UNIVERSIDAD DE CONCEPCIÓN

Facultad de Ingeniería

Departamento de Ingeniería Informática y Ciencias de la Computación

DESARROLLO DE SOFT SENSORS Y HERRAMIENTAS PREDICTIVAS BASADAS EN DATOS Y MACHINE LEARNING PARA EL DIAGNÓSTICO OPERACIONAL Y LA MEJORA DEL DESEMPEÑO DE FILTROS DE LICOR VERDE EN PLANTA INDUSTRIAL

Tesis para optar al título de Ingeniero civil informático

Supervisor en la empresa: Nicolás Vera

Profesor guía: Hugo Garcés

Profesor co-guía: Ricardo Flores

Estudiante: Aníbal Polanco Ochoa

Concepción, Chile
24 de junio de 2026

Índice

Resumen

La reducción en el circuito de licor verde es una variable relevante para la operación de la caldera recuperadora, pero su observación presenta limitaciones prácticas. Por una parte, las mediciones de laboratorio tienen baja frecuencia y latencia. Por otra, el sensor físico en línea disponible en planta presenta problemas de continuidad, disponibilidad y calidad efectiva de señal. En este contexto, esta memoria aborda el problema desde la construcción y evaluación de un *soft sensor* capaz de estimar la reducción a partir de variables historizadas del proceso.

El trabajo se desarrolló sobre datos industriales del área de recuperación, integrando extracción, alineación temporal, depuración del target y construcción de espacios de entrada para modelamiento. Sobre esa base se compararon distintos regímenes de entrenamiento, distintos subconjuntos de variables, variantes con y sin retardos temporales, y dos formas principales de representación del problema: directa y mediante PLS. La evaluación se realizó bajo un esquema temporal estricto, con separación explícita entre entrenamiento, *test* y *holdout*, manteniendo al laboratorio como referencia principal para la medición del desempeño fuera de muestra.

Los resultados muestran que el problema sí admite soluciones predictivas útiles, pero que su comportamiento depende de cómo se combinan espacio de variables, representación, uso de lag y modo de entrenamiento. No aparece una receta única que domine en todos los escenarios. La mejor solución retenida se obtuvo en un subconjunto reducido de variables, con estructura temporal explícita y entrenamiento mixto, mientras que en espacios más amplios o intermedios la representación mediante PLS se mantuvo competitiva. En conjunto, el estudio respalda la factibilidad técnica del enfoque, confirma el valor de una evaluación temporal conservadora y muestra que la señal del sensor aporta más cuando se integra de forma controlada que cuando se la trata como reemplazo directo del laboratorio.

1. Introducción

En el proceso *kraft*, el circuito de recuperación química permite cerrar el ciclo de reactivos y, al mismo tiempo, recuperar energía útil para la operación de la planta. El licor negro proveniente del área de cocción se concentra en evaporadores y luego se quema en la caldera recuperadora. En esa etapa se genera vapor y se recuperan los compuestos inorgánicos del proceso. El fundido inorgánico o *smelt* se disuelve para formar licor verde y, posteriormente, mediante recaustificación, este se transforma en licor blanco para su reutilización en el digestor. En términos industriales, se trata de un circuito maduro, ampliamente utilizado y bien documentado dentro del proceso *kraft* (Smook, 2002; Whitney, 1968).

1.1. Contexto del circuito de recuperación *kraft*

Desde el punto de vista operativo, la secuencia relevante puede resumirse en cuatro etapas. Primero, el licor negro se concentra hasta alcanzar una condición adecuada para combustión. Segundo, en la caldera recuperadora ocurren la combustión de la fracción orgánica y las reacciones de reducción asociadas al azufre, generándose un *smelt* compuesto principalmente por Na_2CO_3 y Na_2S. Tercero, ese *smelt* se descarga hacia el disolvedor, donde se mezcla con licor débil o agua débil para formar licor verde. Finalmente, el licor verde pasa por clarificación y recaustificación para regenerar licor blanco, que vuelve al digestor. En conjunto, este esquema permite recuperar químicos de cocción y aportar al balance térmico de la planta (Smook, 2002; Whitney, 1968).

1.2. Definición operativa de reducción

Dentro de este circuito, una variable de especial interés es la eficiencia de reducción, o simplemente reducción. Esta cuantifica qué tan bien el sistema convierte el azufre oxidado en sulfuro de sodio, que es la forma química útil para el proceso *kraft*. En términos operativos, una formulación habitual es:

$$\eta_{red} = \frac{[Na_2S]}{[Na_2S] + [Na_2SO_4]} \times 100 \tag{1}$$

donde $[Na_2S]$ y $[Na_2SO_4]$ representan las concentraciones de sulfuro de sodio y sulfato de sodio, respectivamente. Un valor bajo de reducción sugiere que la zona de reducción de la caldera no está operando de manera favorable, lo que puede traducirse en mayor carga muerta en el circuito y en un deterioro del desempeño químico global. En la práctica, esta variable suele evaluarse en licor verde, por lo que resume tanto el resultado de las reacciones en la caldera como las condiciones posteriores de disolución del *smelt* (Smook, 2002; Whitney, 1968).

1.3. Problema práctico: latencias, discontinuidades y retardos físicos

Aunque la reducción es una variable relevante para evaluar el desempeño del circuito de recuperación, su observación operativa presenta limitaciones. En primer lugar, la referencia de laboratorio no entrega una señal continua ni inmediata, sino mediciones de menor frecuencia y con latencia respecto del estado real del proceso. Esto reduce su utilidad para seguimiento fino y para apoyo directo a decisiones operacionales en línea.

En segundo lugar, la variable observada no responde de manera instantánea a perturbaciones o cambios ocurridos en la caldera recuperadora. Entre la generación del *smelt*, su disolución y la posterior formación del licor verde, intervienen tiempos de residencia, mezcla y transporte dentro del circuito. Por lo mismo, la reducción medida en licor verde refleja el efecto acumulado de fenómenos ocurridos aguas arriba con cierto desfase temporal. En términos de operación y control, esto implica que la variable de interés combina baja frecuencia de observación con retardo de proceso.

En la práctica, esta combinación vuelve más difícil construir una lectura oportuna del estado del sistema usando solo laboratorio. Por eso, para fines de monitoreo y apoyo operacional, resulta natural explorar señales historizadas del proceso que permitan aproximar el valor de reducción con mayor continuidad, aun cuando la referencia final siga siendo la medición de laboratorio (Bequette, 2003; Smook, 2002).

1.4. Limitaciones del sensor físico en línea (contexto del sitio)

En la instalación estudiada existe un sensor físico en línea para reducción. Sin embargo, su uso como fuente principal de observación presenta limitaciones prácticas. De acuerdo con la experiencia del sitio, el equipo no está disponible de forma continua, muestra episodios de deriva o lecturas poco confiables y requiere contrastes frecuentes con laboratorio. A esto se suma un costo de propiedad alto, tanto por reposición tecnológica como por mantención e intervenciones.

En consecuencia, el problema no consiste solo en que la referencia de laboratorio sea poco frecuente, sino también en que la señal instrumental disponible no siempre entrega continuidad ni calidad suficiente para sostener una observación robusta en línea. Bajo ese contexto, un *soft sensor* resulta atractivo por dos motivos complementarios: puede entregar cobertura cuando el sensor físico no está disponible y, además, puede servir como apoyo para contrastar su comportamiento cuando la señal del equipo presenta degradación o deriva.

2. Marco Teórico

2.1. Conocimientos preliminares.

En esta parte se exponen los conocimientos preliminares considerados fundamentales. Algunos son la base de lo que se desarrolla más adelante en esta sección; otros se mantienen vigentes por sí mismos.

2.1.1. Fundamentos probabilísticos y estadísticos.

La base del aprendizaje estadístico reside en el manejo de la incertidumbre mediante las leyes de la probabilidad. Siguiendo a Bishop (2006), para dos variables aleatorias X (entrada) e Y (objetivo), la regla del producto y la regla de la suma permiten descomponer su distribución conjunta:

$$p(X, Y) = p(Y|X)p(X) = p(X|Y)p(Y) \tag{2}$$

En el contexto de regresión para el sensor de reducción, el objetivo es modelar la distribución condicional $p(y|\mathbf{x}, \mathbf{w})$, donde $\mathbf{w}$ representa los parámetros del modelo. Bajo la suposición de ruido Gaussiano aditivo con precisión β, la verosimilitud de un conjunto de datos $\mathcal{D}$ se expresa como:

$$p(\mathbf{t}|\mathbf{w}, \beta) = \prod_{n=1}^{N} \mathcal{N}(t_n|y(\mathbf{x}_n, \mathbf{w}), \beta^{-1}) \tag{3}$$

La maximización de esta verosimilitud es equivalente a la minimización de la suma de errores cuadrados (SSE), lo que proporciona la justificación probabilística al uso de la métrica MSE en este trabajo.

Convención de notación. En lo que sigue, X, Y denotan variables aleatorias discretas y x, y sus realizaciones; de forma análoga, $\mathbf{X}$ es un vector aleatorio y $\mathbf{x}$ una realización (dato observado). Por ejemplo, $p(x)$ abrevia $P(X = x)$ y $p(y \mid \mathbf{x})$ la masa condicional evaluada en $(y, \mathbf{x})$ (Cover & Thomas, 2006).

Datos y factorización conjunta. Sea $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{N}$ un conjunto de observaciones, donde $\mathbf{x}$ agrupa variables explicativas y y es la realización de la variable objetivo Y. La relación estadística entre ambas se describe mediante una distribución conjunta $p(\mathbf{X}, Y)$, que puede factorizarse como

$$p(\mathbf{X}, Y) = p(Y \mid \mathbf{X})\, p(\mathbf{X}). \tag{4}$$

Esta descomposición separa (i) el modelo condicional que interesa para predicción $p(Y \mid \mathbf{X})$ y (ii) la distribución de condiciones de operación observadas $p(\mathbf{X})$ (Bishop, 2006).

Probabilidad condicional, Bayes e independencia. Para variables discretas X, Y,

$$p(x \mid y) = \frac{p(x, y)}{p(y)} \qquad \text{con } p(y) > 0, \tag{5}$$

y la regla de Bayes permite reexpresar la inferencia como actualización de creencias: $p(y \mid x) \propto p(x \mid y)\, p(y)$ (Bishop, 2006). Decir que X e Y son independientes equivale a $p(x, y) = p(x)p(y)$ (o $p(x \mid y) = p(x)$); este punto es operativo porque condiciona qué dependencias puede capturar un modelo y qué estructura queda en el residuo.

Momentos y dependencia lineal. La esperanza y la varianza resumen nivel y dispersión. Para X discreta:

$$\mathbb{E}[X] = \sum_x x\, p(x), \qquad \mathrm{Var}(X) = \mathbb{E}\Big[(X - \mathbb{E}[X])^2\Big]. \tag{6}$$

Para X, Y, la covarianza cuantifica co-variación lineal:

$$\mathrm{Cov}(X, Y) = \mathbb{E}[(X - \mathbb{E}[X])(Y - \mathbb{E}[Y])], \tag{7}$$

y puede normalizarse mediante el coeficiente de correlación $\rho_{XY} = \mathrm{Cov}(X, Y)/\sqrt{\mathrm{Var}(X)\mathrm{Var}(Y)}$. En forma vectorial, $\Sigma = \mathrm{Cov}(\mathbf{X})$ concentra varianzas y covarianzas entre componentes.

Muestreo y estimación. Dada una muestra $\{x_i\}_{i=1}^N$, el promedio muestral

$$\bar{x} = \frac{1}{N} \sum_{i=1}^{N} x_i \tag{8}$$

se usa como estimador de $\mathbb{E}[X]$. En el mismo sentido, estimaciones de varianza/covarianza y métricas son cantidades calculadas sobre datos finitos; por ello su estabilidad depende del tamaño muestral y del esquema de evaluación (Bishop, 2006).

Modelo condicional, verosimilitud y MAP (regularización). Una formulación estándar propone una familia paramétrica $p(y \mid \mathbf{x}, \boldsymbol{\theta})$. Bajo el supuesto habitual de independencia condicional de las observaciones dado $\mathbf{x}_i$, la verosimilitud factoriza como

$$p(\mathcal{D} \mid \boldsymbol{\theta}) = \prod_{i=1}^{N} p(y_i \mid \mathbf{x}_i, \boldsymbol{\theta}), \qquad \boldsymbol{\theta}_{\mathrm{ML}} = \arg\max_{\boldsymbol{\theta}} \log p(\mathcal{D} \mid \boldsymbol{\theta}). \tag{9}$$

En un enfoque bayesiano, la actualización viene dada por

$$p(\boldsymbol{\theta} \mid \mathcal{D}) \propto p(\mathcal{D} \mid \boldsymbol{\theta})\, p(\boldsymbol{\theta}), \tag{10}$$

y el estimador MAP satisface

$$\boldsymbol{\theta}_{\mathrm{MAP}} = \arg\max_{\boldsymbol{\theta}} \log p(\mathcal{D} \mid \boldsymbol{\theta}) + \log p(\boldsymbol{\theta}) = \arg\min_{\boldsymbol{\theta}} \Big(- \log p(\mathcal{D} \mid \boldsymbol{\theta}) - \log p(\boldsymbol{\theta}) \Big), \tag{11}$$

haciendo explícita la lectura de $-\log p(\boldsymbol{\theta})$ como término de penalización (regularización) (Bishop, 2006).

Predicción y función de pérdida (base de regresión). La predicción puntual puede plantearse como un problema de decisión: para una pérdida $L(y, \hat{y})$, se elige $\hat{y}(\mathbf{x})$ minimizando la pérdida esperada bajo la distribución condicional del objetivo. En particular, con pérdida cuadrática se obtiene como solución óptima el promedio condicional

$$\hat{y}(\mathbf{x}) = \mathbb{E}[\, Y \mid \mathbf{X} = \mathbf{x} \,] = \sum_{y} y \, p(y \mid \mathbf{x}), \tag{12}$$

y al generalizar a pérdidas tipo Minkowski se recuperan soluciones ligadas a cuantiles (por ejemplo, mediana condicional para $q = 1$) (Bishop, 2006). Esta lectura fija el vínculo entre (i) supuestos sobre $p(y \mid \mathbf{x})$, (ii) pérdida optimizada y (iii) criterio con el que se reporta desempeño.

Residuo, error cuadrático y RMS. Dada una muestra y un predictor $\hat{y}_i = \hat{y}(\mathbf{x}_i)$, el residuo es $e_i = y_i - \hat{y}_i$. En regresión por mínimos cuadrados, se trabaja con la suma de errores al cuadrado

$$E(\boldsymbol{\theta}) = \frac{1}{2} \sum_{i=1}^{N} (y_i - \hat{y}_i)^2 , \tag{13}$$

y una forma estándar de reportar error es el RMS, definido como

$$E_{\mathrm{RMS}} = \sqrt{\frac{2\, E(\boldsymbol{\theta})}{N}} = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (y_i - \hat{y}_i)^2}. \tag{14}$$

Esta métrica queda directamente alineada con la pérdida cuadrática y se interpreta en las mismas unidades que y (Bishop, 2006).

Conjuntos de entrenamiento/prueba, validación y *cross-validation*. El error de entrenamiento no es un buen proxy del desempeño fuera de muestra cuando existe sobreajuste. Un esquema base separa los datos en conjunto de entrenamiento y prueba, y puede incorporar un conjunto de validación (*hold-out*) para seleccionar complejidad/hiperparámetros. Cuando los datos son limitados, la validación cruzada permite reutilizar la mayor parte de las muestras para entrenar, rotando particiones y promediando el desempeño para comparar modelos (Bishop, 2006).

Independencia de muestras y particionado en series de tiempo. Los esquemas estándar de evaluación asumen, de forma explícita o implícita, que las observaciones pueden tratarse como aproximadamente independientes al particionar. En series de tiempo con rezagos/ventanas, un particionado aleatorio puede inducir *leakage* por solapamiento informacional entre entrenamiento y validación; por ello, más adelante se privilegian particiones temporales y procedimientos de validación diseñados para dependencias temporales (Bishop, 2006).

2.1.2. Modelos de mezcla gaussianos y algoritmo EM

Un modelo de mezcla gaussiana (GMM) representa una distribución de probabilidad como una combinación lineal de K componentes gaussianas. Siguiendo la formulación de Bishop (2006), la densidad de probabilidad para un vector $\mathbf{x}$ se define como:

$$p(\mathbf{x}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}|\mu_k, \Sigma_k) \tag{15}$$

Donde π_k son los coeficientes de mezcla que cumplen $0 \leq \pi_k \leq 1$ y $\sum_{k=1}^{K} \pi_k = 1$. El ajuste de los parámetros $\{\pi_k, \mu_k, \Sigma_k\}$ se realiza mediante el algoritmo de esperanza-maximización (EM). En el paso de esperanza (E-step), se utilizan los valores actuales de los parámetros para evaluar las responsabilidades, denotadas como $\gamma(z_{nk})$:

$$\gamma(z_{nk}) = \frac{\pi_k \mathcal{N}(\mathbf{x}_n|\mu_k, \Sigma_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(\mathbf{x}_n|\mu_j, \Sigma_j)} \tag{16}$$

En el paso de maximización (M-step), se reestiman los parámetros para maximizar la esperanza de la log-verosimilitud. La actualización de las medias μ_k se define como:

$$\mu_k^{new} = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk}) \mathbf{x}_n \tag{17}$$

Donde $N_k = \sum_{n=1}^{N} \gamma(z_{nk})$ representa el número efectivo de puntos asignados a la componente k. Este procedimiento iterativo garantiza la convergencia hacia un máximo local de la función de verosimilitud.

2.1.3. Teoría de la información: entropía e información mutua

La teoría de la información cuantifica la incertidumbre y la dependencia estadística entre variables aleatorias sin asumir formas funcionales específicas (Cover & Thomas, 2006).

Entropía de Shannon. Para una variable aleatoria discreta X con distribución de probabilidad $p(x)$, la entropía $H(X)$ mide la incertidumbre promedio asociada:

$$H(X) = -\sum_{x \in \mathcal{X}} p(x) \log p(x) \tag{18}$$

Análogamente, para dos variables (X, Y), se define la entropía condicional $H(X|Y)$ como la incertidumbre remanente en X una vez observada Y:

$$H(X|Y) = -\sum_{x,y} p(x, y) \log p(x|y) \tag{19}$$

Información mutua. La información mutua $I(X;Y)$ cuantifica la reducción de incertidumbre en X debida al conocimiento de Y. Se define mediante la divergencia de Kullback-Leibler entre la distribución conjunta y el producto de sus marginales:

$$I(X;Y) = \sum_{x,y} p(x, y) \log \frac{p(x, y)}{p(x)p(y)} \tag{20}$$

Esta métrica es simétrica, no negativa ($I(X;Y) \geq 0$) y se relaciona con la entropía mediante la identidad $I(X;Y) = H(X) - H(X|Y)$.

2.1.4. Taxonomía del aprendizaje automático

El aprendizaje automático constituye un marco de inducción de modelos a partir de datos, permitiendo que un sistema mejore su desempeño en tareas específicas sin ser programado de forma explícita para cada escenario posible. Según la definición formal de Mitchell (1997), se dice que un programa aprende si su medida de rendimiento P en una tarea T se incrementa con la experiencia E. Esta mejora se operacionaliza mediante algoritmos que ajustan parámetros internos de una función para aproximar una estructura latente o minimizar una señal de error. En consecuencia, la clasificación de estos métodos depende directamente de la naturaleza de la experiencia y el tipo de retroalimentación disponible durante el proceso de entrenamiento (Russell & Norvig, 2004).

El aprendizaje supervisado utiliza un conjunto de entrenamiento compuesto por pares de entrada y salida objetivo $(\mathbf{x}, y)$. El proceso busca inducir una función de mapeo capaz de predecir la etiqueta o el valor numérico para nuevos datos no observados, minimizando la discrepancia entre la predicción y el valor real. Esta categoría abarca problemas de clasificación, donde la salida es categórica, y de regresión, donde el objetivo es una variable continua; en ambos casos, la presencia de una señal de error explícita guía el ajuste de los parámetros hacia la generalización del fenómeno estudiado.

El aprendizaje no supervisado prescinde de etiquetas o valores de referencia, enfocándose exclusivamente en identificar regularidades, estructuras o distribuciones de probabilidad

intrínsecas en los datos. Al no existir una noción predefinida de respuesta correcta, el algoritmo agrupa observaciones similares o reduce la dimensionalidad del espacio de entrada para facilitar la interpretación de patrones complejos. Este enfoque es fundamental para tareas de segmentación y detección de densidades, permitiendo descubrir la organización natural de la información sin sesgos impuestos por una supervisión externa.

El aprendizaje por refuerzo se basa en la interacción dinámica entre un agente y su entorno para maximizar una recompensa acumulada a largo plazo. A diferencia del esquema supervisado, el agente no recibe instrucciones sobre la acción óptima, sino que debe explorar diferentes estados y aprender de las consecuencias de sus decisiones mediante señales de refuerzo (premios o penalizaciones). Este paradigma es característico de problemas de control y toma de decisiones secuenciales, donde la retroalimentación suele ser diferida y el éxito depende de la capacidad del sistema para equilibrar la explotación de conocimientos previos con la exploración de nuevas estrategias.

2.2. Fundamentos del *Machine Learning*

El *Machine Learning* (ML) se aborda como el diseño de modelos cuyo comportamiento se ajusta a partir de datos con el objetivo de mejorar el desempeño en una tarea bien definida. En un sentido operacional, se dice que un sistema aprende de la experiencia E, respecto de una clase de tareas T y una medida de desempeño P, si su desempeño en T, medido por P, mejora al incorporar E (Mitchell, 1997).

Formulación del aprendizaje. El proceso de entrenamiento busca minimizar una función de pérdida $\mathcal{L}$ que cuantifica la discrepancia entre el valor real y_i y la predicción $\hat{y}_i$. Para este problema de regresión, se utiliza el error cuadrático medio (MSE) como medida de desempeño P:

$$\text{MSE} = \frac{1}{N}\sum_{i=1}^{N}(y_i - f(x_i;\theta))^2 \tag{21}$$

El ajuste de los parámetros θ se realiza mediante la minimización de 21 sobre el conjunto de entrenamiento $\mathcal{D}_E$. Esta formulación permite tratar el desarrollo del soft-sensor como un problema de optimización numérica, donde el objetivo es la generalización sobre datos no observados durante el entrenamiento.

2.2.1. Regresión: criterio de ajuste y evaluación (sin duplicar definiciones).

En regresión supervisada se aproxima la relación entre entradas $\mathbf{x}$ y una salida real y mediante un modelo $\hat{y} = f(\mathbf{x})$. En este informe se conserva el criterio de ajuste cuadrático ya definido en (13) (y su lectura en unidades del objetivo mediante (14)),

junto con el esquema de entrenamiento/validación/cross-validation descrito en *Conjuntos de entrenamiento/prueba, validación y* cross-validation (Bishop, 2006).

2.2.2. Familias de modelos de regresión: de lineales a árboles y ensambles.

En regresión supervisada se construye un predictor a partir de ejemplos etiquetados, aproximando la relación entre una entrada $x \in \mathbb{R}^p$ y una salida real y. En la práctica, distintas familias de modelos imponen supuestos distintos sobre la forma de $\mathbb{E}[Y \mid X = x]$, el nivel de suavidad del ajuste y la forma en que se controla la complejidad (Hastie et al., 2009).

Modelos lineales. El caso base es la regresión lineal, donde la esperanza condicional del objetivo se modela como una combinación afín de las entradas:

$$f(x) = \beta_0 + x^\top \beta. \tag{22}$$

Este caso es relevante porque entrega una referencia simple e interpretable, y porque varios métodos posteriores pueden leerse como extensiones o relajaciones de esta forma.

Modelos lineales penalizados. Cuando hay multicolinealidad, ruido o muchas variables respecto del tamaño muestral, una forma estándar de estabilizar el ajuste es penalizar los coeficientes. En Ridge, el estimador se obtiene resolviendo

$$\hat{\beta}_{\text{ridge}} = \arg\min_{\beta} \left\{ \sum_{i=1}^{N} (y_i - \beta_0 - x_i^\top \beta)^2 + \lambda \|\beta\|_2^2 \right\}, \tag{23}$$

cuya solución cerrada, para variables centradas, es

$$\hat{\beta}_{\text{ridge}} = (X^\top X + \lambda I)^{-1} X^\top y. \tag{24}$$

En Lasso se reemplaza la penalización cuadrática por una penalización L_1,

$$\hat{\beta}_{\text{lasso}} = \arg\min_{\beta} \left\{ \sum_{i=1}^{N} (y_i - \beta_0 - x_i^\top \beta)^2 + \lambda \|\beta\|_1 \right\}, \tag{25}$$

lo que permite encoger coeficientes y, en muchos casos, llevar algunos exactamente a cero (Hastie et al., 2009).

Métodos de margen y kernels. Otra familia relevante es la de máquinas de soporte para regresión (*Support Vector Regression*, SVR). Su idea central es construir funciones que mantengan buen poder predictivo controlando complejidad mediante regularización por margen y, cuando corresponde, extendiendo el espacio de entrada mediante kernels.

En la práctica, esto permite capturar relaciones no lineales sin abandonar por completo un marco de ajuste bien regularizado (Murphy, 2012).

Árboles de regresión. Un árbol de regresión aproxima la función objetivo particionando el espacio de entrada en regiones disjuntas $R_1, \ldots, R_M$, y asignando una predicción constante en cada región:

$$f(x) = \sum_{m=1}^{M} c_m \, \mathbb{I}(x \in R_m). \tag{26}$$

La idea no es imponer una relación global suave, sino dividir el espacio en zonas donde una regla simple entregue una aproximación razonable.

Ensambles tipo bagging. Una limitación conocida de los árboles individuales es su alta varianza. Bagging la reduce promediando muchos árboles ajustados sobre muestras bootstrap del conjunto de entrenamiento:

$$\hat{f}_{\text{bag}}(x) = \frac{1}{B} \sum_{b=1}^{B} \hat{f}^{(b)}(x). \tag{27}$$

Dentro de esta familia, *Random Forest* agrega aleatorización adicional al restringir, en cada partición, el subconjunto de variables candidatas, lo que disminuye la correlación entre árboles y mejora el efecto del promedio. Variantes como *ExtraTrees* empujan aún más esa aleatorización, introduciendo cortes más aleatorios en la construcción del ensamble (Hastie et al., 2009).

Ensambles aditivos tipo boosting. En boosting, el predictor se construye de manera secuencial como una suma de aprendices débiles, típicamente árboles pequeños:

$$F_M(x) = \sum_{m=1}^{M} \nu \, h_m(x), \tag{28}$$

donde h_m es el aprendiz incorporado en la iteración m y ν es la tasa de aprendizaje. La lógica es que cada nuevo término corrija parte del error residual dejado por los anteriores. En esta familia se ubican implementaciones como *XGBoost*, que refuerzan regularización y eficiencia computacional, y variantes de *gradient boosting* basadas en histogramas, como HGBR, pensadas para entrenamiento más eficiente en datos tabulares (Hastie et al., 2009).

Las familias anteriores no se distinguen solo por desempeño, sino también por el tipo de estructura que pueden representar. Los modelos lineales privilegian simplicidad e interpretabilidad; los modelos penalizados controlan complejidad; SVR introduce una alternativa regularizada con kernels; los árboles capturan umbrales e interacciones; y los ensambles buscan mejorar estabilidad o capacidad predictiva a partir de modelos base. En consecuencia, más que esperar una familia universalmente superior, conviene entender

estas alternativas como sesgos inductivos distintos frente a un mismo problema.

