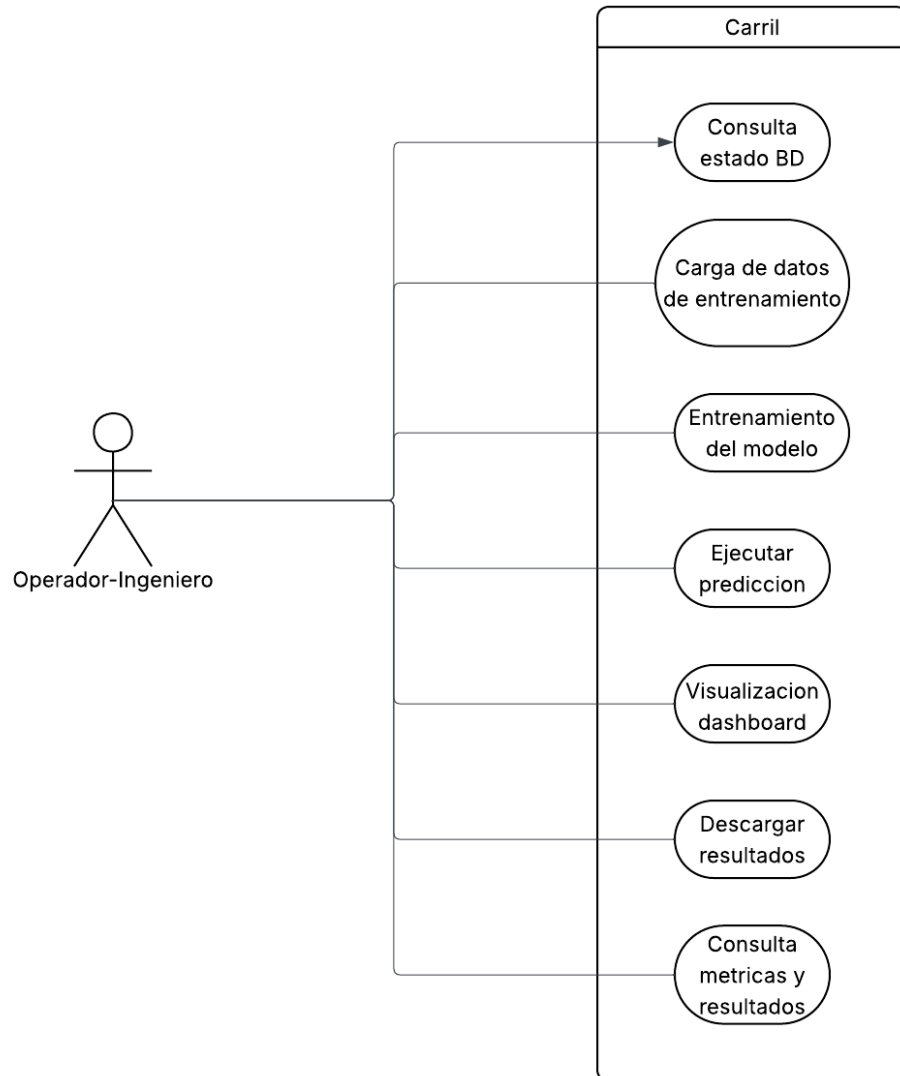


Figura 9: Diagrama de caso de uso

Diseño de interfaces y mockups

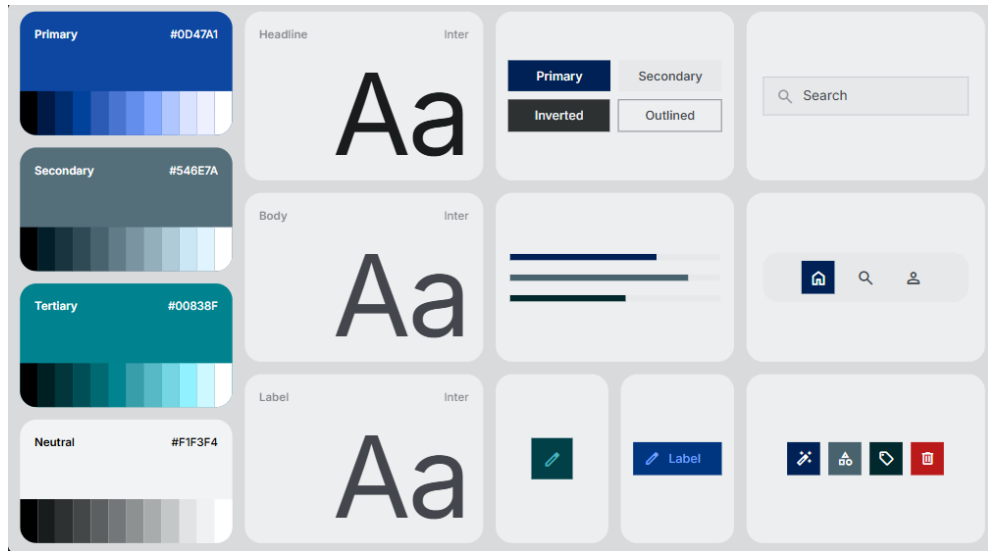


Figura 10: Guía de estilo visual: paleta de colores, tipografía y componentes reutilizables del sistema VFM

Como parte de la fase de diseño, se elaboraron mockups de las principales pantallas de la aplicación web. Estos prototipos permitieron definir la estructura visual, la distribución de componentes y la navegación del sistema antes de iniciar la construcción.

Los mockups fueron utilizados como guía para la fase de construcción de la interfaz en Laravel, permitiendo mantener coherencia entre los módulos de dashboard, entrenamiento, predicción y gestión de modelos. Además, facilitaron la identificación de los elementos necesarios para que el usuario pudiera cargar datos, ejecutar procesos de entrenamiento, generar predicciones y visualizar resultados.

La Figura 10 definió la base visual de la interfaz. Se establecieron cuatro familias cromáticas primario, secundario, terciario y neutro con escalas tonales para fondos, texto y estados de interacción. La tipografía *Inter* organizó la jerarquía en titulares, cuerpo y etiquetas. Además, se especificaron botones (primario, secundario, invertido y contorno), barra de búsqueda, barra de navegación con iconos de inicio, búsqueda y perfil, iconografía funcional (edición, automatización, categorías, etiquetas y eliminación) y barras de progreso. Este catálogo garantizó coherencia entre pantallas y redujo retrabajo en la implementación con Laravel.

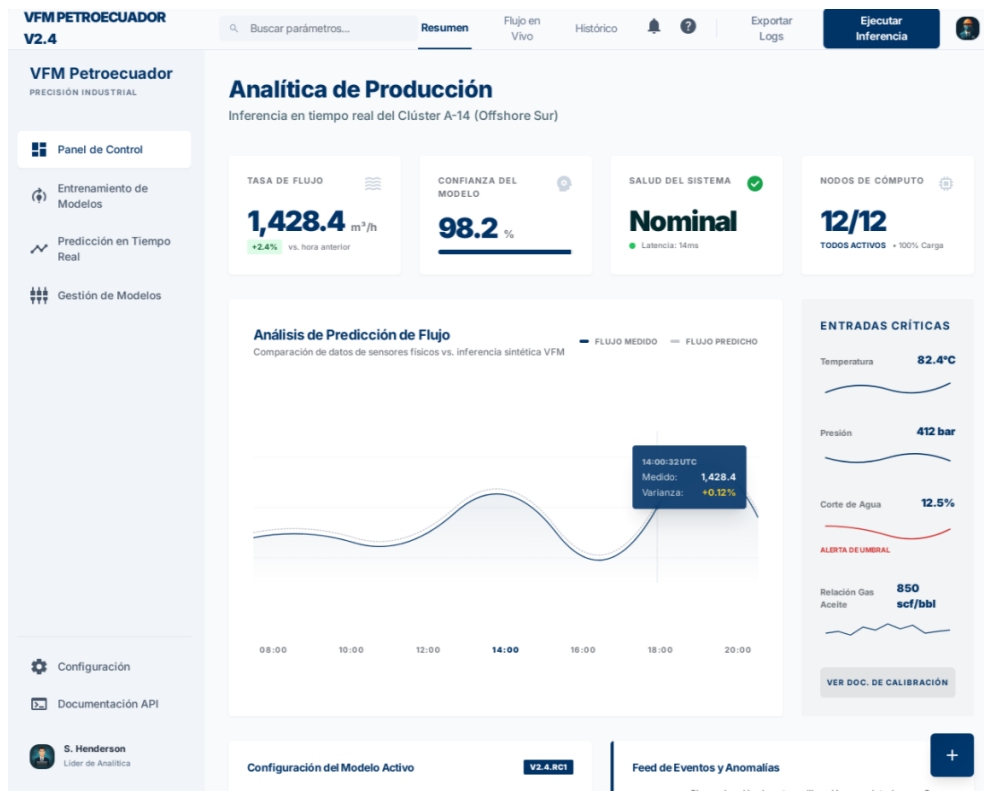


Figura 11: Mockup del dashboard principal del sistema VFM

La Figura 11 presentó el panel de control como vista de entrada al sistema. Incluía menú lateral, barra superior con búsqueda de parámetros, pestañas de resumen, flujo en vivo e histórico, y acciones de exportación de registros y ejecución de inferencia. El área central mostraba indicadores de tasa de flujo, confianza del modelo, salud del sistema y nodos de cómputo; un gráfico comparaba flujo medido frente a flujo predicho por el VFM; y un panel lateral listaba entradas operacionales críticas con alertas de umbral. Esta pantalla concentraba la supervisión operativa y la validación visual del desempeño del modelo.

