

Componentes de software y documentación del caso de uso de economía tradicional

MLEDGE - Aprendizaje automático en la nube y en el borde
(Cloud and Edge Machine Learning)

Diciembre de 2024

Información sobre el entregable

Nombre del documento:

M2 - Componentes de software y documentación del caso de uso de economía tradicional

Versión actual: 1.0

Proyecto: MLEDGE - Aprendizaje automático en la nube y en el borde (Cloud and Edge Machine Learning)

Paquete de trabajo: P3 - Implementación del caso de uso de economía tradicional

Tareas: El entregable es resultado del trabajo en los diversos componentes técnicos:
- A3.2: Implementación y pruebas del caso de uso de economía tradicional

Entregable: E3.2 - Componentes software, informe y documentación preliminares.

Autores: Inmarepro S.L. y Acuratio Europe S.L.

Revisores: Nikolaos Laoutaris (IMDEA) y Javad Dogani (IMDEA)

Historial de Versiones

Versión	Fecha	Resumen de modificaciones
Versión 1.0	31-12-2024	Versión final del documento

Información sobre el entregable	2
Historial de Versiones	2
1. Introducción	4
2. Definición del problema y objetivos	5
2.1. Objetivos.....	5
2.2. La solución	6
2.3. Caso de uso	7
3. Instalación de sensores	8
3.1. Descripción de Sensores.....	8
3.2. Descripción de PLC.....	10
3.3. Conectividad Sensores/PLC.....	11
3.4. Implantación casos reales.	11
4. Transmisión y protección de Datos	20
4.1. Extracción y preparación de datos para la transmisión.....	20
4.2. Transmisión de datos y configuración de la nube	23
4.3. Despliegue Plataforma Federada.....	26
4.3.1. Cloud-Edge	26
4.3.2. Despliegue	27
4.4. Exploración de datos.....	28
5. Solución preliminar	30
5.1. Sistema de control.....	30
5.1.1. Escalabilidad: Un enfoque más amplio.	30
5.1.2. Rendimiento y Eficiencia Operativa	31
5.1.3. Seguridad y Usabilidad Adaptadas a Escenarios Variados	31
5.1.4. Evolución hacia una Solución Generalizada	31
5.2. Desarrollo del modelo ML.....	32
5.2.1. Modelo predictivo de la producción de vapor.....	33
5.2.2. Modelo predictivo del rendimiento de la caldera	35
5.3. Implementación de FL.....	36
5.3.1. Federación del Modelo.....	37
6. Conclusión y siguientes pasos	43

1. Introducción

En diciembre de 2022, el Ministerio de Asuntos Económicos y Transformación Digital del Gobierno de España adjudicó a IMDEA Networks el proyecto titulado “*MLEDGE - Aprendizaje automático en la nube y en el borde (Cloud and Edge Machine Learning)*” (REGAGE22e00052829516, en adelante el ‘Proyecto’ o MLEDGE). Esta iniciativa cuenta con financiación de la Unión Europea a través del Plan de Recuperación, Transformación y Resiliencia (European Union - NextGenerationEU/PRTR).

El objetivo principal del Proyecto es establecer un ecosistema robusto de servicios de aprendizaje federado (FL, por sus siglas en inglés) en el borde, que sean seguros y eficientes. Estos servicios están diseñados para facilitar el uso de datos personales y confidenciales, tanto de tipo B2B como de consumidores, en el entrenamiento de modelos de aprendizaje automático (ML), garantizando en todo momento la privacidad de los datos y de sus propietarios.

Los **objetivos generales del proyecto** se pueden resumir en los siguientes:

1. Facilitar la accesibilidad del aprendizaje federado en el borde mediante el desarrollo de una capa de software intermedio y componentes que simplifiquen la complejidad inherente al procesamiento y al intercambio de datos.
2. Abordar los desafíos técnicos asociados al aprendizaje federado en entornos de nube y borde, optimizando su implementación y rendimiento.
3. Validar la funcionalidad desarrollada a través de casos de uso representativos de problemas reales de la industria, demostrando el impacto práctico de estas tecnologías.
4. Difundir y explotar los resultados del Proyecto, involucrando a agentes externos clave y comunicando los hallazgos a un público más amplio y potencialmente interesado.

Uno de los objetivos fundamentales del proyecto consiste en diseñar, implementar y poner a disposición del público demostradores que trabajen con datos sensibles, tanto de carácter industrial como personal. Estos demostradores están orientados a alimentar modelos de aprendizaje automático aplicables a diversos sectores de la industria. En la etapa inicial del proyecto, se seleccionaron empresas clave para el desarrollo de la plataforma FLaaS (*Federated Learning as a Service*), el monitoreo de costes computacionales, y el diseño e implementación de casos de uso reales que se beneficien del aprendizaje distribuido en el borde de la nube.

La adjudicación del paquete de trabajo P3, enfocado en el diseño e implementación del caso de uso relacionado con la economía tradicional, recayó en la UTE INMAREPRO S.L. - ACURATIO EUROPE SL, con NIF U70842836.

El presente documento corresponde al entregable E3.2, titulado “*Componentes de software y documentación del caso de uso de economía tradicional*”. Su propósito principal es detallar el desarrollo e implementación de los elementos definidos en el entregable E3.1, describiendo el proceso seguido y presentando la solución preliminar alcanzada.

2. Definición del problema y objetivos

En este apartado se presenta un resumen de la definición del problema y los objetivos específicos del proyecto, tal como se detallaron en el entregable E3.1. Asimismo, se describen los casos de uso relevantes, con el objetivo de proporcionar un marco de referencia que facilite la comprensión de las tareas realizadas en este entregable y su integración en el contexto general del proyecto.

2.1. Objetivos

El proyecto tiene como objetivo optimizar el consumo energético de calderas de vapor mediante el ajuste de sus parámetros operativos utilizando modelos entrenados con datos provenientes de calderas instaladas en múltiples clientes, incluidos algunos que son competidores directos. Para preservar la confidencialidad de los datos, los modelos se desarrollarán empleando técnicas de *Federated Learning* (FL). Este enfoque busca superar las barreras asociadas al intercambio de datos y demostrar a los clientes el valor de la colaboración para reducir los costos energéticos.

Aunque las soluciones de MLEDGE diseñadas para clientes industriales son aplicables a diversos tipos de equipos (como calderas, compresores, enfriadoras y sistemas eléctricos), el desarrollo inicial del componente objeto del proyecto se ha centrado en un tipo de equipo específico: las calderas de vapor. Esta decisión responde a su papel crítico como el núcleo operativo y uno de los costos más relevantes en numerosas industrias tradicionales. La elección de este enfoque acotado permite una recopilación más específica de datos y el desarrollo de modelos de ML/FL que generen un impacto significativo en la reducción de costos energéticos de los clientes industriales.

Las calderas de vapor, debido a su operación estandarizada y medible, son especialmente adecuadas para la toma de datos. Estas calderas desempeñan un rol central en la generación de energía y los procesos de producción, lo que las convierte en elementos estratégicos para la mejora de la eficiencia energética. Dichas mejoras pueden lograrse mediante el ajuste de parámetros operativos, la sustitución de quemadores mecánicos por electrónicos, o la incorporación de recuperadores de calor.

Estas características hacen que las calderas de vapor sean idóneas para la implementación de tecnologías federadas, ya que los datos necesarios para optimizar su rendimiento se generan en múltiples ubicaciones y poseen un alto nivel de sensibilidad. Por ejemplo, del análisis de estos datos podría inferirse información estratégica sobre el consumo energético y la producción de las plantas industriales. Por este motivo, el uso de tecnologías federadas resulta fundamental para garantizar la privacidad, la confidencialidad y la eficiencia en el tratamiento de estos datos.

En este contexto, las tecnologías federadas permiten monitorear y optimizar el rendimiento de cada caldera de manera segura y eficiente. Estas soluciones no solo protegen la privacidad de datos sensibles como el consumo de combustible o la producción de vapor, sino que también facilitan el mantenimiento predictivo y la gestión energética integral, contribuyendo a maximizar la eficiencia operativa y reducir los costos de los clientes industriales.

2.2. La solución

El caso de uso se ha desarrollado aprovechando componentes y sistemas ya establecidos, con el objetivo de mejorar la eficiencia operativa y optimizar la toma de decisiones mediante el uso de *Federated Machine Learning*.

Esta solución ofrece un alto valor a los clientes industriales al permitirles evaluar el rendimiento óptimo de sus equipos y tomar decisiones empresariales que mejoren la eficiencia energética en sus procesos productivos, todo ello garantizando la confidencialidad de sus datos.

Desde el punto de vista de la solución, se pueden identificar tres niveles de servicio potencial, cada uno habilitado por una capacidad tecnológica distinta que, a su vez, incrementa la complejidad y el valor en relación con el nivel anterior. En este sentido, la implementación de un sistema de control básico podría verse potenciada con la incorporación de capacidades de *Machine Learning* y, finalmente, con la integración de *Federated Learning*:

Nivel 1: Sistema de Control

El primer peldaño de esta solución tecnológica es un sistema de control avanzado que realiza un monitoreo constante de todas las variables relevantes, como el consumo de gas, electricidad y agua, así como la producción de energía. A través de una plataforma en línea, los operadores pueden:

- Ver la producción en tiempo real.
- Acceder al historial de consumo.
- Monitorear la eficiencia del equipo mediante indicadores como el *Coefficient of Performance* (COP).

Para poder cumplir estos puntos el sistema de control se apoya en los siguientes equipos de medida recogidos en el PLC de manera local:

- Caudalímetro de vapor.
- Caudalímetro de agua de alimentación caldera.
- Cuantometro de gas con correcto PT.
- Sonda de temp. de agua de alimentación caldera.
- Sonda de temp. humos de chimenea.
- Sonda de temp. depósito de condensados.
- Sonda de temp. ambiente entrada de aire al quemador.
- Contador de agua de llenado.
- Sonda de temp. de agua de llenado.

Nivel 2: *Machine Learning*

Al incorporar el *Machine Learning*, el sistema alcanza un nuevo nivel de robustez y precisión. Con el análisis de datos en tiempo real y en base a históricos, se pueden habilitar herramientas basadas en modelos de ML como:

- Detección de anomalías que podrían indicar fallos o ineficiencias.
- Sugerencias de optimización para mejorar el rendimiento del equipo.

- Predicción de la producción para ayudar en la planificación.
- Optimización en inversión en sensores por inferencias de variables obtenidas.
- Análisis de coste-beneficio para el ahorro de costes.

Nivel 3: *Federated Learning*

El nivel más avanzado utiliza el *Federated Learning* para aprovechar la inteligencia colectiva a la hora de entrenar estos modelos de ML. Al comparar su equipo con modelos entrenados con máquinas similares en diferentes instalaciones, los operadores obtienen:

- Un entendimiento más profundo de qué variables afectan la eficiencia del equipo.
- Recomendaciones para mejorar el rendimiento basadas en la eficiencia de equipos similares en otros lugares.
- La posibilidad de tomar decisiones informadas sobre inversiones en nuevos equipos, seleccionando aquellos con niveles de eficiencia superiores.

2.3. Caso de uso

Desde optimizar procesos empresariales, hasta predecir anomalías de alto impacto sobre una instalación, el análisis de datos ofrece un amplio abanico de posibilidades. A través de estos cuatro casos de uso que se presentan, exploraremos cómo esta poderosa herramienta impulsa la toma de decisiones y el ahorro energético-económico sobre las industrias.

Se han confirmado cuatro casos (anónimos) en la cartera de clientes de la compañía INMAREPRO S.L. con distintos sectores económicos o casos de producción de vapor, para presentar un modelo industrial más fortalecido y exportable al mercado.

ID	Caso de uso	Usuario	Objetivo	Beneficio, resultado, razón del caso de uso
US_1	COSMÉTICO	Dirección de Producción y Mantenimiento de la fábrica.	Analizar y mejorar el rendimiento de la caldera de vapor, junto a la detección predictiva de anomalías a tiempo para actuar preventivamente y así prolongar su longevidad y evitar averías de alto impacto.	Aportar datos y cuantificar el impacto económico, ecológico y funcional sobre las acciones anteriormente indicadas, junto a una supervisión continua de los consumos para una detección inmediata de posibles desviaciones.
US_2	ALIMENTICIO			
US_3	LAVANDERIA INDUSTRIAL			
US_4	FARMACEUTICO			

3. Instalación de sensores

Una correcta instalación de sensores constituye un pilar fundamental para el monitoreo eficiente de las calderas industriales en el caso de uso de economía tradicional. Para ello se ha instalado, configurado, desplegado y testeado una nueva infraestructura de adquisición, transmisión y almacenamiento de datos. Desde el Edge, donde se generan los datos hasta la nube donde se almacena, para posteriormente ser tratados. Esta nueva infraestructura ha sido un reto debido a las complejidades inherentes de los entornos industriales.