2.2.3. Medidas de rendimiento clásicas.

Cuando uno evalúa un modelo de regresión, conviene aterrizar el desempeño en métricas comparables. Sea un conjunto de evaluación de tamaño n, con observaciones y_i y predicciones $\hat{y}_i$. Se define el residuo como $e_i = y_i - \hat{y}_i$ (Montgomery & Runger, 2003).

MAE (error absoluto medio). Se usa por su lectura directa en unidades del objetivo:

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|e_i|. \tag{29}$$

RMSE (raíz del error cuadrático medio). Captura una penalización mayor a errores grandes por el cuadrado del residuo:

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}e_i^2}. \tag{30}$$

En la notación clásica de regresión, la suma de cuadrados del error es $\text{SSE} = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2$ y el *mean square error* se obtiene dividiendo una suma de cuadrados por sus grados de libertad (por ejemplo, en regresión lineal simple $\text{MSE} = \text{SSE}/(n-2)$) (Montgomery & Runger, 2003). En esta memoria, RMSE coincide con la lectura en unidades del objetivo ya definida vía una raíz de un promedio cuadrático (cf. (14)) cuando se calcula sobre el conjunto evaluado.

R^2 (coeficiente de determinación). Se define a partir de la descomposición de variabilidad total en variabilidad explicada por el modelo y variabilidad residual:

$$\text{SST} = \sum_{i=1}^{n}(y_i - \bar{y})^2, \qquad \text{SSR} = \sum_{i=1}^{n}(\hat{y}_i - \bar{y})^2, \qquad \text{SSE} = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2, \tag{31}$$

con $\text{SST} = \text{SSR} + \text{SSE}$ en el caso clásico de ajuste lineal con intercepto (Montgomery & Runger, 2003). Bajo esta identidad,

$$R^2 = \frac{\text{SSR}}{\text{SST}} = 1 - \frac{\text{SSE}}{\text{SST}}. \tag{32}$$

En términos prácticos, R^2 se interpreta como una medida de la fracción de variabilidad de y explicada por el modelo (Montgomery & Runger, 2003). En evaluación fuera de muestra, sin embargo, la cantidad $1 - \text{SSE}/\text{SST}$ puede tomar valores negativos si el predictor rinde peor que una referencia basada en la media del conjunto evaluado. Por ello, en esta

memoria R^2 se interpreta como una métrica complementaria a MAE y RMSE, y no como criterio único de desempeño.

En conjunto, estas tres métricas permiten leer el problema desde ángulos complementarios: MAE entrega una medida robusta y directamente interpretable del error medio, RMSE penaliza con mayor fuerza errores grandes y R^2 ayuda a ubicar cuánto de la variabilidad del objetivo logra explicar el modelo. Por eso, en este trabajo se prioriza una lectura conjunta y no el uso aislado de una sola métrica.

2.2.4. Maldición de la dimensionalidad.

En regresión (y, en general, en aprendizaje supervisado) es común disponer de muchas variables de entrada potencialmente útiles para estimar una variable objetivo, pero con un número limitado de observaciones. En ese régimen, aumentar la dimensionalidad del espacio de entrada puede volver problemático el aprendizaje, fenómeno conocido como *maldición de la dimensionalidad*. (Bishop, 2006) Bishop ilustra que, en espacios de alta dimensionalidad, intuiciones geométricas formadas en pocas dimensiones pueden fallar: por ejemplo, en una esfera D-dimensional gran parte del volumen se concentra en una capa delgada cercana a la superficie. De forma análoga, en una distribución Gaussiana en alta dimensión, la masa de probabilidad tiende a concentrarse en una cáscara delgada a cierta distancia del origen. Estas concentraciones implican que nociones como "vecindarios locales" y proximidad pueden comportarse de manera contraintuitiva al crecer D, dificultando la estimación y la predicción basada en proximidad. Aun así, el texto enfatiza que la maldición de la dimensionalidad no impide construir técnicas efectivas en espacios de alta dimensión por dos razones principales: (i) los datos reales suelen estar confinados a regiones de menor dimensionalidad efectiva, y (ii) típicamente presentan propiedades de suavidad local, de modo que pequeños cambios en las entradas inducen pequeños cambios en la variable objetivo, permitiendo explotar aproximaciones tipo interpolación local. (Bishop, 2006)

2.2.5. Regresión por mínimos cuadrados parciales (PLS).

Como alternativa a ajustar la regresión directamente en el espacio original de variables, *Partial Least Squares* (PLS) construye combinaciones lineales ortogonales de las entradas para luego realizar la regresión en ese espacio reducido. A diferencia de la regresión por componentes principales (PCR), PLS utiliza y (además de X) para construir dichas direcciones, por lo que su construcción es supervisada (Hastie et al., 2009). Dado que el método no es invariante a escala, se asume que las variables $\{x_j\}$ han sido estandarizadas (media cero y varianza uno) (Hastie et al., 2009).

En una formulación operacional, PLS inicia calculando pesos proporcionales al efecto

univariado de cada predictor sobre y, y con ellos define la primera dirección derivada:

$$\hat{\phi}_{1j} = \langle x_j, y \rangle, \qquad z_1 = \sum_{j=1}^{p} \hat{\phi}_{1j}\, x_j. \tag{33}$$

Luego se regresa y sobre z_1 y se ortogonalizan los predictores respecto de z_1; el proceso se repite para obtener $z_2, \ldots, z_M$, generando una secuencia de entradas derivadas y ortogonales (Hastie et al., 2009). Si se construyen $M = p$ direcciones, se recupera una solución equivalente a mínimos cuadrados ordinarios; usar $M < p$ produce una regresión reducida (Hastie et al., 2009).

Una caracterización equivalente del m-ésimo paso es que la dirección busca maximizar, bajo restricciones de ortonormalidad, un criterio que combina asociación con y y varianza en X:

$$\max_{\alpha} \ \mathrm{Corr}^2\big(y, X\alpha\big)\, \mathrm{Var}\big(X\alpha\big) \quad \text{s.a.} \quad \|\alpha\| = 1,\ \alpha^\top S \hat{\phi}_\ell = 0,\ \ell = 1, \ldots, m-1, \tag{34}$$

lo que ayuda a entender por qué, en la práctica, PLS tiende a comportarse de manera similar a métodos de *shrinkage* y reducción de dimensión como ridge y PCR (Hastie et al., 2009). En la comparación de métodos, se reporta que PLS suele contraer direcciones de baja varianza, pero puede inflar algunas direcciones de alta varianza, volviéndose algo inestable y con error de predicción ligeramente mayor que ridge en ciertos casos (Hastie et al., 2009). En escenarios con muchas variables ruidosas, se advierte que PLS tiende a *downweight* ese ruido sin eliminarlo, por lo que un gran número de *features* poco informativas puede contaminar la predicción (Hastie et al., 2009).

2.3. Particionado y validación: aleatorio, temporal y *Purged K-Fold* con embargo

En series temporales con *features* basadas en lags/ventanas, la validación debe controlar el *leakage* por dependencia y, especialmente, por solapamiento entre los intervalos de información usados para construir *features* y los intervalos asociados al objetivo (López de Prado, 2018).

Definiciones.

K-fold. Se particiona el conjunto de datos en k subconjuntos; para cada $i = 1, \ldots, k$, se entrena con $k-1$ subconjuntos y se evalúa en el subconjunto restante (López de Prado, 2018).

***Purged* K-fold.** Si las etiquetas abarcan intervalos temporales (p. ej., $Y_i = f([t_{i,0}, t_{i,1}])$), se **purga** el entrenamiento eliminando toda observación cuyo intervalo solape el intervalo

del fold de validación, evitando que observaciones concurrentes queden repartidas entre train y validación (López de Prado, 2018).

***Purged* K-fold con embargo.** Además de purgar, se excluye un **embargo** posterior al intervalo de validación (un *buffer* de duración h o un porcentaje del total) para mitigar dependencia de corto alcance; en la práctica, un embargo pequeño puede ser suficiente (López de Prado, 2018).

En implementación, el ancho de *purge* w se fija al máximo horizonte efectivo inducido por lags/ventanas (y/o por la definición del objetivo), y el embargo e se especifica como un horizonte temporal equivalente o como fracción del rango temporal (López de Prado, 2018). La Figura 1 resume visualmente las diferencias entre estas estrategias de particionado.

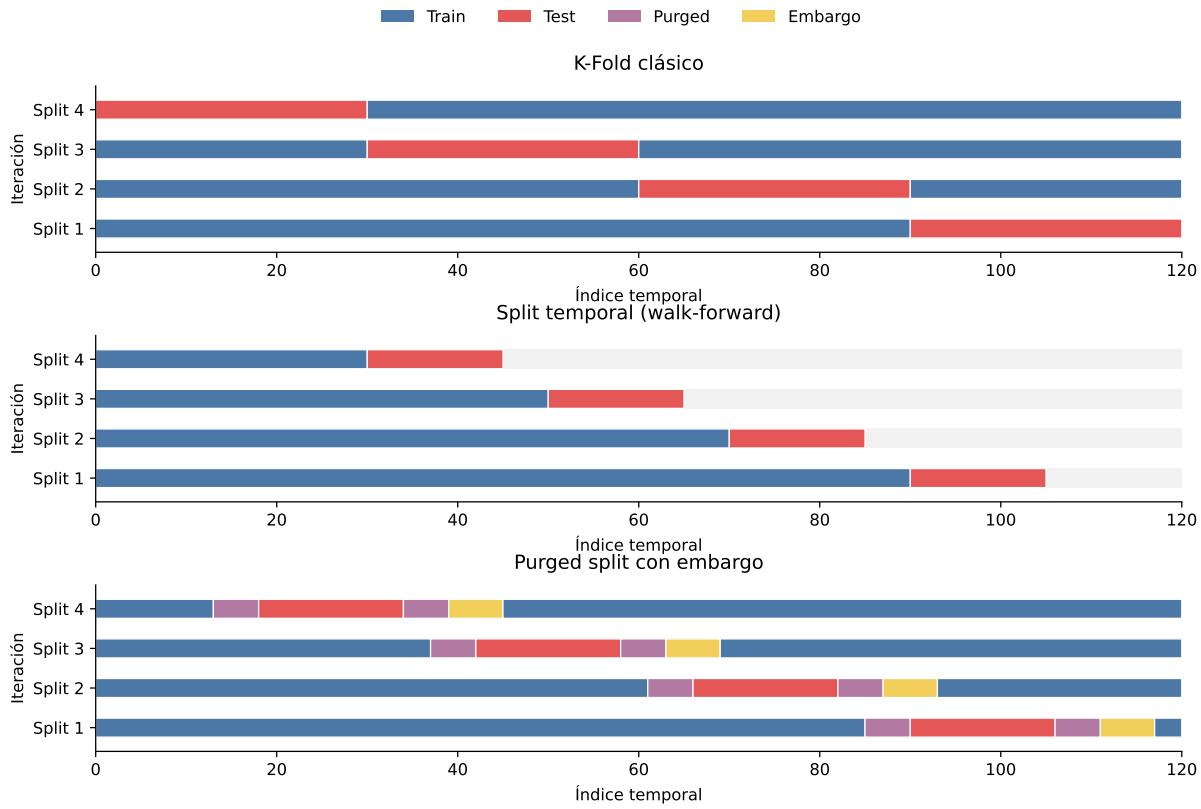

Figura 1: Comparación esquemática entre particionado aleatorio, particionado temporal y *Purged K-Fold* con embargo. La figura ilustra cómo cambia la separación entre entrenamiento y validación cuando se busca controlar solapamiento temporal y dependencia de corto alcance.

2.4. Dependencia estadística: Causalidad

En selección y análisis de variables interesa cuantificar cuánto aporta cada señal sobre el objetivo. En ese contexto conviene distinguir entre dependencia lineal, dependencia estadística general y evidencia exploratoria de direccionalidad temporal. La diferencia

importa porque una asociación alta no implica, por sí sola, interpretación causal en sentido fuerte.

Correlación. El coeficiente de correlación ρ_{XY}, ya introducido en (7), se utiliza como una medida rápida de asociación lineal entre dos variables. Su ventaja es la simplicidad. Su límite es que no captura dependencia no lineal ni entrega direccionalidad temporal.

Información mutua. Para dos variables aleatorias discretas X e Y, la información mutua se define como

$$I(X;Y) = \sum_x \sum_y p(x,y) \log\left(\frac{p(x,y)}{p(x)p(y)}\right). \tag{35}$$

Esta cantidad mide dependencia estadística general: vale cero si y solo si X e Y son independientes, y aumenta cuando conocer una reduce incertidumbre sobre la otra (Cover & Thomas, 2006). En la práctica, sirve para detectar relaciones que pueden no ser lineales y que, por lo mismo, no necesariamente aparecen con claridad al revisar solo correlación.

Entropía de transferencia. Cuando además interesa explorar direccionalidad temporal plausible, una extensión natural es comparar la incertidumbre de Y_{t+1} condicionada en su propio pasado con la incertidumbre restante al incorporar también el pasado de X. En forma discreta, la entropía de transferencia desde X hacia Y puede escribirse como

$$T_{X\to Y} = \sum p(y_{t+1}, y_t^{(k)}, x_t^{(\ell)}) \log\left(\frac{p(y_{t+1} \mid y_t^{(k)}, x_t^{(\ell)})}{p(y_{t+1} \mid y_t^{(k)})}\right), \tag{36}$$

donde $y_t^{(k)}$ y $x_t^{(\ell)}$ representan vectores de historia de longitudes k y ℓ, respectivamente. Operacionalmente, esta medida resulta útil para revisar si una variable parece aportar información temporal adicional sobre el objetivo más allá del propio pasado del objetivo.

Alcance de estas medidas. En este informe, correlación, información mutua y entropía de transferencia se utilizan como herramientas de lectura exploratoria y priorización de variables. No se emplean como prueba concluyente de causalidad fuerte. Lo que sí permiten revisar es si una señal muestra dependencia útil con el objetivo y si esa dependencia aparece concentrada en retardos compatibles con la dinámica del proceso.

2.5. Selección de features óptimas

En selección de variables aparece de inmediato la idea de encontrar un subconjunto "óptimo". El problema es que, si se dispone de p variables candidatas, el número de

subconjuntos posibles es

$$|\mathcal{P}(\{1,\ldots,p\})| = 2^p. \tag{37}$$

Eso vuelve inviable una búsqueda exhaustiva en escenarios realistas, sobre todo si cada subconjunto debe evaluarse bajo validación temporal y no solo con una partición simple.

Por esa razón, en aprendizaje automático suele ser más razonable trabajar con estrategias de reducción y exploración del espacio de búsqueda que aspirar a una solución global por fuerza bruta. Entre las alternativas más habituales aparecen métodos filtro, métodos envoltorio y enfoques embebidos, cada uno con compromisos distintos entre costo computacional, estabilidad e interpretabilidad (Guyon & Elisseeff, 2003).

Aporte incremental en modelos lineales. Cuando el análisis se realiza dentro de un marco lineal, una forma clásica de evaluar si una variable o un bloque de variables agrega información sobre el objetivo es comparar un modelo restringido con uno ampliado. Si R^2_{full} y R^2_{red} son los coeficientes de determinación del modelo completo y del modelo reducido, entonces el aporte incremental puede resumirse como

$$\Delta R^2 = R^2_{\text{full}} - R^2_{\text{red}}. \tag{38}$$

Bajo los supuestos clásicos del modelo lineal, este contraste puede formalizarse mediante un estadístico F:

$$F = \frac{(RSS_{\text{red}} - RSS_{\text{full}})/q}{RSS_{\text{full}}/(n - p_{\text{full}} - 1)}, \tag{39}$$

donde RSS es la suma de cuadrados residuales, q es el número de restricciones impuestas al pasar del modelo completo al reducido, n es el tamaño muestral y p_{full} el número de predictores del modelo completo (Greene, 2018).

La utilidad de esta lectura es directa: aportar no equivale solo a correlacionar con el objetivo, sino a explicar variación adicional una vez controladas las variables ya incluidas en el modelo.

En la práctica, la selección de variables rara vez se cierra con una sola técnica. Lo normal es combinar filtros exploratorios, restricciones operacionales y métodos de búsqueda más costosos cuando el espacio de entrada ya fue reducido a una escala manejable.

2.6. Estado del arte

2.6.1. Feature engineering

Rol de la ingeniería de variables en calderas de recuperación. En el contexto de este estudio, el *feature engineering* no se limita a la transformación matemática, sino a la representación de la inercia térmica y química del proceso. Siguiendo la lógica de Zhang et al. (2023), el espacio de búsqueda para el sensor virtual de reducción se expande

mediante transformaciones no lineales y agregaciones temporales. Esto es crítico debido a que variables como la presión del hogar o la temperatura del lecho no afectan la reducción de forma instantánea, sino a través de dinámicas de transporte que requieren una estructuración de datos tabulares que preserve la causalidad física.

OpenFE (generación automatizada de *features*). OpenFE es una herramienta de generación automatizada de variables para datos tabulares, orientada a reducir el trabajo manual y encontrar transformaciones efectivas en espacios de búsqueda amplios (Zhang et al., 2023). Su eficiencia se basa en (i) un esquema de evaluación incremental denominado *FeatureBoost* y (ii) un algoritmo de poda en dos etapas para filtrar variables candidatas en forma *coarse-to-fine* (Zhang et al., 2023). Evaluaciones empíricas reportan mejoras predictivas al incorporar estas variables en 49 de 68 conjuntos de datos de OpenML-CC18, con un incremento promedio de 1.9 % en desempeño (Zhang et al., 2023). Sin embargo, la principal limitación de OpenFE para este proyecto es su incapacidad nativa para manejar series de tiempo, al no incorporar restricciones de causalidad temporal en el cómputo de nuevas variables (Zhang et al., 2023).

Criterio de priorización en esta memoria (a partir de OpenFE). Se considera a OpenFE como una herramienta válida, pero sus resultados se interpretan como mejoras incrementales frente a un alto costo en complejidad. Dado que la generación masiva de combinaciones reduce la trazabilidad y explicabilidad del modelo, en esta memoria se priorizan métodos clásicos de selección basados en el conocimiento del proceso y métricas de información, asegurando modelos más parsimoniosos e interpretables operacionalmente.

TabPFN (modelos tipo *prior-data fitted network* para tabular). TabPFN introduce una arquitectura *transformer* orientada a problemas de clasificación tabular de baja volumetría, diseñada para entregar predicciones en tiempo real sin requerir ajuste de hiperparámetros (Hollmann et al., 2023). El enfoque reemplaza el flujo clásico de entrenamiento por un único modelo preentrenado que evalúa nuevos conjuntos de datos mediante una pasada hacia adelante (Hollmann et al., 2023). Aunque su evaluación destaca en conjuntos numéricos pequeños (hasta 2 000 muestras y 100 variables sin valores faltantes) (Hollmann et al., 2023), TabPFN se documenta aquí como referencia del avance en arquitecturas fundacionales para datos tabulares, pese a que las características del problema industrial (regresión, ruido temporal, mayor volumetría) limitan su aplicación directa.

2.6.2. Selección de familias de modelos

A pesar del auge reciente de las arquitecturas de aprendizaje profundo (*Deep Learning*), su aplicación directa en datos estructurados no garantiza superioridad frente a métodos

estadísticos y de aprendizaje más clásicos.

Grinsztajn et al. (2022) muestran empíricamente que, para conjuntos de datos tabulares de tamaño medio, los modelos basados en árboles de decisión siguen siendo altamente competitivos frente a diversas arquitecturas neuronales. El estudio incluye ensambles como *Random Forest*, *Gradient Boosting Trees* y *XGBoost*, comparados contra redes estándar y variantes *Transformer* adaptadas para formato tabular, considerando además el costo computacional asociado a la búsqueda de hiperparámetros (Grinsztajn et al., 2022).

Esta diferencia se explica, en parte, por los sesgos inductivos de cada familia. En datos tabulares, los ensambles de árboles suelen manejar bien heterogeneidad, relaciones no lineales, interacciones entre variables, presencia de valores extremos y ruido. En cambio, las redes neuronales pueden resultar más sensibles a variables poco informativas y a configuraciones donde la geometría específica del espacio de entrada contiene información relevante que no conviene suavizar en exceso (Grinsztajn et al., 2022).

En el contexto de esta memoria, el problema se representa finalmente en formato tabular mediante el uso de retardos, ventanas y transformaciones sobre variables historizadas del proceso. Bajo esa representación, resulta metodológicamente razonable tratar a los modelos basados en árboles como una línea base fuerte. Sin embargo, eso no implica asumir superioridad universal de una sola familia. También conviene conservar comparadores lineales, penalizados o basados en kernels, porque distintas representaciones del problema pueden favorecer compromisos distintos entre estabilidad, flexibilidad e interpretabilidad.

Por lo mismo, en este trabajo la selección de familias de modelos no se plantea como una apuesta cerrada por una técnica única, sino como una comparación entre sesgos inductivos distintos dentro de un mismo contrato de evaluación. Bajo esa lógica, los ensambles de árboles se consideran especialmente pertinentes para datos tabulares industriales, pero su lectura debe mantenerse abierta a contrastes con otras familias cuando el espacio de variables o la forma de representación así lo justifiquen.

3. Definición del problema

El presente trabajo se desarrolla en una planta *kraft* (sitio Arauco–Arauco) y se enfoca en la línea de recuperación. En particular, el objetivo es estimar la reducción asociada a la caldera recuperadora, observada en el circuito de licor verde. La reducción se reporta como un indicador porcentual, con un objetivo operacional típico cercano a 95 % (valor de referencia operativo). Para fijar el contexto físico mínimo del fenómeno y del punto de observación, la Figura 3 resume los flujos relevantes alrededor de la caldera recuperadora y el disolvedor.

Objetivo general. Evaluar y desarrollar un enfoque de *soft sensor* para estimar la reducción observada en el circuito de licor verde de una planta *kraft*, a partir de variables historizadas del proceso, bajo un esquema de evaluación temporal fuera de muestra que permita juzgar su factibilidad técnica y el aporte informativo de distintas señales operacionales.

Objetivos específicos.

- Definir la variable objetivo, su contexto físico de observación y las restricciones temporales relevantes del problema, incluyendo punto de medición, fuentes disponibles, alineación temporal y condiciones básicas de causalidad.
- Construir un dataset usable para modelamiento a partir de señales historizadas del proceso, incorporando extracción, integración temporal, limpieza, control de calidad y criterios de trazabilidad suficientes para el desarrollo experimental.
- Evaluar si variables operacionales de distinta naturaleza, incluyendo señales más directas del proceso y otras más indirectas o menos evidentes a priori, aportan información útil para la estimación de la reducción.
- Analizar la presencia de retardos temporales plausibles entre variables del proceso y la variable objetivo, de modo de identificar si existe una lógica temporal consistente con el fenómeno observado y si esta puede aprovecharse en la construcción de *features*.
- Comparar distintos espacios de entrada y formas de representación de variables, considerando el grado de predicción alcanzable y el *trade-off* entre desempeño e interpretabilidad, con el fin de determinar configuraciones más útiles para una eventual aplicación operativa.

Metodología de trabajo. La metodología seguida se organizó en las siguientes etapas:

- Familiarización con el problema, el contexto operacional y las señales disponibles en PI.

- Extracción, integración temporal, depuración y control de calidad de datos.
- Análisis exploratorio de sensibilidad, asociación y retardos temporales entre variables y objetivo.
- Selección y construcción de espacios de entrada para modelamiento.
- Evaluación experimental de modelos predictivos bajo un esquema temporal explícito de entrenamiento, validación y prueba fuera de muestra.

3.1. Variable objetivo y alineación temporal

La variable de interés es la reducción medida en el licor verde, denotada como $y(t)$. Debido a la dinámica del proceso, existe un retardo temporal τ entre los cambios en las variables de operación de la caldera $x(t)$ y su efecto observado en el disolvedor. El problema de estimación se define como encontrar una función f tal que:

$$\hat{y}(t+\tau) = f(x(t), x(t-1), \ldots, x(t-k)) \tag{40}$$

Donde $\hat{y}$ es la estimación de la reducción y $x(t)$ es el vector de variables de proceso (presiones, flujos, temperaturas) en el instante t. La existencia de este τ (tiempo de residencia y transporte) justifica la necesidad de utilizar técnicas de alineación de señales y ventanas de tiempo para que el modelo capture la causalidad física del sistema.

La Figura 2 resume de manera esquemática la lógica general del problema de estimación abordado en esta memoria, distinguiendo las variables de entrada del modelo y la salida objetivo.

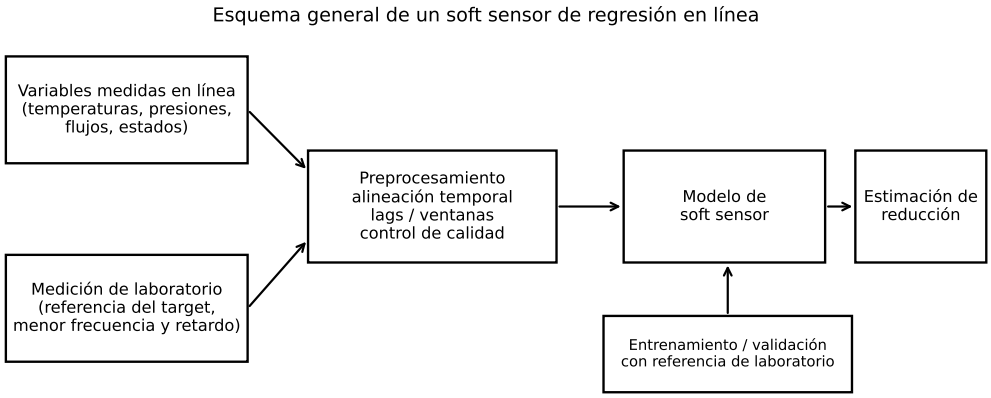

Figura 2: Esquema general del problema de estimación abordado en esta memoria. Variables de proceso medidas en línea se integran mediante alineación temporal, lags, ventanas y control de calidad para alimentar un modelo de soft sensor, entrenado y validado con referencia de laboratorio, que estima la reducción. Fuente: elaboración propia.

3.2. Variable objetivo y definición operativa

Se define la variable objetivo como

$$y(t) \;=\; \text{Reducción}(t)\;[\,\%].$$

En términos operativos, la reducción se expresa como

$$\text{Reducción}(t)\;[\,\%] \;=\; \frac{\text{Na}_2\text{S}(t)}{\text{Na}_2\text{S}(t) + \text{Na}_2\text{SO}_4(t)} \times 100.$$

La reducción se observa a través de dos fuentes complementarias:

- Laboratorio: medición discreta con periodicidad aproximada de 12 horas. El *timestamp* asociado corresponde a la hora de extracción de la muestra.
- Sensor en línea (alcalímetro): medición con frecuencia aproximada entre 15 y 30 minutos.

Para la formulación del problema se consideran ambas fuentes de medición. En particular, la señal de laboratorio se mantiene como referencia principal del objetivo para fines de evaluación, mientras que la señal del sensor en línea puede incorporarse en etapas de construcción del dataset y de entrenamiento cuando ello resulte metodológicamente útil para mejorar continuidad temporal o enriquecer la representación del problema. En consecuencia, la integración de ambas fuentes debe entenderse aquí como una decisión de modelado y entrenamiento, no como reemplazo de la referencia de laboratorio.