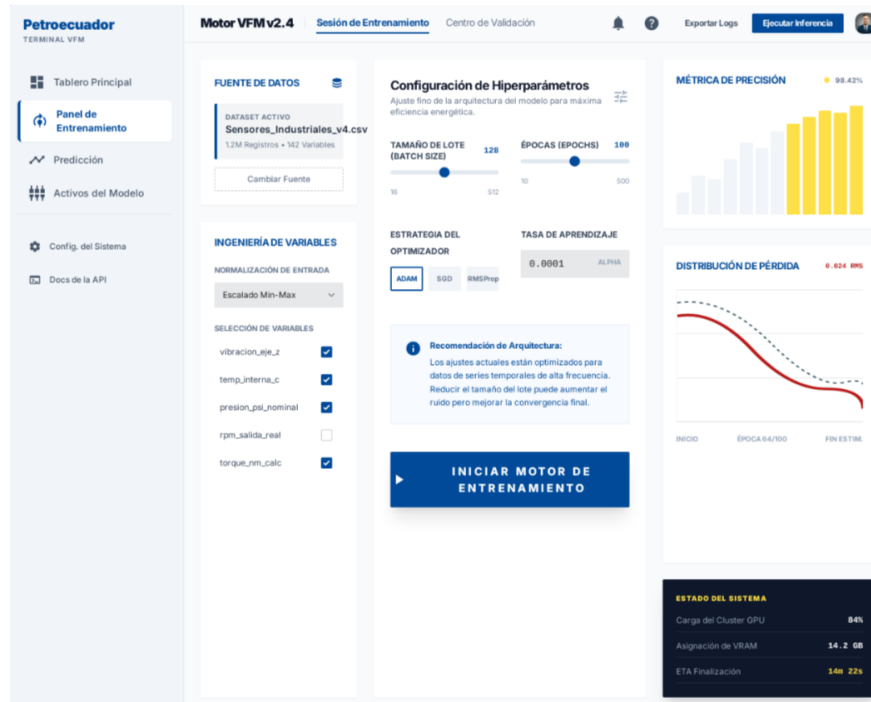


Figura 12: Mockup del panel de entrenamiento del modelo

La Figura 12 correspondió al módulo de entrenamiento. En la columna izquierda se configuraba la fuente de datos, el método de normalización y la selección de variables de sensor. La zona central permitía ajustar hiperparámetros —tamaño de lote, número de épocas, optimizador y tasa de aprendizaje— y mostraba recomendaciones de arquitectura según el tipo de serie temporal. El botón principal iniciaba el motor de entrenamiento. La columna derecha monitorizaba en tiempo real la precisión alcanzada, la curva de pérdida y el estado del sistema (carga de GPU, memoria y tiempo estimado de finalización). Este diseño materializó los requisitos RF2, RF3 y RF4: construcción del dataset, entrenamiento del modelo y carga de datos desde archivos estructurados.

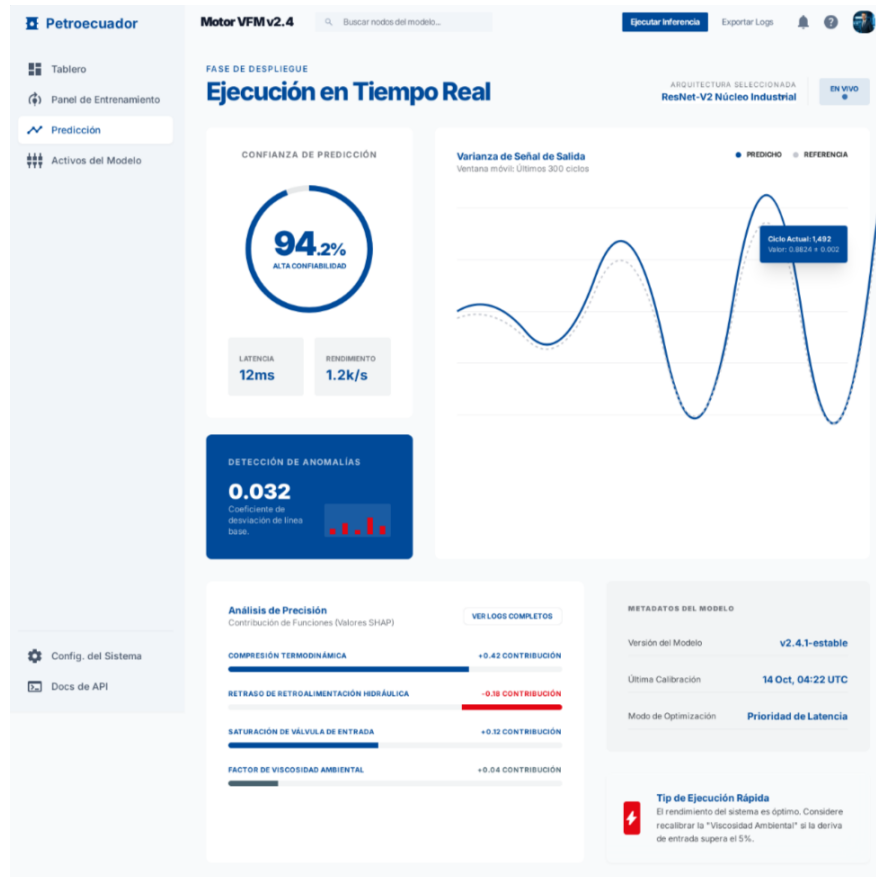


Figura 13: Mockup del módulo de predicción del flujo de gas asociado

La Figura 13 ilustró la pantalla de predicción en tiempo real. Se visualizaba la confianza de la inferencia, la latencia y el rendimiento del servicio; un gráfico central contrastaba la señal predicha con la referencia; y paneles auxiliares reportaban detección de anomalías, contribución de variables y metadatos del modelo activo (versión, última calibración y modo de optimización). El indicador de estado EN VIVO y el acceso rápido a ejecutar inferencia reforzaron el uso operativo del VFM durante el despliegue. Esta vista dio soporte al RF5 (generación de predicciones) y al RF6 (visualización de resultados y métricas).

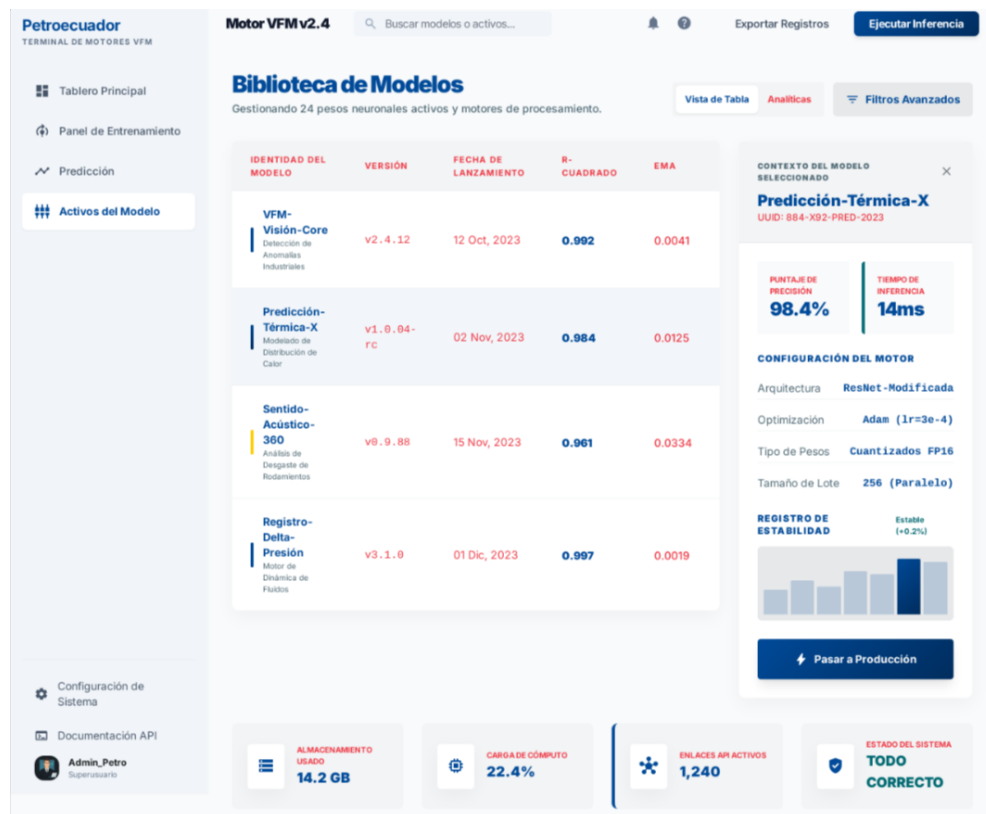


Figura 14: Mockup del módulo de gestión de modelos y activos

La Figura 14 definió la biblioteca de modelos y activos. Una tabla listaba los modelos disponibles con versión, fecha de liberación y métricas de desempeño, además del R^2 y el error medio móvil. Al seleccionar un registro, el panel derecho mostraba identificador, precisión, tiempo de inferencia, configuración del motor (arquitectura, optimizador, tipo de pesos y tamaño de lote), registro de estabilidad y la acción de pasar el modelo a producción. En la parte inferior, tarjetas resumían almacenamiento, carga de cómputo, enlaces API activos y estado general del sistema. Este módulo cubrió la trazabilidad y el control de versiones exigidos por RF7 y RNF3, y facilitó la comparación entre modelos entrenados antes de usarlos en predicción.