Esta etapa asegura la recolección de datos precisos y en tiempo real sobre variables críticas, como consumo energético, temperatura, presión y flujos de vapor, permitiendo un análisis detallado y la optimización del rendimiento de los equipos.

El proyecto MLEDGE ha instalado sensores de última generación y aprovecha una infraestructura preexistente basada en sensores industriales estándares (0-10V y 4-20mA) integrados con controladores lógicos programables (PLCs) a través del protocolo Modbus. Estos sensores garantizan la medición fiable de las condiciones operativas y facilitan la transmisión de datos hacia el sistema en la nube.

En esta sección, se detalla el proceso de selección, instalación y configuración de los sensores y PLCs en los entornos industriales de los clientes, abordando las características técnicas necesarias para su integración con la plataforma MLEDGE.

3.1. Descripción de Sensores.

A continuación, se detalla en una tabla los sensores y dispositivos instalados en las calderas de vapor, incluyendo sus características y variables medidas, que aseguran la recopilación precisa de datos en tiempo real.

Sensor/ Dispositivo	Variable Medida	Rango de Operación	Protocolo	Ubicación	Propósito
Temperatura de llenado	Temperatura del agua	-40 a 150 °C	NTC20k	Tubería entrada de agua de llenado	Monitorizar la temperatura de llenado en el depósito de condensados.
Contador de agua de llenado	Consumo de agua	Cuantometro 0-999999 litros	Pulsos/litro	Entrada de agua de llenado	Monitorizar el volumen de agua que se aporta o consume en instalación.
Temperatura depósito de condensador	Temperatura del agua	-40 a 150 °C	NTC20k	Depósito de condensados	Monitorizar la temperatura del depósito de condensados.

Temperatura agua alimentación caldera	Temperatura del agua	-40 a 150 °C	NTC20k	Tubería alimentación caldera	Monitorizar la temperatura de alimentación a caldera.
Caudalímetro agua alimentación caldera	Consumo de agua	0-1000 m³/h	Modbus	Tubería alimentación caldera	Monitorizar el volumen transcurrido de agua de alimentación a la caldera
Temperatura de humos	Temperatura humos	0 a 300 °C	PT-100	Chimenea	Monitorizar la temperatura de la salida de humos de la caldera de vapor.
Sonda Lambda	Oxigeno	0 a 20%	0-10V	Chimenea	Monitorizar la cantidad de oxígeno en la salida de los gases de la caldera de vapor.
Temperatura ambiente entrada de aire al quemador	Temperatura aire	-40 a 70 °C	NTC20k	Pared cercana al quemador	Monitorizar la temperatura aire ambiente de entrada al quemador.
Contador de gas	Consumo de gas	0-500 m³/h	Modbus	Entrada de suministro de gas	Monitoreo del consumo de gas
Caudalímetro de Vapor	Producción de Vapor	0-1500 m³/h	Modbus	Tubería de vapor	Monitorizar la producción y el volumen transcurrido de vapor a la instalación.
Presión caldera	Presión del vapor	0-16 bar	Modbus	Tubería de vapor	Monitorizar de la presión operativa de vapor.

3.2. Descripción de PLC.

En este apartado, se describe el Controller CLNXEHSERIES26ND, el controlador lógico programable (PLC) utilizado en las distintas instalaciones del proyecto. Este dispositivo destaca por su robustez, capacidades avanzadas de procesamiento y comunicación, y su compatibilidad con múltiples protocolos de automatización industrial.

Características técnicas principales

Procesador y memoria:

- **CPU:** Procesador NXP i.MX 6SoloX, dual core, con núcleo ARM Cortex-A9 a 1 GHz y núcleo ARM Cortex-M4 a 227 MHz.
- **Memoria:**
 - i. RAM: 1 GB DDR3.
 - ii. Flash: 4 GB.
 - iii. MRAM (almacenamiento no volátil): 512 KB, asegurando la persistencia de datos críticos.

Entradas y salidas (I/O):

- Entradas universales (UI): Compatibles con señales **NTC10kΩ**, **NTC20kΩ**, y **0-10V**.
- Entradas digitales (BI): Soporte para señales de 24V con frecuencia máxima de 15 Hz.
- Salidas analógicas (AO): Rango de salida de **0-11V**.
- Salidas digitales (BO): Relés con capacidad de corriente de hasta 10A (para alta potencia).

Protocolo de comunicación:

- **Modbus RTU/TCP:** Permite la integración directa con sensores y sistemas de monitoreo.
- **BACnet/IP y MS/TP:** Ideal para la interoperabilidad en entornos de control HVAC.
- Soporte adicional para **M-Bus**, **LONWORKS**, y otros protocolos estándares.

Capacidades de red y conectividad:

- **Ethernet (2 puertos RJ45):** Comunicación a 10/100 Mbps.
- **Interfaces RS485:** Aisladas y no aisladas, compatibles con Modbus RTU y Panel Bus.
- **USB 2.0:** Para conexión de periféricos como HMIs o módulos de expansión.

Características físicas:

- Dimensiones: 215.5 x 110 x 61 mm.
- Protección: IP20, adecuado para montaje en paredes, armarios o paneles eléctricos.
- Alimentación: 24 V AC/DC, consumo típico de 7W.

Ventajas operativas

- **Procesamiento en tiempo real:** Gracias al sistema operativo embebido **QNX**, el controlador asegura operaciones fiables y seguras, incluso con acceso remoto.
- **Flexibilidad de montaje:** Opciones para instalación en paredes, puertas de armarios, o rieles DIN.
- **Soporte web:** Incluye servidor web integrado para monitoreo y configuración a través de navegadores estándar, permitiendo el acceso desde cualquier dispositivo conectado, **escalabilidad y modularidad**.

El EagleHawk NX permite ampliar su capacidad mediante módulos I/O adicionales conectados a través del Panel Bus, alcanzando más de 1000 puntos I/O. Su arquitectura modular facilita la integración con dispositivos externos y sistemas de gestión centralizada como SCADA.

3.3. Conectividad Sensores/PLC

La conectividad entre los sensores industriales y el controlador lógico programable (PLC) CLNXEHSERIES26ND es esencial para garantizar el monitoreo eficiente de las calderas industriales. Los sensores empleados, como caudalímetros, sondas de temperatura, medidores de presión y dispositivos de medición de oxígeno, están diseñados para transmitir datos críticos en tiempo real mediante protocolos estándar como Modbus RTU/TCP, pulsos/litro o señales analógicas (0-10V, 4-20mA).

Estos datos se integran en el PLC, el cual dispone de amplias capacidades de procesamiento y almacenamiento, entradas y salidas universales, y conectividad con redes Ethernet y protocolos industriales.

En el siguiente esquema eléctrico se detalla cómo se interconectan estos equipos, asegurando una transmisión fiable y robusta hacia el sistema de monitoreo.

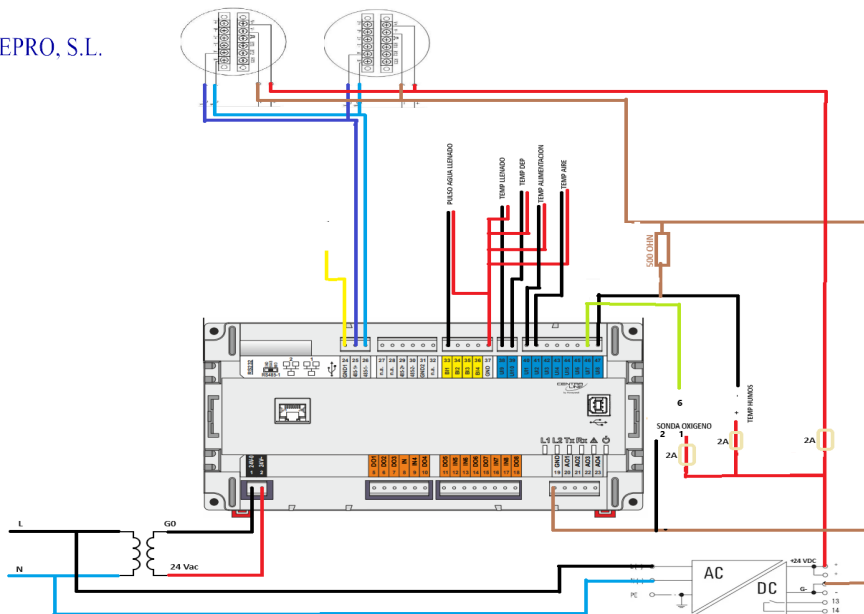


Ilustración 1: Esquema eléctrico PLC

3.4. Implantación casos reales.

La implementación en un entorno industrial real consistió en la instalación del sistema integral de monitoreo y control en calderas de vapor, utilizando los sensores y el controlador lógico programable (PLC) previamente descritos. Este proceso incluyó la integración de los sensores para capturar datos clave de las condiciones operativas de la caldera, como temperatura, presión, consumos y producción de vapor, junto con su conexión al PLC mediante protocolos estándar como Modbus y señales analógicas.

Además, se instaló un cuadro eléctrico centralizado que alberga el PLC y los módulos de expansión necesarios, con su correspondiente sistema de alimentación y protección eléctrica. El diseño y montaje se llevaron a cabo cumpliendo los estándares de seguridad y eficiencia, asegurando una operatividad fiable y de bajo mantenimiento.

A continuación, se presentan imágenes del cuadro eléctrico, el controlador, y los sensores instalados en el sistema, que ilustran el despliegue físico de la solución en un entorno de operación real de algunas de las instalaciones del proyecto:



IMAGEN CUADRO DE CONTROL
CLIENTE US_3

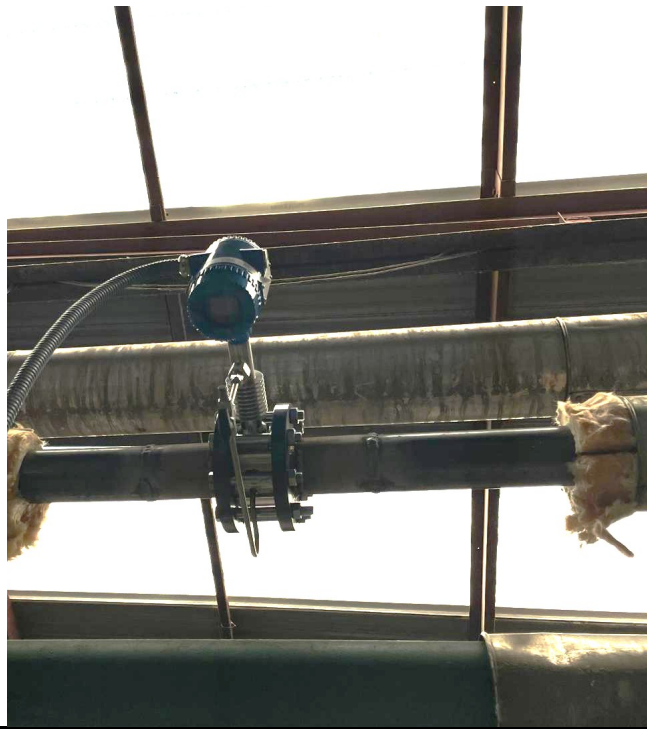


IMAGEN CAUDALIMETRO VAPOR
CLIENTE US_3



IMAGEN CONTADOR DE GAS CON
CORRECTOR DE TEMPERATURA
Y PRESION CLIENTE US_3



IMAGEN CAUDALIMETRO AGUA
ALIMENTACIÓN CALDERA
CLIENTE US_3



IMAGEN SONDA AGUA
ALIMENTACIÓN CALDERA
CLIENTE US_3



IMAGEN SONDA DE HUMOS Y
SONDA LAMBDA (Oxigeno)
CLIENTE US_3



IMAGEN SONDA TEMPERATURA
DEPOSITO CONDENSADOS
CLIENTE US_3



IMAGEN Sonda TEMPERATURA
LLENADO Y CONTADOR AGUA
LLENADO CON EMISOR DE
PULSOS CLIENTE US_3



IMAGEN CUADRO DE CONTROL
CLIENTE US_1

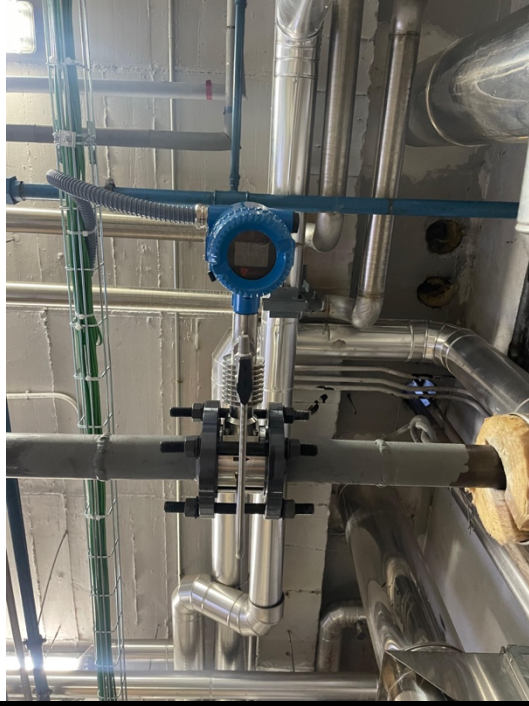


IMAGEN CAUDALIMETRO VAPOR
CLIENTE US_1



IMAGEN CONTADOR DE GAS CON
CORRECTOR DE TEMPERATURA
Y PRESION JUNTO A Sonda
TEMPERATURA AIRE AMBIENTE
CLIENTE US_1



IMAGEN CAUDALIMETRO AGUA
ALIMENTACIÓN CALDERA
CLIENTE US_1



IMAGEN Sonda DE HUMOS Y
SONDA LAMBDA (Oxigeno)
CLIENTE US_1



IMAGEN Sonda AGUA
ALIMENTACIÓN CALDERA
CLIENTE US_1



IMAGEN Sonda TEMPERATURA
LLENADO Y CONTADOR AGUA
LLENADO CON EMISOR DE
PULSOS CLIENTE US_1

4. Transmisión y protección de Datos

4.1. Extracción y preparación de datos para la transmisión

Tras completar las instalaciones y puestas en marcha de las distintas instalaciones descritas en el punto 3, que incluyeron la integración de sensores y PLCs para la monitorización de variables críticas, se inició el proceso de extracción y registro de datos. Este paso es fundamental para garantizar que los datos recolectados representen fielmente las condiciones operativas, permitiendo un análisis detallado del rendimiento de las calderas y optimizando la toma de decisiones basada en métricas fiables.