3.3. Punto de medición y condiciones de observación

La medición de reducción utilizada en el proyecto se ubica operacionalmente entre el disolvedor y el estanque de licor verde crudo, en la línea activa. De acuerdo con la Figura 3, este punto se encuentra aguas abajo del disolvedor, en el circuito de licor verde.

La condición de "línea activa" se identifica mediante una variable operacional (bit/estado) disponible en el sistema de control o en el preprocesamiento efectuado por operación. En esta sección se abstrae dicho detalle: se asume que el dataset ha sido filtrado o anotado de modo de asociar cada observación de reducción a la condición de operación correspondiente.

3.4. Datos disponibles y estructura temporal

Las variables de proceso (*features*) provienen de tags industriales con frecuencias heterogéneas (distintos Δt). Para integrar fuentes con granularidades dispares, se construye una grilla temporal común Δt_{ref} y se alinean todas las series a dicha referencia. En particular, tanto el laboratorio como el alcalímetro se representan en el dataset como señales tipo

sample-and-hold ("step"): si no existe una nueva medición en un instante de la grilla, se mantiene el último valor observado.

Este criterio de alineamiento implica que la cantidad de filas del dataset refleja el número de instantes en la grilla, no necesariamente la cantidad de mediciones independientes de $y(t)$. Por ello, la formulación del problema y la evaluación posterior deben considerar explícitamente la estructura temporal y los tramos constantes inducidos por el muestreo discreto.

3.5. Formulación de *Machine Learning*: nowcasting

El problema se formula inicialmente como regresión supervisada de nowcasting:

$$\hat{y}(t) \;=\; f(\mathbf{x}(t))\,,$$

donde $\mathbf{x}(t) \in \mathbb{R}^p$ representa el vector de variables de proceso (y transformaciones) alineadas al instante t, y $f(\cdot)$ es un modelo estimado a partir de un conjunto de entrenamiento $\mathcal{D} = \{(\mathbf{x}(t_i), y(t_i))\}_{i=1}^{N}$.

Dado que la construcción de *features* incorpora retardos explícitos (*lags*), el mismo planteamiento se puede extender de forma natural a **forecasting** $(t + H)$ como derivación posterior, sin cambiar el diseño base del dataset. En esta etapa, el foco es establecer el problema de estimación en t y su trazabilidad con las señales disponibles.

3.6. Construcción de *features*: lags y ventanas

Las entradas $\mathbf{x}(t)$ se construyen principalmente con variables de la caldera recuperadora y, en menor medida, con algunos indicadores operacionales aguas arriba. Operacionalmente, se han explorado **lags entre 10 y 240 minutos**, de modo que el modelo reciba información contemporánea y retardada:

$$\mathbf{x}(t) = \Big[\ldots,\; x_j(t),\; x_j(t - L_1),\; x_j(t - L_2),\; \ldots\Big], \qquad L_k \in \{10, \ldots, 240\}\ \text{min}.$$

Dependiendo de la naturaleza de cada señal, el diseño puede incorporar agregaciones en ventanas móviles (p. ej., promedio, mediana, mínimo, máximo) como alternativa a *lags* puntuales, siempre que se respete la causalidad (uso exclusivo de información disponible hasta t) y se eviten transformaciones que introduzcan filtraciones.

Nota anti-*leakage*. Se excluye del conjunto de entrada cualquier señal directa o indirectamente derivada del cálculo de reducción o del sistema de muestreo del objetivo. En particular, cualquier tag que replique, postprocese o utilice como insumo la medición de reducción (laboratorio o alcalímetro) no debe formar parte de $\mathbf{x}(t)$.

3.7. Alcance y límites de la definición

En síntesis, se estima Reducción[%] a partir de variables de proceso, con foco en *nowcasting*, usando señales alineadas a una grilla temporal común Δt_{ref} y representadas mediante *sample-and-hold*. La formulación considera tanto la información de laboratorio como la del sensor en línea, manteniendo al laboratorio como referencia principal para evaluación y permitiendo que la señal del sensor participe en la construcción del dataset y en ciertos regímenes de entrenamiento. La medición se asocia al tramo disolvedor → licor verde crudo, en la línea activa. Quedan fuera de esta sección (y se tratan por separado si corresponde) la evaluación económica del sensor físico, decisiones de inversión y cualquier planteamiento de optimización directa de control.

3.7.1. Preguntas de investigación (RQ)

A partir de la definición operacional del objetivo, la estructura temporal de los datos y el diseño de $x(t)$, se plantean las siguientes preguntas de investigación:

- **RQ1.** ¿Con qué precisión puede estimarse Reducción(t) a partir de variables de proceso disponibles hasta el instante t?
- **RQ2.** ¿Existe evidencia de dependencia temporal-operacional útil entre variables del proceso y Reducción(t), considerando exclusivamente información previa a la medición de la reducción?
- **RQ3.** ¿Mejora el desempeño predictivo al incorporar retardos (*lags*) en las variables de entrada?
- **RQ4.** Si se identifica un "mejor retardo" para una variable, ¿coincide dicho retardo con el tiempo físico esperado en que esa variable afecta el licor negro/*smelt*, es decir, con la ventana temporal real del proceso?
- **RQ5.** ¿Se puede construir un modelo que generalice adecuadamente bajo evaluación temporal (fuera de muestra) y ante potenciales cambios de régimen del proceso?
- **RQ6.** ¿Mejora la predicción al utilizar el objetivo proveniente del sensor, en comparación con el objetivo de laboratorio (y/o su combinación, si corresponde)?

3.7.2. Hipótesis principales (H)

Sobre la base del diseño experimental propuesto, se formulan las siguientes hipótesis de trabajo:

H1. Es posible estimar la reducción con un error inferior a la variabilidad intrínseca del sensor físico mediante el uso de modelos no lineales que capturen la dinámica de combustión en la caldera.

H2. La incorporación de retardos temporales (*lags*) puede mejorar el desempeño predictivo en configuraciones donde la dinámica del proceso y el espacio de variables retenido sí logran capturar información temporal útil, en comparación con variantes equivalentes sin retardos.

H3. El desempeño predictivo depende del acoplamiento entre familia de modelo, espacio de variables y forma de representación. En particular, se espera que los modelos no lineales basados en árboles resulten más competitivos cuando se trabaja con representación directa y espacios de variables más controlados, mientras que modelos lineales o penalizados pueden mantenerse competitivos cuando el problema se representa mediante PLS en espacios más amplios o intermedios.

H4. Bajo configuraciones comparables de espacio de variables y representación, un modelo entrenado con datos mixtos (laboratorio y sensor físico) puede mostrar una mayor robustez práctica fuera de muestra que un modelo entrenado exclusivamente con datos de laboratorio.

3.7.3. Hipótesis operacionales y criterios de refutación

Para evitar hipótesis vagas, cada hipótesis se operacionaliza mediante comparaciones explícitas, métricas y un criterio de decisión.

- **Definición de referencia basal y comparadores.** Se utilizará como línea base principal:
  - **Baseline oficial B0:** promedio móvil de una semana del objetivo de laboratorio.

  Además, cuando corresponda, se considerarán como comparadores de referencia variantes lineales entrenadas sobre espacios de entrada equivalentes a los de los modelos no lineales evaluados, con y sin retardos según el contraste experimental de interés.

- **Criterio de refutación (estándar).** Sean M las métricas primarias (por ejemplo MAE/RMSE) calculadas sobre conjuntos de evaluación temporal fuera de muestra, y sea $\Delta M = M(\text{modelo}) - M(\text{referencia})$.
  - **Para H1:** se refuta si el mejor modelo propuesto no logra mejorar al *baseline* oficial en las métricas primarias o si esa mejora no se sostiene de manera razonable bajo el esquema de evaluación temporal adoptado.
  - **Para H2:** se refuta si las variantes con *lags* no muestran una ventaja consistente frente a sus contrapartes equivalentes sin retardos al comparar familias de modelos y espacios de variables bajo el mismo contrato de evaluación, o si esa

ventaja aparece de forma demasiado acotada como para sostener una mejora general del uso de retardos.

- **Para H3:** se refuta si, bajo el mismo contrato de evaluación, no se observa una dependencia apreciable entre familia de modelo, espacio de variables y representación. En particular, se refuta si los modelos basados en árboles no resultan competitivos en variantes con representación directa y espacios más controlados, o si los modelos lineales o penalizados no se mantienen competitivos cuando se utiliza PLS en espacios más amplios o intermedios.
- **Para H4:** se refuta si, bajo el mismo contrato de evaluación y al comparar configuraciones competitivas en espacios de variables y representaciones comparables, los modelos entrenados con datos mixtos no muestran una ventaja apreciable frente a sus contrapartes *lab_only* en desempeño fuera de muestra, o si esa ventaja no se sostiene de manera suficientemente consistente como para respaldar una mayor robustez práctica.

- **Control anti-*leakage*.** La evaluación se realizará con particiones temporales estrictas y un *embargo* de 1 día entre conjuntos para evitar contaminación temporal. Además de la separación *train–test*, se definirá un *holdout* intocable para el reporte final de las métricas fuera de muestra. En todo el diseño experimental se privilegiará un criterio conservador respecto de fuga de información, incluyendo fuga temporal.

3.8. Métricas y criterios de éxito

Esta subsección fija el contrato mínimo de evaluación para responder las RQ y contrastar las hipótesis. Dado que el objetivo es Reducción(t) [%], las métricas se reportarán en términos absolutos y en unidades interpretables (puntos porcentuales), evitando métricas relativas cuando no aporten claridad operativa.

- **Métricas primarias.** Se reportarán MAE, RMSE y R^2 como métricas principales. Para MAE y RMSE, el error se reportará en términos absolutos sobre Reducción(t) (puntos porcentuales).

- **Criterio de comparación.** La comparación entre configuraciones se realizará sobre particiones temporales estrictas, separando explícitamente entrenamiento, *test* y *holdout*, con embargo de 1 día entre bloques cuando corresponda. Para seleccionar configuraciones competitivas se priorizará RMSE mínimo, luego MAE mínimo y, en caso de persistir empate, R^2 mayor.

- **Criterio de éxito operacional.** No existe un umbral oficial de error; el objetivo será comparar alternativas y seleccionar modelos en base a mejora respecto del *baseline* oficial y desempeño fuera de muestra temporal.

- **Reporte mínimo.** El reporte mínimo incluirá tablas comparativas de métricas por *split*, síntesis cruzadas entre *test* y *holdout* y, cuando corresponda, comparaciones específicas entre regímenes de datos, representaciones o familias de modelos.

4. Desarrollo del Proyecto

En esta sección se describe el flujo de trabajo seguido para construir el dataset industrial y habilitar el entrenamiento de modelos de estimación de reducción. El punto de partida es la extracción sistemática de señales historizadas del ecosistema PI; luego (en subsecciones posteriores) se abordan depuración, integración temporal, control de calidad, construcción de *features* y validación.

4.1. Familiarización con el problema

Antes de la extracción masiva y del modelamiento se realizó una etapa de familiarización orientada a reducir ambigüedades operacionales y a fijar un contexto técnico mínimo para interpretar las señales disponibles. El objetivo fue entender cómo se usa la reducción en operación, qué condiciones del proceso pueden afectarla de manera plausible y qué tan trazables y confiables eran los tags historizados asociados al área de caldera recuperadora y disolvedor.

Levantamiento operacional. Se sostuvieron conversaciones con ingenieros de operaciones para precisar el significado práctico de la reducción, su variabilidad esperada y las consecuencias operacionales asociadas a desviaciones. Esta interacción permitió además validar nomenclatura local, identificar señales que, a priori, podían resultar útiles y detectar tags potencialmente redundantes o de interpretación ambigua.

Comprensión del proceso aguas arriba. En paralelo, se revisaron procedimientos y descripciones del proceso con el fin de reconstruir el encadenamiento causal mínimo desde evaporación hacia caldera recuperadora, disolución de *smelt* y circuito de licor verde. Esa revisión se utilizó como guía para formular hipótesis iniciales sobre qué grupos de variables podían contener información útil respecto de la dinámica observada en el punto de medición. Como apoyo a esa familiarización, la Figura 3 resume de forma esquemática el recorrido general del proceso en la zona de interés. En términos simples, el licor negro alimenta la caldera recuperadora, donde las condiciones de combustión y distribución de aire influyen en su comportamiento. A partir de ese proceso se genera *smelt*, que luego pasa al estanque disolvedor, donde se enfría y se mezcla con licor débil para dar origen al licor verde crudo. Esta secuencia permite ubicar mejor el fenómeno de interés y entender por qué pueden aparecer retardos entre variables medidas en la caldera y la reducción observada aguas abajo.

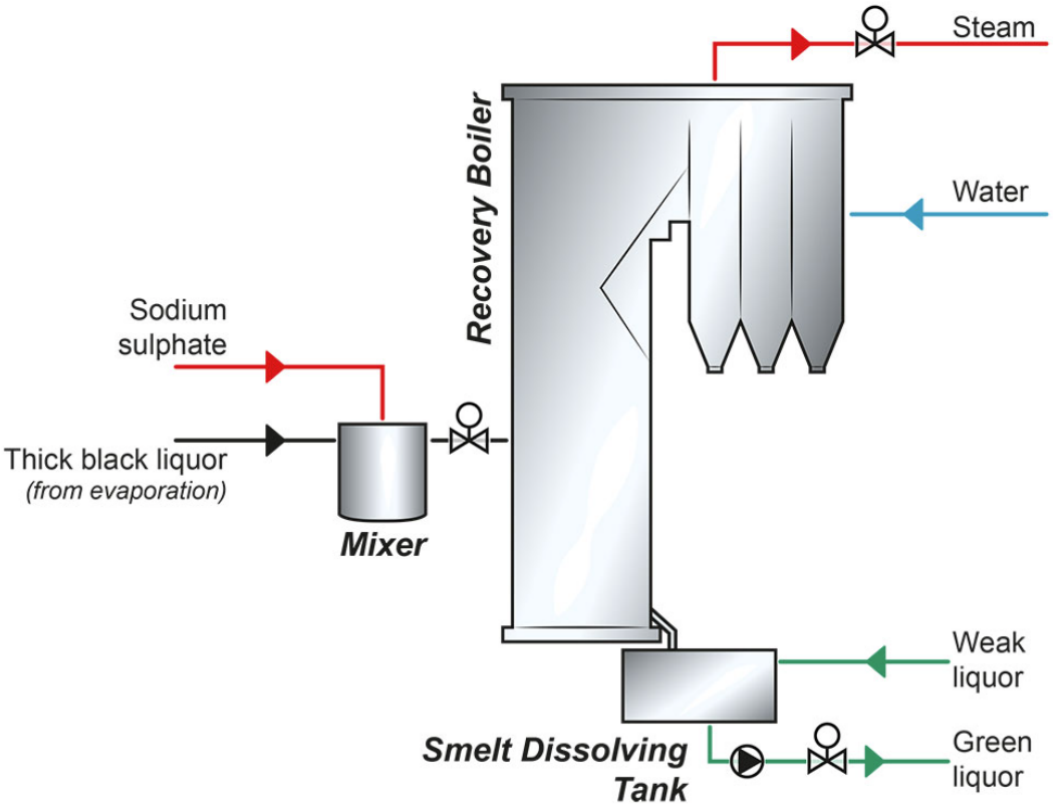

Figura 3: Esquema simplificado del proceso en la zona de interés para la familiarización operacional. El licor negro alimenta la caldera recuperadora, se genera *smelt* y este pasa al estanque disolvedor, donde se mezcla con licor débil para formar licor verde crudo.

Herramientas PI para inspección y trazabilidad. Se utilizó PI Vision para inspeccionar tendencias, unidades, descripciones y consistencia temporal de señales candidatas, y PI Builder como apoyo para exploraciones rápidas orientadas a confirmar disponibilidad, rangos y continuidad de historización. En esta etapa el foco estuvo en trazabilidad, es decir, en entender qué mide cada tag y cómo se comporta, más que en desempeño predictivo.

Exploración preliminar con un pool reducido. Antes de trabajar con el universo completo de tags, se construyó un subconjunto inicial de señales y un intervalo acotado de tiempo para exploración. Sobre ese pool reducido se generaron visualizaciones y chequeos rápidos de series temporales, tramos constantes, faltantes, saltos, saturaciones y coherencia de rangos. Esta revisión permitió familiarizarse con la dinámica típica de variables del área y su relación temporal con el objetivo, identificar problemas evidentes de calidad de datos, como *NaN*, congelamientos o escalones propios del *sample-and-hold*, y anticipar necesidades de preprocesamiento que después se formalizaron en reglas de alineamiento y limpieza.

4.2. Extracción de datos

La extracción se realizó en Python mediante un conector ya disponible en el entorno de la planta, el cual consume la **PI Web API** (acceso de solo lectura). En paralelo, se contó con acceso al portal web corporativo (PI Vision) para inspección y verificación operacional de tendencias, descripciones y contexto de tags. Las series se descargaron y manipularon como ***DataFrames*** (`pandas`), utilizando el sello temporal como índice para habilitar alineamiento y transformaciones posteriores.

Período y zona horaria. Se extrajeron datos desde **enero de 2024** (arranque del período operacional disponible para el proyecto) hasta los **últimos días de noviembre de 2025**. Todos los *timestamps* se manejaron en la zona horaria local **America/Santiago** (UTC-3), evitando conversiones a UTC para mantener trazabilidad con operación.

Resolución de consulta y representación temporal. La extracción se ejecutó sobre una **grilla uniforme de 1 minuto**. Dado que las señales originales presentan frecuencias heterogéneas (y, en varios casos, naturaleza escalonada), el conector utilizado retornó las series ya alineadas a dicha grilla, aplicando interpolación y/o propagación del último valor según correspondiera a cada tipo de medición. En etapas posteriores del desarrollo se evalúan tratamientos específicos por tipo de señal y resoluciones efectivas distintas, pero esta subsección se restringe al criterio de extracción base.

Variables objetivo. Se descargaron explícitamente los tags asociados a las dos fuentes del objetivo descritas en la Sección 3:

- `752AI4546_I10.MV`: **Reducción CR3** [%] medida por alcalímetro. Este tag corresponde a un **cálculo interno** que consolida la medición en la **línea activa**, lo que simplifica el tratamiento operacional (en contraste con manejar manualmente las líneas por separado).
- `752LAB087.QC`: **Análisis Reducción TK Disolvedor** [%] proveniente de laboratorio.

Adicionalmente, y para fines de contraste y trazabilidad, en algún punto del proceso también se descargaron los tags de reducción por línea individual; sin embargo, para el desarrollo del dataset se priorizó el tag calculado de línea activa por consistencia y continuidad.

Universo inicial de tags (`752*`) y criterio "no dejar nada afuera". Como aproximación inicial se adoptó un enfoque deliberadamente inclusivo: **descargar el universo completo** de tags que matchean el patrón `752*`, correspondiente al área de **caldera**

recuperadora y **estanque disolvedor** (incluyendo instrumentación, estados/motores, presiones, flujos, niveles, frecuencias, condiciones de línea activa, variables asociadas y señales historizadas disponibles). El objetivo de esta decisión fue dar oportunidad a todas las señales para evidenciar su utilidad potencial, desplazando la selección fina a etapas posteriores, ya con evidencia empírica de calidad e informatividad.

Volumen y filtro inicial por interpretabilidad. El universo descargado en primera instancia fue del orden de ~ 1000 tags. Posteriormente, se aplicó un filtro inicial basado en revisión de **descripciones y legibilidad operacional** (p. ej., descarte de tags crípticos, no interpretables o con semántica no recuperable desde la documentación disponible), consolidando un set de trabajo de aproximadamente $\sim$ 700–800 tags. Este filtrado se entiende como una depuración mínima por trazabilidad, y no como selección por desempeño predictivo.

Calidad general y persistencia. En términos generales, la calidad de datos observada fue adecuada para modelamiento, aunque con incidencias habituales en historiadores industriales (tramos pegados, valores faltantes, discontinuidades). El control de calidad y las reglas específicas de limpieza se presentan más adelante; en esta etapa se preservó el dataset bruto alineado. Para facilitar iteración y transporte dentro del flujo Python, los datos se persistieron principalmente en formato **PKL**.

Incorporación de variables externas. Además del universo `752*`, se integró un conjunto adicional de variables propuesto por un tercero que exploró el problema con un criterio distinto de selección. Dicho set aportó tags complementarios con solapamiento parcial respecto del universo principal; su aporte se consideró en etapas posteriores de depuración y evaluación.

Ejemplos de comportamiento temporal en intervalos acotados. Como apoyo a la familiarización con las señales descargadas, se seleccionaron dos ejemplos de comportamiento temporal sobre intervalos acotados. En ambos casos se muestran variables con resolución de 1 minuto, consistentes con la grilla base utilizada en la extracción. La idea no es presentar aquí un análisis exhaustivo, sino ilustrar cómo se comportan algunas señales dentro de ventanas concretas asociadas a muestras del objetivo.

La primera variable corresponde al consumo de licor negro hacia la caldera recuperadora. Se trata de una señal operativamente directa, ya que representa el flujo del principal insumo que alimenta la combustión y, en consecuencia, el proceso que termina dando origen al *smelt* y, aguas abajo, al licor verde cuya reducción se busca estimar. La segunda variable se incluye por una razón distinta: durante las iteraciones del análisis apareció repetidamente como señal con posible aporte informativo, pese a que a priori no resulta

tan directa ni tan fácil de interpretar desde la lógica más evidente del proceso. En ese sentido, se incorpora como ejemplo de una variable cuya relación con la reducción parece existir, pero cuya lectura operacional sigue siendo menos inmediata y más difusa.

La Figura 4 muestra ambas señales en el intervalo comprendido entre dos muestras consecutivas de laboratorio. La Figura 5 muestra el mismo tipo de señales en un intervalo acotado entre dos muestras consecutivas del sensor en línea. Estas visualizaciones permiten apreciar, de manera concreta, la diferencia de escalas temporales entre ambas fuentes del objetivo y el tipo de variación que presentan las señales de proceso dentro de cada ventana.

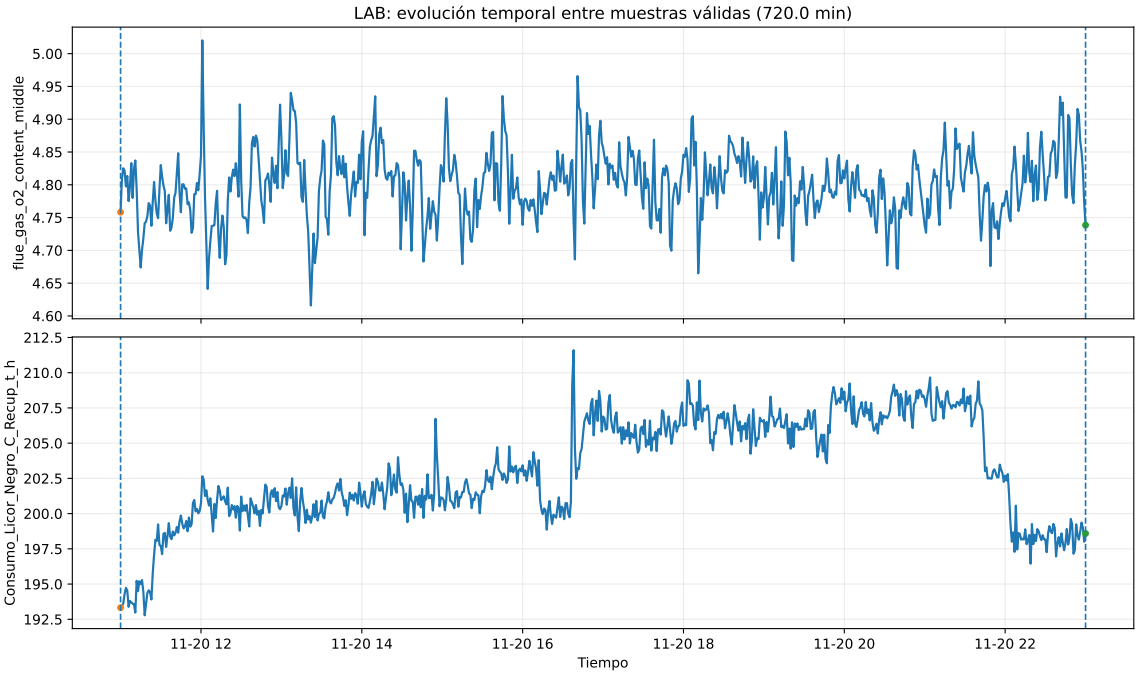

Figura 4: Ejemplo de comportamiento temporal de dos variables de proceso en un intervalo comprendido entre dos muestras consecutivas de laboratorio. Se incluye una variable de lectura más directa, asociada al consumo de licor negro hacia la caldera recuperadora, y otra señal que durante el análisis mostró aporte potencial, aun cuando su interpretación operacional resulta menos inmediata.

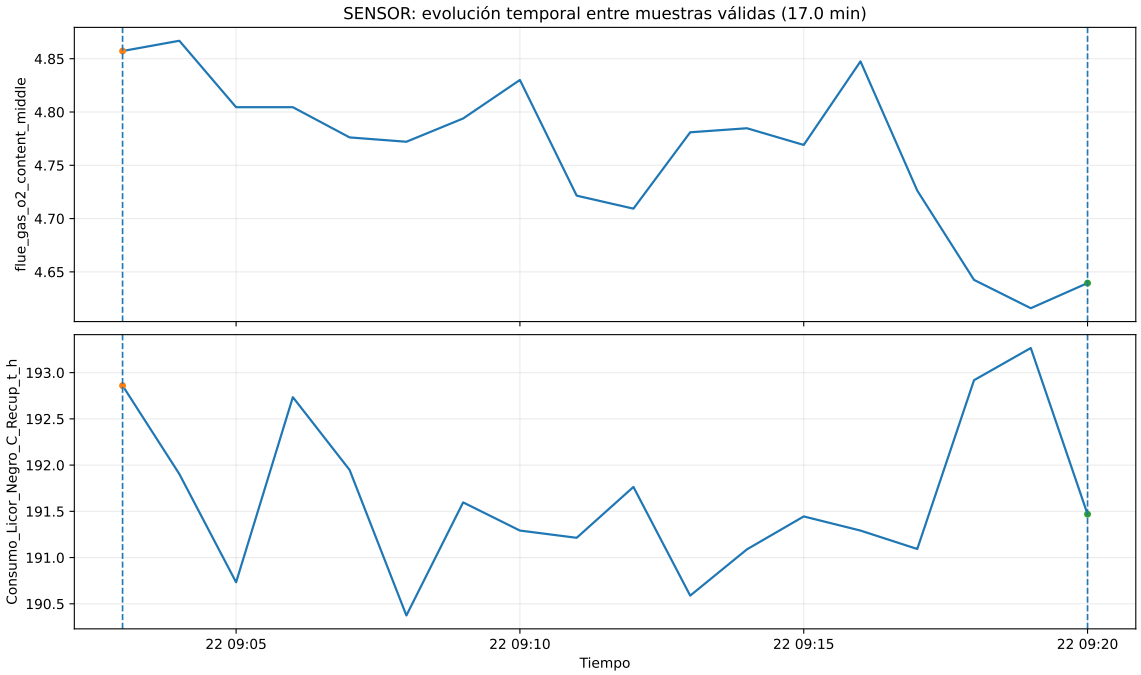

Figura 5: Ejemplo de comportamiento temporal de las mismas señales en un intervalo acotado entre dos muestras consecutivas del sensor en línea. La comparación con la ventana de laboratorio ayuda a visualizar la diferencia de resolución temporal entre ambas fuentes del objetivo.