3.3.4 Fase 3: Construcción

La fase de construcción materializó en código ejecutable lo definido en la Fase 2 (Diseño). A partir de la arquitectura cliente-servidor de la Figura 8 —presentación en Laravel, lógica de inferencia en FastAPI y persistencia en PostgreSQL— se implementó

cada capa sin mezclar responsabilidades: la interfaz web solo orquestaba solicitudes y mostraba resultados, mientras el procesamiento de datos, el entrenamiento con TensorFlow/Keras y la predicción del flujo de gas asociado permanecían en el servicio Python. En paralelo, los mockups y la guía de estilo (Figuras 10 a 14) se tradujeron en vistas Blade con la misma estructura de navegación: panel de control, entrenamiento, predicción en tiempo real y gestión de modelos. El desarrollo técnico siguió el enfoque RAD de la metodología adoptada: prototipos acotados por módulo (datos, API, modelo e interfaz), revisión funcional y refinamiento antes de cerrar la implantación.

Los apartados que siguen describieron cómo cada decisión de diseño se concretó en el repositorio del VFM: infraestructura Docker y base de datos, pipeline del dataset, núcleo de inteligencia artificial, integración HTTP entre Laravel y FastAPI, y pantallas finales alineadas con los requisitos funcionales y no funcionales levantados en la Fase 1.

A) Entrenamiento y predicción del modelo de inteligencia artificial.

En el backend de inteligencia artificial se concentró la lógica de aprendizaje y estimación del flujo de gas asociado:

- **Entrenamiento con validación cruzada:** se entrenó el modelo mediante K-Fold, se registraron métricas por pliegue (RMSE, MAE, R^2 , MAPE) y se conservó la mejor versión mediante *early stopping* y regularización.
- **Red neuronal:** se modelaron las relaciones entre presión y temperatura por activo y el flujo de gas asociado; los pesos entrenados y los escaladores quedaron guardados para reutilización.
- **Predicción:** se prepararon las entradas operacionales, se ejecutó el modelo activo y se obtuvieron estimaciones diarias de GAS por activo, con posibilidad de consolidar el flujo por periodo (consolidación mensual a partir de las predicciones diarias).
- **Servicios de entrenamiento y predicción:** el usuario pudo alimentar el proceso desde la base de datos o desde un archivo Excel; el sistema devolvió métricas de desempeño y un libro de resultados descargable.

- **Monitoreo del servicio:** se consultó el estado del backend, las versiones de modelos disponibles y el desempeño por activo para alimentar el dashboard y la gestión operativa.

El código fuente del módulo de entrenamiento, de la red neuronal, del servicio de predicción y de las rutas del backend se consigna en el [Anexo C](#) (Figuras 24 a 26).

B) Integración entre la aplicación web y el backend de IA.

La aplicación web actuó como capa de presentación y enlace con el backend de inteligencia artificial, sin duplicar la lógica de aprendizaje automático:

- **Coordinación de módulos:** desde la interfaz se centralizaron las operaciones de dashboard, entrenamiento, predicción, consulta de modelos y verificación del estado de la base de datos.
- **Comunicación con el backend:** se enviaron archivos Excel al servicio de IA y se gestionaron tiempos de espera más amplios para el entrenamiento que para las consultas rápidas de estado del sistema.
- **Validación de entradas:** se verificó el formato de los archivos cargados y la correcta configuración de la conexión entre la aplicación web y el backend en el entorno de despliegue.
- **Optimización de consultas:** las respuestas de salud del sistema y de inventario de modelos se reutilizaron de forma temporal para evitar saturar el backend en cada actualización del panel.
- **Persistencia de resultados:** las predicciones generadas se almacenaron en el servidor y quedaron disponibles para su descarga posterior por el usuario.

C) Interfaz web y módulos funcionales.

A partir de los mockups de la Fase 2 se implementaron las pantallas que materializaron los flujos de usuario del VFM:

- **Dashboard:** mostró el estado general del sistema, los modelos disponibles y el acceso a métricas de desempeño (Figura 11).
- **Entrenamiento:** permitió cargar datos desde Excel o desde la base de datos, ejecutar el proceso de entrenamiento y visualizar métricas y curvas de pérdida al finalizar (Figuras 12 y 16).
- **Predicción:** habilitó la carga de entradas operacionales, la ejecución de inferencias y la descarga del libro de resultados (Figuras 13 y 17).
- **Gestión de modelos:** presentó el listado de modelos entrenados, sus métricas por activo y la trazabilidad de versiones (Figura 14).
- **Navegación común:** unificó menú lateral, estilos de la guía visual y mensajes informativos o de error para todo el sistema.

A continuación se presentaron las evidencias de implementación: el entorno contenedorizado y las pantallas finales del sistema, construidas a partir de los mockups de la Fase 2.

The screenshot shows a Docker terminal window with a dark theme. On the left, there's a sidebar with a search bar and a list of services: 'synthetic-da...' (teal icon), 'postgres' (red icon), and 'postgres:17' (green icon). The main area displays the logs for the 'postgres:17' container. The logs show PostgreSQL 17.9 startup on a Debian 12.0-59 system, listening on IPv4 address '0.0.0.0', port 5432, and a Unix socket. It also shows checkpoint logs with details like buffers, WAL files, and sync times. At the bottom, it says 'PostgreSQL Database directory appears to contain a database; Skipping initialization'.

Figura 15: Servicios Docker utilizados en la implementación del sistema

La Figura 15 documenta el despliegue local del stack. Se observa el contenedor `postgres:17` mapeado al puerto `5433:5432` y los servicios de persistencia y

backend del proyecto. Los registros del terminal confirman el arranque de PostgreSQL, la recuperación automática tras una interrupción y el mensaje *database system is ready to accept connections*, lo que valida que la capa de persistencia quedó operativa antes de ejecutar entrenamientos y predicciones (RNF2).

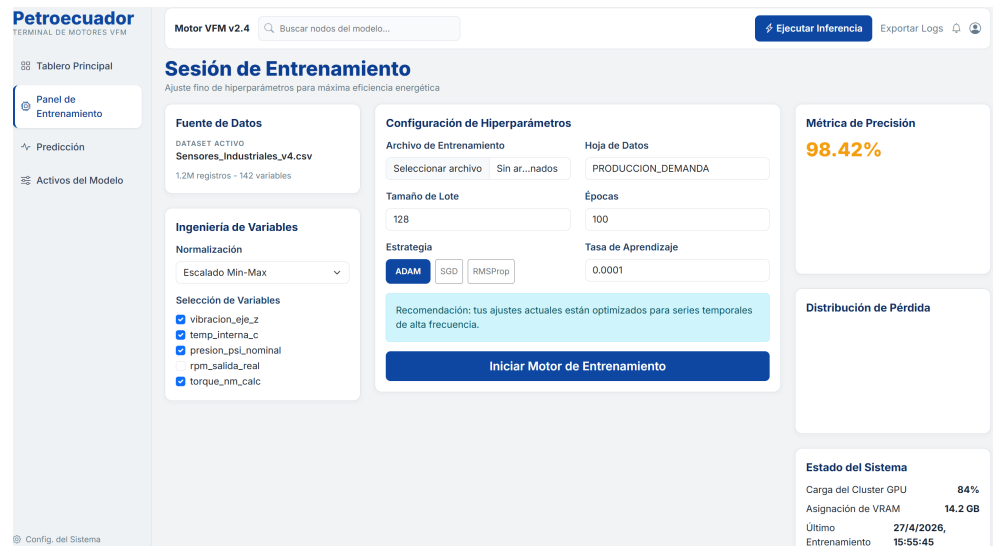


Figura 16: Interfaz entrenamiento del sistema VFM

La Figura 16 mostró la pantalla final del **panel de entrenamiento**, alineada con el mockup de la Figura 12. El usuario seleccionó el archivo y la hoja de cálculo (PRODUCCION_DEMANDA) o, de forma equivalente, cargó los datos desde las tablas históricas en PostgreSQL, definió hiperparámetros (tamaño de lote, épocas, optimizador ADAM y tasa de aprendizaje) e inició el motor de entrenamiento. En el panel derecho se reportaron la métrica de precisión alcanzada, el espacio reservado para la curva de pérdida y el estado del sistema con marca de tiempo del último entrenamiento. Esta vista materializó el flujo de entrenamiento descrito en el apartado C y cubrió RF3 y RF4.

