

Informe de modelos – Guardian de Fraudes - Contugas

Nombre: Juan sebastian perdomo Becerra js.perdomo@unal.edu.co

1 Introducción

El proyecto Guardian de Fraudes, es una solución orientada a la detección temprana de consumos anómalos en clientes industriales de gas natural. Este caso surge como una propuesta aplicada para la empresa Contugas, una empresa del sector energético que opera en el mercado peruano, donde se identificó la necesidad de contar con herramientas más avanzadas para el monitoreo del comportamiento de consumo de sus clientes, especialmente en el segmento industrial, donde pequeñas desviaciones pueden representar grandes pérdidas económicas o incluso fraudes operativos.

Durante los procesos normales de lectura y validación del consumo, se evidencian comportamientos que no siempre son fáciles de explicar. Algunos pueden deberse a fallas técnicas, errores humanos o condiciones de operación atípicas. Sin embargo, también existen patrones que podrían estar asociados con manipulación del sistema o intervención no autorizada. En ese sentido, el propósito de este proyecto es diseñar e implementar un sistema que permita identificar dichas anomalías de manera más eficiente y oportuna, utilizando técnicas de analítica avanzada y modelos de machine learning, que puedan adaptarse a cada cliente y contexto. Para generar una herramienta predictiva que genere alertas para evitar fraudes o problemas operativos

Este desarrollo no parte desde cero, ya que en la primera fase se construyó un prototipo funcional del sistema, incluyendo una tabla de requerimientos y un primer diseño visual de cómo debería verse la herramienta en su versión productiva. En esta segunda etapa se busca validar si la propuesta metodológica efectivamente logra detectar patrones irregulares y si es escalable en un contexto real.

Adicionalmente, uno de los aportes más relevantes de esta solución es el enfoque híbrido que se planteó desde el inicio, en donde no se recurre únicamente a modelos estadísticos clásicos o reglas fijas, sino que se combinan modelos físicos con enfoques no supervisados y también modelos por cliente basados en series de tiempo. Esto permite construir una solución que no solo detecte anomalías, sino que también sea interpretable para el área técnica y suficientemente robusta para adaptarse a distintos tipos de clientes. La idea no es reemplazar el criterio de los analistas, sino ofrecerles una herramienta que priorice los casos más relevantes y facilite el seguimiento operativo.

2 Descripción de los datos

Para el proyecto Guardianes de Fraude se utilizó una base de datos histórica compartida que nos da el consumo de gas natural de clientes industriales anónimos pero que ejemplifican la operación real de Contugas. Esta base cuenta con registros por cliente y por día, lo cual permite construir análisis temporales y comparar comportamientos con base a patrones históricos.

	fecha	presion	temperatura	temperatura_k	volumen	indice_pvt_ideal	indice_pvt_real	indice_vt	indice_pv	cliente
0	2019-01-14 00:00:00	17.732563	28.209354	301.359354	20.969751	1.233900	1.371000	0.069584	0.845626	CLIENTE1
1	2019-01-14 01:00:00	17.747776	28.518614	301.668614	17.845739	1.049901	1.166557	0.059157	0.994511	CLIENTE1
2	2019-01-14 02:00:00	17.758916	28.230191	301.380191	20.975914	1.236012	1.373347	0.069600	0.846634	CLIENTE1
3	2019-01-14 03:00:00	17.727940	27.811509	300.961509	20.592299	1.212976	1.347751	0.068422	0.860901	CLIENTE1
4	2019-01-14 04:00:00	17.746484	27.795293	300.945293	21.690626	1.279077	1.421197	0.072075	0.818164	CLIENTE1

Dentro de las variables principales se encuentran la fecha, el nombre del cliente, el volumen consumido (m³), la presión, la temperatura y variables adicionales necesarias para calcular los índices físicos como el factor de compresibilidad Z o el índice PV/T.

Desde el inicio fue necesario realizar una limpieza y estandarización de los datos. Algunas columnas tenían valores nulos o inconsistentes que podían afectar el entrenamiento de los modelos, especialmente aquellos basados en series de tiempo. Por este motivo se aplicaron transformaciones como la conversión de unidades (por ejemplo, temperatura de °C a °K) y la creación de nuevas variables calculadas que permiten una mejor caracterización del consumo.

Una parte importante de la preparación de los datos fue la generación de índices físicos que sirvieron como línea base para evaluar desviaciones. Entre ellos se destaca:

Índice físico	Fórmula	Interpretación
PV/T	$P * V / T$	Relación esperada según ley de gases ideales
PV/(ZT)	$P * V / (Z * T)$	Ajuste del comportamiento con gases reales

Ambas ecuaciones se incluyeron en la fase exploratoria, ya que ofrecen una forma de representar físicamente si el consumo de gas en un cliente tiene sentido según las condiciones de operación.

La ecuación PV/T se basa en la ley de gases ideales, más simple y directa, mientras que PV/(ZT) incorpora el factor de compresibilidad Z, lo cual permite reflejar condiciones más realistas del gas en operación, especialmente cuando hay presiones altas o temperaturas

bajas. Por eso se decidió trabajar con ambas para luego comparar cuál tenía mejor comportamiento como referencia de normalidad. Este análisis se detalla más adelante.



Imagen 1. Porcentaje de registros sospechosos por cliente

Se observa qué porcentaje del total de registros de cada cliente fueron considerados sospechosos, lo cual ayuda a identificar posibles comportamientos irregulares a nivel agregado.