4.3. Limpieza de datos

En esta etapa se definieron reglas de depuración para las variables objetivo y para las variables independientes. El criterio general fue priorizar consistencia temporal, trazabilidad y correctitud metodológica por sobre cobertura. En este problema eso era necesario, porque el sensor presentaba discontinuidades, tramos sin actualización y episodios que, al contrastarse con laboratorio, no resultaban defendibles para entrenamiento.

Los umbrales y filtros utilizados en esta subsección no deben interpretarse como óptimos formales del problema. Se adoptaron como heurísticas informadas, construidas a partir del comportamiento observado en los datos, de la revisión manual de señales y de la necesidad de conservar un subconjunto usable y metodológicamente defendible. En ese sentido, su propósito fue endurecer el criterio de aceptación del dato más que maximizar cobertura.

Sea $t \in \mathcal{T}$ la grilla temporal uniforme de 1 minuto definida en la extracción. Sea y_t^{lab} el target de laboratorio, sea y_t^{sens} el target de sensor y sea $x_{j,t}$ la variable independiente j-ésima observada en t.

Targets tipo *step* y eventos informativos. Tanto y_t^{lab} como y_t^{sens} se comportan como señales tipo *sample-and-hold*. Entre mediciones reales, el historiador repite el último valor. Por lo mismo, no todas las filas aportan información nueva. El punto informativo es el instante en que el valor efectivamente cambia.

Para ambos targets se construyó un indicador de cambio. Si z_t representa cualquiera de las dos señales objetivo, se definió

$$c_t(z) = \begin{cases} 1, & \text{si } z_t \neq z_{t-1}, \\ 0, & \text{si } z_t = z_{t-1}, \end{cases} \tag{41}$$

tratando el caso `NaN` = `NaN` como ausencia de cambio. Cuando una misma transición generó una racha de verdaderos consecutivos, se conservó solo el último instante de esa racha. Con esto, el dataset queda representado por eventos de cambio y no por mesetas repetidas artificialmente.

Frecuencias heterogéneas en variables independientes. Las variables independientes no comparten una única frecuencia de historización. Algunas señales en línea se descargan con granularidad de segundos, otras con frecuencia del orden de minutos y las variables de laboratorio tienen una frecuencia mucho menor. Bajo estas condiciones, antes de entrar al filtrado específico del sensor se aplicó una depuración inicial para reducir el efecto de señales con cobertura demasiado baja.

Filtro por faltantes en variables independientes. Como primer control, se eliminaron las variables independientes cuyo porcentaje de valores faltantes excedía 40 %. Si m_j es el número de faltantes de la variable x_j, la regla aplicada fue

$$\frac{m_j}{|\mathcal{T}|} > 0.40 \implies x_j \text{ se elimina del universo de trabajo.} \tag{42}$$

La razón es práctica. Una señal con ese nivel de ausencia obliga a perder muchas filas al exigir completitud o, alternativamente, deja al modelo aprendiendo sobre una fracción acotada del período total. En ambos casos se reduce estabilidad. El umbral de 40 % no se fijó como un valor universal ni óptimo, sino como una regla conservadora de higiene del espacio de variables dentro de este pipeline.

Problemas observados en el sensor y necesidad de filtrado específico. Una vez reducido el análisis a eventos informativos, se observó que el sensor no era usable en todo el horizonte. Existían tramos sin mediciones nuevas del orden de 30 % del tiempo y episodios compatibles con cambios de línea, donde el sensor mostraba valores fuera de rango o inconsistentes con la operación. Por eso se construyó una limpieza específica del sensor usando el laboratorio como ancla local. La idea no fue rescatar la mayor cantidad posible de puntos, sino conservar un subconjunto que pudiera defenderse operativamente.

Paso 1: *flags* de cambio para laboratorio y sensor. A partir de (41) se construyeron dos *flags* de cambio, uno para laboratorio y otro para sensor. Luego de consolidar rachas

consecutivas, se obtuvo un conjunto reducido de eventos de laboratorio y un conjunto reducido de eventos de sensor. Estos eventos son la base de todos los pasos posteriores.

Paso 2: normalización de nombres y *shift* de variables independientes. En paralelo a los *flags* de cambio, se normalizaron nombres de columnas descriptivas y se aplicaron desplazamientos temporales predefinidos a variables independientes. Este paso se hizo por higiene del espacio de *features* y por reproducibilidad. Los targets y los *flags* quedaron fuera de estas transformaciones.

Paso 3: reducción del dataset a timestamps relevantes. Sea f_t^{lab} el indicador final de cambio del laboratorio y sea f_t^{sens} el indicador final de cambio del sensor. El dataset crudo de eventos se definió conservando solo los timestamps donde ocurre un cambio en alguno de los dos targets:

$$f_t^{\text{evt}} = \mathbb{I}\Big(f_t^{\text{lab}} = 1 \ \vee \ f_t^{\text{sens}} = 1\Big). \tag{43}$$

Este recorte todavía no corresponde a la limpieza final. Solo construye una base de eventos relevantes sobre la cual se endurece la selección del sensor.

4.3.1. Anclaje local sensor–laboratorio (Paso 4)

El primer filtro real del sensor consiste en definir qué eventos de laboratorio sirven como ancla local confiable. Para cada evento de laboratorio en t_k, se estimó el valor del sensor en ese mismo instante por interpolación temporal, usando solo soporte por ambos lados. Si t_k^- y t_k^+ son, respectivamente, el evento de sensor inmediatamente anterior y posterior a t_k, se definió

$$\widehat{y}^{\text{sens}}(t_k) = y^{\text{sens}}(t_k^-) + \frac{t_k - t_k^-}{t_k^+ - t_k^-}\Big(y^{\text{sens}}(t_k^+) - y^{\text{sens}}(t_k^-)\Big). \tag{44}$$

Luego se calcularon las distancias temporales a soporte

$$d_k^- = t_k - t_k^-, \qquad d_k^+ = t_k^+ - t_k, \tag{45}$$

y la discrepancia local entre sensor interpolado y laboratorio

$$\Delta_k = \widehat{y}^{\text{sens}}(t_k) - y^{\text{lab}}(t_k). \tag{46}$$

Un laboratorio se consideró candidato solo si el soporte del sensor era cercano por ambos lados. La regla usada fue

$$d_k^- \le 22, \qquad d_k^+ \le 22, \qquad d_k^- + d_k^+ \le 22, \tag{47}$$

con tiempos en minutos. Esta condición se dejó estricta porque no bastaba con tener un dato antes y uno después. Ambos debían quedar suficientemente cerca como para que la interpolación fuera razonable en ese entorno local. De nuevo, no se planteó como una frontera óptima en sentido formal, sino como una heurística conservadora para asegurar soporte temporal local suficientemente defendible.

4.3.2. Banda estricta de consistencia y bracketing de eventos del sensor (Paso 5)

Sobre el conjunto de laboratorios candidatos se aplicó una restricción adicional de coherencia local. Se conservó un laboratorio como válido solo si

$$|\Delta_k| < 3. \tag{48}$$

Con esto, no solo se exigía soporte temporal cercano, sino también acuerdo local fuerte entre sensor y laboratorio.

Una vez hecho eso, la validez se traspasó al sensor con una regla de *bracketing*. Sea t_i un evento del sensor. Ese evento se aceptó en esta etapa solo si existía un laboratorio válido inmediatamente anterior t_{k^-} y uno inmediatamente posterior t_{k^+}, ambos dentro del conjunto de eventos de laboratorio, tales que

$$t_{k^-} < t_i < t_{k^+}. \tag{49}$$

Si faltaba alguno de esos dos laboratorios válidos, o si alguno no cumplía las condiciones de soporte y banda estricta, el evento del sensor se descartaba.

4.3.3. Continuidad mínima de la señal aceptada (Paso 6)

El *bracketing* elimina regiones mal ancladas, pero todavía puede dejar eventos del sensor demasiado aislados. Para controlar eso, se impuso un criterio de continuidad local. Sea $A^{(5)}$ el conjunto de eventos del sensor aceptados hasta el Paso 5. Un evento $t_i \in A^{(5)}$ se mantuvo en el Paso 6 si existía continuidad cercana, ya fuera por otro evento aceptado del sensor o por un nuevo cambio real de laboratorio.

En términos operativos, se mantuvo t_i si se cumplía al menos una de estas dos condiciones:

$$\exists\, t_j \in A^{(5)} \text{ tal que } t_j > t_i \text{ y } t_j - t_i \leq 22, \tag{50}$$

o bien

$$\exists\, t_k \in \mathcal{L}_{\text{chg}} \text{ tal que } t_k > t_i \text{ y } t_k - t_i \leq 22, \tag{51}$$

donde $\mathcal{L}_{\text{chg}}$ denota el conjunto de cambios reales del laboratorio. La idea de este paso fue evitar activaciones puntuales sin continuidad mínima, pero sin castigar tramos donde un cambio de laboratorio interrumpe la densidad aparente del sensor.

4.3.4. Filtro de discontinuidades tipo cambio de línea (Paso 7)

Después se buscó remover patrones compatibles con cambios de línea o reconfiguraciones abruptas. Sobre la señal aceptada tras el Paso 6, se comparó cada evento verdadero con el siguiente evento verdadero más cercano. Si dos eventos consecutivos $t_i < t_{i+1}$ satisfacían

$$t_{i+1} - t_i \leq 60 \qquad \text{y} \qquad |y^{\text{sens}}(t_{i+1}) - y^{\text{sens}}(t_i)| > 3, \tag{52}$$

se activaba un *trigger* de cambio de línea. Cuando eso ocurría, el filtrado se aplicaba de forma conservadora: se bloqueaba la aceptación de eventos durante las 4 horas siguientes. Con esto se evitó arrastrar al subconjunto final tramos inmediatamente posteriores a una discontinuidad fuerte.

4.3.5. Diagnóstico de fragmentación por grupos aislados (Paso 8)

Una vez removidas las discontinuidades principales, se evaluó la fragmentación de la señal restante. Para eso se formaron grupos uniendo eventos verdaderos consecutivos cuyos gaps no excedieran 22 minutos. Si t_i y t_{i+1} son eventos consecutivos del subconjunto aceptado en esta etapa, se consideró que pertenecían al mismo grupo si

$$t_{i+1} - t_i \leq 22. \tag{53}$$

A partir de esta partición, un grupo se clasificó como aislado si el gap respecto del verdadero anterior fuera del grupo y el gap respecto del verdadero siguiente fuera del grupo eran ambos mayores o iguales a 4 horas. Además, se registraron pares con gaps intermedios en el rango $(22, 240)$ minutos como insumo de auditoría, pero sin convertirlos todavía en criterio automático de descarte.

4.3.6. Remoción de grupos aislados pequeños (Paso 9)

Con el diagnóstico anterior, se eliminaron grupos aislados de tamaño pequeño. Si G_r denota uno de esos grupos, la regla aplicada fue

$$1 \leq |G_r| \leq 4 \implies G_r \text{ se elimina.} \tag{54}$$

Este paso apunta a remover islotes con densidad baja y soporte débil. A partir de aquí se generó un nuevo flag base del sensor y un flag específico de remoción asociado a pertenencia a grupo aislado pequeño.

4.3.7. Remoción de *singletons* sin banda mínima (Paso 10)

Después del Paso 9 todavía podían quedar grupos de cardinalidad uno. Sea t_i un *singleton*. Ese evento se eliminó si no tenía soporte temporal cercano por ambos lados. En particular, el punto se conservó solo si existían eventos verdaderos t_{i^-} y t_{i^+} tales que

$$0 < t_i - t_{i^-} \leq 60 \qquad \text{y} \qquad 0 < t_{i^+} - t_i \leq 60. \tag{55}$$

Si alguna de estas dos condiciones fallaba, el *singleton* se removía. Este paso produjo un flag *pre-final* del sensor y un flag de trazabilidad para los puntos descartados por falta de vecindad mínima.

4.3.8. Remoción de episodios demasiado cortos entre laboratorios usados (Paso 11)

El último filtro conecta la continuidad del sensor con la estructura de anclaje en laboratorio. Sea $\mathcal{L}^\star = \{\ell_1, \ell_2, \ldots, \ell_m\}$ el conjunto ordenado de laboratorios usados, es decir, laboratorios que sobrevivieron a las condiciones estrictas anteriores. Para cada par consecutivo (ℓ_r, ℓ_{r+1}), se definió el episodio abierto

$$I_r = (\ell_r, \ell_{r+1}), \tag{56}$$

y se contó el número de eventos del sensor *pre-final* contenidos estrictamente en ese intervalo:

$$n_r = \left|\{t_i \in A^{\text{pre}} : \ell_r < t_i < \ell_{r+1}\}\right|. \tag{57}$$

Si un episodio presentaba $1 \leq n_r \leq 3$, todos esos eventos se removían. El criterio acá fue práctico: entre dos laboratorios usados, un tramo con tan pocos eventos del sensor no entregaba continuidad suficiente como para justificar su permanencia en entrenamiento.

4.3.9. Resultado: subconjunto final del sensor

El resultado de los Pasos 4 a 11 es un subconjunto del sensor anclado en laboratorios con soporte temporal cercano y consistencia local estricta, con continuidad mínima, sin discontinuidades abruptas evidentes y con menor fragmentación residual. Cada endurecimiento deja además un flag intermedio y, cuando corresponde, un flag específico de remoción, de modo que el proceso puede seguirse paso a paso. Como cierre de esta subsección, la Figura 6 resume de manera esquemática una parte del procedimiento general de depuración aplicado al universo inicial de variables. No busca representar cada chequeo de sanidad realizado durante el desarrollo, sino mostrar la lógica base con que se redujo el espacio inicial de señales hacia un conjunto de *features* potenciales. En términos prácticos, se

descartaron variables con cobertura insuficiente, nombres no interpretables o señales cuya calidad no permitía sostener su uso en el análisis posterior.

En este problema, además, la limpieza más crítica no recayó solamente en las variables independientes, sino en la variable objetivo y en la forma en que esta quedaba representada en el tiempo. Tanto el laboratorio como el sensor se comportan como señales tipo *snapshot/sample-and-hold*, por lo que la depuración se realizó en torno a sus eventos informativos y a criterios de consistencia temporal. Las Figuras 7 y 8 ilustran ese punto mostrando ejemplos de la reconstrucción y depuración del target en laboratorio y sensor, respectivamente.

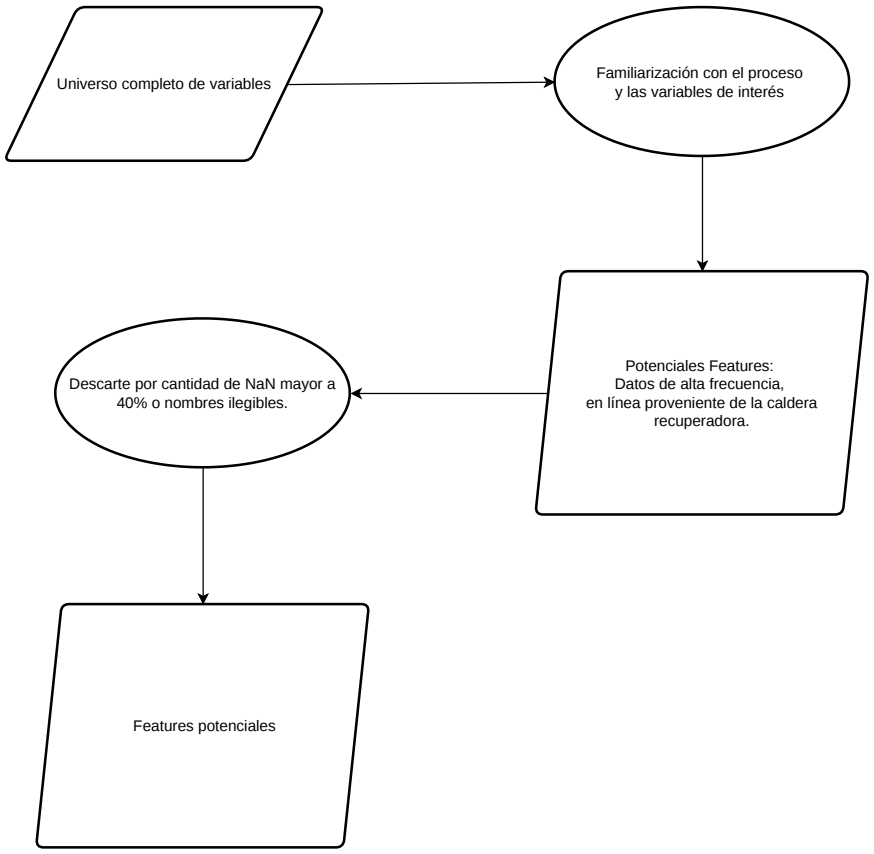

Figura 6: Esquema resumido de depuración del universo inicial de variables. La figura sintetiza la reducción desde el universo completo hacia un conjunto de *features* potenciales, considerando criterios básicos de cobertura e interpretabilidad.

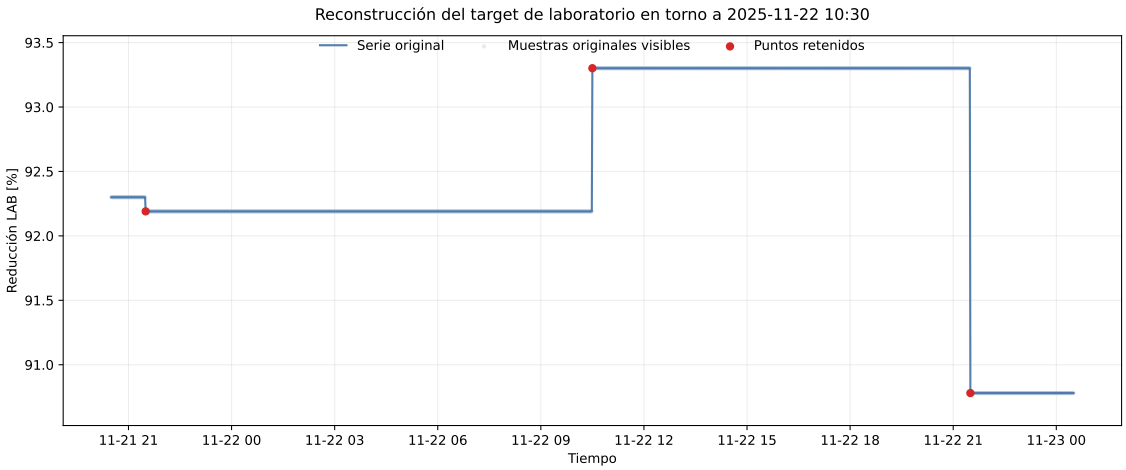

Figura 7: Ejemplo de reconstrucción y depuración del target de laboratorio sobre una ventana temporal acotada. La visualización ilustra cómo la representación final se apoya en eventos informativos y no en la repetición artificial de mesetas inducidas por el historiador.

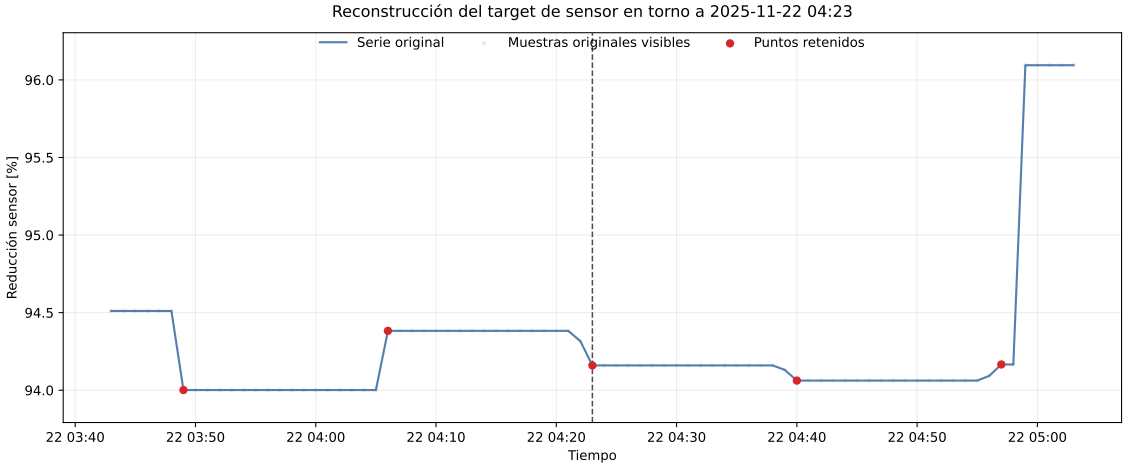

Figura 8: Ejemplo de reconstrucción y depuración del target de sensor sobre una ventana temporal acotada. La figura permite visualizar el efecto de los criterios de consistencia y continuidad aplicados sobre la señal antes de su uso en entrenamiento.

4.4. Análisis exploratorio de sensibilidad y dependencia temporal

Esta etapa se utilizó como filtro exploratorio sobre un universo amplio de variables. El objetivo fue identificar señales con asociación útil respecto del target y, al mismo tiempo, revisar si esa asociación tendía a concentrarse en ciertos retardos. En la práctica, esta subsección sirvió para recortar el espacio inicial de candidatas antes de pasar a etapas de selección y modelamiento más costosas.

En versiones previas del trabajo esta etapa se describía de forma abreviada como un análisis de “causalidad”. En esta memoria se mantiene esa referencia solo como continuidad histórica del desarrollo, pero el alcance real del análisis es más acotado. Las medidas

calculadas permiten estudiar dependencia lineal, dependencia no lineal y direccionalidad temporal plausible, pero no constituyen por sí solas una demostración de causalidad en sentido fuerte ni una validación factual de mecanismos físicos del proceso.

4.4.1. Objetivo y datos utilizados en esta etapa

Esta etapa se usó como filtro exploratorio sobre un universo amplio de variables. El objetivo fue identificar señales con asociación útil respecto del target y revisar si esa asociación tendía a concentrarse en ciertos retardos. En la práctica, sirvió para reducir el espacio inicial de candidatas antes de pasar a etapas más costosas de selección y modelamiento.

El análisis se realizó únicamente con el objetivo de laboratorio, por ser la fuente más trazable disponible en esa fase y la más adecuada para una primera lectura conservadora del problema. Además, se trabajó con un subconjunto temporal del período que más adelante se utilizó en el resto del desarrollo. Eso explica que en esta etapa aparezcan del orden de ~ 800 muestras válidas de laboratorio: al momento de ejecutar este barrido todavía se estaba iterando con una ventana que iba aproximadamente desde marzo o abril de 2024 hasta agosto de 2025. Más adelante se incorporó un horizonte mayor, hasta fines de noviembre de 2025, con el consiguiente aumento en el número total de muestras.

4.4.2. Diseño del barrido de *lags*

La idea fue evaluar cada variable independiente en distintos retardos respecto del target. Sea $y(t)$ el objetivo de laboratorio y sea $x_j(t)$ la variable independiente j-ésima. Para cada retardo ℓ se construyó una versión desplazada de cada variable,

$$x_j^{(\ell)}(t) = x_j(t - \ell), \tag{58}$$

y la comparación se realizó contra el target en el mismo instante t, es decir, sobre el par $\left(x_j^{(\ell)}(t), y(t)\right)$, restringido a los *timestamps* válidos del laboratorio.

En esta etapa no se asignó un *lag* distinto por variable. Lo que se hizo fue aplicar, para cada corrida, un mismo retardo fijo a toda la matriz de variables independientes y recalcular las métricas sobre ese dataset desplazado. En una corrida representativa, el barrido consideró retardos en el rango aproximado de ~ 80 a ~ 140 minutos, con valores discretos. Ese rango no salió de un ajuste formal. Se tomó como una aproximación práctica basada en la dinámica esperada del encadenamiento caldera recuperadora $\rightarrow$ disolvedor $\rightarrow$ punto de observación del licor verde.

4.4.3. Métricas calculadas por variable y por *lag*

Para cada variable candidata x_j y para cada *lag* ℓ evaluado se calcularon tres medidas complementarias.

La primera fue la correlación de Pearson, usada como indicador rápido de asociación lineal y manteniendo el signo de la relación:

$$\rho_{x_j^{(\ell)},y} = \frac{\mathrm{Cov}\left(x_j^{(\ell)}, y\right)}{\sigma_{x_j^{(\ell)}}\sigma_y}. \tag{59}$$

La segunda fue la información mutua entre la variable desplazada y el target,

$$I\left(x_j^{(\ell)}; y\right), \tag{60}$$

utilizada para capturar dependencia estadística más general, incluyendo relaciones no lineales.

La tercera fue *Transfer Entropy* (TE), calculada sobre el par variable–objetivo para cada *lag*. En este caso se usó como apoyo para revisar direccionalidad temporal plausible entre una variable y el target, comparando cuánto aporta el pasado de la variable sobre la incertidumbre del objetivo más allá del propio pasado del objetivo. Su uso en esta etapa fue estrictamente exploratorio y de priorización relativa, no de inferencia causal fuerte. En conjunto, estas tres métricas se usaron como criterio exploratorio de sensibilidad temporal. La lógica fue directa: si una variable mostraba valores relativamente altos de MI o TE en un rango acotado de retardos, y además ese patrón era consistente visualmente, entonces pasaba a ser una candidata razonable para la selección inicial.

4.4.4. Consolidación visual: mapas de calor con escala local y referencia global

Dado que en esta etapa el universo todavía estaba en el orden de 700 a 800 variables, la revisión caso a caso no era práctica. Por eso los resultados se consolidaron en mapas de calor agrupados por bloques de aproximadamente 10 variables por figura.

Cada figura se construyó de la siguiente manera. Para cada variable se dispusieron tres filas, correspondientes a CORR, MI y TE, y como columnas se usaron los distintos *lags* evaluados. En cada celda se anotó tanto el valor numérico como una versión escalada para facilitar lectura rápida entre métricas y entre retardos.

La escala de color se definió usando una normalización tipo z y una referencia global compartida entre grupos. Eso permitió mantener comparabilidad visual entre figuras distintas. En el caso de CORR, el cero se mantuvo centrado, de modo que relaciones negativas y positivas conservaran peso visual comparable. En MI y TE, donde los valores son no negativos, la escala operó como un ranking relativo: los valores bajos quedaron en la parte inferior y los altos en la superior, manteniendo la misma referencia global entre grupos.

Como resultado, se generó un compendio de aproximadamente $\sim$ 76 figuras. Ese compendio permitió ver con relativa rapidez qué variables mostraban señal útil de forma persistente y en qué retardos tendían a concentrarse sus máximos relativos. La Figura 9 no

corresponde a una reproducción literal de una de esas salidas, sino a una ilustración construida para mostrar de manera más clara la lógica de lectura de este tipo de consolidación visual.