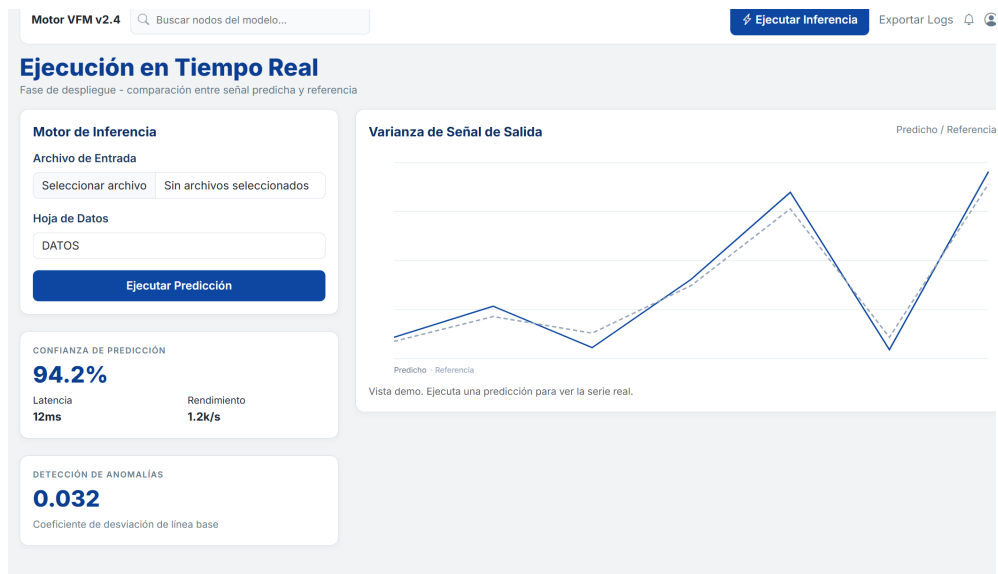


Figura 17: Interfaz predicción del sistema VFM

La Figura 17 correspondió al módulo de **predicción en tiempo real**. Permitted cargar el archivo de entrada, indicar la hoja DATOS y ejecutar la predicción mediante el botón principal; adicionalmente, el mismo flujo pudo tomar registros operacionales directamente desde la base de datos PostgreSQL para inferencia sin carga manual de archivos. Se exhibieron indicadores de confianza (94,2 %), latencia, rendimiento del servicio y un coeficiente de detección de anomalías. El gráfico central contrastó la señal *predicha* frente a la *referencia*, facilitando la validación visual del modelo en fase de despliegue. La pantalla implementó el flujo de predicción y satisfizo RF5.

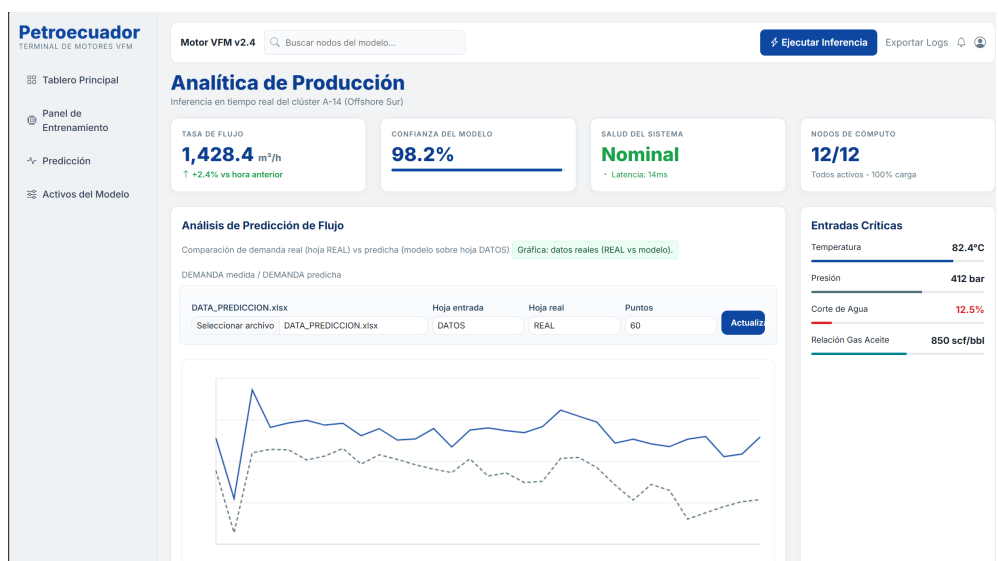


Figura 18: Interfaz web desarrollada para el sistema VFM

La Figura 18 presentó el **tablero principal de analítica de producción**, coherente con el mockup del dashboard (Figura 11). En la parte superior se mostraron tarjetas de tasa de flujo, confianza del modelo, salud del sistema y nodos de cómputo. El área central permitió elegir el libro `DATA_PREDICCION.xlsx`, las hojas de entrada y de datos reales, el número de puntos a graficar y actualizar la serie; además, habilitó la consulta de series históricas almacenadas en PostgreSQL para construir los mismos gráficos sin depender únicamente de carga manual.

El gráfico comparó el gas asociado real frente al gas asociado predicho (GAS). El panel lateral listó entradas críticas —temperatura, presión, corte de agua y relación gas—aceite— con resaltado de alertas. Esta vista integró consulta, visualización y contraste de resultados (RF5 y RF6) desde un único punto de acceso para el usuario operativo.

3.3.5 Fase 4: Implementacion

La fase de implantación comprendió el despliegue del sistema, la verificación integral de su funcionamiento y la validación frente a los requisitos definidos. Con el objetivo de garantizar la confiabilidad y estabilidad del Medidor Virtual de Flujo (VFM), se definió una estrategia de pruebas basada en diferentes niveles y enfoques de validación. Esta fase permitió evaluar tanto la lógica interna del sistema como el comportamiento observable desde la perspectiva del usuario final, asegurando que los módulos desarrollados en la construcción operaran de manera correcta e integrada en el entorno objetivo (Docker). Se consideró el **cierre de entregables** (salidas descargables, evidencias y reproducibilidad del flujo) como parte de la puesta en operación del producto refinado en los ciclos RAD de construcción.

Para ello, se aplicaron pruebas unitarias, pruebas de integración, pruebas de caja blanca y pruebas de caja negra, debido a que estas permitieron analizar el sistema desde distintos niveles de complejidad, validando componentes individuales, comunicación entre módulos, comportamiento interno del código y funcionamiento funcional del sistema.

Clasificación general de las pruebas

- **Pruebas unitarias:** permitieron validar funciones o componentes individuales del sistema de manera aislada.
- **Pruebas de integración:** verificaron la correcta comunicación e intercambio de datos entre los diferentes módulos del sistema.
- **Pruebas de caja blanca:** evaluaron la lógica interna del sistema con conocimiento de la implementación del código.
- **Pruebas de caja negra:** validaron el comportamiento observable del sistema desde la perspectiva del usuario, sin analizar el código fuente.

Pruebas unitarias. Las pruebas unitarias fueron implementadas principalmente en Python para validar la lógica interna del backend de inteligencia artificial y del módulo de preparación de datos. Estas pruebas permitieron verificar que los componentes individuales del sistema funcionaran correctamente antes de ser integrados con el resto de la aplicación.

En el sistema VFM, las pruebas unitarias se enfocaron en funciones relacionadas con el procesamiento del dataset, validación de columnas, carga y transformación de datos operacionales, cálculo de correlaciones y predicción del modelo.

Objetivos de las pruebas unitarias

- Verificar la normalización y validación de columnas del dataset.
- Validar el comportamiento de la función `predict_dataframe()` frente a datos válidos e inválidos.
- Comprobar el funcionamiento de la validación cruzada implementada en `train_model_cv()`.
- Verificar la correcta construcción del dataset ancho y la exportación de archivos de salida.

- Validar el cálculo de correlaciones entre variables operacionales.

Comando de ejecución

```
python -m tests.python.test_app_prediction_api -v
```

Cuadro 45: Pruebas unitarias aplicadas al sistema VFM

Componente evaluado	Prueba realizada	Resultado esperado
Carga de datos operacionales	Verificar la ingesta de registros históricos para todos los activos definidos.	La base debe contener datos completos y consistentes para el entrenamiento del modelo.
Construcción del dataset	Validar que el dataset contenga las columnas requeridas por el modelo.	El dataset debe contener variables de presión, temperatura y flujo.
Predicción del modelo	Ejecutar la función <code>predict_dataframe()</code> con datos válidos.	El modelo debe retornar predicciones numéricas sin errores.
Validación cruzada	Comprobar el entrenamiento mediante K-Fold.	El sistema debe seleccionar el mejor fold del entrenamiento.
Matriz de correlación	Verificar el cálculo de correlaciones de Pearson.	Deben generarse correctamente las matrices de correlación.
Exportación de archivos	Validar generación de archivos Excel y gráficos PNG.	Los archivos deben generarse correctamente y ser accesibles.