Los sensores instalados comenzaron a generar datos en tiempo real sobre parámetros clave como temperatura, presión, caudal y consumo energético. Estos datos fueron recopilados por el PLC, que actúa como el núcleo de procesamiento y almacenamiento temporal del sistema.

A continuación, se adjuntan pantallazos de varias de las instalaciones sobre las que hemos realizado la instalación con las lecturas reales de ese momento.

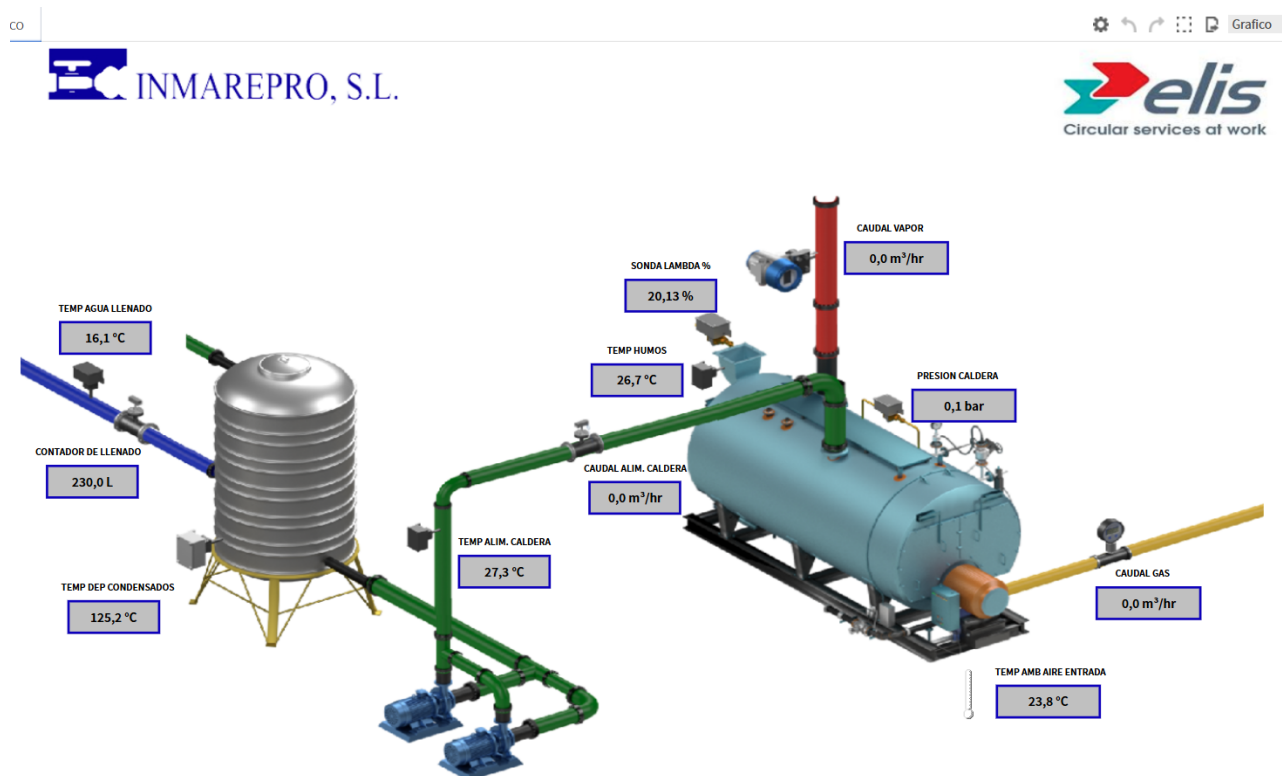


Ilustración 2: Ejemplo toma de datos instalación I

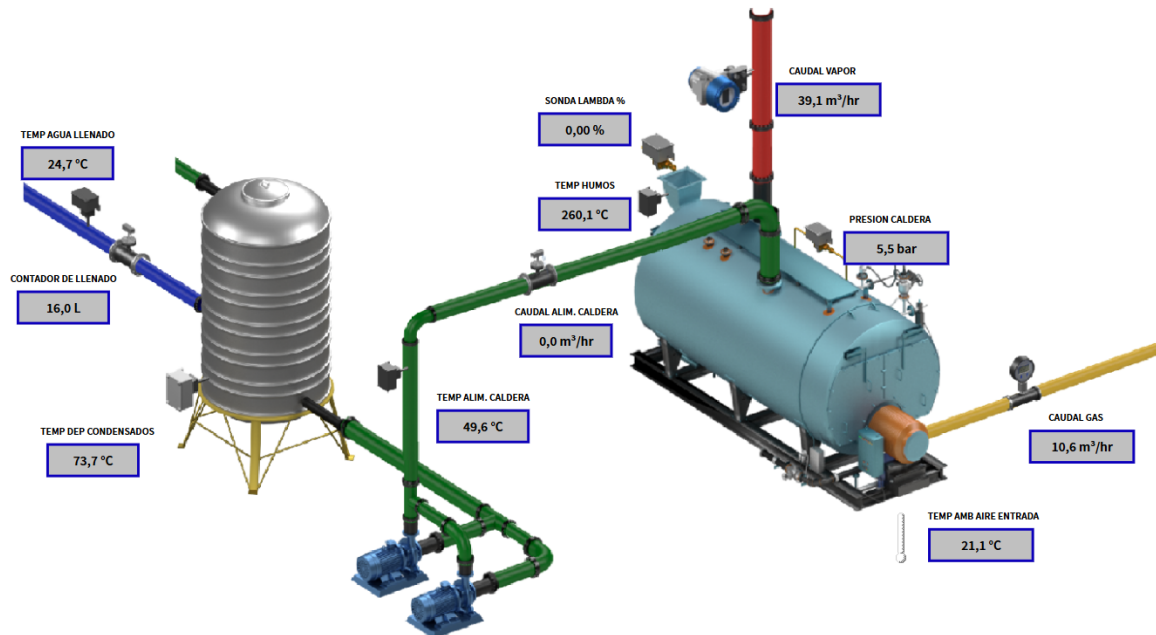


Ilustración 3: Ejemplo toma de datos instalación II

Para definir la mejor estrategia de registro, se llevó a cabo un análisis exhaustivo de las distintas frecuencias de muestreo, evaluando intervalos de de:

- 15 minutos.
- 5 minutos.
- 1 minuto.
- 30 segundos.

Este análisis buscaba identificar la periodicidad óptima para capturar la dinámica operativa del sistema sin generar redundancia o sobrecargar la infraestructura de almacenamiento.

Durante el análisis, se observaron importantes variaciones en las mediciones en lapsos cortos de tiempo. Estas variaciones, especialmente en variables como la temperatura de los humos de la chimenea y el caudal de vapor producido, son indicativas de los cambios rápidos en las condiciones operativas que se producen durante los ciclos normales de trabajo de las calderas.

Al evaluar el intervalo de 30 segundos, se determinó que, aunque lograba captar las fluctuaciones del sistema con alta frecuencia, aportaba poco valor adicional en comparación con el intervalo de 1 minuto. Además, este nivel de granularidad implicaba una mayor demanda de recursos de almacenamiento y procesamiento, lo que no se justificaba en términos de beneficios analíticos. Por estas razones, se descartó el intervalo de 30 segundos como una opción viable.

En cambio, al optar por un intervalo de 1 minuto, se logró capturar fielmente las variaciones operativas significativas sin generar una sobrecarga innecesaria en los recursos. Los intervalos de 15 y 5 minutos, por otro lado, mostraron una suavización excesiva de los datos,

lo que podría llevar a interpretaciones inexactas sobre el comportamiento del sistema. Por ello, se determinó que el intervalo de 1 minuto era el más adecuado para capturar con precisión las dinámicas del sistema y proporcionar una base sólida para el análisis.

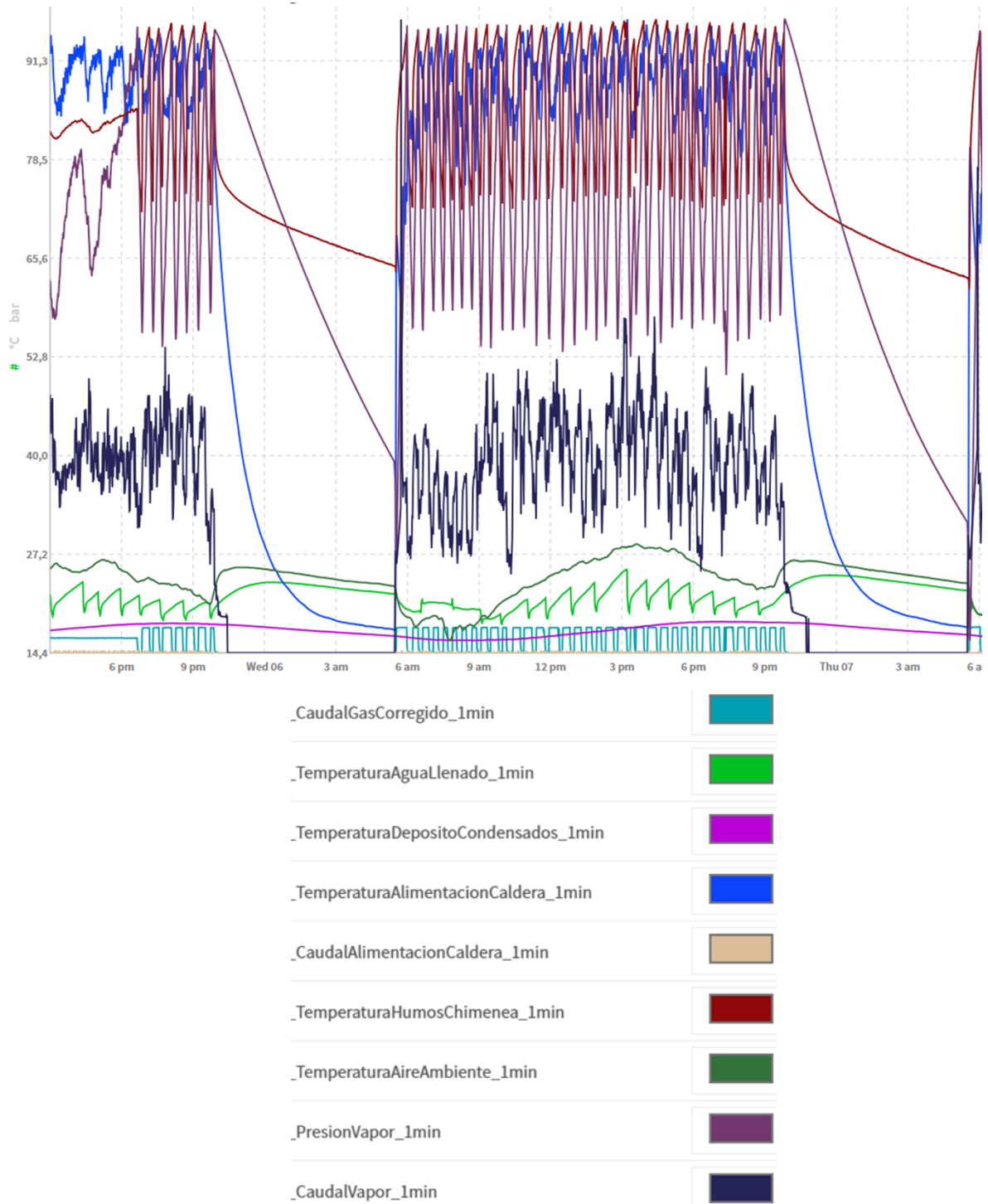


Ilustración 4: Ejemplo toma de datos en sensores instalados

Esta decisión no solo asegura una mayor precisión en el monitoreo, sino que también permite identificar patrones operativos detallados, detectar anomalías en tiempo real y establecer relaciones más precisas entre variables clave. Por ejemplo, al correlacionar la temperatura

de los humos con el caudal de gas corregido y el caudal de vapor producido, es posible identificar ajustes en la mezcla de combustible o cambios en la eficiencia de intercambio térmico. Estas correlaciones no serían evidentes con registros menos frecuentes.