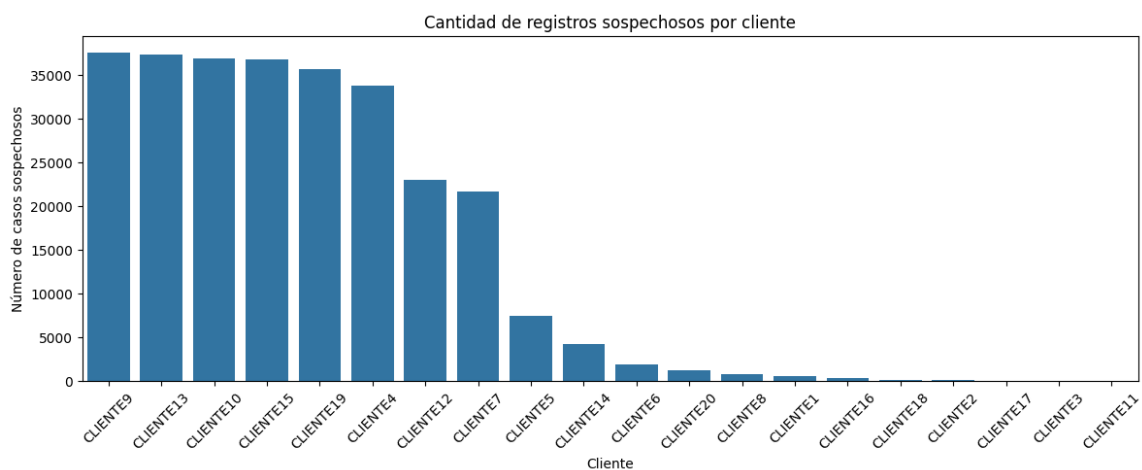


Imagen 2. Cantidad de registros sospechosos por cliente

Complementa la imagen anterior pero ahora se expresa en volumen absoluto de casos, lo cual ayuda a priorizar clientes con más eventos críticos detectados.

Cabe aclarar que los datos no cambian entre fases del proyecto, es decir, se trabaja con un mismo conjunto estático para todas las pruebas. Esto se decidió ya que no están disponibles más datos como tal, sino que es la única muestra que se nos comparte. En producción real esta lógica puede cambiar de acuerdo con la posibilidad de conectarnos a datos en tiempo real o con etiquetas para entrenar modelos de otro tipo. Por esta razón también se usa como línea base los modelos físicos, es decir, los relacionados a la ecuación de los gases tanto ideal como real.

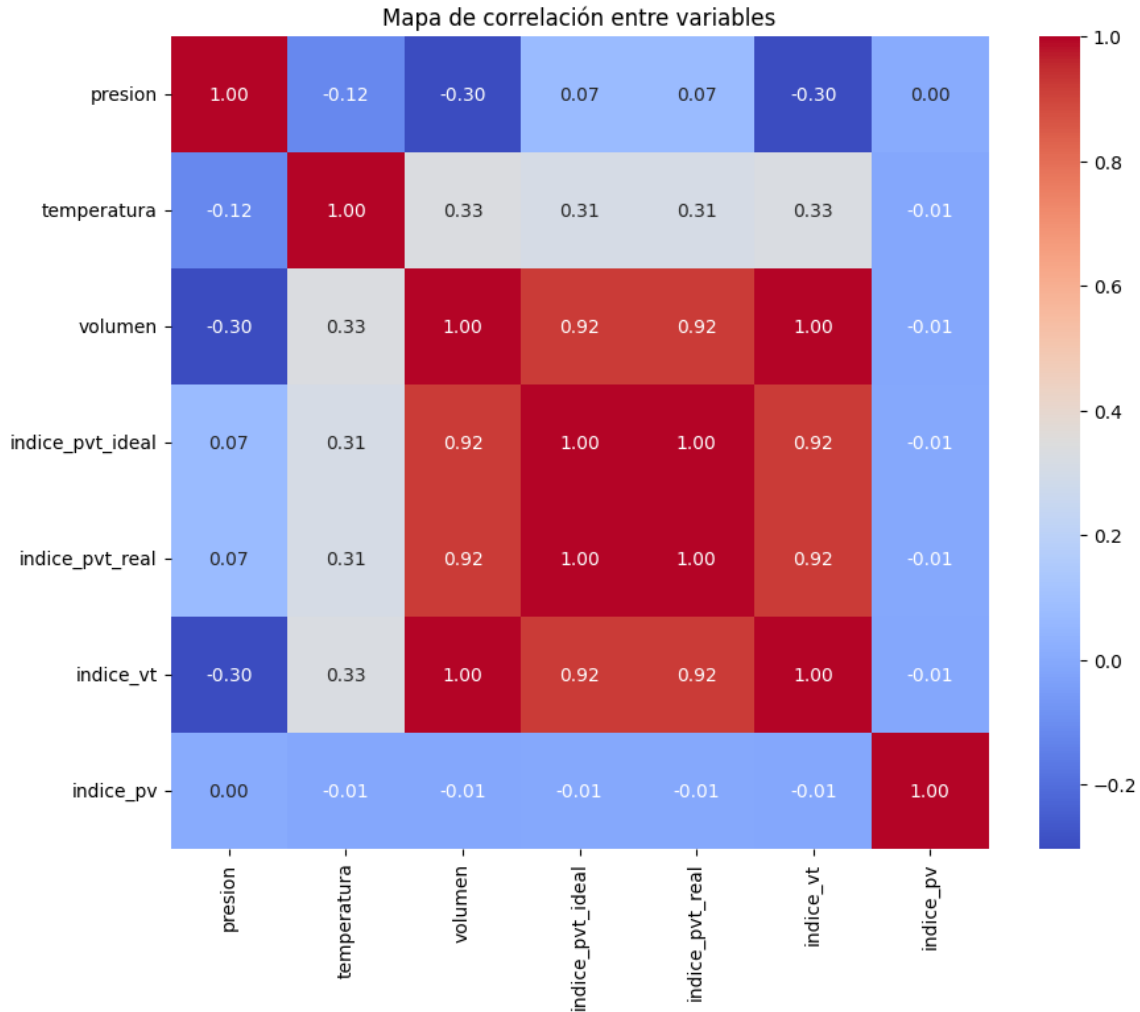


Imagen 3. Correlacion de variables

Esta gráfica muestra cómo se relacionan entre sí las principales variables físicas. Se observa una alta correlación entre presión y volumen, como era esperado, mientras que la temperatura muestra variabilidad más independiente.

3 Enfoque metodológico

Para abordar el problema de forma integral, se propuso dividir el análisis en dos ejes complementarios. Esta decisión permitió atacar el problema tanto desde una perspectiva de detección directa de anomalías sin necesidad de etiquetas, como desde un enfoque predictivo centrado en la evolución individual de cada cliente. De esta manera, se logró evaluar no solo qué registros eran atípicos, sino también entender si el comportamiento de un cliente estaba cambiando en relación con lo que se esperaba de él.