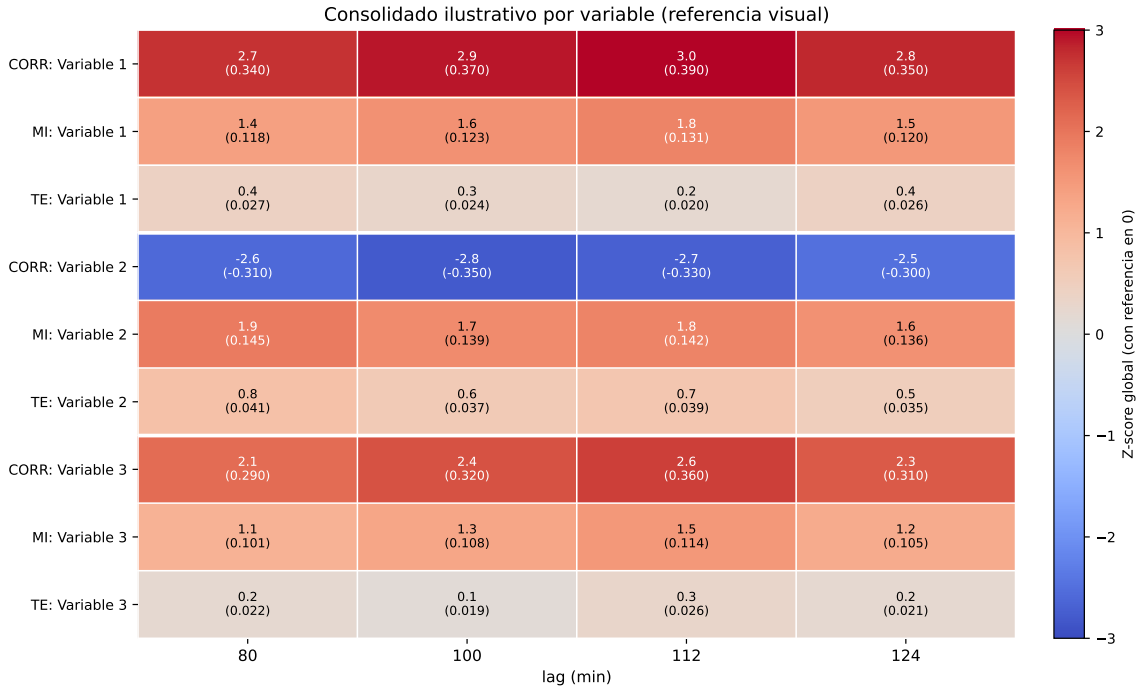

Figura 9: Ilustración del esquema de consolidación por variable y *lag*. Cada variable se representa con tres filas (CORR, MI y TE) y columnas correspondientes a los retardos evaluados. La escala de color se muestra con una referencia global para facilitar la comparación entre grupos distintos.

4.4.5. Criterio práctico de lectura y selección inicial

La lectura de los mapas se hizo de forma manual y asistida por las tres métricas. En términos prácticos, se priorizaron variables que mostraban valores relativamente altos y estables de MI y TE en un rango acotado de *lags*. Cuando aparecían varios retardos competitivos para una misma variable, se tendió a privilegiar aquel con mayor TE dentro del conjunto razonable. La correlación de Pearson se utilizó como complemento, sobre todo para identificar relaciones lineales claras y para no perder el signo de la asociación.

La selección inicial no salió de una regla automática única ni de un ranking ciego. Salió de una lectura guiada por los mapas y por la coherencia del patrón temporal observado. En esta etapa resultó más útil descartar rápido variables sin señal clara y retener candidatas defendibles desde el comportamiento observado que imponer una optimización cerrada demasiado temprano.

4.4.6. Iteraciones y nota histórica

Este análisis se repitió en tres ciclos. En ciclos posteriores, además de rehacer el barrido completo, se probaron subconjuntos de variables que, según la revisión previa, parecían tener mayor potencial. La expectativa era refinar la selección antes de entrar a etapas más pesadas del pipeline.

Visto en retrospectiva, esa iteración por subconjuntos no fue especialmente determinante. El problema principal no estaba solo en encontrar mejores candidatas dentro del barrido, sino en la madurez general del pipeline, en la forma de construir el espacio experimental y en cómo evaluar después los subconjuntos. Bajo esas condiciones, el valor principal de esta etapa fue dejar trazabilidad sobre sensibilidad temporal y producir una primera reducción razonable del universo de variables.

4.4.7. Aclaraciones adicionales: nomenclatura y frecuencia de descarga

En las figuras de esta etapa aparece en varias variables el sufijo `_mean_mean`. Ese sufijo corresponde al esquema de descarga y nombramiento usado en ese momento. No implica que sobre la variable ya se haya aplicado un promedio adicional distinto del considerado en la extracción original. En esta subsección tiene solo un efecto de nomenclatura.

Además, por el volumen de variables evaluadas en esta fase, la descarga se ejecutó con una resolución de 2 minutos en lugar de 1 minuto. Por eso los *lags* aparecen como valores pares. Esta diferencia corresponde a una etapa previa del trabajo y explica la discrepancia respecto de la resolución general descrita en la subsección de extracción. Fuera de este análisis puntual, el desarrollo posterior se realizó con los criterios y la resolución ya fijados para el resto del pipeline.

4.4.8. Evaluación de unimodalidad vs. bimodalidad del *target* mediante GMM

Motivación. Durante el desarrollo del modelo apareció con frecuencia una tendencia a predecir hacia la media. Una explicación posible era que el target estuviera mezclando dos regímenes operacionales relativamente separados. Bajo esa hipótesis, esta subsección se utilizó para revisar si la distribución del target se ajustaba mejor a un modelo unimodal o a una mezcla bimodal simple.

Datos. Para este diagnóstico se utilizó únicamente el subconjunto de laboratorio del `target_unificado`, por ser la fuente más estable y trazable para estudiar la forma de la distribución. En ese subconjunto se tuvo $n = 1215$ observaciones, con media $\bar{y} = 89.645$ y desviación estándar $s_y = 3.096$, en escala original de porcentaje.

Método. Se ajustó un *Gaussian Mixture Model* unidimensional comparando dos hipótesis: $K = 1$ componente y $K = 2$ componentes. El ajuste se realizó sobre la variable

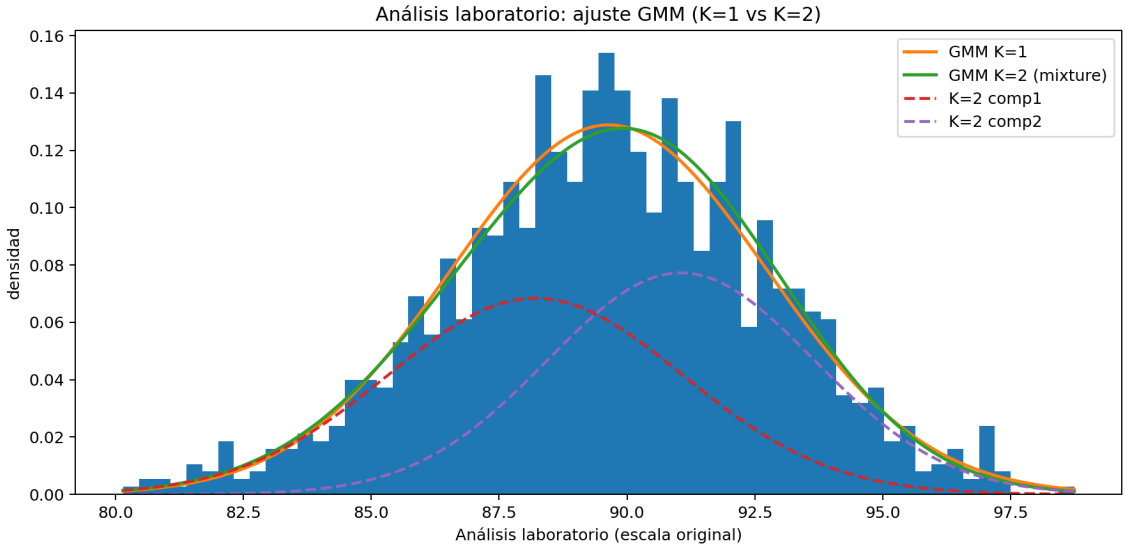

Figura 10: Histograma del *target* en el subconjunto de laboratorio y ajuste de GMM 1D para $K = 1$ y $K = 2$, mostrado en escala original.

estandarizada para mejorar estabilidad numérica, pero los parámetros se reportaron luego en escala original. La estimación se hizo mediante EM, con 40 reinicios y selección de la mejor solución según verosimilitud.

La comparación entre ambos modelos se realizó con AIC y BIC. Si p es el número de parámetros libres, n el tamaño muestral y LL la log-verosimilitud total evaluada en el óptimo, entonces

$$\mathrm{AIC} = 2p - 2LL, \qquad \mathrm{BIC} = p\log(n) - 2LL. \tag{61}$$

En ambos casos se prefieren valores menores. La diferencia práctica es que BIC penaliza con más fuerza modelos más complejos cuando crece el tamaño muestral.

Resultados. Los valores obtenidos fueron los siguientes:

- **Log-verosimilitud total:** $\ell_{\mathrm{tot}}(K = 1) = -1724.010$ y $\ell_{\mathrm{tot}}(K = 2) = -1722.427$.
- **AIC:** 3452.021 para $K = 1$ y 3454.854 para $K = 2$.
- **BIC:** 3462.226 para $K = 1$ y 3480.367 para $K = 2$.
- **Parámetros para $K = 1$:** $\mu = 89.645$, $\sigma = 3.096$.
- **Parámetros para $K = 2$:** $\pi = (0.495,\ 0.505)$, $\mu = (88.211,\ 91.054)$, $\sigma = (2.891,\ 2.605)$.
- **Ambigüedad de asignación en $K = 2$:** la cantidad máx(γ) tuvo media 0.696 y mediana 0.689. La fracción ambigua fue 0.527 con umbral 0.7 y 0.773 con umbral 0.8.

Visualmente, el ajuste con $K = 2$ no mostró una separación marcada entre modos. Eso es consistente con las responsabilidades obtenidas: una parte importante de las observaciones quedó asignada de forma blanda entre ambas componentes, más cerca de una transición continua que de una partición limpia en dos grupos.

Conclusión. Bajo esta comparación, la evidencia favoreció el modelo unimodal. Tanto AIC como BIC prefirieron $K = 1$, por lo que no se justificó modelar el target con una mezcla gaussiana bimodal simple. Esto no descarta la existencia de regímenes operacionales. Lo que indica es algo más acotado: si esos regímenes existen, no aparecen aquí como dos modos claramente separados en la distribución del objetivo. En términos prácticos, este resultado sirvió para descartar la explicación más simple de "bimodalidad marginal" y dejar abierto que una eventual estructura de regímenes dependa de contexto, tiempo o combinaciones de otras variables, más que de la forma marginal aislada del objetivo.

4.5. Selección inicial de características

En este punto se consolidó el primer universo formal de variables independientes con el que se trabajó en adelante. La idea fue unir dos fuentes de candidatas: por un lado, las señales que habían mostrado mejor comportamiento en el análisis de sensibilidad por *lags*; por otro, un listado complementario propuesto por un tercero con un criterio distinto de selección. El objetivo no fue cerrar todavía el set final, sino reducir el riesgo de dejar afuera señales potencialmente útiles por depender de una sola ruta de filtrado.

4.5.1. Conjunto base desde el análisis de sensibilidad/causalidad

A partir del compendio de mapas de calor se construyó un listado de candidatas prioritarias, del orden de ~ 300 señales. La retención se hizo usando el criterio manual aplicado en esa etapa, conservando las variables que alcanzaban una puntuación de al menos 5 en la lectura del análisis. Este listado pasó a ser el núcleo del conjunto base, ya que resume la primera reducción sobre el universo original de ~ 700 a ~ 800 variables evaluadas.

4.5.2. Incorporación de un set complementario propuesto por un tercero

En paralelo, se incorporó un conjunto adicional de variables sugerido por un tercero que trabajó el mismo problema con otra lógica de selección y con conocimiento operativo del área. Ese set tenía solapamiento parcial con el conjunto base, pero también incluía señales que no habían quedado retenidas en la etapa exploratoria inicial. Su incorporación se hizo como una ampliación deliberada del universo de candidatas, asumiendo que más adelante varias de ellas serían descartadas en etapas de depuración o ablación.

4.5.3. Unión de listados y tratamiento práctico de retardos

La consolidación se hizo como una unión de ambos listados, eliminando duplicados cuando había *overlap*. En las variables que venían del análisis de sensibilidad se mantuvieron los *lags* seleccionados en esa etapa, según el criterio cualitativo ya descrito.

En cambio, para las variables aportadas por el tercero que no estaban en el listado base y, por tanto, no tenían un *lag* asociado desde ese análisis, se usó un retardo fijo representativo del rango observado, de aproximadamente 100 minutos. Ese valor no se planteó como un óptimo ni como una estimación cerrada del retardo físico real para cada señal. Se usó como una aproximación práctica para dejar el dataset en una forma homogénea y operable en esa fase del proyecto, privilegiando continuidad metodológica y viabilidad del pipeline por sobre ajuste fino variable a variable.

4.5.4. Resultado: universo de variables para etapas posteriores

Como resultado se obtuvo un dataset consolidado de características, cuya cardinalidad total se reporta más adelante, y que se adoptó como *pool* oficial de variables independientes para las etapas posteriores del flujo: preparación del espacio experimental, ablación, selección fina y modelamiento. Para mantener trazabilidad, los listados de origen y su consolidación se conservaron como artefactos reproducibles.

4.6. Preparación del espacio experimental para la ablación de características

Antes de ejecutar la ablación se implementaron dos pasos de preparación. El primero buscó estandarizar el uso del target bajo distintos niveles de exigencia. El segundo buscó depurar el universo de variables por trazabilidad, gobernanza e higiene mínima. En esta etapa el objetivo fue dejar preparado un espacio de trabajo consistente para la búsqueda posterior.

4.6.1. Control por *tiers* y target unificado

Para poder trabajar con distintas fuentes del target bajo una misma estructura, se construyó una representación unificada con lógica por *tiers*. En términos prácticos, esto permitió manejar sobre una misma base observaciones de laboratorio, observaciones de sensor y variantes mixtas, y luego definir por configuración qué subconjunto de filas o qué fuente del target se usaría en cada caso.

La ventaja de este diseño fue operativa. En vez de reconstruir un dataset distinto para cada escenario, se dejó preparada una base común con la información necesaria para distinguir entre observaciones de laboratorio, observaciones de sensor y variantes

mixtas. Con eso se redujo fricción al momento de cambiar configuraciones y se mantuvo trazabilidad sobre qué régimen de datos se estaba usando en cada corrida.

Ahora bien, aunque en esta preparación general se trabajó con las tres variantes, la ruta que finalmente se retuvo para la ablación y selección de variables quedó anclada en laboratorio. Esa decisión respondió principalmente a criterios de correctitud metodológica, trazabilidad del target y prioridades del estudio en esa etapa. No implica asumir que una ablación apoyada en datos mixtos carezca de valor, sino solo que no fue la ruta retenida en esta memoria.

4.6.2. Depuración inicial del universo de variables: higiene, trazabilidad y prevención de sesgos

Además del filtrado cuantitativo por faltantes descrito en la limpieza, se aplicó una depuración cualitativa y estructural sobre el universo de variables. El objetivo fue sacar señales que, aun pudiendo correlacionar con el objetivo, fueran difíciles de justificar operativamente o introdujeran riesgos metodológicos innecesarios.

Las exclusiones se agruparon en cuatro clases. La primera corresponde a descartes por revisión experta u operacional: variables eliminadas por indicación de contrapartes técnicas del área, ya sea por irrelevancia operativa, semántica ambigua o baja confianza en su trazabilidad. La segunda corresponde a descartes por criterio del autor, asociados al riesgo de *proxies*: señales demasiado agregadas o contextuales que podían inflar desempeño sin aportar interpretabilidad útil para operación. La tercera corresponde a variables con nombres no descriptivos o semántica no recuperable, en particular algunas incorporadas al sumar listados externos. La cuarta corresponde a variables contemporáneas al objetivo o derivadas del mismo dominio de análisis, que podían actuar como atajos informativos. Aunque ese último caso no implica necesariamente fuga formal en todos los escenarios, en esta etapa se excluyó para mantener una separación más limpia entre objetivo y predictores.

Además de estas reglas por revisión directa, se definieron exclusiones automáticas por **substrings**. Con esto se eliminaron familias de señales que correspondían al propio objetivo o a variantes nominales del mismo, y también variables demasiado globales de planta, como utilidades o agregados extensos, que podían contener señal pero no eran defendibles operativamente para este caso. Esta limpieza no se usó como argumento de desempeño, sino como control de gobernanza previo a la experimentación.

4.7. Ablación de características: loop de búsqueda iterativa

Esta subsección describe la mecánica de trabajo usada para la ablación y selección de características en la etapa experimental. El interés acá no está en adelantar qué subconjuntos terminaron siendo mejores, sino en dejar explícito el procedimiento con que se fueron proponiendo, evaluando y aceptando cambios sobre el set de variables.

4.7.1. Antecedente: una ruta *greedy* previa y el set inicial reducido

Antes del loop que finalmente se consolidó, se probó una ruta *greedy* de ablación que no resultó especialmente efectiva para converger a un set satisfactorio. Aun así, esa exploración dejó un producto útil: un conjunto inicial reducido respecto del universo completo. Ese subconjunto se tomó como punto de partida del loop posterior. En otras palabras, esta etapa no arranca desde todas las variables, sino desde un espacio ya recortado por una iteración previa.

4.7.2. Validación temporal: *folds* purgados con embargo

Las evaluaciones del loop se basaron en una partición temporal con tres niveles: *holdout*, calibración y *CV pool*. La referencia final se dejó en laboratorio. El *holdout* se definió como el tramo LAB más reciente y se mantuvo intocable. Antes de él se reservó una ventana de calibración también construida con mediciones LAB válidas. La exploración interna de variables se realizó sobre el *CV pool*, es decir, sobre los datos previos al inicio de la calibración.

Dentro de ese *CV pool* se generaron *folds* temporales purgados con embargo, con el fin de reducir fuga temporal y dependencia espuria entre entrenamiento y prueba. En esta implementación se usó un embargo fijo de un día. Los bloques de test interno también se definieron sobre *timestamps* LAB válidos, de modo que la comparación entre subconjuntos se mantuviera alineada con la fuente de referencia adoptada para esta etapa.

Aunque el script contempla otros modos en la configuración y durante el desarrollo se trabajó también con variantes de sensor y mixtas, la ruta retenida para ranking y ablación quedó apoyada en laboratorio. Bajo estas condiciones, esta validación debe leerse como un esquema temporal LAB-only usado para comparar subconjuntos de variables dentro del *CV pool*.

4.7.3. Preparación del pool de candidatas y reglas de exclusión

A partir del dataset numérico se construyó un *pool* de variables candidatas. Para formarlo se excluyeron el target y sus columnas auxiliares, como *flags*, columnas de trazabilidad y variables vetadas por lista explícita o por familias definidas mediante *substrings*. También se excluyeron variables no numéricas o con conversión numérica insuficiente.

Este *pool* constituye el universo desde el cual el algoritmo puede proponer adiciones o *swaps*. La gracia de dejarlo fijado al inicio de cada corrida es que la búsqueda opera sobre un espacio controlado y no sobre una mezcla cambiante de señales disponibles. En paralelo, se mantuvo un conjunto de variables bloqueadas que no se eliminaban del set activo, por considerarse ancladas al proceso y de alta confiabilidad operacional.

4.7.4. Ranking proxy para proponer movimientos

En cada iteración se utilizó un ranking proxy de candidatas para priorizar propuestas de cambio. En esta ruta, dicho ranking se construyó a partir de una lógica tipo RMI, calculando información mutua entre cada *feature* y el residual *out-of-fold* obtenido en el esquema LAB-only. La idea fue usar una señal relativamente barata para ordenar el espacio de búsqueda antes de pasar a evaluaciones más costosas.

Este ranking no se interpretó como decisión final. Se usó como una heurística de priorización. En la práctica, permitía acercar al frente de la búsqueda variables con mayor probabilidad de aportar información complementaria respecto del residuo, dejando más atrás aquellas que, bajo este criterio preliminar, ofrecían menos expectativa de mejora.

Bajo esta formulación, el ranking deja de depender solo de la relación directa *feature*–target y pasa a mirar qué información adicional podría aportar una variable una vez considerado el error del modelo base. Para esta etapa, ese criterio resultó suficiente como filtro inicial antes de entrar a la evaluación iterativa del subconjunto.

4.7.5. Loop iterativo de búsqueda: acciones, evaluación por etapas y aceptación

Sobre el conjunto actual de variables se propusieron movimientos de tres tipos: agregar una variable, remover una variable o hacer un *swap* entre una variable del conjunto activo y una del *pool*. No todas las propuestas pasaban directo a evaluación completa. Primero se usó una revisión más liviana para filtrar movimientos evidentemente malos y reservar la evaluación cara para los casos con alguna posibilidad razonable.

Si una propuesta mostraba una mejora suficientemente clara en la métrica objetivo, se aceptaba directamente. Si la mejora era marginal, o incluso levemente desfavorable, aún podía aceptarse con cierta probabilidad. Esa regla se dejó para evitar que la búsqueda se quedara detenida demasiado pronto en óptimos locales. Cuando la exploración se estancaba por varias iteraciones, se gatillaban reinicios controlados desde el mejor conjunto conocido de igual cardinalidad o desde el mejor global, aplicando luego mutaciones parciales para reactivar la búsqueda.

No se planteó esto como un algoritmo exacto de optimización. Fue una heurística de búsqueda guiada, pensada para recorrer un espacio amplio de subconjuntos sin pagar el costo de una exploración exhaustiva.

4.7.6. Persistencia, trazabilidad y reproducibilidad del proceso

El procedimiento dejó trazas sistemáticas de ejecución. Se almacenó historial por iteración, registro de propuestas aceptadas y rechazadas, mejores conjuntos observados, estados aleatorios y archivos de salida para reconstruir el espacio de variables, las particiones y los rankings generados. Esto era importante porque la búsqueda tenía un componente

heurístico y además un costo computacional alto. Sin ese registro, después habría sido difícil reconstruir por qué se llegó a cierto subconjunto y no a otro.

Con estas trazas, la evolución del proceso quedó auditable dentro de las condiciones fijadas para cada corrida, aun cuando la búsqueda no siguiera una ruta determinista única.

4.7.7. Esquema operativo del algoritmo de búsqueda

En términos operativos, el algoritmo seguía el siguiente patrón general: partir desde un conjunto inicial reducido, construir el *pool* de candidatas permitido, proponer movimientos usando el ranking proxy, filtrar rápidamente propuestas poco prometedoras, evaluar con mayor costo las candidatas supervivientes, aceptar o rechazar según mejora y regla de exploración, y finalmente registrar el estado antes de pasar a la siguiente iteración. Cuando la búsqueda se estancaba, se aplicaban reinicios controlados y nuevas mutaciones parciales.

La importancia de dejar este esquema explícito no está en su sofisticación formal, sino en que permite entender qué tipo de proceso generó los subconjuntos evaluados más adelante.

4.7.8. Alcance metodológico de la ablación respecto de los regímenes de datos

El script utilizado para esta etapa fue preparado con soporte para más de un modo de entrenamiento. En la configuración aparecen explícitamente las variantes `lab_only`, `sensor_only` y `mixed`. Durante el desarrollo general también se trabajó con esas tres variantes. Sin embargo, la ruta que finalmente quedó retenida para la ablación y el ranking de variables fue la anclada en laboratorio.

Esto se refleja en varios puntos del procedimiento. El *holdout* final se define sobre mediciones LAB válidas, la calibración también se construye con LAB, el *CV pool* se recorta a partir del inicio de esa calibración, y los bloques usados para evaluación interna se seleccionan sobre tiempos LAB válidos. Además, el ranking tipo RMI utilizado para priorizar candidatas se planteó sobre residuales *out-of-fold* en un esquema LAB-only.

La razón de fondo fue metodológica y práctica. En esta etapa se privilegió una referencia de target más trazable y más fácil de defender, de modo de evitar que la selección de variables quedara demasiado acoplada a una señal cuya continuidad y calidad efectiva ya habían requerido una limpieza más agresiva. A eso se sumaron restricciones de tiempo y foco del estudio: dado el costo computacional y la cantidad de decisiones abiertas en el pipeline, se optó por cerrar primero una ruta de ablación consistente sobre laboratorio antes de abrir una comparación completa entre regímenes.

Por esta razón, en esta etapa no conviene presentar la ablación como una comparación consolidada entre `lab_only`, `sensor_only` y `mixed`. Lo que sí puede afirmarse es algo más acotado: el procedimiento se dejó preparado para trabajar con esos tres regímenes, pero la ruta efectivamente retenida para selección de variables quedó anclada en laboratorio. Bajo

estas condiciones, la ablación opera acá como una búsqueda guiada sobre una referencia más trazable del target, mientras que la comparación entre regímenes se deja para los resultados experimentales posteriores.

Núcleo inamovible. A lo largo de la búsqueda se mantuvo un conjunto de variables que no se eliminaban del subconjunto activo. Se trató de señales consideradas ancladas al proceso y de alta confiabilidad operacional. En esta etapa ese núcleo incluía contenido de O_2 en gases de combustión de la zona media, flujo de licor negro a boquillas, sólidos secos quemados y flujo de agua de alimentación.

Heurística muestra–características y consideraciones de dimensionalidad. Durante la ablación se intentó preservar una relación mínima entre cantidad de muestras efectivas y número de variables. Como guía práctica, se buscó idealmente un orden de magnitud cercano a ~ 50 muestras por característica. Este umbral no se trató como garantía formal ni como regla universal, sino como una heurística de prudencia para no aumentar complejidad demasiado rápido dentro de un espacio experimental todavía grande.

Costo computacional. En el hardware utilizado, una corrida completa de este *pipeline* de ablación y búsqueda podía tomar del orden de ~ 12 horas. Ese costo se asumió como aceptable en esta etapa, dado que el objetivo era explorar el espacio de variables de forma controlada antes de fijar el protocolo experimental definitivo.

5. Resultados experimentales

Esta sección consolida los resultados de las ejecuciones experimentales del soft-sensor, utilizando artefactos reproducibles generados por el pipeline. Los resultados se reportan sobre los conjuntos *test* y *holdout*, definidos mediante partición temporal basada en mediciones de laboratorio.

5.1. Marco común de evaluación

En esta sección se comparan las configuraciones experimentales bajo un mismo contrato de evaluación. Todas las corridas usan la misma partición temporal, definida sobre mediciones LAB elegibles. La lógica general es fija para toda la sección: se reserva un bloque final de holdout, se deja una guarda temporal previa a ese bloque y, dentro de la zona pre-holdout, se define un conjunto test también basado en mediciones LAB. Bajo estas condiciones, la lectura principal de desempeño se apoya en holdout, ya que representa el tramo final fuera de muestra.

Cuadro 1: Parámetros estructurales comunes del esquema de evaluación temporal.

Parámetro	Valor	Descripción
MIN_TARGET	65.0	Umbral mínimo de elegibilidad del target
TEST_N_LABS	180	Número de mediciones LAB elegibles en test
HOLDOUT_N_LABS	36	Número de mediciones LAB elegibles en holdout
HOLDOUT_GUARD_DAYS	1.0 día	Guarda temporal previa a holdout
TEST_EMBARGO_DAYS	1.0 día	Embargo temporal alrededor de test

Cuadro 2: Instanciación temporal del esquema común de evaluación.