Una vez establecida la periodicidad de registro de 1 minuto, se programaron los controladores PLC para crear históricos detallados que incluyeran todas las variables seleccionadas. Este registro local se diseñó para optimizar el uso de la memoria del equipo, garantizando al mismo tiempo que la calidad de los datos no se vea comprometida. Los históricos se almacenan temporalmente en el controlador por un tiempo máximo de 7 días, desde donde se preparan para su transmisión al sistema en la nube. La configuración incluye un formato estandarizado que asegura la compatibilidad con las herramientas de análisis utilizadas por los equipos de ACURATIO, lo que facilita la integración y el procesamiento posterior de la información.

La visualización de las instalaciones no solo muestra el alcance del proyecto, sino que también pone en evidencia la importancia de un diseño robusto y la atención a los detalles técnicos en el despliegue de sistemas de monitoreo avanzados.

4.2. Transmisión de datos y configuración de la nube

Los datos generados por las distintas calderas se almacenarán cifrados en *Google Cloud Storage*, controlando el acceso desde cada una de las Máquinas Virtuales (VMs) asignadas a las empresas propietarias de dichas calderas. Para esto se emplearán permisos granulares definidos para cada archivo o grupo de archivos. Esto asegura que solo las VMs autorizadas puedan acceder a archivos específicos, utilizando políticas robustas y cifrado para proteger la información.

Para transmitir los datos desde el PLC industrial al *bucket* de almacenamiento en GCP, el sistema utiliza emails como medio de transporte y *App Engine* como procesador que organiza y transfiere los datos al *bucket*. En la siguiente figura se muestra un resumen de la arquitectura desplegada para implementar y monitorizar el proceso de transmisión de datos.



Ilustración 5: Diagrama arquitectura adquisición de datos del Edge al Cloud

El PLC recopila datos de los sensores y forma un mensaje de texto con un archivo *xlsx* con las métricas. A continuación, envía un correo electrónico a la dirección predefinida en *App Engine* usando *Simple Mail Transfer Protocol* (SMTP). SMTP es un protocolo estándar

utilizado para enviar correos electrónicos a través de redes IP. Es la base para la transmisión de mensajes de correo electrónico entre servidores de correo y hacia el cliente de correo final.

Configurar *Google App Engine* para recibir correos electrónicos y transmitir su contenido a un bucket de *Google Cloud Storage* (GCS) implica varias etapas que integran diferentes servicios de GCP. Para lograr que *App Engine* pueda procesar correos electrónicos se utiliza *Mail API*, desarrollando un script de *Python* capaz de extraer los archivos adjuntos del correo y redirigirlos a GCS.

Los componentes principales del sistema de gestión de datos en la nube son:

- **Repositorio de datos:** *Google Cloud Storage* (GCS), los datos se almacenan en *buckets* con cifrado en reposo.
- **Servicio de gestión de claves (KMS):** Administra las claves de cifrado utilizadas para proteger los datos.
- **Gestión de Identidades y Accesos (IAM):** Gestiona los permisos a nivel de archivo, *bucket* y clave.

En la siguiente figura se puede apreciar el flujo de datos recibido por *App Engine* el día 27 de noviembre durante las pruebas de despliegue. Cada vez que el PLC envía un correo electrónico, el *backend* en *Google Cloud* procesa los archivos adjuntos y al localizar un fichero *xlsx* lo deposita en GCS.

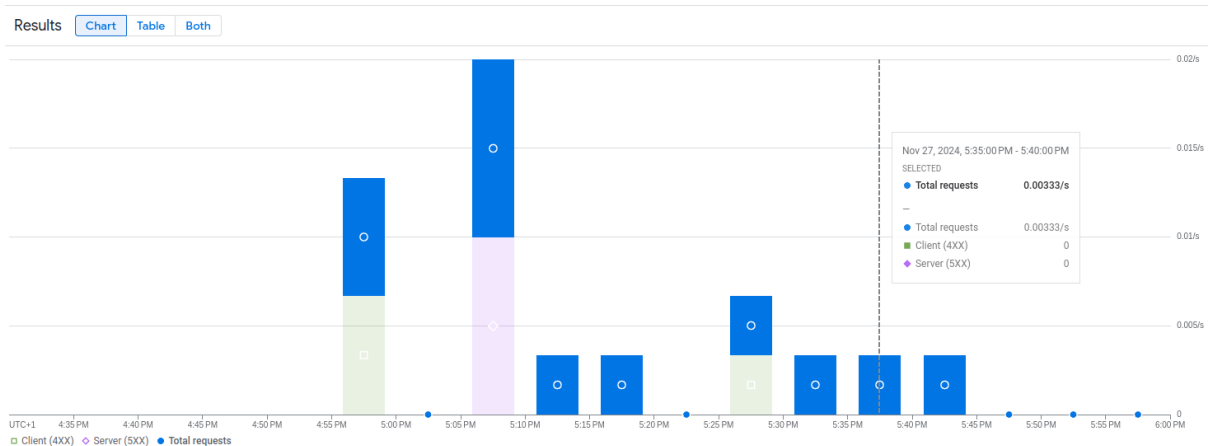


Ilustración 6: Muestra de registros de adquisición datos

A continuación, se define el flujo de datos una vez han sido almacenados en GCS. Todos los datos en GCS están cifrados en reposo automáticamente empleando una de las siguientes soluciones:

- **CMEK (Customer-Managed Encryption Keys):** Permite usar claves administradas por el cliente a través de *Cloud KMS*. Da control total al cliente sobre la rotación, habilitación o inhabilitación de claves.
- **CSEK (Customer-Supplied Encryption Keys):** Claves proporcionadas por el cliente. GCP nunca almacena las claves; se proporcionan en cada acceso.
- **Claves administradas por Google:** Configuración predeterminada; GCP administra el ciclo de vida de las claves.

En este escenario se ha optado por utilizar las claves administradas por Google por simplicidad de gestión, sin embargo, también sería posible utilizar CMEK para asignar una clave diferente para cada cliente. Desde el punto de vista de la seguridad de la información no es necesario emplear claves diferentes para cada archivo ya que el acceso a cada fichero se controla de forma independiente a través de IAM.

El uso de claves gestionadas por Google en Google Cloud Storage ofrece numerosas ventajas frente a las claves proporcionadas por el cliente. Estas claves, administradas automáticamente por Google, eliminan la necesidad de que los usuarios gestionen, almacenen y roten manualmente las claves, reduciendo así el riesgo de errores humanos y brechas de seguridad. Además, Google aplica automáticamente las mejores prácticas de seguridad, como el cifrado avanzado y la rotación regular, garantizando la protección de los datos en todo momento. También proporcionan una integración más sencilla con otros servicios de Google Cloud, mejorando la experiencia del usuario y reduciendo la complejidad operativa. En resumen, utilizar claves gestionadas por Google mejora la seguridad, simplifica la administración y optimiza la eficiencia en la gestión de claves de cifrado.

Una vez los datos están almacenados de forma segura en GCS, hay que definir los permisos de acceso para garantizar la separación lógica de dichos datos. Para ello se crean roles IAM personalizados o se utilizan roles predeterminados para acceso a GCS (por ejemplo, *roles/storage.objectViewer* para solo lectura). A continuación, se asignan estos roles a las Identidades de Servicio asociadas a las distintas VMs. Si se usa cifrado con claves de cliente (CMEK), se configuran permisos adicionales en Cloud KMS para controlar qué identidades pueden usar las claves de cifrado.

Con los permisos configurados, las VMs utilizan sus credenciales IAM para autenticarse con GCS, al intentar acceder a un archivo el sistema verifica los permisos IAM en el *bucket* y el archivo y los permisos de uso de claves si se usa Cloud KMS. Solo las VMs que cumplan todos los criterios podrán acceder a los archivos específicos.

Adicionalmente el uso de GCS nos permite activar los *Cloud Audit Logs* y el *Cloud Monitoring* para registrar el acceso a cada archivo y poder tener una trazabilidad de todos los datos generados por las distintas instalaciones. En definitiva, se ha diseñado un sistema de almacenamiento con seguridad en tres capas:

- **Cifrado:** Los datos se cifran en reposo con AES-256 y en tránsito forzando la transmisión por TLS.
- **Control de acceso:** Se emplea el principio de mínimo privilegio utilizando IAM para gestionar accesos granulares:
- **Auditoría:** Se registran y monitorizan todos los accesos.

Finalmente, recogemos a continuación los requisitos que debía cumplir el módulo de extracción de datos, y aportamos la justificación correspondiente:

TPD1	Seguridad de Comunicaciones	La transmisión de datos debe utilizar protocolos de comunicación encriptados y seguros, garantizando la confidencialidad y la integridad de los datos durante todo el proceso de transferencia.
------	-----------------------------	---

Como se ha explicado los datos se transmiten por SMTP, que no incluye cifrado por defecto, sin embargo, se utiliza TLS para proteger este canal de comunicación. Una vez recibidos los datos en GCP se almacenan cifrados en reposo tal y como se ha descrito anteriormente.

TPD2	Rapidez en la transmisión de los datos	Los datos deben ser transmitidos de manera eficiente, minimizando la latencia y asegurando que la información esté disponible para su procesamiento sin demoras significativas.
------	--	---

Al tratarse de un proceso de transmisión automatizado y utilizar SMTP la latencia será mínima, solo limitada por el ancho de banda de la conexión del PLC. La frecuencia de las cargas será configurable e inicialmente se acuerda hacer una transmisión semanal para cada instalación.

TPD3	Seguridad en el almacenamiento de los datos	Los datos recibidos deben ser almacenados en infraestructuras seguras, con medidas de protección que eviten accesos no autorizados, pérdidas o alteraciones de la información.
------	---	--

Como se ha explicado anteriormente los datos se almacenan en Google Cloud Storage con cifrado en reposo y controlando el acceso a los mismos mediante premisos granulares a nivel de recurso.

TPD4	Administración y gestión	Debe tener capacidades robustas para administrar y visualizar los datos en cada nodo donde se hayan guardado, permitiendo la supervisión de la información almacenada.
------	--------------------------	--

Cada nodo tendrá acceso a los datos de su instalación en GCS, desde su *Jupyter Notebook* es posible visualizar, analizar y procesar estos datos. De este modo, la gestión de permisos de la aplicación garantiza que el acceso al nodo esté correctamente restringido.

4.3. Despliegue Plataforma Federada.

Para este caso de uso, se ha diseñado una arquitectura *multi-cloud*, dado que no se puede garantizar que todos los clientes cuenten con una solución en la nube homologada. Algunos clientes requerirán que el sistema se despliegue en sus propios servidores locales. Como resultado, será necesario soportar una variedad de entornos y modalidades de conexión entre los distintos clientes, adaptando la infraestructura para garantizar su interoperabilidad y eficiencia.

4.3.1. Cloud-Edge

En el proyecto actual se utilizó Google Cloud como infraestructura para el demostrador, pero es importante destacar que el software desarrollado tiene la flexibilidad de ser desplegado en diferentes entornos, adaptándose a las necesidades del cliente. Esto incluye la posibilidad de implementarlo en nodos Edge (servidores, ordenadores) cercanos al cliente o incluso dentro de las instalaciones del propio cliente. En el primer caso, el cliente solo necesitaría solicitar un nodo de Cloud Edge a un proveedor de nube pública, como Google u otros equivalentes en el mercado. En el segundo caso, el cliente tendría la opción de implementar y operar su propio nodo de Edge en sus instalaciones utilizando hardware especializado. Esta flexibilidad garantiza que el software pueda adaptarse a distintas configuraciones y demandas de operación, ofreciendo una mayor escalabilidad y personalización en su uso.

Para el actual demostrador, la implementación en un nodo local (on-premises) no resulta práctica, dado que no aporta beneficios significativos en el contexto actual del proyecto. Esto podría cambiar en futuras versiones del software, especialmente si se plantea utilizar los resultados del aprendizaje automático (ML) en tiempo real para ajustar el estado operativo de diferentes máquinas. En un escenario así, la reducción de la latencia gracias a un nodo local

podría ser crítica para mejorar la capacidad de respuesta y la eficiencia operativa. Aunque esta funcionalidad va más allá de los objetivos originales del proyecto, representa una dirección interesante para explorar en futuras fases, ampliando aún más las capacidades y aplicaciones del software desarrollado.