3.1 Predicción y análisis de comportamientos anómalos (eje1)

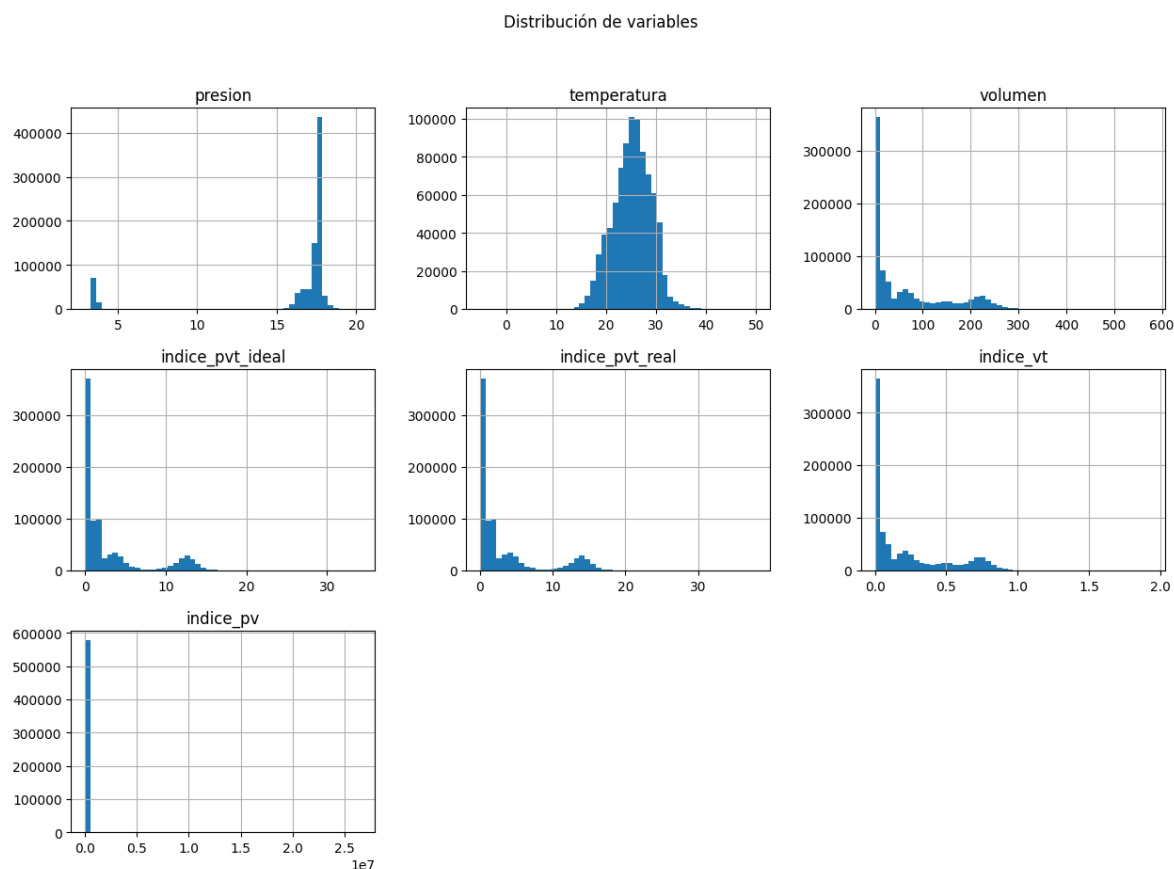
Este eje tiene como objetivo identificar registros que se desvían del comportamiento normal sin necesidad de conocer una etiqueta previa (anómalo o no anómalo). Se trabajó con tres tipos de modelos:

3.1.1 Modelos físicos (baseline)

Se implementaron dos fórmulas basadas en la ley de los gases:

- **Ley de gases ideales:** $PV/TPV/TPV/T$
- **Ley de gases reales:** $PV/(ZT)PV/(ZT)PV/(ZT)$

Ambas fórmulas buscan capturar si el comportamiento del consumo está dentro del rango físico esperado. En la fase inicial se probaron las dos, pero al comparar su comportamiento se evidenció que el índice **PV/(ZT)** ofrecía mejor estabilidad y menor dispersión, especialmente en clientes con presiones más altas. Esto se puede ver en la siguiente comparación:

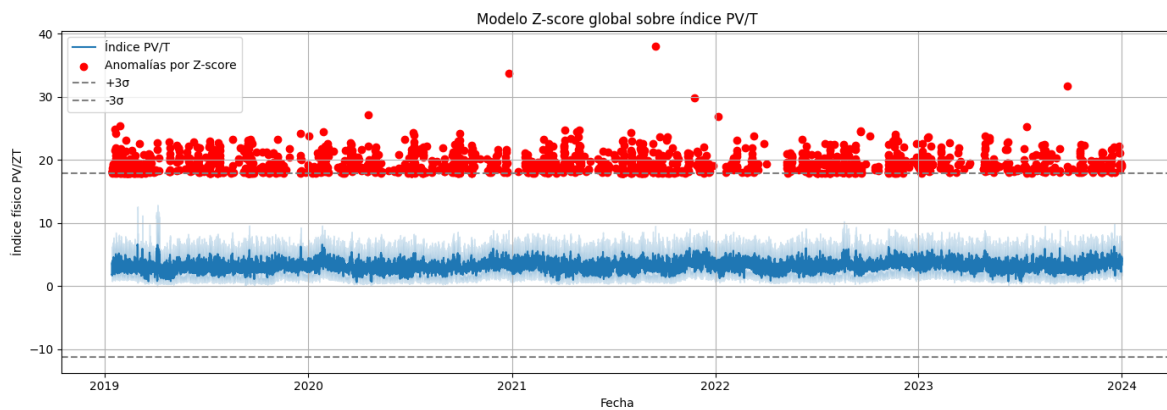
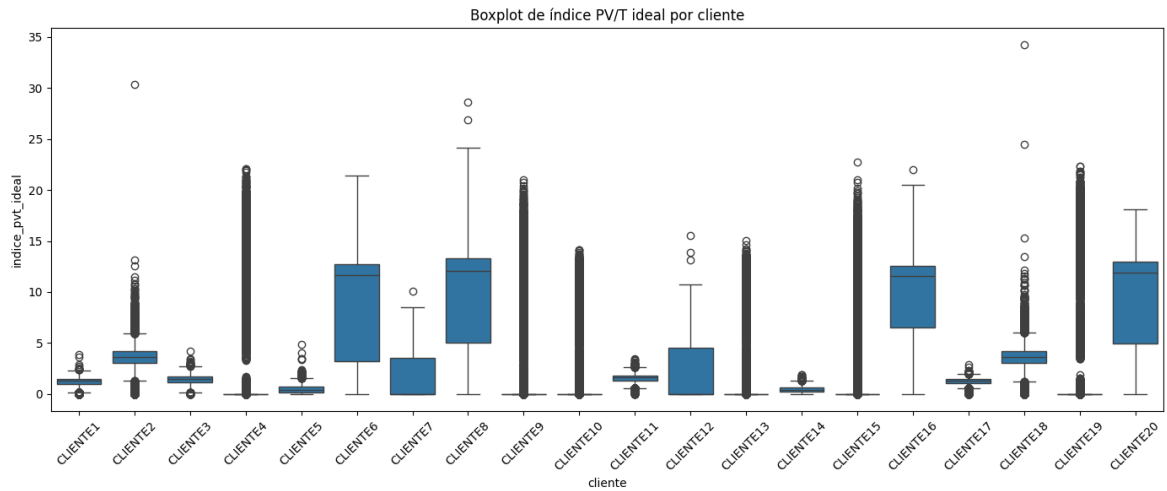