Autor del cuadro: Marco Montesdeoca

Estas pruebas permitieron confirmar que los componentes internos del sistema funcionaron correctamente de manera individual y generaron resultados coherentes para el entrenamiento y predicción del modelo.

```
PS C:\Users\Felipe\Desktop\Proyectos\TesisComposerV2> python -m tests.python.test_app_prediction_api -v
test_column_lookup_normalizes_spaces_case_and_underscores (__main__.AppPredictionApiTest.test_column_lookup_normalizes_spaces_case_and_underscores) ... ok
test_predict_dataframe_generates_demanda_and_reference_sum (__main__.AppPredictionApiTest.test_predict_dataframe_generates_demanda_and_reference_sum) ... ok
test_predict_dataframe_rejects_missing_required_columns (__main__.AppPredictionApiTest.test_predict_dataframe_rejects_missing_required_columns) ... ok
test_train_model_cv_selects_and_saves_lowest_mse_fold (__main__.AppPredictionApiTest.test_train_model_cv_selects_and_saves_lowest_mse_fold) ... 2026-05-06 22:54:29,343 - app_prediction_api - INFO - Iniciando entrenamiento con K-Fold para activo: AU
2026-05-06 22:54:29,343 - app_prediction_api - INFO - Fold 1/5 - Train size: 8, Val size: 2
2026-05-06 22:54:29,343 - app_prediction_api - INFO - Fold 1 - MSE: 9.0000, MAE: 3.0000, R²: -17.0000
2026-05-06 22:54:29,344 - app_prediction_api - INFO - Fold 2/5 - Train size: 8, Val size: 2
2026-05-06 22:54:29,344 - app_prediction_api - INFO - Fold 2 - MSE: 4.0000, MAE: 2.0000, R²: -7.0000
2026-05-06 22:54:29,344 - app_prediction_api - INFO - Fold 3/5 - Train size: 8, Val size: 2
2026-05-06 22:54:29,344 - app_prediction_api - INFO - Fold 3 - MSE: 0.0000, MAE: 0.0000, R²: 1.0000
2026-05-06 22:54:29,344 - app_prediction_api - INFO - Fold 4/5 - Train size: 8, Val size: 2
2026-05-06 22:54:29,344 - app_prediction_api - INFO - Fold 4 - MSE: 16.0000, MAE: 4.0000, R²: -31.0000
2026-05-06 22:54:29,344 - app_prediction_api - INFO - Fold 5/5 - Train size: 8, Val size: 2
2026-05-06 22:54:29,344 - app_prediction_api - INFO - Fold 5 - MSE: 25.0000, MAE: 5.0000, R²: -49.0000
2026-05-06 22:54:29,345 - app_prediction_api - INFO - Modelo guardado para AU: C:\Users\Felipe\AppData\Local\Temp\tmp5davnjsa\AU_model.keras
ok
ok

Ran 4 tests in 0.341s

OK
```

Figura 19: Ejecución de pruebas unitarias del backend de predicción

Pruebas funcionales y de caja negra. Las pruebas funcionales fueron desarrolladas desde el enfoque de caja negra, evaluando el comportamiento observable del sistema desde la perspectiva del usuario final. Estas pruebas permitieron validar que las funcionalidades implementadas respondieran correctamente ante diferentes escenarios de uso.

Las pruebas funcionales se ejecutaron principalmente sobre la aplicación web desarrollada en Laravel, validando rutas, formularios, carga de archivos y generación de resultados.

Casos funcionales verificados

- Carga de archivos Excel para entrenamiento.
- Flujo completo de predicción desde la interfaz web.
- Rechazo de archivos con formatos no permitidos.
- Descarga de resultados generados por el sistema.
- Visualización de métricas y gráficos del dashboard.

Comando de ejecución

```
php artisan test
```

Cuadro 46: Pruebas de caja negra aplicadas al sistema VFM

Funcionalidad evaluada	Entrada o acción del usuario	Resultado esperado
Acceso al dashboard	Ingresar a la página principal del sistema.	El sistema debe mostrar correctamente métricas y gráficos.
Carga de archivo de entrenamiento	Subir un archivo Excel válido.	El sistema debe aceptar el archivo e iniciar el entrenamiento.
Carga de archivo inválido	Subir archivos con formato incorrecto.	El sistema debe rechazar el archivo y mostrar un error.
Predicción de flujo	Subir datos de presión y temperatura.	El sistema debe generar predicciones correctamente.
Descarga de resultados	Solicitar descarga del archivo generado.	El sistema debe entregar un archivo Excel descargable.
Consulta de modelos	Ingresar al módulo de modelos entrenados.	El sistema debe mostrar modelos y métricas disponibles.

Autor del cuadro: Marco Montesdeoca

Estas pruebas permitieron validar que el sistema responde adecuadamente a las acciones del usuario y cumple con las funcionalidades esperadas.

```
PS C:\Users\Felipe\Desktop\Proyectos\TesisComposerV2> php artisan test

PASS Tests\Unit\ExampleTest
✓ that true is true

PASS Tests\Feature\ExampleTest
✓ the application returns a successful response

PASS Tests\Feature\PredictionControllerTest
✓ predict uploads excel to backend and stores download
✓ train uploads excel to backend and returns training result
✓ predict rejects non excel files before calling backend

Tests: 5 passed (19 assertions)
Duration: 0.36s
```

Figura 20: Ejecución de pruebas funcionales y de caja negra en la interfaz web

Pruebas de integración. Las pruebas de integración permitieron validar la comunicación entre los diferentes módulos que conforman el sistema VFM, incluyendo la aplicación web en Laravel, el backend de inteligencia artificial desarrollado en Python/FastAPI, la base de datos PostgreSQL y los archivos de entrada/salida.

Debido a que el sistema está compuesto por múltiples tecnologías integradas, este tipo de pruebas resultó fundamental para garantizar el flujo correcto de información entre componentes.

Alcance de las pruebas de integración

- Verificación de conectividad entre Laravel y la API de predicción.
- Validación de endpoints como /ping y /health.
- Confirmación del procesamiento de archivos Excel.
- Validación de almacenamiento y recuperación de resultados.
- Verificación del funcionamiento de PostgreSQL mediante Docker.

Comandos de ejecución

```
composer test:integration
```

```
$env:INTEGRATION_API_TESTS='1'
```

```
python tests/python/test_integration_prediction_api.py -v
```

Cuadro 47: Pruebas de integración aplicadas al sistema VFM

Módulos integrados	Prueba realizada	Resultado esperado
Docker + PostgreSQL	Levantar contenedores y verificar conexión.	La base de datos debe estar disponible y operativa.
Laravel + FastAPI	Enviar archivos desde la interfaz web.	La API debe recibir y procesar correctamente los datos.
FastAPI + Modelo IA	Ejecutar entrenamiento y predicción.	El sistema debe generar modelos y predicciones válidas.
Backend + Archivos Excel	Generar archivos de salida descargables.	Los archivos deben generarse correctamente.
API Python	Verificar endpoints HTTP.	Los endpoints deben responder correctamente.

Autor del cuadro: Marco Montesdeoca

Estas pruebas permitieron confirmar que todos los módulos del sistema trabajan de manera integrada y estable.

```

PS C:\Users\Felipe\Desktop\Proyectos\TesisComposerV2> composer test:integration
You are running Composer with SSL/TLS protection disabled.

[INFO] Configuration cache cleared successfully.

PHPUnit 11.5.55 by Sebastian Bergmann and contributors.

Runtime:      PHP 8.2.12
Configuration: C:\Users\Felipe\Desktop\Proyectos\TesisComposerV2\phpunit.integration.xml

....                                                4 / 4 (100%)

Time: 00:01.249, Memory: 36.00 MB

OK (4 tests, 13 assertions)

```

Figura 21: Ejecución de pruebas de integración entre módulos del sistema

Pruebas de caja blanca. Las pruebas de caja blanca se realizaron con conocimiento de la estructura interna del sistema, revisando directamente el comportamiento del código fuente y la lógica implementada.

Este enfoque se aplicó sobre funciones del backend, arquitectura de la red neuronal, validaciones del dataset y procesos de entrenamiento del modelo.

Cuadro 48: Pruebas de caja blanca aplicadas al sistema VFM

Elemento evaluado	Aspecto revisado	Resultado esperado
Validación de columnas	Revisión de identificación de variables requeridas.	El backend debe detectar correctamente las columnas necesarias.
Construcción del dataset	Transformación de datos para entrenamiento.	Las variables deben estructurarse correctamente.
Arquitectura MLP	Revisión de capas y entrenamiento del modelo.	El modelo debe entrenarse correctamente.
Validación cruzada	Aplicación de K-Fold en entrenamiento.	El sistema debe evaluar múltiples particiones del dataset.
Generación de gráficos	Revisión de funciones de correlación.	Los gráficos deben generarse sin errores.

Autor del cuadro: Marco Montesdeoca

El enfoque de caja blanca permitió validar que la lógica interna del sistema se encontraba alineada con los objetivos planteados en el proyecto.

Checklist operativo de validación. Con el fin de mantener una validación continua del sistema, se estableció el siguiente orden de ejecución de pruebas:

1. `php artisan test`
2. `composer test:integration`
3. `python tests/python/test_integration_prediction_api.py`
`-v`

Cuadro 49: Resumen general de pruebas del sistema VFM

Tipo de prueba	Propósito	Aplicación en el sistema VFM
Pruebas unitarias	Validar componentes individuales.	Funciones internas del backend y dataset.
Pruebas de integración	Verificar comunicación entre módulos.	Laravel, FastAPI, PostgreSQL y Docker.
Caja blanca	Analizar lógica interna del sistema.	Entrenamiento, validaciones y procesamiento.
Caja negra	Validar comportamiento desde el usuario.	Carga de archivos, predicción y descarga.

Autor del cuadro: Marco Montesdeoca

Resumen de pruebas aplicadas.

Conclusión de la estrategia de pruebas. La aplicación de pruebas unitarias, funcionales, de integración, caja blanca y caja negra permitió validar el correcto funcionamiento del sistema de Medidor Virtual de Flujo (VFM) en diferentes niveles de análisis.

Las pruebas realizadas permitieron detectar errores tempranos, validar la lógica del modelo de inteligencia artificial, comprobar la comunicación entre módulos y asegurar que los flujos principales del sistema funcionaran correctamente de extremo a extremo.

Además, se confirmó que el sistema fue capaz de procesar archivos de entrada, entrenar modelos de predicción, generar resultados y presentar información de manera adecuada para el usuario final, garantizando así la confiabilidad y estabilidad de la solución desarrollada.