Conjunto / ventana	Tamaño	Instanciación temporal
Test	180 LAB	desde 2025-06-14 11:00 hasta 2025-10-05 11:00
Holdout	36 LAB	desde 2025-11-06 12:00 hasta 2025-11-24 21:30
Guarda previa a holdout	1.0 día	inicio de guarda en 2025-11-05 12:00
Embargo de test	1.0 día	exclusión temporal antes y después de test

En términos cronológicos, el conjunto test ocupa el último tramo evaluable de la zona pre-holdout, mientras que holdout corresponde al bloque final fuera de muestra, separado por una guarda explícita de un día. Tanto test como holdout se definen exclusivamente sobre mediciones LAB elegibles. Esto fija una referencia común para todas las comparaciones de la sección, aun cuando el tamaño efectivo del conjunto de entrenamiento cambie según el modo de datos utilizado.

Las métricas que se reportan en esta sección son RMSE, MAE y R^2. La comparación entre configuraciones se realiza priorizando RMSE mínimo, luego MAE mínimo y, si todavía persiste empate, mayor R^2. Para compactar la presentación, las tablas reportan la familia del modelo y no el preset ni el detalle completo de hiperparámetros. Esa información queda fuera del cuerpo principal, porque acá interesa concentrar la lectura en qué configuraciones responden mejor al problema bajo el mismo esquema de evaluación.

Las abreviaciones usadas en tablas son las siguientes: Vars para espacio de variables, Rep para representación, Modo para modo de entrenamiento y Fam para familia del modelo. Los espacios de variables se resumen como All, Fix y Ext. La representación se resume como Raw o PLS. Cuando sirve para compactar, el modo de entrenamiento se indica como L para laboratorio, S para sensor y M para mixto.

Una vez fijado este marco, cada bloque experimental se limita a una comparación puntual. La idea es evitar repetir la misma estructura por laboratorio, sensor y mixto en cada subsección, y concentrar la lectura en los resultados retenidos y en cómo estos se relacionan con el problema planteado.

5.2. Baseline de referencia

Antes de comparar modelos supervisados, se fija una referencia base simple. En este caso se usa el promedio móvil de la última semana del target de laboratorio. Se mantiene exactamente el mismo esquema de evaluación definido en el marco común: test con 180 muestras LAB, holdout con 36 muestras LAB y una separación temporal de 1 día entre los bloques. La idea de este baseline no es competir con las configuraciones más elaboradas, sino dejar un piso comparativo claro para el resto de la sección.

Cuadro 3: Desempeño del baseline de referencia en test y holdout.

Split	RMSE	MAE	R^2
Test	3.4858	2.3204	-0.2948
Holdout	3.6590	2.3635	-0.1440

El baseline entrega un RMSE de 3.4858 en test y 3.6590 en holdout. La diferencia entre ambos bloques es acotada, por lo que funciona como una referencia relativamente estable. Sin embargo, el nivel de error sigue siendo alto para dejarlo como solución final. Bajo estas condiciones, el baseline se usa como piso comparativo: una configuración retenida más adelante debería mejorar esta referencia en holdout, o al menos mostrar una ventaja clara que justifique su complejidad.

5.3. Espacio All: comparación entre representación directa y PLS

En este bloque se revisa el comportamiento del espacio All, es decir, el universo completo de variables retenidas para la etapa experimental. La comparación se centra en dos formas de representar ese espacio: uso directo de las variables y proyección PLS antes del modelo final. La idea acá es ver si, en un espacio grande y más expuesto a redundancia, la compresión supervisada de PLS ayuda a mejorar la generalización fuera de muestra.

Cuadro 4: Mejor configuración en test para el espacio All.

Rep	Modo	Fam	RMSE
Raw	M	Random Forest	2.7069
PLS	M	Ridge	2.8062

Cuadro 5: Mejor configuración en holdout para el espacio All.

Rep	Modo	Fam	RMSE	R^2
Raw	M	HGBR	2.9871	0.2376
PLS	L	Ridge	2.6924	0.3806

En test, la mejor configuración del bloque All aparece con representación directa, modo mixto y Random Forest. PLS también se mantiene competitivo en esa partición, pero queda algo por detrás, con una configuración mixta basada en Ridge.

La lectura cambia al pasar a holdout. Ahí la mejor configuración del bloque aparece con PLS, modo laboratorio y Ridge, con RMSE de 2.6924 y R^2 de 0.3806. La mejor variante directa válida en holdout no queda en laboratorio, sino en modo mixto con HGBR, y alcanza un RMSE de 2.9871 con R^2 de 0.2376. Bajo estas condiciones, en el espacio All la representación PLS sí entrega una ventaja clara en el tramo final de evaluación.

También cambia la familia que domina según el split. En test, las variantes más competitivas se apoyan en modelos de árboles o ensambles sobre una representación más directa. En holdout, en cambio, la mejor salida queda en una combinación más estable, con PLS y Ridge. Esto sugiere que, en el espacio All, una representación más compacta ayuda a controlar mejor la degradación fuera de muestra.

En cualquier caso, tanto la mejor variante directa como la mejor variante con PLS mejoran el baseline de referencia en holdout. Sin embargo, la diferencia entre ambas ya no es marginal. En este bloque, la mejor señal final queda del lado de PLS.

La comparación entre variantes con lag y sin lag se deja para la subsección específica de ese contraste. Acá lo que se retiene es algo más acotado: dentro de la representación

base usada en esta parte, el espacio All favorece PLS por sobre la representación directa cuando la decisión se apoya en holdout.

5.4. Espacio Fix: subconjunto fijo de variables

En este bloque se revisa el subconjunto fijo de variables, denotado como Fix. A diferencia del espacio All, acá no se trabaja con el universo completo, sino con un set más acotado y ya bastante curado. La idea es ver si esa reducción del espacio de entrada permite sostener o incluso mejorar la generalización fuera de muestra sin recurrir todavía a una representación latente. En esta parte se resume la variante directa. La comparación con PLS sobre este mismo espacio se deja para la subsección siguiente.

Cuadro 6: Configuraciones principales del espacio Fix con representación directa.

Split	Modo	Fam	RMSE
Test	M	Random Forest	2.6048
Holdout	M	XGBoost	2.5085

En test, la mejor configuración del bloque aparece con representación directa, modo mixto y Random Forest. Eso muestra que el subconjunto fijo no pierde competitividad al recortar el espacio de entrada. Sigue dejando configuraciones fuertes, incluso antes de pasar al bloque final de evaluación.

La lectura principal, sin embargo, está en holdout. Ahí la mejor configuración del bloque queda en modo mixto con XGBoost y alcanza un RMSE de 2.5085. Bajo el criterio fijado para esta sección, este pasa a ser el mejor resultado global del paquete válido. No solo mejora con claridad al baseline, sino que además queda por delante de las mejores configuraciones retenidas en All y Ext.

También conviene notar que el buen comportamiento de Fix no depende solo de una combinación puntual. En la síntesis general del capítulo, la variante con laboratorio y representación directa también se mantiene competitiva dentro de este mismo espacio, lo que sugiere que el resultado no se explica únicamente por mezclar fuentes, sino también por una selección de variables más contenida y mejor alineada con el problema.

En términos prácticos, este bloque deja una señal clara. Cuando el espacio de entrada se reduce a un subconjunto fijo y manejable, la representación directa sigue funcionando muy bien y no necesita, al menos de entrada, una transformación adicional para sostener el mejor resultado final. Por eso este bloque queda como eje central de la sección, y la comparación con PLS que viene a continuación debe leerse como un contraste puntual sobre este mismo espacio, no como un experimento separado.

5.5. Espacio Fix: comparación entre representación directa y PLS

En esta parte se compara el mismo subconjunto fijo de variables, pero cambiando la forma de representarlo. Por un lado está la variante directa, donde las 21 variables se usan tal como entran al modelo. Por otro lado está la variante con PLS, donde primero se proyecta el espacio a un número reducido de componentes y luego se entrena el modelo final. La pregunta acá es directa: dado que el espacio Fix ya es acotado, hay que ver si PLS todavía aporta algo o si la representación directa sigue siendo suficiente.

Cuadro 7: Mejor configuración por representación en test para el espacio Fix.

Rep	Modo	Fam	Comp.	RMSE test
Raw	M	Random Forest	–	2.6048
PLS	S	Ridge	4	2.8367

Cuadro 8: Mejor configuración por representación en holdout para el espacio Fix.

Rep	Modo	Fam	Comp.	RMSE holdout	R^2 holdout
Raw	M	XGBoost	–	2.5085	0.4623
PLS	M	HGBR	20	2.6503	0.3998

La primera diferencia se ve de inmediato en test. La mejor variante directa queda en 2.6048 de RMSE, mientras que la mejor variante con PLS queda en 2.8367. La brecha no es enorme, pero sí consistente. En este subconjunto, la compresión previa no mejora el mejor resultado intermedio.

En holdout pasa algo parecido. La mejor salida con representación directa sigue siendo la configuración mixta con XGBoost, con RMSE de 2.5085 y R^2 de 0.4623. La mejor salida con PLS también queda en modo mixto, pero con HGBR y 20 componentes, alcanzando un RMSE de 2.6503 y un R^2 de 0.3998. Es un resultado competitivo, pero no supera a la variante directa.

Esto sugiere algo simple. En el espacio All, donde el número de variables es mayor y la redundancia también puede ser mayor, PLS sí tenía una justificación más clara. En Fix, en cambio, el espacio ya viene bastante controlado. Bajo esas condiciones, la proyección PLS pierde parte de su ventaja y la representación directa sigue capturando mejor la relación con el target.

También cambia la familia que aparece como mejor configurada bajo cada representación. En la variante directa domina XGBoost en holdout. En PLS la mejor salida queda en HGBR. No parece ser un cambio casual. Más bien da la impresión de que la proyección

latente modifica la estructura del problema y favorece familias algo más estables, pero sin alcanzar el mismo nivel final de ajuste que logra la variante directa en este subconjunto.

Si se revisa el holdout por mitades, la diferencia tampoco desaparece. La mejor configuración directa logra RMSE de 1.1275 en la primera mitad y 3.3637 en la segunda. La mejor configuración con PLS queda en 1.3509 y 3.4962, respectivamente. Es decir, la representación directa queda por delante en ambas partes del bloque final.

Bajo este contraste, la lectura que se retiene es directa. En el espacio Fix, PLS no mejora el resultado principal. Sigue siendo una alternativa válida, pero la mejor señal del bloque queda en la representación directa. Por eso, para la síntesis final, Fix se interpreta principalmente a través de su variante directa, mientras que PLS queda como referencia secundaria dentro del mismo espacio.

5.6. Espacio Ext: subconjunto ampliado de variables

En este bloque se revisa el espacio Ext, que corresponde a un subconjunto ampliado de 149 variables. La idea acá es intermedia respecto de los bloques anteriores. No es un espacio tan grande como All, pero tampoco tan acotado como Fix. Por eso interesa ver si este punto medio entrega un mejor compromiso entre tamaño del espacio de entrada y desempeño fuera de muestra.

Cuadro 9: Mejor configuración en test para el espacio Ext.

Rep	Modo	Fam	Comp.	RMSE test
Raw	S	ExtraTrees	–	2.8935
PLS	S	Ridge	4	2.9742

Cuadro 10: Mejor configuración en holdout para el espacio Ext.

Rep	Modo	Fam	Comp.	RMSE holdout	R^2 holdout
Raw	M	HGBR	–	2.8196	0.3207
PLS	L	Ridge	12	2.7342	0.3612

En test, la mejor configuración del bloque aparece con representación directa, modo sensor y ExtraTrees, con un RMSE de 2.8935. PLS queda cerca en esa partición, también en modo sensor, pero con Ridge y 4 componentes. La diferencia entre ambas no es grande.

La lectura principal cambia al pasar a holdout. Ahí la mejor salida del bloque no queda en la representación directa, sino en PLS, modo laboratorio y Ridge con 12 componentes. Esa configuración alcanza un RMSE de 2.7342 y un R^2 de 0.3612. La mejor variante

directa en holdout también se mantiene competitiva, con HGBR en modo mixto y RMSE de 2.8196, pero queda algo por detrás.

Bajo estas condiciones, el espacio Ext muestra un comportamiento parecido al que ya aparecía en All. Cuando el espacio de entrada vuelve a crecer y arrastra más redundancia, PLS recupera parte de su ventaja. No se ve una mejora transversal en todas las particiones, pero sí una ventaja concreta en el tramo final de evaluación, que es el que más pesa en la lectura de esta sección.

También conviene notar que este bloque no desplaza al subconjunto fijo como mejor alternativa global. El mejor holdout de Ext mejora al baseline y deja una salida válida, pero sigue por debajo de la mejor configuración retenida en Fix. Por eso Ext se interpreta acá como una solución intermedia: más contenida que All, competitiva en holdout, y con una ventaja razonable para PLS, pero sin alcanzar el mejor resultado global del capítulo.

La comparación detallada entre variantes con lag y sin lag se deja para la subsección siguiente. En esta parte, la lectura que se retiene es más acotada: dentro del espacio Ext, la mejor salida final vuelve a quedar del lado de PLS.

5.7. Comparación entre variantes con lag y sin lag

En esta parte se compara el efecto de incorporar lag en las variables de entrada. La revisión se hace por espacio de variables, manteniendo el mismo esquema de evaluación usado en el resto de la sección. La pregunta acá no es si el lag mejora siempre, sino algo más acotado: ver en qué espacios ayuda y en cuáles no.

Cuadro 11: Mejor configuración en holdout para las variantes con lag y sin lag, separadas por espacio de variables.

Vars	Variante	Rep	Modo	Fam	RMSE holdout	MAE holdout	R^2 holdout
All	con lag	PLS	L	Ridge	2.6924	1.4439	0.3806
All	sin lag	PLS	L	HGBR	2.5233	1.4885	0.4560
Fix	con lag	Raw	M	XGBoost	2.5085	1.3184	0.4623
Fix	sin lag	PLS	L	SVM	2.7143	1.5499	0.3705
Ext	con lag	PLS	L	Ridge	2.7342	1.4653	0.3612
Ext	sin lag	PLS	L	Ridge	2.8303	1.4573	0.3155

La comparación deja una señal clara, pero no una regla única. En el espacio All, la mejor configuración final aparece sin lag. Esa variante alcanza un RMSE de 2.5233 en holdout, mientras que la mejor variante con lag queda en 2.6924. Bajo estas condiciones, en el universo completo de variables el uso de lag no entrega la mejor salida final.

En Fix ocurre lo contrario. La mejor configuración con lag, en representación directa,

modo mixto y XGBoost, alcanza un RMSE de 2.5085. La mejor configuración sin lag queda en 2.7143. La diferencia no es menor. En este subconjunto, incorporar lag sí ayuda a mejorar la generalización final.

En Ext vuelve a pasar algo parecido a Fix, aunque con una brecha más acotada. La mejor configuración con lag llega a un RMSE de 2.7342, mientras que la mejor variante sin lag queda en 2.8303. Acá el efecto no es tan marcado, pero sigue favoreciendo a la variante con lag.

Si se mira el bloque completo, la conclusión es directa. El efecto del lag depende del espacio de variables. En All gana sin lag. En Fix gana con lag. En Ext también gana con lag, aunque por una diferencia menor. Por lo mismo, no se puede sostener que el lag mejora siempre ni que empeora siempre.

También conviene notar que el mejor resultado sin lag aparece en All y no en los subconjuntos reducidos. En cambio, cuando se incorporan lag, el mejor resultado global pasa a quedar en Fix. Esto sugiere que el lag no actúa de manera aislada. Su efecto depende de con qué espacio de entrada se combina y de cuánto ruido o redundancia arrastra ese espacio.

Bajo esta comparación, la lectura que se retiene para la discusión es práctica. El uso de lag sí puede aportar, pero no como decisión universal. En este problema conviene tratarlo como una decisión dependiente del espacio de variables y no como una mejora automática.

5.8. Síntesis transversal de resultados

En esta parte se resume la lectura final de la sección. La idea no es repetir los bloques anteriores, sino dejar en una vista compacta qué configuraciones siguen en pie después de comparar espacio de variables, representación, modo de entrenamiento y uso de lag bajo el mismo contrato de evaluación. Para esta síntesis se consideran solo las corridas válidas del paquete consolidado. La corrida `exp_lag_core__20260411_171335` se deja fuera, ya que fue descartada previamente por problemas en la rama PLS.

Cuadro 12: Mejor configuración en holdout por espacio de variables, separando variantes con lag y sin lag.

Vars	Lag	Rep	Modo	Fam	RMSE	MAE	R^2
All	Sí	PLS	L	Ridge	2.6924	1.4439	0.3806
All	No	PLS	L	HGBR	2.5233	1.4885	0.4560
Fix	Sí	Raw	M	XGBoost	2.5085	1.3184	0.4623
Fix	No	PLS	L	SVM	2.7143	1.5499	0.3705
Ext	Sí	PLS	L	Ridge	2.7342	1.4653	0.3612
Ext	No	PLS	L	Ridge	2.8303	1.4573	0.3155

Cuadro 13: Mejor configuración en holdout por espacio de variables, modo de entrenamiento y uso de lag.

Vars	Modo	Lag	Rep	Fam	RMSE	R^2
All	L	Sí	PLS	Ridge	2.6924	0.3806
All	L	No	PLS	HGBR	2.5233	0.4560
All	M	Sí	PLS	Ridge	2.9293	0.2668
All	M	No	PLS	Ridge	2.8730	0.2947
All	S	Sí	PLS	Random Forest	3.1679	0.1425
All	S	No	Raw	XGBoost	3.1343	0.1606
Fix	L	Sí	Raw	XGBoost	2.6583	0.3962
Fix	L	No	PLS	SVM	2.7143	0.3705
Fix	M	Sí	Raw	XGBoost	2.5085	0.4623
Fix	M	No	Raw	HGBR	2.7983	0.3309
Fix	S	Sí	PLS	HGBR	2.6994	0.3774
Fix	S	No	PLS	HGBR	2.9722	0.2452
Ext	L	Sí	PLS	Ridge	2.7342	0.3612
Ext	L	No	PLS	Ridge	2.8303	0.3155
Ext	M	Sí	PLS	HGBR	2.7690	0.3449
Ext	M	No	PLS	SVM	2.8943	0.2842
Ext	S	Sí	PLS	HGBR	3.1723	0.1401
Ext	S	No	PLS	HGBR	2.8409	0.3104

La primera tabla deja una señal bastante clara. No aparece una receta única que gane en todos los espacios. En All, la mejor salida final queda sin lag, con PLS, modo laboratorio y HGBR. En Fix, la mejor configuración retenida queda con lag, representación directa, modo mixto y XGBoost. En Ext vuelve a aparecer una solución con PLS, esta vez con lag, modo laboratorio y Ridge.

La mejor salida global del paquete válido queda en Fix. La configuración con lag, representación directa, modo mixto y XGBoost alcanza un RMSE de 2.5085 y un R^2 de 0.4623 en holdout. Bajo el criterio fijado para esta sección, esa variante pasa a ser la referencia principal del informe. Mejora al baseline con claridad y también queda por delante de las mejores configuraciones retenidas en All y Ext.

La comparación lag versus no lag ahora queda explícita en toda la síntesis. En All, las variantes sin lag superan a las variantes con lag en los tres modos de entrenamiento. En Fix ocurre lo contrario: lag mejora en L, M y S, y además deja la mejor salida global del paquete válido. En Ext también se observa una ventaja consistente de lag en los tres modos, aunque más acotada que en Fix. Bajo estas condiciones, entendemos que el efecto

del lag depende del espacio de variables. No conviene tratarlo como una mejora automática, pero tampoco como una decisión irrelevante.

La segunda tabla ayuda a ordenar mejor la lectura por conjunto de datos. En All, el mejor resultado aparece con laboratorio y sin lag. El modo mixto también favorece la variante sin lag, mientras que en sensor la mejor salida cambia a una representación directa con XGBoost. En Fix, el modo mixto con lag entrega la mejor salida del bloque y también la mejor salida global. En ese mismo espacio, lag también supera a no lag en laboratorio y sensor. En Ext vuelve a dominar laboratorio con lag, seguido por mixto con lag, mientras que sensor queda más atrás y muestra la mayor degradación en la variante con lag.

En cuanto a la representación, PLS sigue mostrando utilidad cuando el espacio de entrada es amplio o intermedio. Eso se ve con claridad en All y Ext, donde las mejores configuraciones retenidas quedan en PLS tanto con lag como sin lag. En Fix ocurre algo distinto. Una vez reducido el problema a un subconjunto más controlado, la representación directa sigue funcionando mejor en las salidas principales, en particular cuando se combina con lag y XGBoost.

Por familia de modelos tampoco aparece una dominancia única. XGBoost queda como la mejor familia en Fix y sostiene la mejor configuración global. HGBR se comporta bien en All sin lag y también aparece como mejor alternativa en varios modos de Ext. Ridge se mantiene competitivo cuando se combina con PLS, sobre todo en All con lag y en Ext. SVM logra resultados razonables en algunas variantes sin lag, pero no alcanza a desplazar a las configuraciones principales fuera de esos casos puntuales.

Si se mira la mejor configuración retenida con algo más de detalle, además se observa que el holdout no es homogéneo. En la configuración ganadora de Fix, el RMSE baja a 1.1275 en la primera mitad del holdout y sube a 3.3637 en la segunda. Esto sugiere que el tramo final del período sigue siendo más exigente y que incluso la mejor salida retenida enfrenta un bloque más difícil hacia el cierre.

Bajo estas condiciones, la lectura final de la sección queda bastante directa. El subconjunto Fix entrega la mejor solución del paquete válido. PLS aporta cuando el espacio de entrada sigue siendo amplio, pero no desplaza a la representación directa en el subconjunto fijo. El lag ayuda de forma clara en Fix y Ext, pero no en All. Con esto ya queda un overview completo de la sección y se puede pasar a discusión con una síntesis más ordenada.

6. Discusión de resultados

Esta sección interpreta de forma integrada los hallazgos presentados en la Sección 5. A diferencia del capítulo de resultados experimentales, cuyo foco estuvo en reportar comparaciones empíricas entre configuraciones, espacios de variables, modos de entrenamiento y uso de lag, el objetivo aquí es discutir qué significan esos resultados dentro del problema abordado por esta memoria. En consecuencia, esta sección no introduce nuevas corridas ni nuevas tablas principales, sino que organiza la evidencia ya presentada para extraer una lectura global, contrastarla con las preguntas de investigación e hipótesis planteadas previamente, y delimitar con mayor claridad el alcance efectivo de los hallazgos obtenidos.

La discusión se estructura en seis partes complementarias. Primero, se propone una lectura global de los resultados, orientada a identificar los patrones empíricos más consistentes del estudio. Luego se discuten los hallazgos según el modo de datos utilizado, distinguiendo entre entrenamiento con datos de laboratorio, datos de sensor y datos mixtos. En tercer lugar, se analiza el papel de la representación y del espacio de variables, contrastando el comportamiento de los espacios All, Fix y Ext, junto con el uso de representación directa y PLS. Posteriormente, se revisan las implicancias metodológicas de la comparación entre test y holdout, pero ahora desde una lectura prudente del contrato de evaluación efectivamente utilizado. Después, se contrastan los resultados con las preguntas de investigación y las hipótesis formuladas en el planteamiento del estudio. Finalmente, se explicitan los principales alcances, límites y proyecciones inmediatas que se desprenden de esta memoria.

6.1. Lectura global de los resultados

La lectura global de la Sección 5 cambia de manera importante respecto de versiones anteriores del informe. La síntesis final ya no apunta a una configuración dominante en todos los escenarios, sino a un comportamiento claramente condicionado por el espacio de variables, el uso de lag, la forma de representación y el modo de entrenamiento. En ese sentido, el primer hallazgo relevante es que no aparece una receta universal. Lo que aparece, más bien, es un conjunto acotado de combinaciones que funcionan bien bajo condiciones específicas y que obligan a leer el problema como un espacio de diseño, no como una competencia entre componentes aislados.

Dentro de ese espacio, el subconjunto Fix entrega la mejor solución global retenida en holdout. La configuración con lag, representación directa, entrenamiento mixto y XGBoost alcanza el mejor desempeño final del paquete válido y pasa a ser la referencia principal del informe. Esto no significa que el subconjunto fijo resuelva por sí solo el problema ni que deba entenderse como una verdad definitiva sobre selección de variables. Lo que sí muestra es que una reducción más controlada del espacio de entrada puede mejorar de forma efectiva la generalización cuando se combina con una representación y una familia de modelos que aprovechen bien esa estructura.

Un segundo punto importante es que el efecto del lag no puede leerse como una mejora automática. En All, las variantes sin lag son las que terminan imponiéndose. En Fix ocurre lo contrario: las variantes con lag no solo superan a sus pares sin lag, sino que además dejan la mejor salida global. En Ext también se observa una ventaja a favor de lag, aunque más moderada que en Fix. La conclusión práctica no es que el lag siempre ayude ni que deba descartarse cuando no mejora en todos los casos. La conclusión es más acotada: su efecto depende del espacio de variables con el que se combine y del tipo de redundancia o estructura que ese espacio arrastre.

La comparación entre espacios también deja una lectura útil sobre representación. En All y Ext, las configuraciones más competitivas quedan asociadas a PLS, lo que sugiere que la compresión supervisada sigue teniendo valor cuando el espacio de entrada es amplio o intermedio y todavía existe ruido, colinealidad o redundancia relevante. En Fix, en cambio, la representación directa sigue funcionando mejor en las salidas principales. Eso sugiere que, una vez que el problema se reduce a un subconjunto más controlado, la ventaja de proyectar a un espacio latente deja de ser evidente y puede ceder frente a una representación más simple.

Por familia de modelos tampoco aparece una dominancia única. XGBoost sostiene la mejor configuración global dentro de Fix. HGBR se comporta bien en All sin lag y también aparece como alternativa competitiva en varios bloques. Ridge sigue siendo una opción válida cuando se combina con PLS, especialmente en Ext. Esto refuerza la idea de que el problema no se deja resumir en la superioridad universal de una sola familia. Lo que manda, otra vez, es el acoplamiento entre espacio de variables, representación temporal y régimen de entrenamiento.

Mirado en conjunto, entonces, el capítulo deja una señal relativamente clara. La mejor salida global queda en Fix, pero eso no invalida el aporte de All ni de Ext. All muestra que, en un espacio amplio, todavía se puede obtener una solución fuerte sin incorporar lag explícito. Ext muestra que una solución intermedia, con PLS y lag, también puede sostener desempeño competitivo. Fix, por su parte, concentra la mejor combinación final entre reducción del espacio, generalización y capacidad predictiva. La interpretación más defendible no es que uno de los tres espacios "gane" de forma abstracta, sino que cada uno revela algo distinto sobre cómo representar el problema y sobre qué tipo de señal conviene preservar.

Bajo esta lectura, la Sección 5 deja una conclusión metodológica bastante directa. El desempeño observado depende menos de elegir un componente supuestamente superior en abstracto y más de cómo se combinan datos, representación, espacio de variables y modelo dentro de un mismo contrato de evaluación. Esa es, probablemente, la principal idea que conviene arrastrar a la discusión: no hay una receta única, pero sí hay una estructura de resultados lo bastante consistente como para sostener decisiones más precisas sobre qué configuraciones conviene priorizar en este problema.

6.2. Discusión por modo de datos

Los resultados de la Sección 5 muestran que una parte importante del comportamiento observado no depende solo de la familia de modelo o del espacio de variables, sino también del modo de datos con que se entrena cada configuración. En esta memoria se trabajó con tres regímenes principales: entrenamiento con datos de laboratorio (*lab_only*), entrenamiento con datos del sensor físico (*sensor_only*) y entrenamiento mixto (*mixed*), donde ambas fuentes se incorporan bajo el mismo contrato de evaluación temporal. Por lo mismo, la comparación entre modos no conviene leerla solo como una competencia por el menor RMSE. También conviene leerla como una comparación entre distintas formas de anclar el problema, distintas densidades de información y distintos compromisos entre trazabilidad y continuidad temporal.