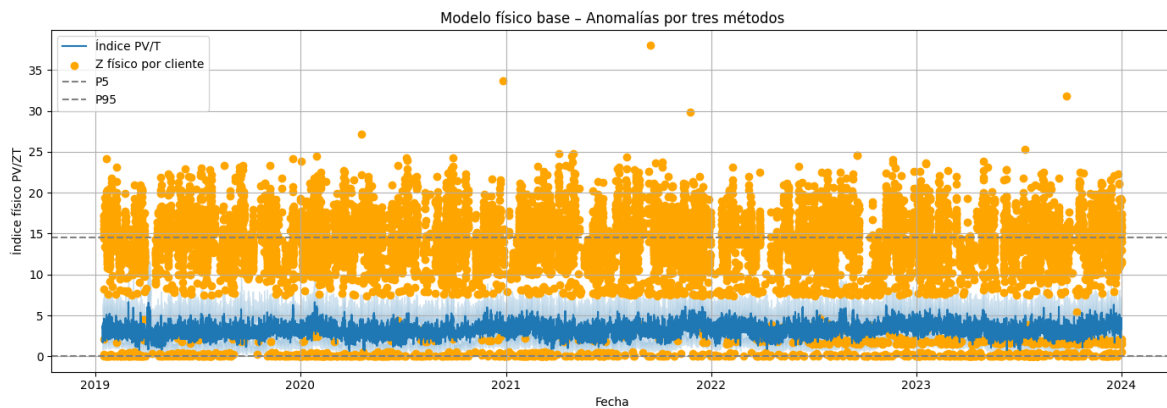
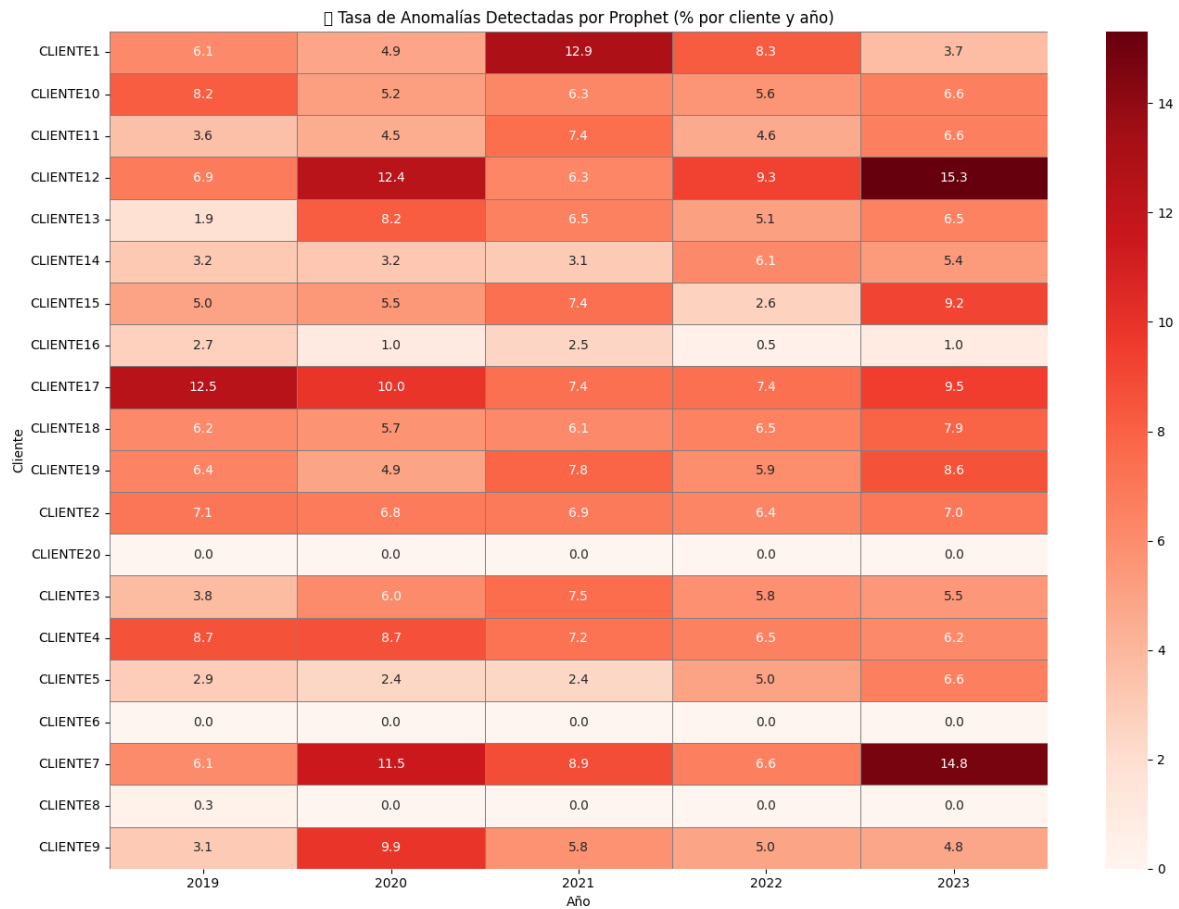
Con base a esto se decidió usar **PV/(ZT)** como índice físico principal. A partir de ese índice se construyeron varios métodos para marcar registros sospechosos:

- Cálculo del Z-score global
- Percentiles (p5 y p95)

- Derivada y Z-score de la derivada
- Z-score por cliente

Estos enfoques sirvieron como línea base para comparar con los modelos más sofisticados.