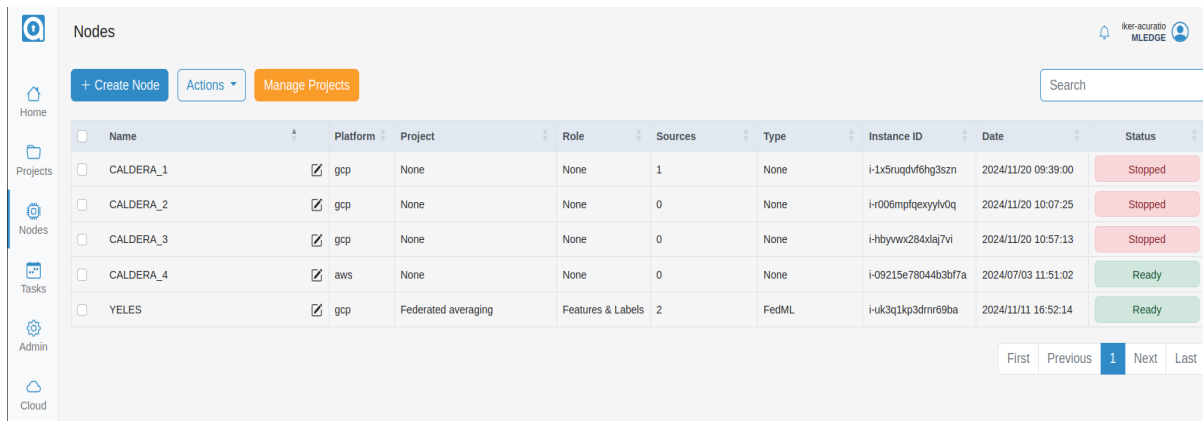
4.3.2. Despliegue

El proceso de despliegue se ha llevado a cabo siguiendo una serie de pasos meticulosamente definidos, con el fin de asegurar una implementación exitosa y alineada con los requisitos técnicos y operativos del proyecto:

1. **Instalación del software:** Se ha procedido a la instalación del software necesario para el funcionamiento de la solución en todos los entornos previstos.
2. **Instalación de los nodos de procesamiento:** Se ha configurado y desplegado la infraestructura de nodos de procesamiento, garantizando su capacidad para procesar los datos de manera eficiente.
3. **Configuración de bases de datos:** Se han instalado y configurado las bases de datos, asegurando que los datos se almacenen y gestionen de forma segura.
4. **Conexión con las APIs de coordinación:** Se han establecido las conexiones que permiten la coordinación entre los diferentes módulos del sistema federado.
5. **Configuración del *front-end*:** Se ha diseñado y configurado la interfaz de usuario para que los usuarios puedan interactuar de forma intuitiva y eficaz con el sistema.
6. **Integración con los servicios de cada cliente/nube:** Se ha integrado el sistema con los servicios específicos de cada cliente, en este caso se han lanzado todos los recursos en la nube pública de Acuratio.
7. **Configuración de las comunicaciones:** Se han establecido los protocolos de comunicación necesarios para asegurar una transmisión de datos segura y eficiente entre los distintos componentes del sistema.
8. **Testeo de carga y resiliencia:** Se han realizado pruebas de carga y resiliencia para verificar que el sistema sea capaz de manejar grandes volúmenes de datos y garantizar su estabilidad bajo diferentes condiciones operativas.
9. **Pruebas de seguridad:** Se ha llevado a cabo un exhaustivo análisis de seguridad para identificar y mitigar posibles vulnerabilidades, asegurando que el sistema sea robusto frente a amenazas externas e internas.
10. **Configuración de usuarios y permisos de acceso:** Se han configurado los roles de usuario y los permisos de acceso, garantizando un control adecuado sobre la información y los recursos del sistema.
11. **Formación y capacitación de usuarios:** Se han desarrollado materiales formativos, seminarios y sesiones de capacitación para asegurar que los usuarios finales comprendan y puedan utilizar correctamente la solución.
12. **Documentación:** Se ha generado toda la documentación técnica y de usuario necesaria, proporcionando una guía completa sobre la configuración, el uso y el mantenimiento del sistema.

Adicionalmente, como parte del compromiso con el proyecto, Acuratio proporcionará un soporte continuo a lo largo de todo su ciclo de vida. Esto incluirá:

1. **Soporte:** Asistencia técnica continua para resolver cualquier incidencia que pueda surgir durante la operación del sistema.
2. **Mantenimiento continuo:** Servicios de mantenimiento para garantizar que el sistema se mantenga actualizado, optimizando su rendimiento y seguridad.
3. **Monitorización:** Vigilancia constante del sistema para asegurar su correcto funcionamiento, detectar posibles fallos y mejorar la eficiencia operativa.



Name	Platform	Project	Role	Sources	Type	Instance ID	Date	Status
CALDERA_1	gcp	None	None	1	None	i-1x5ruqdvf0hg3szn	2024/11/20 09:39:00	Stopped
CALDERA_2	gcp	None	None	0	None	i-006mpfqxyyv0q	2024/11/20 10:07:25	Stopped
CALDERA_3	gcp	None	None	0	None	i-hbyvwx284xla7vi	2024/11/20 10:57:13	Stopped
CALDERA_4	aws	None	None	0	None	i-09215e78044b3bf7a	2024/07/03 11:51:02	Ready
YELES	gcp	Federated averaging	Features & Labels	2	FedML	i-uk3q1kp3dmr69ba	2024/11/11 16:52:14	Ready

Ilustración 7: Nodos desplegados en la plataforma de Acuratio.

El *Federated Learning* ayuda a proteger los datos sensibles porque las informaciones personales nunca abandonan los dispositivos donde se generan, lo que minimiza el riesgo de exposición o mal uso. En lugar de enviar los datos, el aprendizaje federado comparte únicamente actualizaciones del modelo en forma de parámetros agregados, que son menos susceptibles de ser utilizados para identificar a individuos. Esto es especialmente valioso en el contexto de protección de datos bajo el Reglamento General de Protección de Datos (RGPD) de la Unión Europea.

4.4. Exploración de datos

En esta fase preliminar la exploración de datos se ha limitado a la selección de variables en base a la experiencia previa sobre el funcionamiento de calderas de vapor. De las variables disponibles se han seleccionado seis:

1. **Caudal de vapor producido:** Mide la cantidad de vapor producido por la caldera en tiempo real. Este dato es fundamental para evaluar el trabajo de la caldera y asegurar que la producción de vapor cumple con los requerimientos del proceso sin incurrir en excesos o deficiencias. Permite calcular la relación entre el combustible consumido y el vapor generado (eficiencia térmica), ayudando a identificar ineficiencias en la conversión de energía. Además, monitorizar el caudal de vapor ayuda a detectar posibles fugas o pérdidas en la red de distribución, permitiendo su corrección oportuna para evitar desperdicios.
2. **Sonda lambda:** Mide la concentración de oxígeno en los gases de combustión, lo que permite ajustar la mezcla aire-combustible para lograr una combustión óptima. Una combustión eficiente reduce el consumo de combustible y minimiza emisiones contaminantes. Detecta desviaciones en la proporción de oxígeno, que pueden indicar exceso de aire o combustible, lo que ayuda a evitar pérdidas energéticas y optimizar el consumo.
3. **Temperatura de los humos de la chimenea:** Proporciona un indicador clave de las pérdidas térmicas en los gases de escape. Temperaturas excesivamente altas pueden sugerir que una parte significativa de la energía no está siendo aprovechada. Permite identificar la necesidad de ajustes en la operación o en el mantenimiento, como la limpieza de tubos de intercambio térmico, para mejorar la transferencia de calor y reducir el desperdicio energético.

4. **Presión de vapor:** Monitoriza la presión operativa de la caldera, asegurando que el sistema trabaja dentro del rango óptimo para evitar sobrecargas o ineficiencias en el proceso de generación de vapor. Ayuda a identificar patrones de consumo de vapor y posibles fugas en la instalación, lo que contribuye a reducir pérdidas y mejorar la planificación energética.
5. **Caudal de gas corregido:** Registra el consumo exacto de gas teniendo en cuenta variables como temperatura y presión, lo que mejora la precisión en el control de combustible. Facilita el análisis detallado del consumo de gas en función de la demanda de vapor, permitiendo identificar excesos y áreas donde se puede optimizar el uso del combustible.
6. **Temperatura de agua de alimentación de la caldera:** Mide la temperatura del agua que alimenta la caldera, un parámetro crítico para garantizar que el sistema opera de manera eficiente. Al conocer la temperatura del agua de entrada, se pueden optimizar los ciclos de calentamiento, minimizando el esfuerzo requerido por la caldera para alcanzar la temperatura de operación. Permite evaluar si el agua de alimentación proviene adecuadamente del retorno de condensados (que debería estar a una temperatura más elevada) o si es necesario recalentarla. Optimizar esta variable reduce el consumo de combustible, ya que el agua precalentada requiere menos energía para convertirla en vapor.

5. Solución preliminar

5.1. Sistema de control

El diseño e implementación del sistema de control desarrollado en este proyecto destacan por su adaptabilidad y escalabilidad, características que lo posicionan como una solución versátil y aplicable a diversos entornos industriales más allá del monitoreo de calderas de vapor. Esta solución, fundamentada en los requerimientos no funcionales definidos en el marco del proyecto MLEDGE, permite que las instalaciones evolucionen de manera eficiente y segura, respondiendo a las necesidades dinámicas de los clientes y garantizando un alto rendimiento incluso en escenarios futuros.

5.1.1. Escalabilidad: Un enfoque más amplio.

La arquitectura del sistema, inicialmente diseñada para la monitorización de calderas, ha sido concebida con una flexibilidad que permite extrapolarla a otros tipos de instalaciones industriales o terciarias. Esto se logra mediante la maleabilidad de los componentes del sistema de control, que pueden integrarse en entornos diversos como plantas de producción, sistemas de refrigeración, redes de distribución de energía, entre otros.

La capacidad de escalar el sistema incluye:

- **Ampliación de nodos:** Es posible agregar sensores y equipos de monitoreo en nuevas áreas o instalaciones, aumentando progresivamente el alcance del sistema sin comprometer el rendimiento.
- **Adaptación a distintos entornos:** La solución puede ser personalizada para capturar y procesar datos específicos de cada aplicación, como la calidad del aire en sistemas HVAC, niveles de líquidos en tanques de almacenamiento o eficiencia energética en redes de distribución eléctrica.
- **Incremento en la capacidad de procesamiento:** La infraestructura en la nube está preparada para gestionar un aumento en la cantidad de datos recopilados, sin importar el volumen generado por nuevas instalaciones o sistemas conectados.



Ilustración 8: Ilustración sistema de control.

5.1.2. Rendimiento y Eficiencia Operativa

El sistema mantiene un enfoque en el rendimiento, garantizando que los datos recolectados sean procesados y transmitidos en tiempo real, independientemente de la naturaleza de la instalación. Por ejemplo, en una planta de ensamblaje, los tiempos de respuesta críticos se pueden mantener dentro de los límites aceptables para no interrumpir los ciclos de producción.

El control y la optimización de variables clave, como el consumo energético, la eficiencia térmica o la monitorización de parámetros ambientales, se traducen en un valor tangible para los clientes al permitirles reducir costes operativos y tomar decisiones informadas basadas en datos confiables.

5.1.3. Seguridad y Usabilidad Adaptadas a Escenarios Variados

La solución ha sido diseñada considerando estrictos protocolos de seguridad para proteger la integridad de los datos y la confidencialidad de la información recopilada. Al expandirse a nuevas aplicaciones, el sistema continúa utilizando cifrado avanzado y permisos granulares para asegurar que los datos solo sean accesibles por usuarios autorizados.

Asimismo, se mantiene un enfoque en la usabilidad: las interfaces y herramientas de análisis se adaptan fácilmente a diferentes industrias, lo que facilita su adopción por clientes con distintos niveles de experiencia técnica. Este diseño versátil asegura que el sistema sea útil tanto para usuarios expertos como para aquellos que requieren soluciones intuitivas y accesibles.

5.1.4. Evolución hacia una Solución Generalizada

El verdadero potencial del sistema se revela en su capacidad de evolucionar más allá de su uso inicial en calderas. La integración con otras infraestructuras permite a los clientes monitorear y optimizar instalaciones de diferente índole, como:

- **Sistemas de transporte y logística:** Rastreo y análisis de eficiencia en vehículos industriales.
- **Plantas de tratamiento de agua:** Monitorización de caudal, pH y conductividad en tiempo real.
- **Centros de datos:** Control de temperatura, humedad y consumo energético para optimizar el funcionamiento de servidores.