Cuadro 50: Pruebas realizadas en la implantación del sistema VFM

Prueba	Resultado esperado	Estado
Creación de base PostgreSQL	Tablas creadas correctamente	Cumplido
Construcción del dataset	Columnas requeridas disponibles	Cumplido
Exportación Excel/CSV	Archivos generados correctamente	Cumplido
Matriz de correlación	Gráficos generados en PNG	Cumplido
Entrenamiento	Modelos generados por activo	Cumplido
Predicción	Estimación de flujo de gas	Cumplido
Visualización	Resultados visibles en interfaz	Cumplido

Autor del cuadro: Marco Montesdeoca

3.3.6 Resultados

Pipeline del sistema

La Figura 22 resumió de extremo a extremo el flujo de datos y modelos del VFM para el Campo Pucuna (activo P). El proceso inició con la **ingesta** de registros operacionales desde archivos Excel hacia PostgreSQL; continuó con el control de **calidad** de datos; incorporó **ingeniería de características** (rezagos, ventanas móviles y estacionalidad); ejecutó el **entrenamiento** del MLP con validación cruzada temporal K-Fold; expuso la **predicción** de gas asociado; y culminó en la publicación de **resultados** en base de datos, libros Excel y dashboard.

Las entradas del modelo fueron petróleo, agua, presión y temperatura; la salida principal fue el flujo de gas (GAS). La capa de presentación (Laravel), el servicio de inteligencia artificial (FastAPI) y la persistencia (PostgreSQL) se mantuvieron separadas, de acuerdo con la arquitectura definida en la Fase 2, lo que permitió trazabilidad del dato y reentrenamiento del modelo sin rediseñar la interfaz de usuario.



Figura 22: Pipeline del sistema VFM

Requerimientos funcionales y no funcionales

Los requerimientos del sistema VFM se clasificaron en funcionales (RF), que describían *qué* debía hacer el sistema, y no funcionales (RNF), que especificaban *cómo* debía comportarse en términos de calidad, arquitectura y despliegue, en línea con ISO/IEC 25010:2011 ([Anexo A](#)). Las Tablas 51 y 52 detallaron cada requerimiento, su evidencia de cumplimiento y el estado al cierre de la implementación.

Cuadro 51: Matriz de requerimientos funcionales del sistema VFM

N°	Tipo	Requerimiento	Evidencia de cumplimiento	Estado
RF1	RF	Gestionar datos operacionales (presión, temperatura y flujo): carga, almacenamiento y consulta en un esquema estructurado.	Carga desde archivos Excel, tablas PostgreSQL y archivo histórico.	Cumplido
RF2	RF	Construir un dataset de entrenamiento a partir de variables de presión y temperatura por activo.	Tabla <code>training_dataset</code> , pivote ancho (<code>*_P</code> , <code>*_A</code>) y <code>dataset_feature_map</code> .	Cumplido
RF3	RF	Entrenar modelos de inteligencia artificial para estimar el flujo de gas asociado.	API FastAPI (<code>/train</code>), modelos MLP por activo con validación K-Fold y TensorFlow/Keras.	Cumplido
RF4	RF	Permitir la carga de archivos Excel para entrenamiento y predicción.	Módulos Laravel de entrenamiento y predicción; endpoints <code>/train</code> y <code>/predict</code> .	Cumplido
RF5	RF	Generar predicciones diarias de flujo de gas asociado (GAS) y valores agregados de GAS por periodo.	Salidas <code>*_D</code> , GAS y exportación de resultados Excel desde el backend.	Cumplido
RF6	RF	Visualizar resultados, métricas y gráficos en una interfaz web.	Dashboard Laravel, gráficos comparativos con Chart.js y capturas de entrenamiento/predicción.	Cumplido
RF7	RF	Consultar tablas de la base de datos y la trazabilidad del dataset desde el origen hasta el entrenamiento.	Módulo de gestión de modelos, <code>dataset_feature_map</code> , consultas a PostgreSQL y control de versiones de modelos entrenados.	Cumplido

Autor del cuadro: Marco Montesdeoca

Cuadro 52: Matriz de requerimientos no funcionales del sistema VFM

N°	Tipo	Requerimiento	Evidencia de cumplimiento	Estado
RNF1	RNF	Modularidad: separar capa web, API de IA, base de datos y módulo de visualización.	Arquitectura distribuida en cuatro componentes principales (Figura 8).	Cumplido
RNF2	RNF	Reproducibilidad del entorno mediante contenedores Docker.	Servicios Docker para PostgreSQL e inicialización automatizada de datos (Figura 15).	Cumplido
RNF3	RNF	Trazabilidad de variables desde tablas fuente hasta el dataset de entrenamiento.	dataset_feature_map, catálogo de activos y documentación completa del pipeline de datos.	Cumplido
RNF4	RNF	Usabilidad: interfaz comprensible para personal técnico y operativo.	Mockups y pantallas finales de entrenamiento, predicción y dashboard del sistema.	Cumplido
RNF5	RNF	Capacidad de procesamiento de datos y entrenamiento en el backend.	Backend Python/FastAPI con Pandas, StandardScaler y validación cruzada implementada.	Cumplido
RNF6	RNF	Escalabilidad para incorporar datos reales de campo en el futuro.	Esquema relacional extensible y carga de archivos Excel sin modificar la arquitectura base.	Cumplido
RNF7	RNF	Integración entre capas mediante servicios HTTP estandarizados.	Comunicación Laravel ↔ FastAPI y pruebas de integración (Tabla 47).	Cumplido
RNF8	RNF	Confiabilidad y validación del sistema antes de uso operativo.	Pruebas unitarias, de integración, caja blanca y caja negra (Tabla 49).	Cumplido

Autor del cuadro: Marco Montesdeoca

La Tabla 53 resume el nivel de cumplimiento por tipo de requerimiento. Todos los requerimientos planificados fueron implementados y verificados durante la fase de pruebas del sistema.

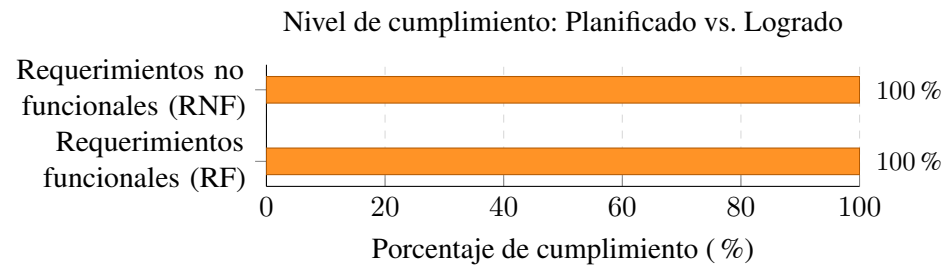
Cuadro 53: Resumen de cumplimiento de requerimientos del sistema VFM

Categoría	Total planificados	Cumplidos	Porcentaje (%)
Requerimientos funcionales (RF)	7	7	100
Requerimientos no funcionales (RNF)	8	8	100
Total general	15	15	100

Autor del cuadro: Marco Montesdeoca

En la Figura 23, el sistema alcanzó un cumplimiento del 100 % en ambas categorías:

Figura 23: Nivel de cumplimiento: planificado vs. logrado



los siete requerimientos funcionales (RF1–RF7) y los ocho no funcionales (RNF1–RNF8), validados mediante la implementación de los módulos descritos en las fases RAD de requisitos, diseño, construcción e implantación del capítulo de resultados.

4. CONCLUSIONES Y RECOMENDACIONES

4.1 Conclusiones

- Se identificaron las variables operacionales críticas de los sistemas y bases de datos históricas de Petroecuador mediante entrevistas ([Anexo B](#)), revisión bibliográfica de Virtual Flow Metering y análisis de las bases de datos históricas del campo Pucuna, y con ello se construyó el dataset de entrenamiento del modelo de inteligencia artificial. El conjunto resultante comprende 2 192 observaciones diarias sin valores faltantes, con GAS en torno a 5 562,95 MSCFD de media; el esquema relacional en PostgreSQL permite vincular cada variable del entrenamiento con su origen documentado y sostener la trazabilidad desde el registro de campo hasta el modelado supervisado.
- En el análisis de algoritmos de regresión Ridge, MLP, Random Forest y SVR evaluados con MAE, RMSE, R^2 , MAPE y mejora frente a un modelo *naive* sobre el 25 % de datos de prueba, el **MLP** obtuvo el mejor desempeño global: $R^2 = 97,02\%$, MAE = 0,13 %, RMSE = 0,18 %, MAPE = 0,13 % y mejora del 86,33 % respecto al *naive*. Ridge ocupó el segundo lugar ($R^2 = 93,55\%$), lo que indica que parte de la relación entre presión, temperatura, producción líquida y gas es aproximadamente lineal; Random Forest y SVR quedaron por debajo del 88 % de R^2 . La comparación día a día del Cuadro de predicciones confirma el ajuste fino del MLP en fechas representativas (por ejemplo, 18/12/2025: gas real 9 411,94 MSCFD, predicción 9 411,41 MSCFD). Con validación cruzada K-Fold, *early stopping* y regularización, se justifica el MLP como núcleo del VFM y se cumple el segundo objetivo específico.
- Se implementó un **sistema integral de Medidor Virtual de Flujo** basado en algoritmos de inteligencia artificial para la predicción del flujo de gas asociado en los flares del Campo Pucuna. Para su desarrollo se adoptó la metodología RAD, seleccionada mediante el modelo de Simon con puntaje ponderado de **4,70** frente a **3,25** del modelo en cascada, al exigir integración progresiva entre datos, inteligencia artificial, API e interfaz; la pila tecnológica se definió con el mismo

criterio: **Python** (5,00/5,00) y **FastAPI** (5,00) para procesamiento, entrenamiento e inferencia; **Laravel** (5,00) para la capa web; **PostgreSQL** (5,00) como motor relacional; y **Docker** (5,00) para reproducibilidad del entorno. Bajo la arquitectura de la Fase 2, se cubre la carga de datos desde Excel hacia PostgreSQL, la construcción del dataset, el entrenamiento por activo, la predicción diaria y su consolidación mensual, la exportación de resultados y la visualización en *dashboard*, con backend TensorFlow/Keras y separación entre la carga computacional del modelo y la presentación al usuario. Los módulos fueron verificados con pruebas unitarias, de integración, caja blanca y caja negra; los 15 requerimientos planificados alcanzaron **100 %** de cumplimiento (Tabla 53), lo que demuestra la viabilidad técnica de estimar el flujo sin depender exclusivamente de medidores físicos en cada punto y aporta una base de software replicable para una fase piloto con recalibración periódica del modelo.