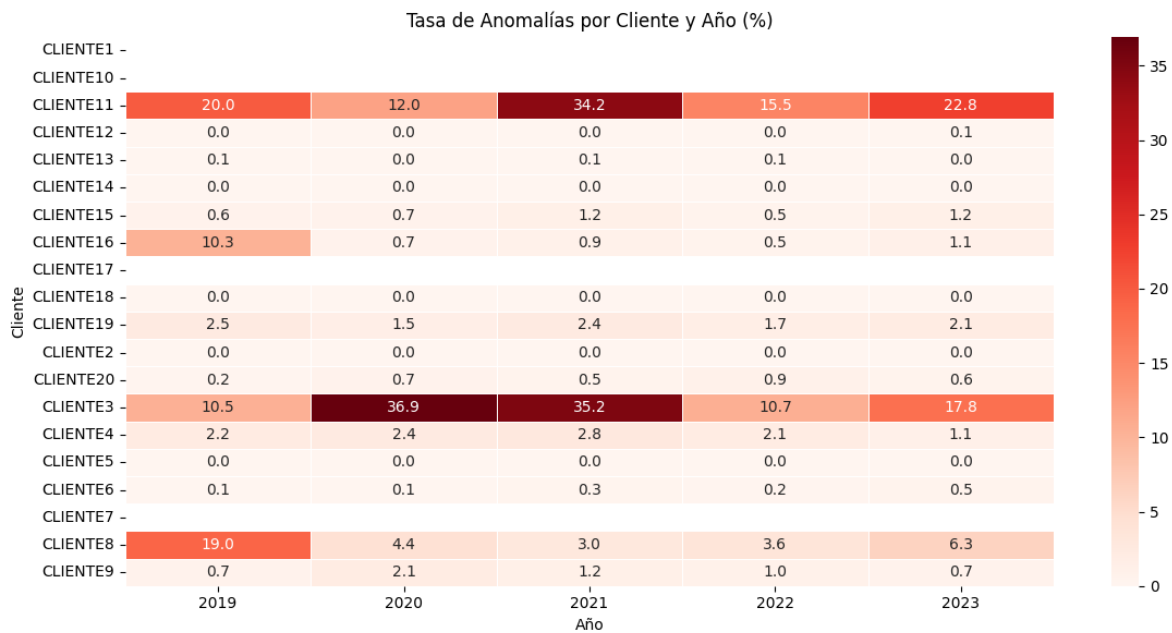
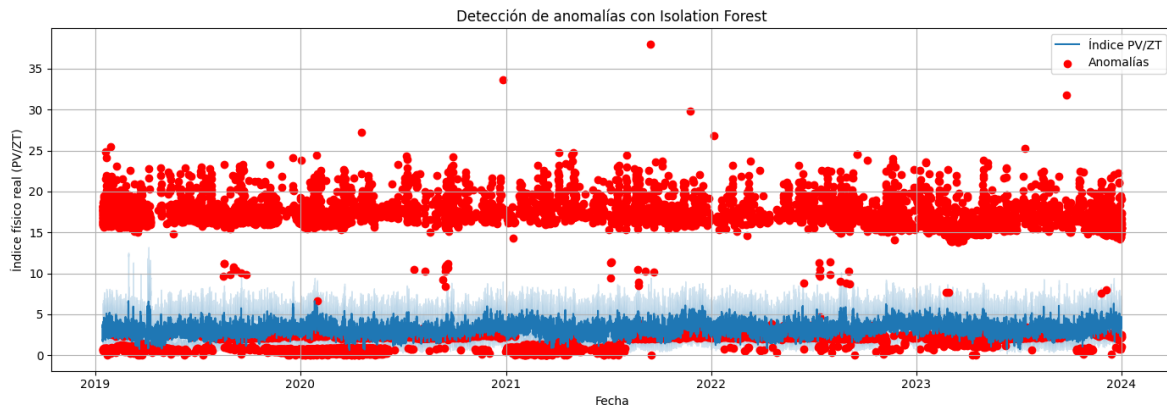




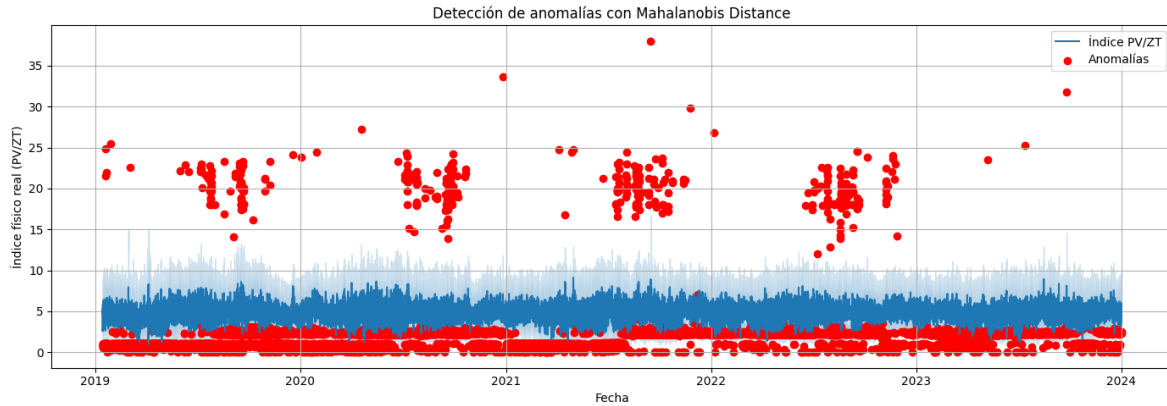
3.1.2 Modelos no supervisados

Se entrenaron los siguientes modelos sin etiquetas, cada uno con una lógica diferente que aporta valor al sistema:

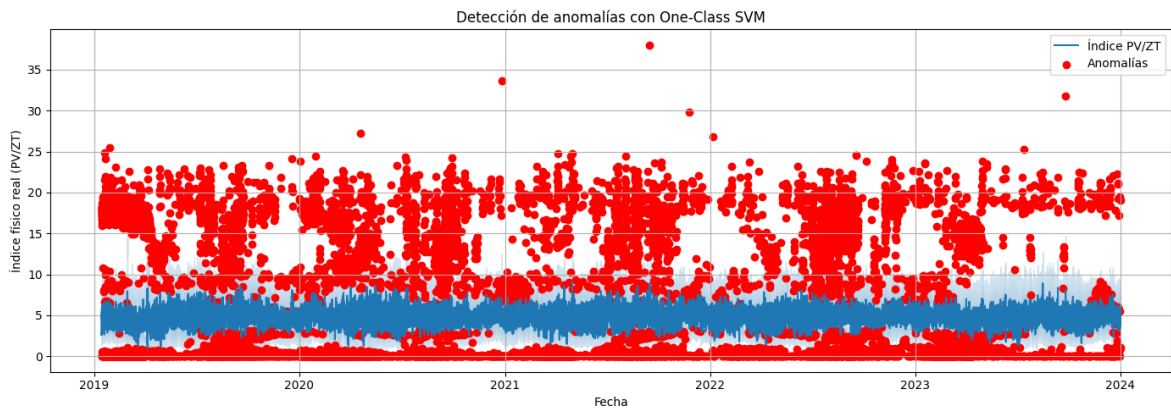
- **Isolation Forest:** Este modelo fue escogido por su capacidad para detectar valores atípicos en grandes volúmenes de datos sin requerir etiquetas. Funciona bien en espacios multivariados y tiene buena interpretabilidad operativa porque devuelve una puntuación de "rareza". Además, su desempeño fue consistente y con buena cobertura en las pruebas iniciales.



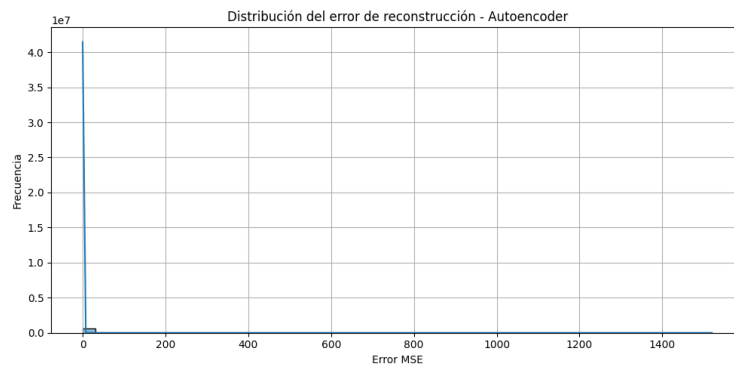
- Mahalanobis Distance:** Se incluyó por su capacidad para detectar observaciones fuera de una nube multivariada normal. Es especialmente útil cuando los datos siguen una estructura elíptica o tienen cierta correlación entre variables. Aunque es más simple, fue una buena referencia para comparar resultados con modelos más complejos.

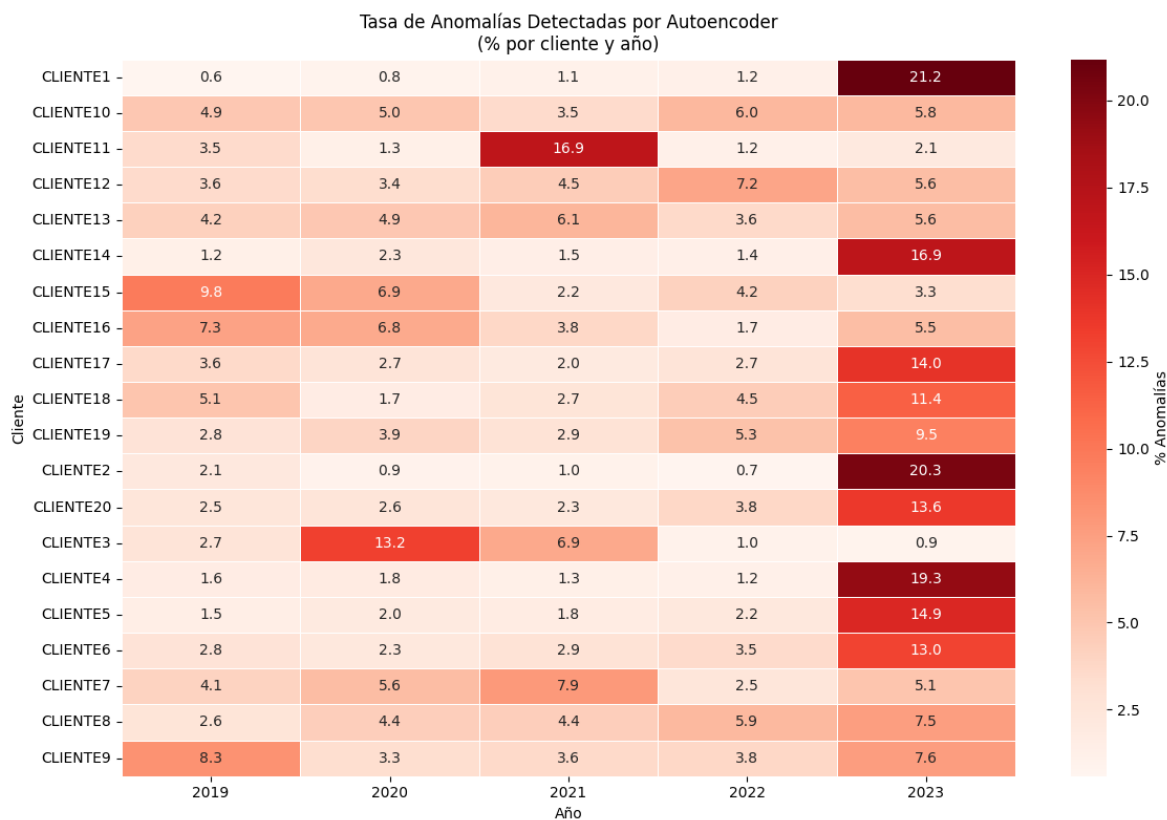


- **One-Class SVM:** Este modelo se utiliza cuando se quiere entrenar sobre un solo tipo de clase (normal) y luego identificar desviaciones. Se probó como alternativa para tener una lógica distinta basada en márgenes.