Este enfoque generalizado transforma el sistema en una herramienta esencial para la industria 4.0, donde la interconectividad y el análisis de datos en tiempo real son fundamentales para la competitividad.

En conclusión, la solución desarrollada no solo cumple con los objetivos originales del proyecto, sino que también abre la puerta a una amplia gama de aplicaciones industriales.

5.2. Desarrollo del modelo ML

Tal como se indicó en el Entregable 3.1 titulado “*Diseño de un sistema de supervisión de eficiencia energética en calderas de vapor (caso de uso de economía tradicional) y el uso de componentes MLEDGE*”, el desarrollo del modelo de ML tiene como objetivo principal transformar los datos recopilados en información valiosa que permita optimizar el rendimiento, identificar anomalías y predecir necesidades de mantenimiento en calderas de vapor y otros equipos industriales.

La gestión eficiente de la producción y el consumo de vapor es un factor clave en numerosos procesos industriales, ya que su optimización representa una ventaja competitiva significativa. En este sentido, la previsión de la producción de vapor se convierte en un aspecto crítico para asegurar el rendimiento óptimo de las calderas industriales. Al aplicar técnicas avanzadas como el uso de redes neuronales, es posible desarrollar un modelo predictivo robusto capaz de anticipar con precisión las fluctuaciones en la demanda de vapor. Un funcionamiento subóptimo de las calderas de vapor puede producirse si la producción de vapor no se ajusta a los parámetros predefinidos para los cuales la máquina fue configurada inicialmente.

Para el desarrollo de este modelo se han utilizado datos provenientes de la instalación correspondiente al caso de uso del sector de la lavandería industrial. Esta instalación ha sido la primera en ser capaz de aportar datos y por lo tanto se ha decidido utilizar esta información para comprobar el correcto funcionamiento del proceso y la viabilidad del caso de uso propuesto. En el próximo entregable se integrarán datos de todas las instalaciones.

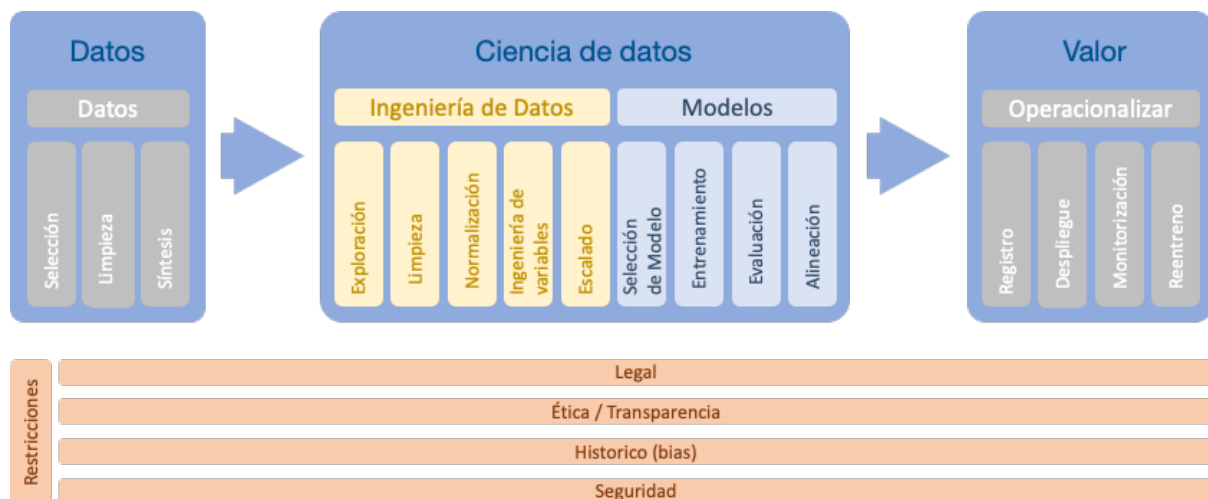


Ilustración 9: Ciencia de datos, proceso de transformación datos a modelos avanzados de ML.

Contar con una predicción precisa del rendimiento de la máquina en función de su configuración y producción de vapor es esencial. Esta capacidad de anticipación facilita la adaptación proactiva de la configuración de la máquina ante posibles variaciones en la producción, ya sea un incremento o una disminución, que puedan afectar su rendimiento.

El desarrollo de este modelo se ha llevado a cabo en los nodos de Acuratio utilizando *Jupyter Lab*, una herramienta que ofrece gran flexibilidad para ejecutar código Python directamente en el nodo. El proceso de programación se ha realizado en Python utilizando notebooks de *Jupyter Lab*.

Python es un lenguaje altamente versátil, adecuado para una amplia gama de tareas, desde la limpieza y preparación de datos hasta el desarrollo de modelos avanzados de *Machine Learning*. Además, Python dispone de una amplia variedad de bibliotecas especializadas en este ámbito. Para el desarrollo de este modelo, se han utilizado bibliotecas como *pandas*, *matplotlib*, *numpy*, *scikit-learn* y *TensorFlow*.

Por otro lado, los notebooks de Jupyter ofrecen múltiples ventajas, como la posibilidad de ejecutar y modificar código de manera interactiva. También permiten combinar código ejecutable con texto explicativo, lo que facilita tanto la documentación como la visualización de los resultados de manera dinámica.

5.2.1. Modelo predictivo de la producción de vapor

Para el desarrollo del modelo, el primer paso fue la selección de las variables. De entre las variables disponibles, se han elegido la sonda lambda, el caudal del gas corregido, la temperatura de los humos de la chimenea, la presión del vapor y el caudal de vapor. Esta selección se ha basado en el conocimiento teórico sobre el funcionamiento de las calderas, aunque está sujeta a posibles ajustes en entregables posteriores.

En cuanto a la división de los datos, estos se han separado en conjuntos de entrenamiento, validación y prueba. Determinar el porcentaje de datos que no se utilizarán durante el proceso de entrenamiento es fundamental para evaluar adecuadamente la eficiencia del modelo y evitar que los resultados sean artificialmente optimistas.

Finalmente, se ha optado por el uso de algoritmos de redes neuronales recurrentes (RNN), conocidas por su eficacia en la modelización de datos secuenciales, como es el caso de las series temporales. En particular, se desarrollará un modelo de *Long Short-Term Memory* (LSTM), reconocido por su capacidad para aprender, procesar y clasificar datos secuenciales, lo que lo hace especialmente adecuado para predecir series no lineales a largo plazo.

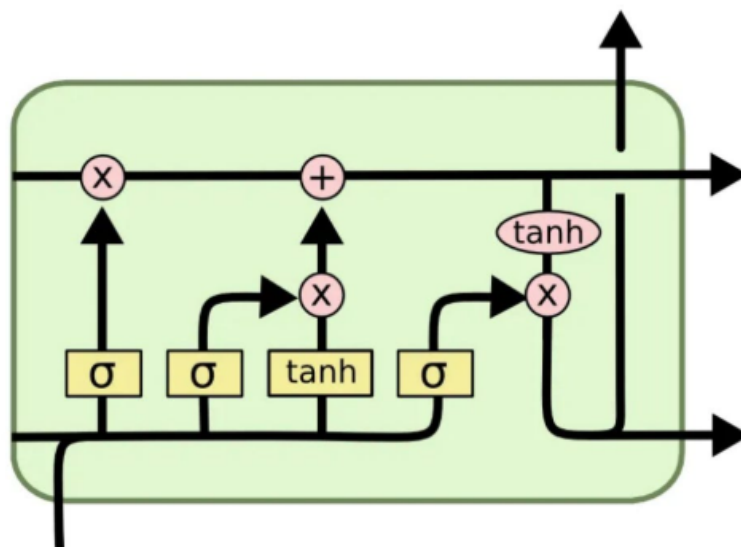


Ilustración 10: Diagrama de una celda LSTM

Una LSTM es un tipo de red neuronal recurrente diseñada para aprender dependencias a largo plazo en secuencias de datos. Se compone de "celdas" que tienen puertas para controlar el flujo de información: la puerta de entrada decide qué información agregar al

estado de la celda, la puerta de olvido regula qué información descartar, y la puerta de salida determina qué parte del estado de la celda usar como salida. Esta estructura permite a las LSTM mantener información relevante durante muchas iteraciones, evitando problemas comunes como el desvanecimiento del gradiente en RNN estándar. En la figura anterior se puede observar el diseño de una de estas celdas.

En el contexto de predicción de series temporales, las LSTM son especialmente útiles porque pueden aprender patrones complejos en los datos, como tendencias, estacionalidades y relaciones no lineales. Al procesar la serie temporal como una secuencia de puntos, las LSTM pueden captar cómo los valores pasados influyen en los futuros, proporcionando predicciones precisas incluso en datos con dependencias a largo plazo.

En la siguiente imagen se puede observar la comparación entre el valor real del caudal de vapor real y su predicción, usando los datos recogidos de una caldera durante un día aproximadamente. Para entrenar este modelo se ha utilizado la técnica de *sliding window*.

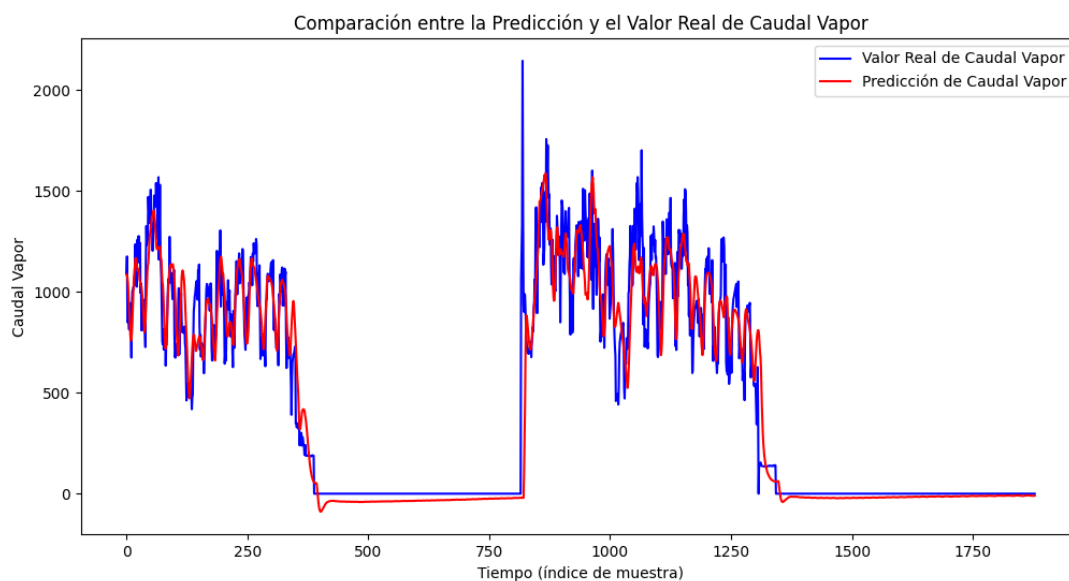


Ilustración 11: Comparación entre la predicción del caudal de vapor y los valores registrados reales

La técnica de *sliding window* para predicción de series temporales transforma los datos secuenciales en un conjunto de instancias supervisadas. Se divide la serie en ventanas de un tamaño fijo, donde cada ventana contiene un número determinado de valores pasados (historia), que se usan como características para predecir el valor siguiente (o futuros). Por ejemplo, en una ventana de tamaño 3, los valores $[x_1, x_2, x_3]$ servirían para predecir x_4 . En la siguiente figura se muestra una representación de la segmentación de un conjunto de datos mediante esta técnica.

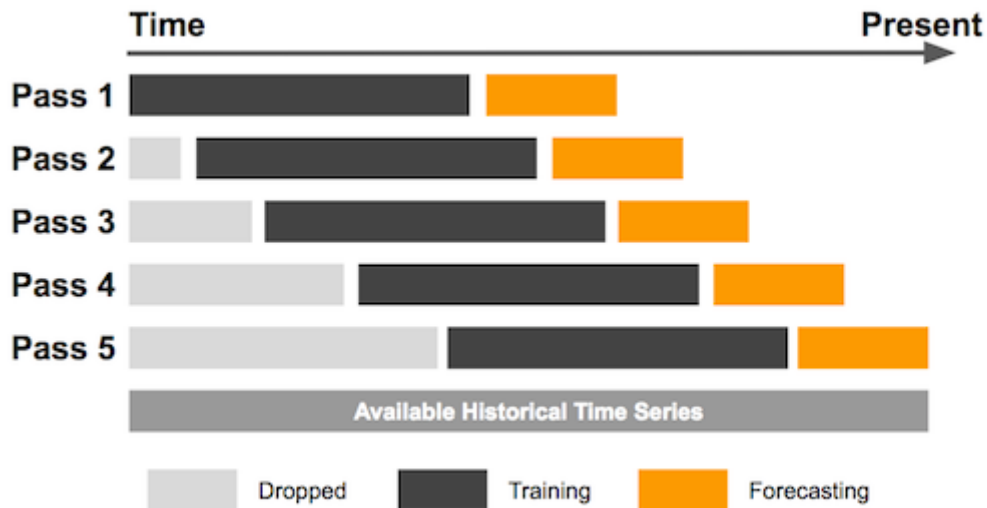


Ilustración 12: Diagrama método Sliding Window.