- El sistema de Medidor Virtual de Flujo (VFM) desarrollado para el Campo Pucuna de Petroecuador durante el periodo académico 2026 unificó en una plataforma web la carga y trazabilidad de datos operacionales, el entrenamiento del modelo, la predicción del caudal de gas asociado y la visualización de resultados, sustituyendo estimaciones manuales y la dependencia exclusiva de medidores físicos en cada punto de medición, e integró un perceptrón multicapa como núcleo predictivo con $R^2 = 97,02 \%$ en el conjunto de prueba, verificado con 100 % de cumplimiento de los quince requerimientos planificados. El resultado es una herramienta operativa y analítica que no existía previamente en el activo y que responde directamente a la problemática de medición del gas asociado diagnosticada en el estudio, en cumplimiento del objetivo general de desarrollar un VFM basado en algoritmos de inteligencia artificial para la predicción del flujo de gas en los pozos de producción del campo Pucuna.

4.2 Recomendaciones

- Se recomienda implementar el sistema de forma progresiva, iniciando con un escenario piloto acotado y ampliando su uso una vez validada la estabilidad de

los datos, el desempeño del modelo y el funcionamiento integral del software en condiciones operativas reales.

- Conviene capacitar al personal técnico y de supervisión vinculado a la medición y al control de producción, mediante sesiones de formación práctica que faciliten la adopción del sistema y el uso adecuado de sus funcionalidades principales.
- Es pertinente evaluar periódicamente el comportamiento del sistema en operación, contrastando la calidad de la información, los resultados del modelo y los tiempos de respuesta con lo establecido en las pruebas de validación, a fin de orientar acciones de mejora continua; conviene revisar también la coherencia de balances mensuales obtenidos por agregación de las predicciones diarias.
- Se sugiere profundizar la línea de investigación sobre medición virtual de flujo mediante estudios que incorporen nuevos periodos de datos, escenarios operativos adicionales y enfoques metodológicos complementarios que fortalezcan la estabilidad y la generalización de las predicciones.
- Resulta conveniente establecer lineamientos de actualización, trazabilidad y documentación del modelo predictivo, de modo que cada versión quede respaldada con evidencia metodológica reproducible para trabajos académicos o de aplicación industrial posteriores.
- Sería oportuno explorar la aplicación del enfoque de medidor virtual en otros contextos industriales donde la medición directa sea limitada o costosa, siempre que exista una relación consistente entre variables operacionales observables y la magnitud que se desea estimar.
- Se aconseja considerar la arquitectura de integración entre datos, inteligencia artificial e interfaz de usuario como referencia para el desarrollo de otros sistemas que requieran entrega incremental, validación temprana con usuarios y toma de decisiones basada en información.
- Se recomienda ampliar el alcance del sistema implementado mediante el desarrollo de un modelo de inteligencia artificial orientado a la predicción del caudal de petróleo, reutilizando la infraestructura de datos, entrenamiento e inferencia

ya construida para el VFM de gas y aprovechando variables operacionales como presión, temperatura y producción líquida, con el fin de ofrecer una estimación integral de los flujos del activo y fortalecer la toma de decisiones operativas en el Campo Pucuna.

5. REFERENCIAS BIBLIOGRÁFICAS

- [1] L. S. Hansen, S. Pedersen, and P. Durdevic, “Multi-phase flow metering in offshore oil and gas transportation pipelines: Trends and perspectives,” *Sensors*, vol. 19, no. 9, p. 2184, 2019.
- [2] E. M. Wokoma, “Multi-phase flow metering in offshore oil and gas pipelines,” *International Journal of Petroleum and Gas Engineering Research*, vol. 7, no. 1, pp. 80–108, 2024.
- [3] T. Bismukhametov and J. Jäschke, “First principles and machine learning virtual flow metering: A literature review,” *Journal of Petroleum Science and Engineering*, vol. 184, p. 106487, 2020. [PDF] First Principles and Machine Learning Virtual Flow Metering A Literature Review.pdf.
- [4] B. Grimstad, M. Hotvedt, A. T. Sandnes, O. Kolbjørnsen, and L. S. Imsland, “Bayesian neural networks for virtual flow metering: An empirical study,” *arXiv preprint*, 2022. Preprint.
- [5] G. S. Rao, K. V. L. Narayana, V. N. Kumar, and P. Sanghavi, “Development of virtual flow sensor using artificial neural networks,” *International Journal of Innovative Technology and Exploring Engineering (IJITEE)*, vol. 9, no. 2, pp. 4081–4084, 2019.
- [6] S. G. Nnabuike, K. E. S. Pilario, L. Lao, Y. Cao, and M. Shafiee, “Identification of gas-liquid flow regimes using a non-intrusive doppler ultrasonic sensor and virtual flow regime maps,” *Flow Measurement and Instrumentation*, vol. 68, 2019. Disponible en: <https://kar.kent.ac.uk/79732/>.
- [7] A. S. Nagoo, “First wholly-analytical gas volume fraction model for virtual multiphase flow metering petroleum industry applications,” *International Journal of Energy Production and Management*, vol. 4, no. 3, pp. 245–253, 2019. [PDF] First Wholly-Analytical Gas Volume Fraction.pdf.

- [8] M. R. Hassan, S. Alkhalaf, A. M. Hemeida, M. Ahmed, and E. Mahmoud, “Artificial intelligent applications for estimating flow network reliability,” *Ain Shams Engineering Journal*, vol. 14, p. 102055, 2023.
- [9] A. T. Sandnes, B. Grimstad, and O. Kolbjørnsen, “Multi-task learning for virtual flow metering,” *Knowledge-Based Systems*, vol. 232, p. 107458, 2021.
- [10] T. Bikmukhametov and J. Jaschke, “Oil production monitoring using gradient boosting machine learning algorithm,” *IFAC PapersOnLine*, vol. 52, no. 1, pp. 514–519, 2019.
- [11] M. Hotvedt, B. A. Grimstad, and L. S. Imsland, “Passive learning to address nonstationarity in virtual flow metering applications,” *Expert Systems with Applications*, vol. 210, p. 118382, 2022.
- [12] H. Ghorbani, D. A. Wood, A. Choubineh, A. Tatar, P. G. Abarghoyi, M. Madani, and N. Mohamadian, “Prediction of oil flow rate through an orifice flow meter: Artificial intelligence alternatives compared,” *Petroleum*, vol. 6, no. 4, pp. 404–414, 2020.
- [13] H. A. Arief, P. J. Thomas, K. Constable, and A. K. Katsaggelos, “Towards building a distributed virtual flow meter via compressed continual learning,” *Sensors*, vol. 22, no. 24, p. 9878, 2022.
- [14] S. Sanzo, M. Montini, L. Cadel, M. Giuliani, and A. Bianco, “Virtual metering and allocation using machine learning algorithms,” in *International Petroleum Technology Conference*, (Dhahran, Saudi Arabia), January 2020. IPTC-20185-MS.
- [15] M. A. Ishak, T. A. H. Al-Qutami, and I. Ismail, “Virtual multiphase flow meter using combination of ensemble learning and first principle physics based,” *International Journal on Smart Sensing and Intelligent Systems*, vol. 15, no. 1, pp. 1–21, 2022.
- [16] R. Vylegzhanin, A. Cheremisin, B. Kolchanov, P. Lykhin, R. Kurmangaliev, M. Kozlov, E. Usov, and V. Ulyanov, “Development of virtual flow-meter concept techniques for ground infrastructure management,” *Energies*, vol. 16, no. 1, p. 400, 2023.