- **Autoencoder (general y por cliente):** Se seleccionó porque permite reconstruir patrones esperados y marcar como sospechosas las observaciones que no logran ser reconstruidas correctamente. Fue útil para validar comportamientos y tuvo buen desempeño cuando se entrenó de forma personalizada por cliente.





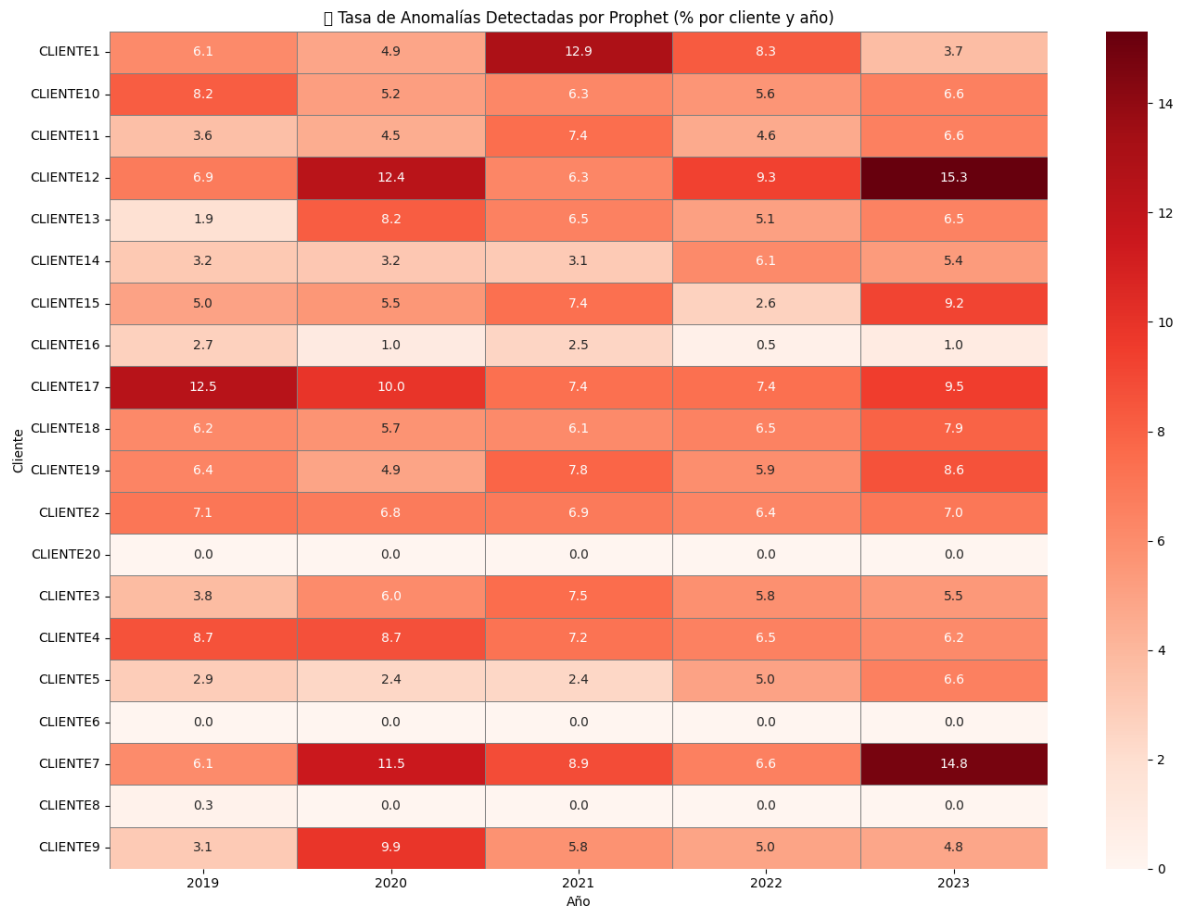
3.2 Predicción personalizada por cliente - modelos de series de tiempo (eje 2)

En este eje se entrenó un modelo por cliente con base en sus consumos históricos, para luego comparar el valor real con el valor predicho día a día. Si la diferencia era muy alta, se marcaba como posible anomalía. Este eje fue clave para generar alertas más personalizadas y adaptadas al comportamiento real de cada cliente.

Se utilizaron tres tipos de modelos:

- **Prophet**

Se eligió por ser un modelo robusto desarrollado por Meta, que permite capturar estacionalidades y tendencias de forma automática. Es fácil de implementar y de interpretar, y se adapta bien a series con comportamiento estable. Sirvió como modelo base para todos los clientes.



- **NeuralProphet**

Fue seleccionado como una mejora del Prophet tradicional, ya que combina componentes lineales con redes neuronales simples (Feedforward). Se comporta mejor en casos donde Prophet no logra captar ciertas fluctuaciones. Aunque su entrenamiento es más lento, se decidió incluirlo como modelo complementario.

- **LSTM**

univariado

Este modelo se escogió por su capacidad de aprender secuencias complejas y detectar desviaciones secuenciales. Aunque requiere más datos y ajuste, permite una detección más sensible en clientes con patrones volátiles. Fue usado como modelo de validación para verificar si los otros enfoques fallaban en capturar ciertos comportamientos.

- **SARIMA**

Este modelo fue incluido por su capacidad para capturar componentes estacionales y tendencias a través de un enfoque estadístico tradicional. Se ajusta bien cuando las series tienen patrones repetitivos, como ciclos semanales o mensuales, y fue útil para validar si un modelo más clásico podía detectar comportamientos anómalos con base en el índice físico $PV/(ZT)$. Aunque su escalabilidad es limitada, permitió establecer una comparación con modelos más modernos como Prophet.