Este enfoque es especialmente útil para predecir series temporales, ya que permite manejar datos secuenciales como entradas fijas. Al mover la ventana de forma deslizante a lo largo de la serie (de ahí su nombre), se generan múltiples instancias de entrenamiento, capturando patrones locales y relaciones entre los puntos temporales. En este caso se ha definido una ventana de 60 minutos que se emplea para predecir la producción de vapor en los 5 minutos inmediatamente posteriores.

5.2.2. Modelo predictivo del rendimiento de la caldera

Al igual que en el modelo anterior, se han seleccionado las variables de la sonda lambda, el caudal del gas corregido, la temperatura de los humos de la chimenea, la presión del vapor y el caudal de vapor. Para este modelo, además, es necesario crear una nueva variable que represente el rendimiento de la caldera. Esta variable se calculará a partir de las variables de caudal de vapor y caudal de gas corregido, definiéndose el rendimiento de la caldera como el cociente entre el caudal acumulado de vapor y el caudal acumulado de gas.

En cuanto a la división de los datos, estos se segmentan en tres conjuntos: entrenamiento, validación y prueba. Se ha establecido que un 20% de los datos no se expondrán al modelo durante el entrenamiento, siendo estos datos reservados para el conjunto de prueba. Este último se utilizará para evaluar la eficiencia del modelo.

Finalmente, en cuanto a la elección del algoritmo, se ha optado por el modelo LSTM, al igual que en el caso anterior. Además, se ha implementado la técnica de *sliding window*, dado que seguimos trabajando en la predicción de series temporales a largo plazo.

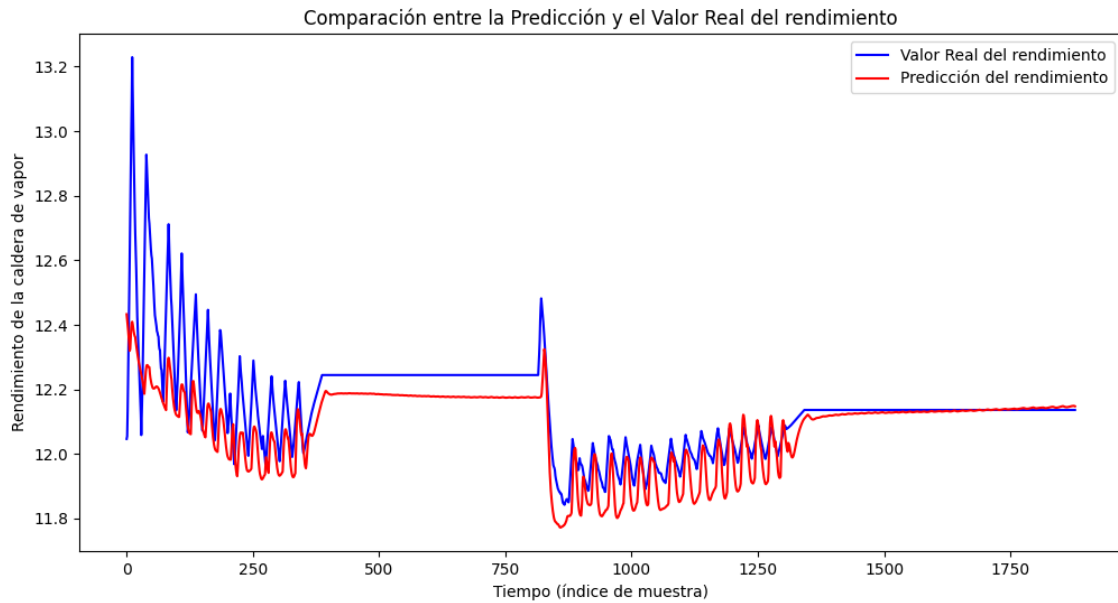


Ilustración 13: Comparación de la predicción del rendimiento y su valor real registrado.

En la imagen anterior se puede observar la comparación entre el rendimiento real de la caldera de vapor y su predicción, usando los datos recogidos de una caldera durante un día aproximadamente. Este modelo se ha entrenado para que teniendo en cuenta lo sucedido durante una hora pueda predecir qué pasará en los siguientes cinco minutos.

En la siguiente tabla se presentan los valores Mean Squared Error (MSE) que es el error cuadrático medio y el Mean Absolute Error (MAE) que indica el error absoluto medio de ambos modelos. Para interpretar estos datos hay que tener en cuenta que para entrenar el modelo se han escalado los datos entre 0 y 1, para ello se usó 'MinMaxScaler'.

	Predicción caudal de vapor		Rendimiento de la caldera	
	Entrenamiento	Validación	Entrenamiento	Validación
MSE	0.0062	0.0008	0.0001	0.0001
MAE	0.0479	0.0283	0.0046	0.0048

5.3. Implementación de FL

En esta fase del proceso, el modelo previamente diseñado y validado de manera local en la etapa anterior será trasladado a un entorno federado. Esto implica replicar un escenario en el que diversas fuentes de datos contribuyen al entrenamiento de un modelo compartido, mientras se preserva la privacidad de cada conjunto de datos, evitando su transferencia directa.

Dado que actualmente se encuentra en fase de finalización el despliegue y la adquisición de datos para iniciar el diseño y las pruebas del modelo federado, se ha optado por realizar una

simulación utilizando datos sintéticos. Este enfoque permite avanzar en el desarrollo del modelo sin esperar la completa disponibilidad de datos reales. Para ello, los datos disponibles se han dividido en dos subconjuntos, cada uno de los cuales simula los datos provenientes de una caldera diferente. Este método emula un entorno federado, ofreciendo una base para analizar y evaluar el comportamiento del modelo en este tipo de escenario.

El proceso de federación del modelo se implementará utilizando el método conocido como *Federated Averaging*.

5.3.1. Federación del Modelo

Federated Learning (o aprendizaje federado) es un enfoque de aprendizaje automático descentralizado que permite entrenar modelos utilizando datos que permanecen localizados en dispositivos o servidores distribuidos, sin necesidad de transferirlos a un servidor central. A continuación, se describe de manera detallada el proceso de entrenamiento.

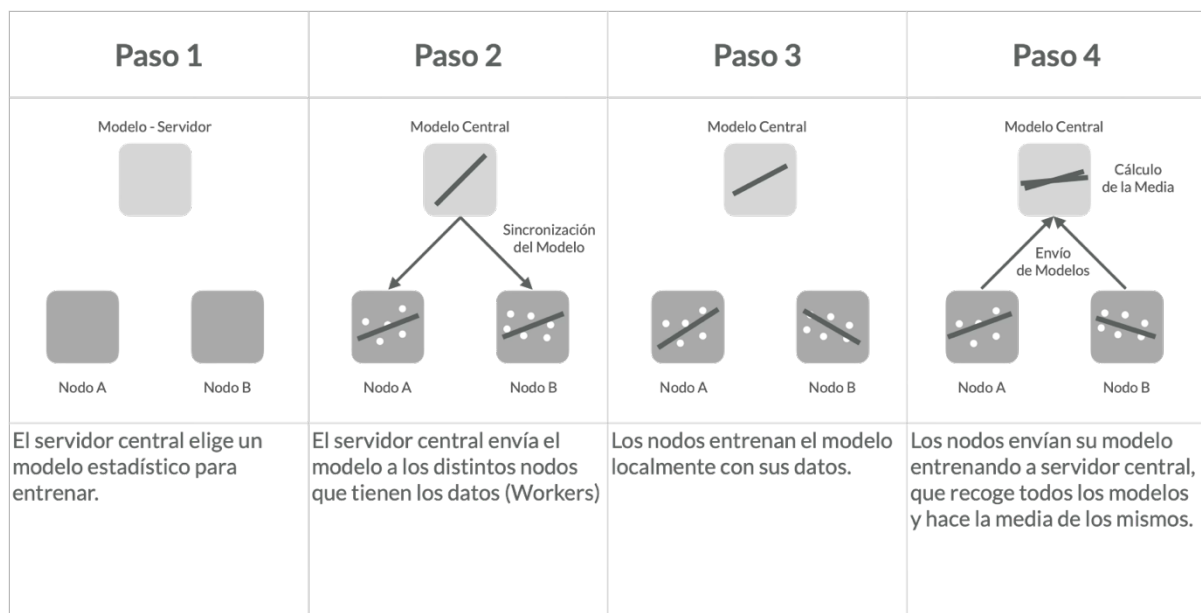


Ilustración 14: Proceso de entrenamiento Federated Learning

El entrenamiento en cada nodo no difiere mucho de lo que es el entrenamiento de una red neuronal local. Lo que lo diferencia es la agregación de las actualizaciones de los pesos en el servidor cada cierto número de pasos de entrenamiento a elección del usuario.

En primer lugar, se inicializan los pesos del modelo de manera aleatoria en todos los nodos participantes. Una vez inicializados los pesos, cada nodo seleccionado realiza la propagación hacia adelante utilizando los datos locales que tiene disponibles.

Durante este proceso, el nodo calcula las activaciones de cada capa del modelo y genera una predicción basada en su conjunto de datos local. A continuación, se calcula el error correspondiente y se actualizan los pesos del modelo en el nodo. Este procedimiento de entrenamiento local se repite en cada nodo durante varias iteraciones hasta alcanzar el número de pasos definidos para el entrenamiento antes de realizar la agregación.

Tras completar el entrenamiento local, cada nodo envía las actualizaciones de sus pesos al servidor central. Estas actualizaciones pueden incluir los gradientes del error local o los pesos

ajustados del modelo después de aplicar un algoritmo de optimización, como el descenso de gradiente estocástico. Este intercambio de información es esencial para combinar los avances realizados en cada nodo individual.

En el servidor central, se lleva a cabo un proceso de agregación de las actualizaciones de pesos recibidos de los nodos participantes. Este promedio ponderado considera el tamaño del conjunto de datos local de cada nodo, lo que permite que las contribuciones reflejen adecuadamente la importancia relativa de los datos disponibles en cada nodo. Este paso es crucial para construir un modelo global que represente de manera equilibrada los datos distribuidos.



Ilustración 15: Diagrama de entrenamiento federado.

Con el promedio ponderado calculado, el servidor central actualiza los pesos del modelo global. Los nuevos pesos son luego enviados de regreso a los nodos participantes, asegurando que todos inicien la siguiente iteración de entrenamiento con la misma configuración de pesos. Este proceso fomenta la convergencia del modelo global al integrar progresivamente las contribuciones de todos los nodos. En la figura anterior puede verse una representación del proceso de aprendizaje federado entre servidores distribuidos al rededor del mundo.

Los pasos descritos (entrenamiento local, envío de actualizaciones, agregación y redistribución de pesos) se repiten durante varias rondas de entrenamiento. Este enfoque iterativo permite que el modelo global se ajuste gradualmente a los datos distribuidos en los nodos participantes, mejorando su capacidad predictiva con cada iteración.

Finalmente, el proceso de entrenamiento se considera completo una vez que se han alcanzado un número predefinido de rondas o se cumple un criterio de convergencia específico. En este punto, el modelo global está listo para ser utilizado en la predicción de nuevos datos, aprovechando el conocimiento obtenido a partir del entrenamiento en los nodos distribuidos.

En un entorno federado, tenemos dos tipos de nodo: los que entrenan los modelos locales y el coordinador central. En este estudio cada caldera es un nodo que entrena un modelo local.

A continuación, se muestra el código del coordinador. En él se define el modelo y se federa con el algoritmo *Federated Averaging*, estableciendo la frecuencia con la que se actualiza el modelo global. Es decir, la actualización del modelo global se realiza en este caso, tras 5 entrenamientos locales.

```

DEFINE MODEL

[2]: input_layer = layers.InputLayer(input_shape=(60, 5))
net = LSTM(units=50, return_sequences=True, input_shape=(60, 5))(input_layer.output)
net = LSTM(units=50, return_sequences=False)(net)
net = Dense(units=25)(net)
net = Dense(units=1)(net)

model = federatedmodel.FederatedModel(
    inputs=input_layer.output,
    outputs=net,
)

# Compilar el modelo
model.compile(optimizer=Adam(0.001), loss='mean_squared_error')

1 workers connected. Waiting for 1 workers to connect
2 workers connected.
✓ Connecting workers

TRAIN MODEL

[3]: history = model.fit(
    avg_frequency=5,
)
    
```

Ilustración 16: Ejemplo del código del nodo coordinador.