- [17] A. F. Ibrahim, R. Al-Dhaif, S. Elkatatny, and D. Al Shehri, “Applications of artificial intelligence to predict oil rate for high gas–oil ratio and water-cut wells,” *ACS Omega*, vol. 6, no. 29, pp. 19484–19493, 2021.
- [18] R. S. Farias, B. J. Cardoso, and D. J. Schiozer, “Virtual flow metering of oil wells for a pre-salt field,” *Journal of Petroleum Science and Engineering*, vol. 203, p. 108586, 2021.
- [19] T. McClure, M. Kårstad, R. Lorentzen, and O. M. Aamo, “Developing a hybrid data-driven, mechanistic virtual flow meter – a case study,” *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 11692–11697, 2020.
- [20] P. Beynon-Davies, C. Carne, H. Mackay, and D. Tudhope, “Rapid application development (rad): an empirical review,” *European Journal of Information Systems*, vol. 8, no. 3, pp. 211–223, 1999.
- [21] J. Martin, *Rapid Application Development*. New York: Macmillan, 1991.
- [22] C. Larman and V. R. Basili, “Iterative and incremental developments. a brief history,” *Computer*, vol. 36, no. 6, pp. 47–56, 2003.
- [23] B. W. Boehm, “A spiral model of software development and enhancement,” *Computer*, vol. 21, no. 5, pp. 61–72, 1988.
- [24] L. Breiman, “Random forests,” *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.
- [25] C. Cortes and V. Vapnik, “Support-vector networks,” *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995.
- [26] H. Drucker, C. J. C. Burges, L. Kaufman, A. Smola, and V. Vapnik, “Support vector regression machines,” *Advances in Neural Information Processing Systems*, vol. 9, pp. 155–161, 1997.
- [27] A. E. Hoerl and R. W. Kennard, “Ridge regression: Biased estimation for nonorthogonal problems,” *Technometrics*, vol. 12, no. 1, pp. 55–67, 1970.
- [28] R. Kohavi, “A study of cross-validation and bootstrap for accuracy estimation and model selection,” *Proceedings of the 14th International Joint Conference on Artificial Intelligence*, vol. 2, pp. 1137–1143, 1995.

- [29] C. M. Bishop, *Pattern Recognition and Machine Learning*. New York: Springer, 2006.
- [30] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA: MIT Press, 2016.

6. ANEXOS

6.1 Anexo A. Normas ISO utilizadas

En el desarrollo del sistema VFM se tomaron como referencia dos normas de la serie ISO/IEC de ingeniería de sistemas y software, citadas en el apartado de selección del ciclo de vida del Capítulo de metodología. A continuación se consigna la denominación oficial, el título íntegro, la edición y el uso que tuvo cada norma en el proyecto.

ISO/IEC/IEEE 12207:2017

- **Denominación completa:** ISO/IEC/IEEE 12207:2017.
- **Título oficial (inglés):** *Systems and software engineering — Software life cycle processes.*
- **Título en español:** *Ingeniería de sistemas y del software — Procesos del ciclo de vida del software.*
- **Organismo:** ISO/IEC JTC 1/SC 7 (Software and systems engineering), en colaboración con IEEE.
- **Fecha de publicación:** noviembre de 2017.
- **Alcance:** Establece un marco común de procesos para el ciclo de vida del software (adquisición, suministro, desarrollo, operación y mantenimiento), con actividades y tareas aplicables a proyectos y organizaciones.
- **Aplicación en el proyecto:** Sustentó la organización del trabajo en fases RAD (requisitos, diseño, construcción e implantación) y la trazabilidad entre requisitos, diseño, implementación y validación del VFM.

ISO/IEC 25010:2011

- **Denominación completa:** ISO/IEC 25010:2011.

- **Título oficial (inglés):** *Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — System and software quality models.*
- **Título en español:** *Ingeniería de sistemas y del software — Requisitos y evaluación de la calidad de sistemas y software (SQuaRE) — Modelos de calidad de sistemas y software.*
- **Organismo:** ISO/IEC JTC 1/SC 7.
- **Fecha de publicación:** marzo de 2011 (sustituida en parte por ISO/IEC 25010:2023, que actualiza el modelo de calidad del producto).
- **Alcance:** Define un modelo de calidad para productos software y sistemas, con características y subcaracterísticas (por ejemplo, mantenibilidad, confiabilidad, compatibilidad, usabilidad, eficiencia del desempeño y seguridad) que orientan requisitos no funcionales y evaluación.
- **Aplicación en el proyecto:** Orientó la definición de los requisitos no funcionales del VFM (modularidad, reproducibilidad, trazabilidad, usabilidad, escalabilidad, entre otros) y su verificación en la matriz de cumplimiento.

Nota: Las normas ISO/IEC son documentos con derechos de autor; en esta tesis se citan únicamente su identificación oficial y su aporte conceptual. El texto completo de cada norma debe consultarse en el organismo nacional de normalización o en la tienda oficial de ISO/IEC.

6.2 Anexo B. Guía de preguntas de las entrevistas

Las entrevistas se aplicaron al personal del Departamento de Instrumentación y Control del Campo Pucuna (supervisor, supervisor de instrumentación y control, e ingenieros de instrumentación y control). Se utilizó una guía semiestructurada en dos momentos: una primera ronda común a todos los entrevistados y una segunda ronda complementaria al supervisor para profundizar en datos operacionales y flujo de gas en

flares. Los resultados detallados por entrevistado figuran en el Capítulo de metodología (Cuadros de entrevistas).

Primera ronda — Preguntas comunes

1. ¿Qué problemas existen en la medición actual del flujo de gas?
2. ¿Qué variables son más relevantes para estimar el flujo?
3. ¿Cómo es la calidad de los datos actuales?
4. ¿Qué opina del uso de inteligencia artificial en este contexto?
5. ¿Qué beneficios tendría un sistema VFM?

Segunda ronda — Preguntas al supervisor (justificación de datos y flujo de gas)

6. ¿Por qué es necesario recolectar datos operacionales para el desarrollo del sistema VFM?
7. ¿Qué tipo de datos considera más importantes para el entrenamiento del modelo?
8. ¿Por qué es importante medir o estimar el flujo de gas en los flares?
9. ¿Qué problemas existen actualmente al no contar con medición directa del flujo de gas?
10. ¿Cómo contribuiría un sistema VFM basado en inteligencia artificial en este contexto?

6.3 Anexo C. Código fuente del sistema VFM

Este anexo reúne las capturas del código implementado en la Fase 3 de construcción del Medidor Virtual de Flujo: módulo de entrenamiento con validación cruzada, arquitectura de la red neuronal, servicio de predicción y definición de rutas del backend FastAPI.

```

def train_model_cv(X, y, model_name: str, activo: str):

    logger.info(f"Iniciando entrenamiento con K-Fold para activo: {activo}")

    scaler = StandardScaler()
    X_scaled = scaler.fit_transform(X).astype(np.float32)

    n_folds = max(2, min(TRAIN_FOLDS, 10))

    kf = KFold(n_splits=n_folds, shuffle=False)
    models = []
    histories = []
    mse_scores = []
    mae_scores = []
    r2_scores = []

    fold = 1
    for train_index, val_index in kf.split(X_scaled):
        X_train, X_val = X_scaled[train_index], X_scaled[val_index]
        y_train, y_val = y[train_index], y[val_index]

        logger.info(f" Fold {fold}/{n_folds} - Train size: {len(X_train)}, Val size: {len(X_val)}")

        model = build_model(input_dim=X.shape[1])
        early_stopping = tf.keras.callbacks.EarlyStopping(
            monitor='val_loss',
            patience=TRAIN_EARLY_STOPPING_PATIENCE,
            restore_best_weights=True,
            verbose=0
        )

        history = model.fit(
            X_train, y_train,
            epochs=TRAIN_EPOCHS,
            batch_size=min(32, len(X_train)),
            validation_data=(X_val, y_val),
            callbacks=[early_stopping],
            verbose=0
        )

        # Calcular métricas
        y_pred = model.predict(X_val, verbose=0).flatten()
        mse = mean_squared_error(y_val, y_pred)
        mae = mean_absolute_error(y_val, y_pred)

```

Figura 24: Módulo de entrenamiento del modelo de inteligencia artificial

```
def predict_dataframe(df: pd.DataFrame):

    df = df.copy()
    df.columns = df.columns.astype(str).str.strip()
    if len(df.columns) > 0 and (str(df.columns[0]).startswith('Unnamed') or str(df.columns[0]).strip() == ''):
        df = df.iloc[:, 1:]

    norm_map = _make_column_map(df.columns)

    for req in REQUIRED_COLS:
        try:
            _ = _find_column(req, norm_map)
        except KeyError:
            raise

    preds = {}
    for activo in ACTIVOS:
        col_a = _find_column(f"{activo}_A", norm_map)
        col_p = _find_column(f"{activo}_P", norm_map)

        model_file = os.path.join(MODEL_DIR, f"{activo}_model.keras")
        if not os.path.exists(model_file):
            raise FileNotFoundError(f"Modelo no encontrado para activo '{activo}': {model_file}")

        base_model = tf.keras.models.load_model(model_file)
        X, clean_report = prepare_asset_predict_frame(df, col_a, col_p)
        if clean_report["nulos_imputados_a"] or clean_report["nulos_imputados_p"]:
            logger.info(
                "Limpieza predict %s: A=%s P=%s nulos imputados",
                activo,
                clean_report["nulos_imputados_a"],
                clean_report["nulos_imputados_p"],
            )

        scaler_file = scaler_path_for(model_file)
        if os.path.exists(scaler_file):
            scaler = joblib.load(scaler_file)
            X = scaler.transform(X)
        y_pred = base_model.predict(X, verbose=0).flatten()
        preds[f"{activo}_D"] = y_pred

    out_df = pd.DataFrame(preds)

    if 'FECHA' in df.columns:
        out_df.insert(0, 'FECHA', pd.to_datetime(df['FECHA'], errors='coerce'))
```

Figura 25: Servicio de predicción del flujo de gas asociado

```
@app.get('/ping')
@app.get('/health')
@app.get('/models')
@app.get('/metrics/{activo}')
```

Figura 26: Rutas del backend FastAPI (entrenamiento, predicción y monitoreo)

6.4 Anexo D. Manual de usuario de la Terminal VFM

El manual se publicó en línea en la siguiente dirección:

<https://felixmf.github.io/manual/>