6.2.1. Entrenamiento con datos de laboratorio (*lab_only*)

El régimen *lab_only* sigue siendo la referencia metodológica más limpia del estudio, porque se apoya exclusivamente en la fuente que esta memoria adopta como verdad de evaluación. Esa condición no garantiza por sí sola el mejor resultado global, pero sí entrega una base especialmente útil para juzgar qué tan bien una configuración es capaz de aproximar la señal de laboratorio sin mezclar fuentes de target durante el entrenamiento.

Bajo la síntesis final de la Sección 5, el mejor resultado de este régimen queda en el espacio All, sin lag, con representación PLS y HGBR. Ese resultado no solo deja a *lab_only* en una posición competitiva, sino que además muestra algo importante sobre el problema: cuando el entrenamiento se restringe a la fuente más trazable, todavía es posible obtener una generalización fuerte sin necesidad de que la mejor salida pase por el espacio más reducido ni por una variante con lag.

Esa lectura conviene hacerla con cuidado. Que *lab_only* funcione bien no significa que el problema quede resuelto de manera operacional. Su principal fortaleza es la consistencia con el target de referencia, pero esa misma decisión mantiene la menor densidad temporal entre los tres regímenes. En otras palabras, *lab_only* ayuda a responder bien la pregunta de si existe señal predictiva defendible respecto de laboratorio, pero no agota por sí solo la necesidad práctica de avanzar hacia esquemas con mayor continuidad de información.

También hay una lectura adicional que vale la pena retener. El hecho de que la mejor salida de *lab_only* aparezca en All y sin lag refuerza que el comportamiento del modo no puede separarse del espacio de variables. En este caso, el régimen de laboratorio parece beneficiarse de una representación más amplia y comprimida mediante PLS, sin que el agregado explícito de retardos entregue la mejor configuración final.

6.2.2. Entrenamiento con datos del sensor físico (*sensor_only*)

El régimen *sensor_only* es probablemente el más exigente de interpretar, porque concentra la principal promesa operativa del estudio, pero también la fuente con mayores problemas de calidad efectiva. En esta memoria, el sensor no se descarta, pero tampoco se trata como una señal equivalente al laboratorio. Su uso exige asumir más riesgo de ruido, discontinuidad y sesgo, incluso después de la limpieza aplicada en la construcción del dataset.

Bajo la síntesis transversal actual, la mejor salida de *sensor_only* ya no queda asociada al subconjunto ampliado ni a la lectura heredada del informe anterior. La mejor configuración retenida aparece en Fix, con lag, PLS y HGBR. Eso cambia la interpretación de este modo de manera importante. Lo que muestra el capítulo no es que el sensor gane cuando se amplía el espacio de variables, sino más bien que, cuando la fuente de entrenamiento es más frágil, conviene trabajar sobre un espacio más controlado y con una estructura temporal que ayude a estabilizar la señal útil.

Aun así, *sensor_only* no alcanza la mejor salida global del capítulo. Eso era esperable. Entrenar solo sobre la señal del sensor obliga al modelo a absorber una fuente más densa, pero también más expuesta a imperfecciones de medición y a problemas de continuidad. Por lo mismo, el valor de este régimen no está en demostrar superioridad absoluta sobre laboratorio o sobre el esquema mixto, sino en mostrar que incluso bajo esas restricciones la información de sensor no queda anulada y puede sostener resultados competitivos cuando se combina con decisiones de representación razonables.

En términos prácticos, este modo deja una conclusión útil. El sensor, por sí solo, no reemplaza la función metodológica del laboratorio como referencia principal de verdad. Pero tampoco aparece como una fuente estéril o irrelevante. Más bien, se comporta como una señal que todavía contiene aporte predictivo, aunque más sensible a cómo se controla el espacio de entrada y a cómo se incorpora la dimensión temporal.

6.2.3. Entrenamiento mixto (*mixed*)

El régimen *mixed* es el que mejor representa la idea de aprovechar simultáneamente la trazabilidad del laboratorio y la mayor continuidad temporal del sensor. En teoría, ese balance era atractivo desde el inicio del estudio, pero no bastaba con asumir que mezclar ambas fuentes mejoraría automáticamente el desempeño. Había que ver si esa combinación realmente entregaba una salida más robusta bajo evaluación temporal estricta.

La síntesis final de la Sección 5 muestra que la mejor configuración global del capítulo queda precisamente en este régimen. El mejor resultado retenido aparece en Fix, con lag, representación directa y XGBoost. Esa salida no solo supera al resto en la comparación transversal final, sino que además le da al modo mixto un peso interpretativo especial dentro de la discusión: no se trata solo de un régimen competitivo, sino del que mejor

logra articular, dentro del paquete válido, continuidad de información, control del espacio de entrada y generalización final.

Este resultado también ayuda a poner límites a una lectura simplista. El mejor desempeño de *mixed* no prueba que combinar fuentes siempre vaya a ser mejor que trabajar con una sola. Lo que muestra es algo más acotado y más útil: en este problema, cuando la mezcla se hace dentro de un espacio reducido y con una configuración que aprovecha bien la estructura disponible, sí puede aparecer una ventaja real frente a los otros regímenes.

Otro punto importante es que la mejor salida de *mixed* queda en representación directa y no en PLS. Eso conversa bien con la lectura global del espacio Fix. Una vez que el problema se reduce a un subconjunto más controlado, la mezcla de fuentes parece beneficiarse más de una representación directa y de una familia flexible como XGBoost que de una compresión adicional del espacio.

6.2.4. Síntesis de la comparación entre modos

Si se comparan los tres modos en conjunto, la lectura que queda no es la de un ganador universal desligado del resto de decisiones del capítulo. *lab_only* sigue siendo la referencia más limpia desde el punto de vista metodológico y conserva una salida muy fuerte en All. *sensor_only* no alcanza el mejor desempeño absoluto, pero muestra que la señal del sensor no debe descartarse, sobre todo cuando se trabaja con un espacio más controlado y con estructura temporal explícita. *mixed*, en tanto, es el régimen que termina entregando la mejor solución global del paquete válido.

La conclusión práctica es bastante directa. El laboratorio sigue siendo indispensable como ancla de verdad y como referencia de evaluación. El sensor, tomado por sí solo, todavía exige cautela. Pero la combinación de ambas fuentes, bien acoplada a un espacio de variables reducido y a una familia de modelo adecuada, es la que deja la mejor salida final del estudio. Bajo este resultado, la discusión por modo no empuja a reemplazar una fuente por otra, sino a entender mejor qué rol cumple cada una dentro de una estrategia predictiva más útil para el problema real.

6.3. Discusión por representación y espacio de variables

Una parte importante de los resultados observados en la Sección 5 no se explica solo por la familia de modelo o por el modo de entrenamiento. También depende de cómo se representa el problema y de cuánto espacio de variables se decide conservar. En esta memoria, esa discusión queda organizada en torno a tres espacios principales: All, que preserva el universo más amplio de variables; Fix, que trabaja con un subconjunto reducido y más controlado; y Ext, que funciona como una alternativa intermedia. A eso se suma, en parte de las corridas, el uso de una representación latente mediante PLS.

Mirado así, el problema no se reduce a escoger entre “más variables” o “menos variables”,

ni entre representación directa o representación proyectada. Lo que muestran los resultados es algo más condicionado: distintas combinaciones de espacio, representación y estructura temporal capturan ventajas distintas, y no todas esas ventajas aparecen al mismo tiempo en un solo bloque. Por eso, esta comparación conviene leerla como una discusión sobre compromisos entre retención de señal, complejidad, estabilidad y utilidad práctica, más que como una competencia abstracta entre experimentos.

6.3.1. Universo completo versus subconjuntos reducidos

La comparación entre All, Fix y Ext deja una señal clara, aunque no lineal. Trabajar con el universo completo no impide obtener las mejores configuraciones de todo el capítulo. De hecho, All sigue siendo competitivo y en laboratorio deja una de las salidas más fuertes del informe. Eso importa porque muestra que el universo amplio todavía contiene señal útil y que una estrategia de compresión adecuada puede aprovecharla sin necesidad de pasar primero por una reducción manual agresiva.

Al mismo tiempo, los resultados también muestran que reducir el espacio de entrada no equivale automáticamente a perder capacidad predictiva. Fix deja la mejor salida global del capítulo, lo que sugiere que una reducción más controlada del problema puede mejorar la generalización final cuando elimina redundancia y deja un espacio más manejable para el modelo. La lectura correcta no es que el subconjunto fijo sea el mejor en todo escenario, sino que una reducción bien orientada sí puede capturar una fracción importante de la señal sin arrastrar toda la complejidad de All.

Ext, por su parte, queda como una solución intermedia en el sentido más útil del término. No alcanza la mejor salida global, pero tampoco queda desplazado. Su comportamiento sugiere que entre conservar todo el universo y restringirse a un subconjunto pequeño existe un punto medio razonable, donde todavía se preserva parte de la señal adicional del espacio amplio sin volver por completo al costo de complejidad de All. En ese sentido, Ext no aparece como una simple concesión metodológica, sino como evidencia de que el tamaño del espacio retenido afecta de forma no trivial la capacidad del modelo para generalizar.

La conclusión que deja esta comparación es práctica. En este problema no conviene plantear el diseño del espacio de entrada como una oposición binaria entre usar todo o usar muy poco. Lo que conviene es entender que el tamaño del espacio interactúa con la representación y con la forma de incorporar la estructura temporal. Por eso, ni All queda validado como solución universal por conservar más información, ni Fix queda validado como receta general por dejar la mejor salida global.

6.3.2. Representación directa versus representación PLS

La comparación entre representación directa y PLS también deja una lectura más matizada que la del informe anterior. PLS sigue mostrando utilidad, pero esa utilidad no

aparece de manera uniforme en todos los espacios. En All y Ext, las mejores configuraciones retenidas quedan asociadas a PLS. Eso sugiere que, cuando el espacio de entrada todavía arrastra amplitud, colinealidad o redundancia, una compresión supervisada puede ayudar a ordenar mejor la señal disponible y sostener salidas competitivas.

En Fix ocurre algo distinto. Una vez reducido el problema a un subconjunto más controlado, la representación directa sigue funcionando mejor en la salida principal del capítulo. Ese resultado importa porque muestra que la ventaja de PLS no es automática. Cuando el espacio ya fue depurado de manera más agresiva, proyectarlo de nuevo a un espacio latente no necesariamente agrega valor y puede incluso dejar de ser la alternativa más conveniente frente a una representación más simple.

Esto ayuda a evitar dos sobrelecturas. La primera sería concluir que PLS mejora siempre el problema por el solo hecho de introducir una representación más elaborada. La segunda sería descartarlo solo porque no domina en Fix. Ninguna de las dos lecturas queda respaldada por la evidencia de esta memoria. Lo que sí queda respaldado es algo más acotado: PLS se vuelve especialmente útil cuando el espacio todavía es amplio o intermedio, mientras que en un espacio ya reducido y mejor controlado la representación directa puede seguir siendo suficiente e incluso superior.

6.3.3. Desempeño versus interpretabilidad y accionabilidad

La discusión por representación no se agota en el RMSE final. También importa qué tipo de lectura permite cada solución sobre el problema. All conserva más señal potencial, pero también deja una configuración más difícil de traducir a una lectura operacional directa, sobre todo cuando esa señal pasa además por una proyección PLS. Desde el punto de vista predictivo eso puede ser válido, pero desde el punto de vista de trazabilidad e interpretabilidad no siempre es la alternativa más cómoda.

Fix ofrece el contrapunto más claro. Aunque no necesariamente representa todo lo que ocurre en el proceso, sí deja una solución más compacta y metodológicamente más fácil de defender. Eso no convierte automáticamente al subconjunto fijo en la mejor opción para cualquier uso futuro, pero sí lo vuelve especialmente atractivo cuando interesa mantener una conexión más clara entre desempeño, parsimonia y posibilidad de inspección posterior.

Ext vuelve a quedar entre ambos extremos. No ofrece la compacidad de Fix ni la amplitud de All, pero precisamente por eso ayuda a mostrar que la discusión no tiene que resolverse solo entre máxima interpretabilidad y máxima retención de señal. También existen soluciones intermedias que, sin ser las mejores del capítulo, mantienen una relación razonable entre complejidad y capacidad predictiva.

Desde una perspectiva aplicada, esto importa bastante. En un problema como este no basta con preguntar qué configuración gana por una métrica. También importa cuánto cuesta sostener esa configuración, cuánto se entiende de la señal que preserva y qué tan defendible resulta si más adelante se quiere revisar, ajustar o auditar su comportamiento.

Bajo ese criterio, la comparación entre All, Fix y Ext no solo ordena resultados; también ordena tipos distintos de compromiso metodológico.

6.3.4. Síntesis interpretativa

La lectura que conviene retener de esta subsección es directa. Los resultados no respaldan la idea de que conservar más variables sea siempre mejor, pero tampoco respaldan que reducir agresivamente el espacio garantice por sí solo una mejor solución. Del mismo modo, PLS sí muestra utilidad en espacios amplios o intermedios, pero no aparece como mejora universal frente a la representación directa.

Bajo la evidencia disponible, All y Ext muestran que la compresión supervisada sigue siendo una herramienta útil cuando el espacio conserva bastante amplitud. Fix muestra que, una vez reducido el problema a un subconjunto más controlado, la representación directa todavía puede funcionar mejor y, en este caso, sostener la mejor salida global del estudio. En consecuencia, la discusión por representación y espacio no lleva a una receta única, sino a una conclusión más prudente: la forma de representar el problema debe decidirse en conjunto con el espacio de variables, el uso de lag y el modo de entrenamiento, porque esos componentes no actúan de manera independiente.

6.4. Implicancias metodológicas de la comparación entre test y holdout

La comparación entre test y holdout conviene leerla, antes que todo, dentro del contrato de evaluación efectivamente usado en esta memoria. En la Sección 5 todas las configuraciones se comparan sobre una misma partición temporal definida exclusivamente a partir de mediciones LAB elegibles. Bajo ese esquema, test corresponde al último tramo evaluable dentro de la zona pre-holdout, mientras que holdout queda reservado como bloque final fuera de muestra, separado por una guarda temporal explícita. Por lo mismo, la comparación entre ambos no debe entenderse como una duplicación de evidencia, sino como dos niveles distintos de exigencia dentro del mismo protocolo.

La primera implicancia de ese diseño es directa: holdout tiene más peso interpretativo que test. Test sigue siendo útil para filtrar configuraciones, revisar estabilidad relativa y detectar rápidamente qué combinaciones merecen ser retenidas en la discusión. Pero la afirmación final sobre generalización no descansa en test, sino en holdout, precisamente porque ese bloque queda más lejos del entrenamiento y representa el tramo temporal más exigente del esquema. En términos prácticos, esto obliga a evitar una lectura excesivamente optimista de configuraciones que se ven fuertes en test pero no sostienen esa ventaja al pasar al bloque final.

La segunda implicancia es que una brecha entre test y holdout no debe leerse de forma automática como contradicción metodológica. Ambos bloques están construidos sobre la

misma lógica general, pero no representan exactamente el mismo contexto temporal. Test cubre un tramo más largo y con mayor cantidad de mediciones LAB elegibles, mientras que holdout concentra un bloque final bastante más pequeño. Bajo esas condiciones, pequeñas variaciones de desempeño pueden amplificarse más fácilmente en holdout, no solo porque ese bloque está más lejos del entrenamiento, sino también porque su tamaño es menor y su lectura queda más expuesta a la composición puntual de ese tramo final.

Esto también ayuda a poner en contexto los casos en que una configuración no empeora en holdout, o incluso se comporta de forma comparable a test. Ese tipo de resultado no debería usarse como prueba fuerte de estabilidad estructural del problema, pero tampoco como algo sospechoso por sí mismo. Dentro de un diseño temporal como este, lo razonable es interpretarlo con prudencia: puede reflejar que la configuración logró generalizar de manera razonable al bloque final, pero no basta por sí solo para concluir que la dificultad de ambos bloques fue equivalente ni que el modelo quedó blindado frente a cambios de régimen.

Otro punto importante es que la referencia de evaluación se mantiene fija en laboratorio, aunque el tamaño y composición del entrenamiento cambien según el modo de datos. Esa decisión fortalece bastante la comparación transversal del capítulo. Permite contrastar configuraciones entrenadas en regímenes distintos bajo una misma vara de evaluación y evita que la discusión por modo se contamine con targets distintos al momento de medir desempeño final. A la vez, esa misma decisión obliga a recordar que igualdad de evaluación no significa igualdad de entrenamiento: las configuraciones se juzgan sobre el mismo test y el mismo holdout, pero no aprenden con el mismo volumen ni con la misma naturaleza de datos.

Desde esa perspectiva, la comparación entre test y holdout deja una lectura metodológica bastante concreta. Test sirve como instancia útil de contraste interno dentro del tramo pre-holdout, pero holdout sigue siendo el bloque que manda en la interpretación final. Las diferencias entre ambos deben leerse como parte normal de una evaluación temporal estricta, no como una anomalía a corregir con sobreexplicaciones. Lo importante, entonces, no es exigir una coincidencia artificial entre ambos splits, sino revisar si las configuraciones retenidas sostienen en holdout un desempeño suficientemente consistente como para justificar la lectura que se hace de ellas en la discusión.

En esta memoria, esa cautela es especialmente importante porque buena parte de las conclusiones dependen de contrastes relativamente finos entre espacios de variables, uso de lag, representación y modo de entrenamiento. Bajo un escenario así, sobreinterpretar diferencias pequeñas entre test y holdout puede llevar a conclusiones más tajantes de lo que el propio diseño permite sostener. La lectura más defendible es más simple: test ordena, holdout valida, y la evidencia final debe quedar anclada en ese último bloque sin perder de vista el tamaño acotado y la naturaleza temporal del split.

6.5. Contraste con preguntas de investigación e hipótesis

Esta subsección retoma las preguntas de investigación y las hipótesis formuladas en la Sección 3.7 a la luz de la evidencia reunida en la Sección 5 y de la discusión desarrollada en las subsecciones anteriores. El objetivo no es forzar una lectura concluyente donde la evidencia no alcanza, sino dejar explícito qué preguntas quedan razonablemente respondidas, cuáles solo reciben apoyo parcial y qué hipótesis deben leerse con mayor cautela dentro del contrato de evaluación efectivamente utilizado en esta memoria.

6.5.1. Respuesta a las preguntas de investigación

RQ1. ¿Con qué precisión puede estimarse Reducción(t) a partir de variables de proceso disponibles hasta el instante t? La evidencia permite responder esta pregunta de manera afirmativa, pero con una formulación prudente. A lo largo de la Sección 5 aparecen configuraciones que superan la referencia basal y que sostienen desempeño competitivo fuera de muestra bajo evaluación temporal estricta. En ese sentido, esta memoria sí muestra que es posible estimar la reducción a partir de variables de proceso disponibles hasta el instante t con una precisión útil dentro del marco comparativo utilizado. Al mismo tiempo, esa precisión no es uniforme ni independiente de las decisiones de diseño. Depende del modo de entrenamiento, del espacio de variables, del uso de lag y de la familia de modelo. Por eso, lo defendible aquí es afirmar viabilidad predictiva, no suficiencia operacional cerrada.

RQ2. ¿Existe evidencia de dependencia temporal-operacional útil entre variables del proceso y Reducción(t), considerando exclusivamente información previa a la medición de la reducción? La evidencia es consistente con una dependencia temporal-operacional útil entre variables del proceso y el objetivo. Esa lectura conversa bien con la formulación del problema como *nowcasting*, con la construcción de *features* basada en información disponible hasta t, y con el hecho de que en una parte importante de los contrastes la incorporación de estructura temporal aporta señal predictiva adicional. Sin embargo, esta respuesta no debe inflarse más de la cuenta. Lo que el capítulo sostiene es compatibilidad con precedencia temporal y utilidad predictiva, no una prueba causal fuerte en sentido físico-experimental. En otras palabras, la memoria muestra que existe señal explotable antes de la medición del *target*, pero no cierra de manera concluyente el mecanismo causal del proceso.

RQ3. ¿Mejora el desempeño predictivo al incorporar retardos (lags) en las variables de entrada? La respuesta es afirmativa, pero solo de forma parcial. La comparación entre variantes con y sin lag muestra que los retardos pueden aportar señal predictiva útil, y eso se nota con bastante claridad en Fix y Ext. Sin embargo, la mejora no es transversal a todo el capítulo. En All, la mejor salida final queda precisamente sin lag. Por eso, la conclusión correcta no es que el lag mejore siempre el problema, sino que

puede aportar de manera importante cuando se combina con ciertos espacios de variables y ciertas configuraciones. Bajo la evidencia disponible, el lag aparece como una herramienta útil, no como una mejora universal.

RQ4. Si se identifica un "mejor retardo" para una variable, ¿coincide dicho retardo con la ventana temporal real del proceso? Aquí la evidencia no alcanza para una afirmación fuerte. La memoria sí muestra que existen retardos empíricamente útiles y que la incorporación de estructura temporal puede mejorar el desempeño en parte relevante del espacio explorado. Pero eso no basta para sostener que el mejor lag predictivo coincida de forma exacta con el retardo físico real del proceso en sentido ingenieril estricto. La selección de retardos tuvo un componente práctico y exploratorio, orientado a construir representaciones plausibles y comparables, más que a validar una correspondencia física cerrada variable por variable. En consecuencia, esta pregunta queda respondida solo parcialmente: hay compatibilidad razonable entre la idea de desfase físico y el valor predictivo de ciertos lags, pero no evidencia suficiente para tratar esa coincidencia como demostrada.

RQ5. ¿Se puede construir un modelo que generalice adecuadamente bajo evaluación temporal y ante potenciales cambios de régimen operacional? La respuesta es afirmativa en el marco acotado del estudio. La existencia de configuraciones competitivas en *holdout* muestra que el problema no queda restringido a un ajuste local sobre entrenamiento o *test*, y que sí existe capacidad de generalización temporal fuera de muestra dentro del contrato utilizado. Sin embargo, también acá conviene ser precisos. Esa generalización debe leerse siempre en relación con el diseño de partición, el horizonte cubierto y el tamaño del bloque final. La memoria sí demuestra viabilidad de generalización temporal en el marco evaluado, pero no una estabilidad universal desligada del período observado ni de las condiciones específicas del estudio.

RQ6. ¿Mejora la predicción al utilizar el objetivo proveniente del sensor, en comparación con el objetivo de laboratorio, o su combinación? La comparación entre modos no muestra una superioridad universal del sensor puro frente al laboratorio, pero tampoco justifica descartarlo como fuente útil. El régimen *lab_only* sigue siendo la referencia más limpia desde el punto de vista metodológico y conserva una salida fuerte en la comparación general. *sensor_only*, por su parte, muestra que la señal del sensor puede sostener resultados competitivos en ciertos contextos, aun cuando no actúe como sustituto directo del laboratorio. El caso más interesante es *mixed*, porque justamente ahí queda la mejor salida global del capítulo. La respuesta, entonces, no es que el sensor supere al laboratorio en forma general, sino que la señal del sensor sí abre una vía metodológicamente relevante y que su mayor valor aparece cuando se la considera como complemento dentro de esquemas mixtos.

6.5.2. Balance de hipótesis

Las hipótesis formuladas en la Sección 3.7.2 pueden contrastarse ahora con una lectura más precisa. En todos los casos conviene evitar etiquetas demasiado tajantes, porque varias de las formulaciones ya fueron deliberadamente suavizadas para mantener coherencia con el contrato experimental y con la evidencia efectivamente obtenida. Por eso, la revisión que sigue distingue entre apoyo razonable, apoyo parcial y evidencia insuficiente.

H1. Es posible estimar la reducción con un error inferior a la variabilidad intrínseca del sensor físico mediante modelos no lineales que capturen la dinámica de combustión en la caldera. Esta hipótesis recibe apoyo parcial, pero no conviene darla por plenamente confirmada. La memoria sí muestra que existen configuraciones no lineales con desempeño competitivo fuera de muestra y que la estimación de la reducción es viable bajo el marco comparativo utilizado. Eso es consistente con el espíritu de la hipótesis. Sin embargo, la formulación original habla de un error inferior a la variabilidad intrínseca del sensor físico, y esa comparación no quedó establecida aquí como contraste directo, sistemático y explícito dentro del contrato de evaluación principal. Por eso, lo más defendible es decir que la evidencia apoya la viabilidad predictiva del enfoque no lineal, pero no alcanza para presentar H1 como cerradamente confirmada en todos sus términos.

H2. La incorporación de retardos temporales (lags) puede mejorar el desempeño predictivo en configuraciones donde la dinámica del proceso y el espacio de variables retenido sí logran capturar información temporal útil, en comparación con variantes equivalentes sin retardos. Esta hipótesis recibe apoyo parcial. El contraste específico entre variantes con y sin lag muestra que los retardos pueden aportar señal predictiva útil y que, en ciertos espacios de variables, esa mejora es relevante. Eso se observa con bastante claridad en Fix y, de forma más moderada, también en Ext. Sin embargo, la mejora no es transversal ni uniforme en todos los espacios evaluados. En All, por ejemplo, la mejor salida final queda sin lag. Por eso, la lectura correcta no es que el uso de retardos quede confirmado como mejora general, sino algo más acotado: su aporte depende de cómo se combinan la estructura temporal, el espacio de variables y la representación del problema.

H3. El desempeño predictivo depende del acoplamiento entre familia de modelo, espacio de variables y forma de representación. Esta hipótesis recibe apoyo razonable dentro del espacio experimental explorado. La evidencia de la Sección 5 no muestra una familia universalmente superior en todos los escenarios. En cambio, deja un patrón más condicionado. En el espacio Fix, la mejor salida global queda en representación directa con una familia basada en árboles. En All y Ext, en cambio, siguen apareciendo configuraciones competitivas cuando el problema se representa mediante PLS y se combina con familias más estables como Ridge o HGBR. Por eso, la lectura más defendible no es que exista un ganador único entre árboles y modelos lineales o penalizados, sino que

la conveniencia de cada familia depende de cómo se representa el problema y de cuánto espacio de variables se conserva. Bajo esa lógica, la evidencia sí respalda la hipótesis en términos generales, aunque sin convertir ese patrón en una regla absoluta desligada del contrato de evaluación y del conjunto de configuraciones efectivamente probado.

H4. Bajo configuraciones comparables de espacio de variables y representación, un modelo entrenado con datos mixtos puede mostrar una mayor robustez práctica fuera de muestra que un modelo entrenado exclusivamente con datos de laboratorio. Esta hipótesis recibe apoyo parcial. El principal argumento a su favor es que la mejor salida global del capítulo queda en el régimen *mixed*, lo que muestra que combinar laboratorio y sensor puede entregar una ventaja real dentro del paquete válido final. Sin embargo, esa evidencia no basta para convertir la hipótesis en una regla general sobre robustez práctica fuera de muestra. El régimen *lab_only* sigue siendo fuerte y metodológicamente más limpio, y la ventaja de *mixed* no aparece como una dominancia universal en todos los espacios y configuraciones. La formulación más prudente es que la mezcla de fuentes sí muestra potencial para mejorar la generalización y la robustez práctica en este problema, pero la evidencia actual todavía no permite sostener H4 como confirmada en términos absolutos.