Las librerías que se utilizan en este código para federar el modelo han sido desarrolladas como parte del proyecto MLEDGE dentro de las tareas asociadas al paquete de trabajo P2 por Acuratio. El soporte con la implementación, desarrollo y adaptación del modelo para su entrenamiento federado también forma parte de las actividades de dicho paquete de trabajo.

A continuación, se muestra también el código desplegado en el nodo cliente que va a entrenar sobre sus datos locales.

```

PREPROCESS DATA

[3]: #Lee el archivo CSV en un DataFrame.
df = pd.read_csv('./data/MLEDGE_YELES_DEM01.csv')
df = df.iloc[750:, :]

[4]: df = df.loc[:, ['CaldVapor_SondaLambdaOxigeno_1min(%)', 'CaldVapor_CaudalGasCorregido_1min(mA³/hr)',
                  'CaldVapor_TemperaturaHumosChimenea_1min(Å°C)', 'CaldVapor_PresionVapor_1min(bar)',
                  'CaldVapor_CaudalVapor_1min(mA³/hr)']]

[5]: # Escala los datos para mejorar el rendimiento del LSTM
scaler = MinMaxScaler(feature_range=(0, 1))
scaled_data = scaler.fit_transform(df)
num = len(scaled_data[0])-1

[6]: #Configura secuencias de entrada y salidas para el modelo LSTM
sequence_length = 60 # Número de pasos de tiempo (puedes ajustarlo)
X, y = [], []

for i in range(len(scaled_data)-sequence_length - 5):
    X.append(scaled_data[i:i + sequence_length, :])
    y.append(scaled_data[i + sequence_length + 5, -1])

X, y = np.array(X), np.array(y)

DEFINE MODEL

[7]: model = federatedmodel.FederatedModel()
    
```

Ilustración 17: Ejemplo de código del nodo cliente.

En primer lugar, se procesan los datos y se definen las variables de entrada y la salida esperada. En este caso, se seleccionan las variables antes comentadas y al modelo se le

proporciona una ventana temporal de datos correspondiente a los últimos 60 minutos para realizar una predicción sobre el comportamiento dentro de 5 minutos.

```

TRAIN MODEL

[9]: history = model.fit(
    X,
    y,
    epochs=15,
    batch_size=32,
    validation_split=0.2)

✓ Receiving initial weights
Epoch 1/15
29/29 [=====] - 15s 510ms/step - loss: 0.0771 - federated_loss: 0.1863 - val_loss: 7.5855e-05
Epoch 2/15
29/29 [=====] - 15s 531ms/step - loss: 0.0135 - federated_loss: 0.0163 - val_loss: 7.3037e-04
Epoch 3/15
29/29 [=====] - 14s 498ms/step - loss: 0.0119 - federated_loss: 0.0117 - val_loss: 0.0018
Epoch 4/15
29/29 [=====] - 15s 535ms/step - loss: 0.0110 - federated_loss: 0.0113 - val_loss: 0.0019
Epoch 5/15
29/29 [=====] - 15s 529ms/step - loss: 0.0114 - federated_loss: 0.0107 - val_loss: 0.0019
Epoch 6/15
29/29 [=====] - 14s 497ms/step - loss: 0.0106 - federated_loss: 0.0094 - val_loss: 3.7824e-04
Epoch 7/15
29/29 [=====] - 15s 526ms/step - loss: 0.0101 - federated_loss: 0.0095 - val_loss: 0.0025
Epoch 8/15
29/29 [=====] - 14s 498ms/step - loss: 0.0103 - federated_loss: 0.0107 - val_loss: 6.1566e-06
Epoch 9/15
29/29 [=====] - 16s 541ms/step - loss: 0.0101 - federated_loss: 0.0093 - val_loss: 1.3774e-05
Epoch 10/15
29/29 [=====] - 6s 205ms/step - loss: 0.0058 - federated_loss: 0.0067 - val_loss: 1.3537e-04
    
```

Ilustración 18: Proceso de entrenamiento del modelo.

Posteriormente, el modelo definido en el coordinador central se entrena de manera local. En este punto, se establecen los siguientes parámetros:

- *epoch*: Indica el número de veces que el modelo pasará por todo el conjunto de datos durante el entrenamiento.
- *batch_size*: Indica el número de datos que se procesarán en cada iteración de entrenamiento. En este caso, el conjunto de datos se divide en lotes de 32 unidades.
- *validation_split*: Representa el porcentaje de datos que no serán utilizados durante el entrenamiento, reservándose para la validación del modelo.

Al entrenar el modelo el software desarrollado muestra valores que permiten hacer el seguimiento del entrenamiento y comprobar si el modelo converge correctamente. En este caso se hace el seguimiento de las siguientes métricas:

- *loss*: Indica el promedio de las pérdidas locales calculadas en los nodos participantes durante una ronda de entrenamiento.
- *federated_loss*: Representa la pérdida calculada en el modelo global después de promediar las actualizaciones provenientes de los nodos.
- *val_loss*: Representa la pérdida del modelo global en un conjunto de validación independiente. Este conjunto de datos no se usa durante el entrenamiento y sirve únicamente para evaluar el desempeño del modelo.

Finalmente se evalúa el modelo comparando los valores que se han predicho con los valores reales, y se hace una gráfica para apreciar la comparativa de una manera más visual. Tras federar ambos modelos estos son los resultados obtenidos:

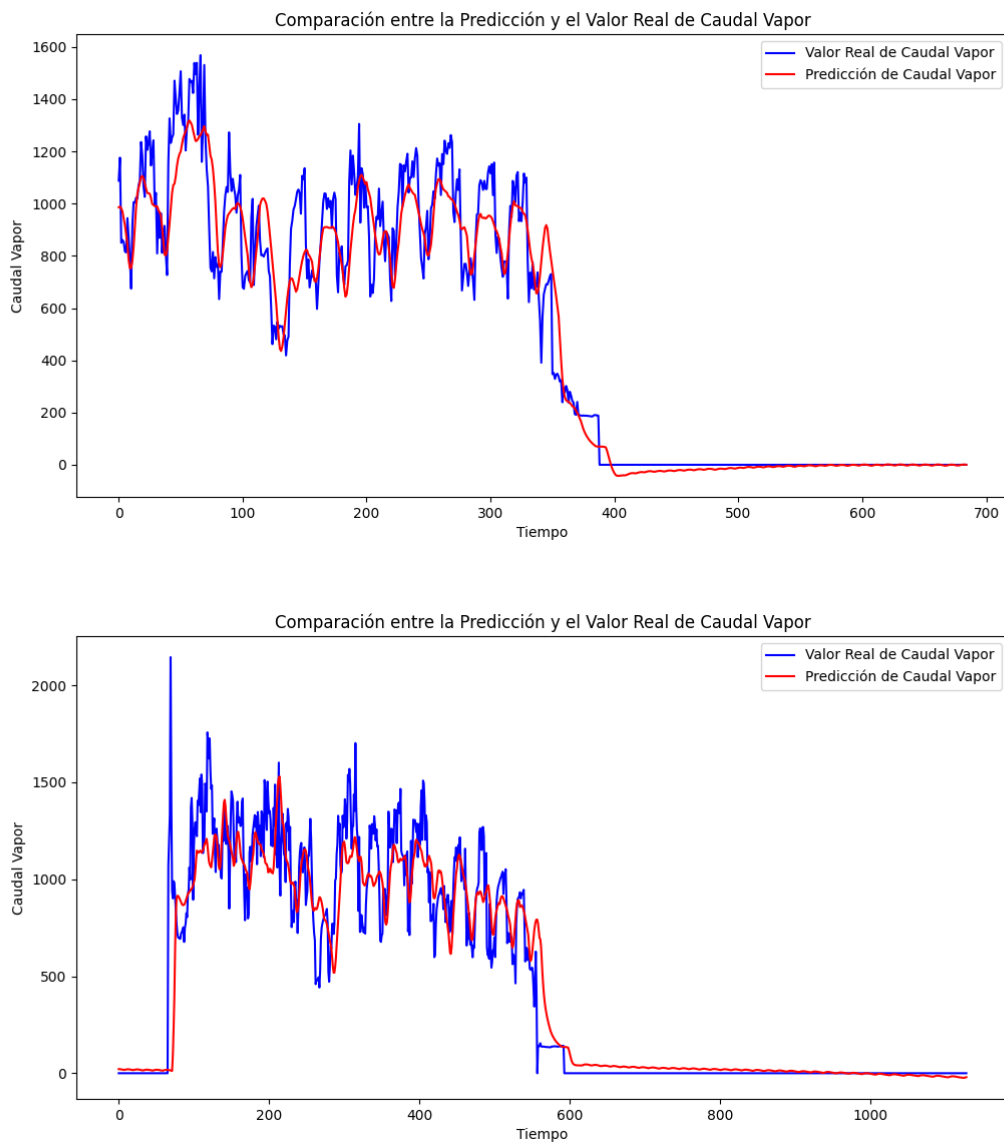


Ilustración 19: Predicciones del modelo federado sobre ambos conjuntos de datos.

Las gráficas anteriores representan los modelos de la predicción del caudal de vapor tras federarse. En este caso, como ya se ha mencionado se dividieron los datos en dos para simular dos calderas diferentes.

A continuación, se presentan las tablas donde se presentan los valores Mean Squared Error (MSE) que es el error cuadrático medio y el Mean Absolute Error (MAE) que indica el error absoluto medio del Worker A y Worker B que representan dos calderas diferentes.

En las tablas se reportan los errores de los modelos locales sobre los conjuntos de entrenamiento y validación, así como los errores del modelo federado sobre el conjunto de entrenamiento de cada worker. Para interpretar estos datos hay que tener en cuenta que para entrenar el modelo se han escalado los datos entre 0 y 1, para ello se usó `'MinMaxScaler'`.

Worker A			
	Entrenamiento	Validación	Modelo Federado
MSE	0.0067	0.0009	0.0076
MAE	0.0557	0.0302	0.0566

Worker B			
	Entrenamiento	Validación	Modelo Federado
MSE	0.0075	0.0008	0.0072
MAE	0.0551	0.0296	0.0560

6. Conclusión y siguientes pasos

La arquitectura técnica diseñada para este proyecto integra hardware y software para optimizar tanto la eficiencia operativa como la toma de decisiones. Los sensores instalados en las calderas recopilan datos críticos en tiempo real, los cuales son gestionados por Controladores Lógicos Programables (PLCs) que operan mediante el protocolo Modbus. La transmisión de datos se realiza de manera segura a través de la infraestructura de Google Cloud Platform (GCP), donde se almacenan en Google Cloud Storage (GCS) con cifrado para garantizar la confidencialidad y seguridad de la información. Los nodos de procesamiento han sido implementados en máquinas virtuales de GCP, las cuales cuentan con almacenamiento cifrado en reposo para proteger los datos sensibles.

Actividades realizadas en el marco de este entregable:

- Instalación de sensores.
- Configuración del PLC para la recopilación y transmisión de datos.
- Puesta en marcha de las instalaciones.
- Configuración de la nube para la recepción y almacenamiento de datos.
- Despliegue de la plataforma federada.
- Exploración de datos.
- Implementación del sistema de control.
- Desarrollo preliminar del modelo de aprendizaje automático (ML).
- Federación inicial del modelo de aprendizaje automático (ML).

Para alcanzar los objetivos establecidos, Acuratio ha implementado una capa de software distribuido FLaaS que permite a Inmarepro recolectar y analizar los datos generados en las instalaciones de sus clientes. Adicionalmente, se ha proporcionado la infraestructura necesaria para el despliegue de entornos federados del demostrador, así como soporte técnico para el entrenamiento y explotación de los modelos federados.

Los modelos de aprendizaje automático (*Machine Learning*) se han empezado a diseñar y trabajar con los datos iniciales de una de las calderas, demostrando la viabilidad del caso de uso planteado y la capacidad de la plataforma implementada para federar estos modelos. El proyecto tiene como objetivo validar que la aplicación del aprendizaje federado en entornos industriales es tanto factible como altamente beneficiosa. Con este propósito, se escalará para agregar los datos de todas las calderas instaladas y continuaremos trabajando en el perfeccionamiento de los modelos actuales, la limpieza de datos, y el diseño de variables que nos ayuden a mejorar iterativamente el modelo.

El siguiente reto es demostrar la escalabilidad de la plataforma, garantizando su capacidad para añadir participantes, resiliencia de las comunicaciones, testeo intensivo y entrenamiento automatizado de los modelos federados. Estas capacidades serán validadas en el próximo entregable; sin embargo, los avances alcanzados en esta etapa son prometedores.

En síntesis, MLEDGE busca generar un impacto significativo en la sostenibilidad y la reducción de emisiones, mientras promueve la competitividad industrial y fomenta la innovación en la gestión de recursos energéticos. La implementación de esta solución representará un avance crucial hacia una industria más eficiente y sostenible, con beneficios tangibles tanto para la economía como para el medio ambiente.