En conjunto, el contraste con las RQ y las hipótesis deja una lectura bastante nítida. La memoria sí demuestra viabilidad predictiva y sí muestra que existe señal temporal útil en el problema. También deja evidencia consistente con que el uso de lag puede aportar, con que el sensor no debe descartarse y con que los esquemas mixtos merecen especial atención. Lo que no deja es una receta universal ni una validación cerrada de todas las formulaciones iniciales. Y eso, más que un problema, ayuda a dejar la discusión en un terreno más honesto y más útil para lo que realmente se logró demostrar.

6.6. Alcances, límites y proyecciones inmediatas

6.6.1. Alcances de lo que esta memoria sí permite sostener

A esta altura del trabajo, sí hay un conjunto de afirmaciones que se pueden sostener con respaldo razonable. La primera es que el problema de estimar la reducción a partir de variables de proceso sí admite soluciones predictivas competitivas bajo evaluación temporal fuera de muestra. La Sección 5 no deja un resultado aislado ni una mejora accidental, sino varias configuraciones que superan la referencia basal y que mantienen desempeño defendible bajo el mismo contrato de evaluación.

La segunda es que el problema no se deja resolver con una receta única. La evidencia muestra que el comportamiento depende de cómo se combinan el espacio de variables, la forma de representación, el uso de lag y el modo de entrenamiento. En ese marco, la mejor salida global del capítulo queda en Fix, con entrenamiento mixto, representación directa, lag y XGBoost. Esa combinación pasa a ser la referencia principal del estudio, pero no

como verdad universal, sino como la mejor solución retenida dentro del paquete válido evaluado.

La tercera es que el laboratorio sigue ocupando un rol metodológico central. No solo porque actúa como referencia principal de evaluación, sino también porque permite mantener una vara común para comparar configuraciones entrenadas bajo regímenes distintos. Esa decisión no elimina el valor del sensor. Lo que hace es ordenar el problema: el sensor aparece como fuente útil, pero su aporte se vuelve más defendible cuando se interpreta en relación con la referencia de laboratorio y no como sustituto directo de ella.

La cuarta es que la incorporación de estructura temporal sí merece mantenerse como parte del diseño del problema, aunque no como regla automática. Los resultados muestran que los retardos pueden aportar en una parte importante del espacio explorado, sobre todo en Fix y Ext. Eso basta para sostener que la dimensión temporal no fue un agregado decorativo, sino un componente real del comportamiento predictivo observado.

En conjunto, entonces, esta memoria sí permite sostener viabilidad predictiva, valor práctico de una evaluación temporal estricta, utilidad de explorar distintos espacios de variables y relevancia metodológica de trabajar con más de un régimen de entrenamiento. Lo que no permite sostener es que el problema haya quedado cerrado ni que la mejor configuración observada pueda trasladarse sin más a cualquier escenario operacional futuro.

6.6.2. Límites del estudio y del contrato de evaluación

Junto con esos alcances, también quedan límites bastante claros. El primero es que toda la lectura final depende del contrato de evaluación efectivamente utilizado. Las configuraciones se comparan bajo una misma partición temporal, con test y holdout definidos sobre mediciones LAB elegibles, y con un bloque final de holdout relativamente acotado. Ese diseño es metodológicamente defendible y conservador, pero al mismo tiempo impone un marco específico. Las conclusiones valen dentro de ese marco y no deben extrapolarse como si equivalieran a validación operacional ilimitada.

El segundo límite es la calidad efectiva de la señal del sensor. Aunque el trabajo de limpieza permitió rescatar un subconjunto más defendible para entrenamiento, el sensor sigue siendo una fuente con discontinuidades, episodios ruidosos y una confiabilidad menor que la del laboratorio como referencia. Por eso, incluso cuando aparece aportando señal útil, conviene mantener una lectura cuidadosa de su rol dentro del problema.

El tercer límite tiene que ver con la interpretación del lag. La memoria muestra que incorporar retardos puede ayudar, pero no entrega una validación cerrada de equivalencia entre el mejor lag predictivo y el retardo físico real del proceso. Esa distancia importa, porque evita sobredimensionar la lectura causal de una mejora que, en la práctica, sigue estando mediada por decisiones de representación, selección de variables y familia de modelo.

El cuarto límite es que la discusión se apoya en un conjunto acotado de espacios de

variables y familias de modelos, no en una exploración exhaustiva de todo el espacio metodológico posible. Eso no invalida los resultados, pero sí obliga a recordar que la mejor configuración retenida es la mejor dentro del paquete evaluado, no necesariamente la mejor concebible en abstracto. En el mismo sentido, el estudio muestra valor predictivo de árboles, ensambles y combinaciones con PLS, pero no agota otras rutas posibles de modelamiento.

Por último, también hay un límite práctico de interpretación. Esta memoria evalúa capacidad predictiva y compara configuraciones bajo un diseño temporal explícito, pero no constituye por sí sola una validación de despliegue industrial cerrado. No resuelve, por ejemplo, todo lo relativo a monitoreo en línea, recalibración continua, umbrales de alarma, integración con operación o mantenimiento futuro del sistema. Esas capas quedan fuera del alcance principal del estudio y no conviene insinuar que ya quedaron resueltas.

6.6.3. Proyecciones inmediatas y líneas lógicas de continuación

Las líneas de continuación más razonables salen bastante directo de los hallazgos ya obtenidos. La primera es mejorar la calidad efectiva del sensor físico y revisar con más detalle su rol dentro del dataset. No porque el trabajo dependa de rescatarlo a toda costa, sino porque una parte importante del valor del régimen mixto parece venir justamente de combinar continuidad temporal con una referencia más trazable. Si la señal del sensor mejora en consistencia y auditabilidad, ese espacio probablemente se vuelve todavía más interesante.

La segunda línea es revisar representaciones temporales más ricas que un esquema basado solo en lags puntuales. Los resultados ya muestran que la dimensión temporal importa, pero también que su efecto depende del espacio de variables. Eso sugiere que todavía hay margen para explorar agregaciones causales, ventanas móviles más informativas o representaciones temporales diferenciadas por tipo de señal, sin romper la trazabilidad del problema.

La tercera es abandonar la idea implícita de que todo el sistema de variables debería responder al mismo desfase. Una continuación natural es revisar retardos por fuente, por grupo de variables o por mecanismo operacional, en lugar de tratar el tiempo como un bloque uniforme. Esa línea conversa mejor con la física del proceso y podría ayudar a capturar relaciones temporales más específicas que las vistas en esta memoria.

La cuarta es extender el problema hacia forecasting de corto plazo. La formulación base del estudio se trabajó como nowcasting, pero desde el propio planteamiento metodológico ya quedaba abierta la posibilidad de derivar hacia horizontes $t+H$. Dado que una parte del capítulo muestra que la estructura temporal sí contiene señal útil, avanzar hacia predicción de corto plazo aparece como una continuación lógica y no como un salto artificial.

También queda abierta una línea más aplicada: revisar con más detalle el equilibrio entre desempeño, interpretabilidad y mantenibilidad del modelo. La mejor salida global

del capítulo es importante, pero no necesariamente agota la discusión sobre qué solución sería más conveniente en un entorno de uso sostenido. Ahí todavía puede haber valor en comparar configuraciones no solo por RMSE final, sino por robustez práctica, facilidad de auditoría y costo de mantenimiento metodológico.

6.6.4. Síntesis de cierre

La lectura final de esta subsección es simple. Esta memoria sí deja evidencia suficiente para sostener que el problema vale la pena y que existe una ruta metodológica defendible para abordarlo con modelos predictivos. También deja bastante claro que las mejores decisiones no aparecen de forma aislada, sino en la interacción entre espacio de variables, estructura temporal, representación y régimen de entrenamiento.

Al mismo tiempo, el estudio también deja límites nítidos. No hay una configuración que gane en todo escenario, el sensor no reemplaza al laboratorio como referencia principal y la mejor solución observada no debe venderse como despliegue ya resuelto. Lo que sí queda es una base bastante más clara para decidir por dónde seguir, qué componentes merecen priorizarse y qué preguntas conviene refinar en una etapa posterior.

6.6.5. Síntesis del aporte

Sin sobreprometer el alcance de los resultados, esta memoria sí deja algunos aportes que vale la pena explicitar. El primero es que muestra que existe señal predictiva útil entre variables historizadas de la caldera recuperadora y la reducción observada aguas abajo, incluso cuando una parte importante de esas variables corresponde a señales indirectas respecto del punto final de medición. En ese sentido, el trabajo no resuelve por completo la relación entre proceso y reducción, pero sí aporta evidencia de que esa relación puede explotarse de manera metodológicamente defendible bajo evaluación temporal estricta.

El segundo aporte es metodológico. Los resultados muestran que la representación del problema no es un detalle menor, sino una decisión que cambia de forma real el comportamiento del modelo. En particular, cuando el espacio de variables se mantiene amplio o intermedio, la representación mediante PLS sigue siendo capaz de sostener soluciones competitivas fuera de muestra. Esto no la convierte en una receta universal, pero sí muestra que cumple un rol relevante para ordenar señal, redundancia y estabilidad en espacios de entrada más grandes.

El tercer aporte es práctico. La mejor solución global retenida no solo resulta competitiva, sino que aparece en un subconjunto reducido de variables. Eso es importante por dos razones. Por una parte, muestra que una reducción controlada del espacio de entrada puede mantener o incluso mejorar el desempeño dentro del paquete evaluado. Por otra, deja una base mucho más favorable para interpretación posterior, auditoría y análisis de importancia de variables, abriendo una ruta concreta para estudiar con mayor detalle

qué señales están empujando la predicción y cómo podrían relacionarse con decisiones operacionales o con estudios posteriores del proceso.

Un cuarto aporte, más acotado, es que bajo la definición metodológica adoptada en esta memoria una fuente de menor trazabilidad efectiva, como el sensor físico, igualmente puede aportar valor cuando se integra de forma controlada. El estudio no respalda tratar al sensor como reemplazo directo del laboratorio, pero sí muestra que su incorporación dentro de esquemas mixtos puede mejorar el desempeño predictivo en este problema. En conjunto, estos resultados no cierran el problema, pero sí dejan una base más clara para continuar la investigación y para orientar desarrollos aplicados de mayor alcance.

6.7. Cierre de la discusión

La discusión de resultados deja una conclusión general que atraviesa todo el capítulo. El problema de estimar la reducción a partir de variables de proceso sí admite soluciones predictivas útiles, pero su comportamiento depende de decisiones de diseño que interactúan entre sí y que no pueden evaluarse por separado de manera simplista. La comparación entre espacios de variables, uso de lag, representación y modo de entrenamiento no lleva a una receta universal, pero sí ordena el problema de una forma bastante más clara que al inicio del estudio.

Dentro de esa lectura, la mejor solución global retenida aparece en el espacio Fix, con entrenamiento mixto, representación directa y lag. All sigue mostrando que un espacio amplio puede sostener resultados fuertes cuando se combina con PLS y sin lag. Ext, por su parte, confirma que también existen rutas intermedias competitivas. Esa diversidad de salidas no debilita la discusión. Al contrario, ayuda a entender que el problema es sensible a cómo se representa la señal y a qué tipo de estructura se preserva en cada caso.

En términos más prácticos, el capítulo permite cerrar dos ideas importantes. La primera es que el laboratorio sigue siendo indispensable como ancla metodológica y como referencia de evaluación. La segunda es que el sensor, aun con sus limitaciones, no debe descartarse, porque su aporte aparece con más claridad cuando se integra de forma controlada dentro de esquemas mixtos. Bajo esa combinación, el estudio no solo mejora la lectura del problema, sino que también deja una base más realista para sus continuaciones.

Con eso, la Sección 6 cumple su propósito principal: no repetir resultados, sino interpretar su significado, ordenar sus límites y dejar claro qué tipo de afirmaciones sí quedan respaldadas por la evidencia reunida en esta memoria.

7. Propuesta de arquitectura de software

A partir de la discusión desarrollada en la Sección 6.7, resulta razonable traducir los hallazgos experimentales a una propuesta simple de arquitectura de software orientada a desplegar un *soft sensor* de reducción en un entorno industrial. El propósito de esta sección no es presentar una plataforma definitiva ni una implementación cerrada en todos sus detalles, sino proponer una arquitectura de alto nivel que sea coherente con el problema estudiado, con las restricciones operacionales del caso y con la necesidad de mantener trazabilidad entre datos de proceso, inferencia y visualización. En ese sentido, la arquitectura debe entenderse como una propuesta de integración técnica razonable y no como la materialización de un modelo universalmente superior.

Desde el punto de vista funcional, la arquitectura propuesta busca habilitar dos usos compatibles con el trabajo desarrollado en esta memoria: *nowcasting* de la reducción en el instante actual y, de manera natural, *forecasting* de corto plazo en horizontes del orden de hasta una hora, siempre que la representación temporal del modelo y la disponibilidad de variables lo permitan. Esta dualidad no exige una arquitectura completamente distinta, sino una separación clara entre adquisición de datos, preparación de entradas, ejecución del modelo y publicación del resultado inferido.

7.1. Vista general de la arquitectura

La propuesta considera un flujo simple de cinco bloques. En primer lugar, las variables de proceso se obtienen desde el ecosistema PI mediante **PI Web API**, manteniendo el mismo principio de acceso historizado ya utilizado durante la construcción del dataset. En segundo lugar, un módulo intermedio de preparación recibe esas variables, aplica las transformaciones necesarias para construir el vector de entrada del modelo y asegura consistencia temporal en la ventana de inferencia. En tercer lugar, un módulo de inferencia ejecuta el modelo predictivo y genera una estimación de reducción. En cuarto lugar, el valor estimado se publica nuevamente hacia la infraestructura PI mediante un mecanismo de escritura controlada. Finalmente, el resultado puede visualizarse en **PI Vision**, de manera similar a otras señales operacionales del entorno.

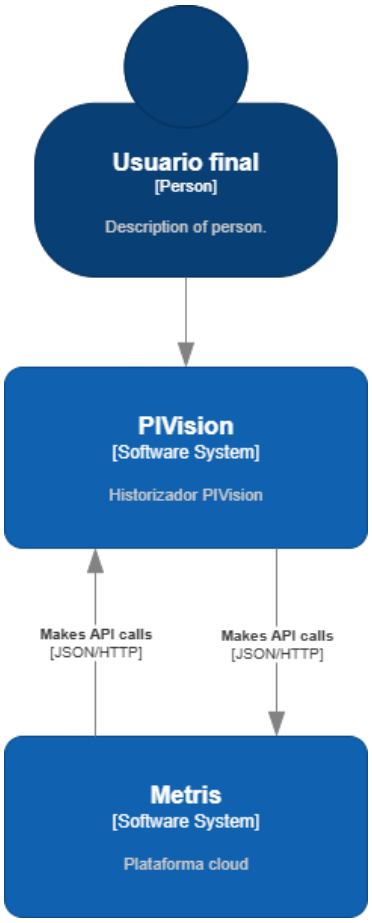

Figura 11: Vista general de la propuesta de arquitectura de software para integración del *soft sensor* con el ecosistema PI.

Esta organización tiene la ventaja de mantener desacopladas las responsabilidades principales del sistema. La adquisición de datos queda separada de la lógica del modelo; la inferencia queda separada de la visualización; y la publicación del resultado puede tratarse como una salida adicional del sistema, sin modificar la lógica de captura original. Desde la perspectiva de mantenibilidad, esta separación reduce acoplamiento y facilita reemplazar o actualizar el modelo sin rehacer toda la integración.

7.2. Componentes principales

El primer componente es la **capa de adquisición**, encargada de consultar variables desde PI Web API. Su función es recuperar las señales requeridas por el modelo en el horizonte temporal necesario para construir la entrada, incluyendo variables contemporáneas y, cuando corresponda, retardos o agregaciones temporales ya definidos por la representación elegida.

El segundo componente es la **capa de preparación e inferencia**. En esta capa se ejecuta la lógica necesaria para transformar las variables crudas en el espacio de entrada efectivo del modelo: alineamiento temporal, selección de columnas, construcción de *lags* o

resúmenes de ventana y aplicación del modelo entrenado. En términos prácticos, esta capa puede implementarse mediante una librería o servicio en Python, reutilizando la lógica ya desarrollada en el entorno experimental y apoyándose en el stack de trabajo utilizado en la memoria.

El tercer componente es la **capa de publicación del resultado**. Una vez generada la estimación, el valor inferido puede escribirse nuevamente al ecosistema PI para quedar disponible como una señal adicional dentro del entorno operacional. Esta decisión tiene una ventaja clara: evita crear una vía paralela de consumo y permite que la visualización se realice en la misma infraestructura donde los usuarios ya observan tendencias y variables de proceso.

El cuarto componente corresponde a la **capa de visualización**, resuelta aquí mediante PI Vision. Bajo este esquema, la salida del *soft sensor* puede observarse como una tendencia adicional, integrable con el resto de las señales de proceso sin exigir una interfaz nueva dedicada. Desde el punto de vista de adopción, esta decisión favorece simplicidad y continuidad operacional.

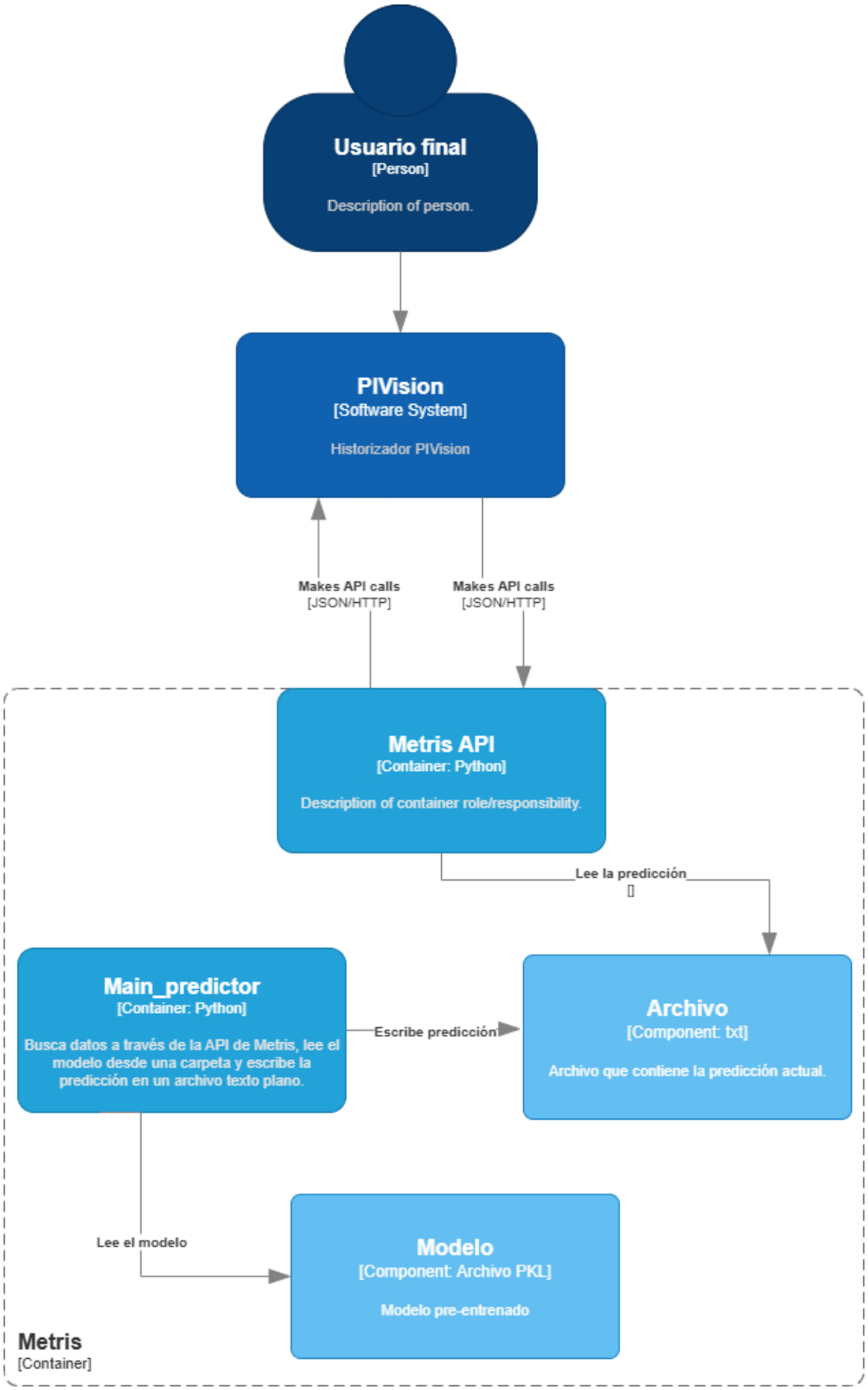

Figura 12: Esquema de componentes principales de la propuesta de arquitectura.

7.3. Consideraciones de diseño y operación

La arquitectura propuesta se apoya en un principio de simplicidad deliberada. Dado que el objetivo de esta memoria no fue desarrollar una plataforma de software extensa, sino

estudiar la viabilidad predictiva del problema y comparar configuraciones de modelado, la propuesta prioriza una integración mínima pero funcional con la infraestructura ya disponible. Esto resulta coherente con la discusión previa: el aporte principal del estudio no es la identificación de un único modelo indiscutible, sino la delimitación de configuraciones defendibles dentro del espacio explorado.

Desde el punto de vista operacional, la arquitectura también es consistente con la lectura metodológica del trabajo. El laboratorio sigue siendo la referencia principal para validación y contraste de desempeño, mientras que las señales del sensor y de proceso operan como insumo para la inferencia según el régimen de modelado adoptado. En consecuencia, la arquitectura no presupone que una única fuente reemplace por completo a otra, sino que debe ser capaz de convivir con la estructura de datos y con los criterios de trazabilidad que guiaron la memoria.

Otro aspecto relevante es que esta arquitectura puede sostener tanto *nowcasting* como *forecasting* de corto plazo. En el primer caso, la inferencia se interpreta como una estimación del estado actual de la reducción a partir de variables de proceso disponibles en el instante de consulta. En el segundo, el mismo flujo puede reutilizarse para horizontes cortos hacia adelante, siempre que el modelo haya sido entrenado con una representación temporal compatible con dicho objetivo. Desde el punto de vista de software, esta diferencia afecta principalmente la lógica del modelo y de preparación de entradas, más que la estructura general de integración.

Finalmente, aunque no se incorpora aquí como bloque central del diseño, una medida razonable para operación futura sería considerar **reentrenamiento periódico** o actualización programada del modelo. Esta recomendación surge de la naturaleza dinámica del proceso y de la propia discusión metodológica sobre estabilidad temporal, pero se plantea solo como práctica complementaria de mantenimiento y no como requisito indispensable para comprender la arquitectura propuesta.

7.4. Síntesis

En conjunto, la arquitectura planteada ofrece una forma simple y trazable de integrar un *soft sensor* de reducción al ecosistema tecnológico ya disponible en planta. Su valor no radica en complejidad tecnológica adicional, sino en traducir los hallazgos de esta memoria a un esquema de integración razonable, mantenible y compatible con la operación. Bajo esta lectura, la arquitectura constituye el cierre aplicado del trabajo: una propuesta concreta de cómo llevar los resultados experimentales hacia una implementación técnicamente coherente, sin presentar esa implementación como una solución definitiva ni cerrada a evolución futura.

8. Bibliografía

Referencias

Barnett, L., Barrett, A. B., & Seth, A. K. (2009). Granger causality and transfer entropy are equivalent for Gaussian variables. *Physical Review Letters*, *103*(23), 238701. https://doi.org/10.1103/PhysRevLett.103.238701

Bequette, B. W. (2003). *Process Control: Modeling, Design and Simulation*. Prentice Hall.

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. Springer.

Cover, T. M., & Thomas, J. A. (2006). *Elements of Information Theory* (2.ª ed.). Wiley-Interscience.

Greene, W. H. (2018). *Econometric Analysis* (8.ª ed.). Pearson.

Grinsztajn, L., Oyallon, E., & Varoquaux, G. (2022). Why do tree-based models still outperform deep learning on tabular data? *arXiv preprint arXiv:2207.08815*.

Guyon, I., & Elisseeff, A. (2003). An Introduction to Variable and Feature Selection. *Journal of Machine Learning Research*, *3*, 1157-1182.

Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction* (2.ª ed.). Springer.

Helm, J. M., Swiergosz, A. M., Haeberle, H. S., Karnuta, J. M., Schaffer, J. L., Krebs, V. E., Spitzer, A. I., & Ramkumar, P. N. (2020). Machine Learning and Artificial Intelligence: Definitions, Applications, and Future Directions. *Current Reviews in Musculoskeletal Medicine*, *13*, 69-76. https://doi.org/10.1007/s12178-020-09600-8

Hollmann, N., Müller, S., Eggensperger, K., & Hutter, F. (2023). A Transformer That Solves Small Tabular Classification Problems in a Second. *International Conference on Learning Representations*.

Hyndman, R. J., & Koehler, A. B. (2006). Another look at measures of forecast accuracy. *International Journal of Forecasting*, *22*(4), 679-688.

Kuncheva, L. I., Matthews, C. E., Arnaiz-González, Á., & Rodríguez, J. J. (2020, agosto). Feature Selection from High-Dimensional Data with Very Low Sample Size: A Cautionary Tale [arXiv:2008.12025v1].

López de Prado, M. (2018). *Advances in Financial Machine Learning*. John Wiley & Sons.

Mitchell, T. M. (1997). *Machine Learning*. McGraw-Hill.

Montgomery, D. C., & Runger, G. C. (2003). *Applied Statistics and Probability for Engineers* (3.ª ed.). John Wiley & Sons.

Murphy, K. P. (2012). *Machine Learning: A Probabilistic Perspective*. MIT Press.

Pearson, K. (1895). Note on regression and inheritance in the case of two parents. *Proceedings of the Royal Society of London*, *58*, 240-242. https://doi.org/10.1098/rspl.1895.0041

Runge, J., Nowack, P., Kretschmer, M., Flaxman, S., & Sejdinovic, D. (2019). Detecting causal associations in large nonlinear time series datasets. *Science Advances*, *5*(11), eaau4996. https://doi.org/10.1126/sciadv.aau4996

Russell, S. J., & Norvig, P. (2004). *Inteligencia artificial: Un enfoque moderno* (2.ª ed.). Pearson Educación.

Schreiber, T. (2000). Measuring information transfer. *Physical Review Letters*, *85*(2), 461-464. https://doi.org/10.1103/PhysRevLett.85.461

Shannon, C. E. (1948). A Mathematical Theory of Communication. *Bell System Technical Journal*, *27*(3-4), 379-423, 623-656. https://doi.org/10.1002/j.1538-7305.1948.tb01338.x

Smook, G. A. (2002). *Handbook for Pulp & Paper Technologists* (3.ª ed.). Angus Wilde Publications.

Valve & Automation. (n.d.). *Recovery Boiler* [Consultado el 31 de marzo de 2026]. https://www.valve.co.za/recovery-boiler/

Whitney, R. P. (Ed.). (1968). *Chemical Recovery in Alkaline Pulping Processes*. Technical Association of the Pulp; Paper Industry.

Zhang, T., Zhang, Z., Fan, Z., Luo, H., Liu, F., Liu, Q., Cao, W., & Li, J. (2023). OpenFE: Automated Feature Generation with Expert-level Performance. *Proceedings of the 40th International Conference on Machine Learning*, *202*.