4 Evaluación comparativa de modelos

Después de construir todos los modelos planteados en la fase metodológica, se realizó un proceso de evaluación donde se compararon los resultados obtenidos por cada uno frente a la referencia definida desde el modelo físico basado en el índice $PV/(ZT)$. Este modelo fue el punto de partida para marcar registros sospechosos, usando criterios como el Z-score, los percentiles y las desviaciones por cliente, y con eso se construyó la variable `sospechoso_binario`.

El objetivo en esta fase no era solo mirar si un modelo detecta o no anomalías, sino entender cuál es su comportamiento general: si marca a todos como sospechosos o solo a unos pocos, si se ajusta al comportamiento del cliente o si actúa como una regla general. Por eso se calcularon las métricas de precisión, recall, F1-score y el porcentaje de clientes donde el modelo marcó al menos un registro como sospechoso (que acá se llama clientes cubiertos).

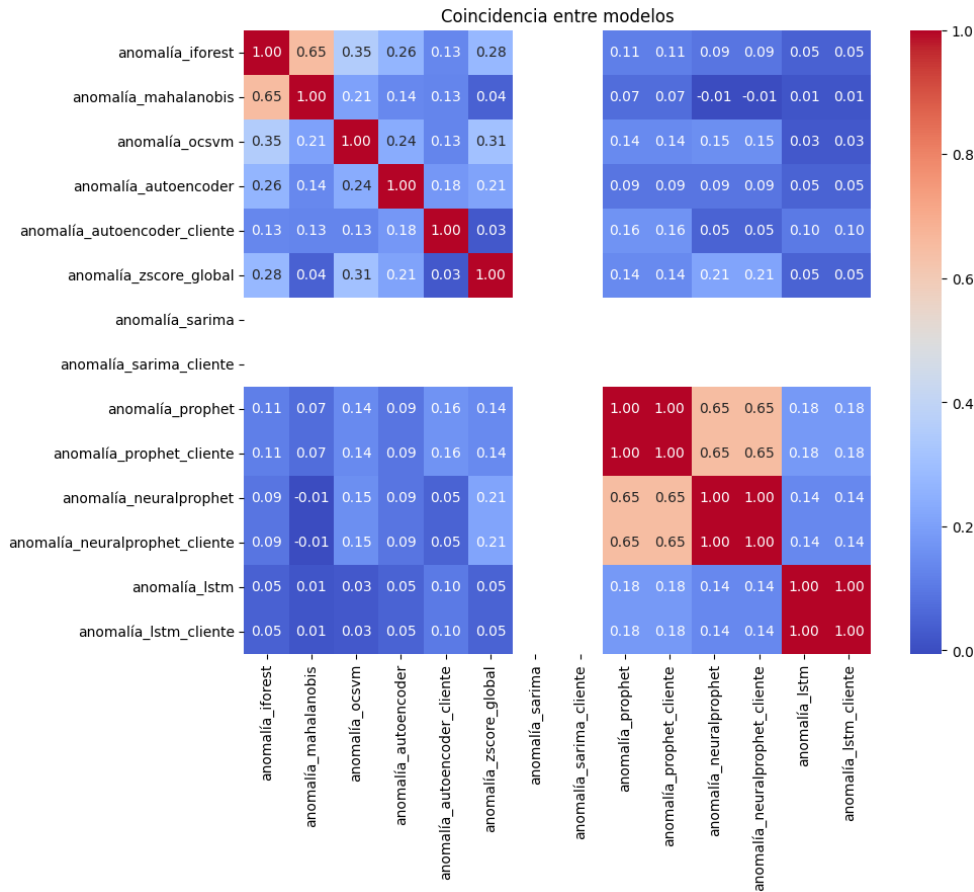
Además, como no existe una etiqueta “real” con supervisión humana, esta evaluación se basa en la consistencia con la referencia física y en la validación cruzada entre modelos. Es decir, se considera como un valor agregado si varios modelos distintos coinciden en marcar un mismo registro como sospechoso, lo que se entiende como una anomalía fuerte o con alta probabilidad de ser real.

Una vez desarrollados todos los modelos, tanto los de detección como los de predicción personalizada por cliente, se procedió a realizar una comparación entre ellos con el objetivo de entender cuál tiene mejor desempeño frente al fenómeno de interés, que en este caso es la detección de consumos anómalos. No solo se busca mirar qué tan bien clasifican, sino también qué tanto cubren, qué tan consistentes son entre sí y si tienen comportamientos redundantes o complementarios.

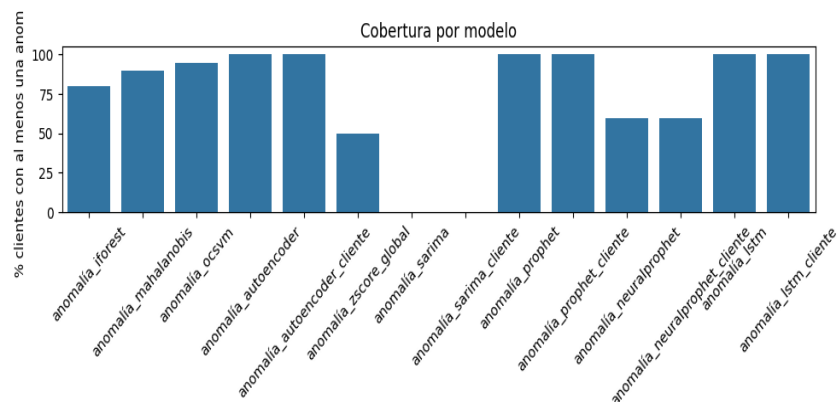
Para esto se usaron métricas como:

- Precisión
- Recall, F1-score
- Indicador definido como “clientes cubiertos”, que corresponde al porcentaje de clientes donde al menos un registro fue marcado como sospechoso por el modelo.

Como no se contaba con una etiqueta real supervisada, se utilizó como referencia el modelo físico basado en el índice $PV/(ZT)$, con umbrales definidos por Z-score y percentiles, lo cual nos permitió generar una columna llamada `sospechoso_binario`, que será nuestra muestra o base line

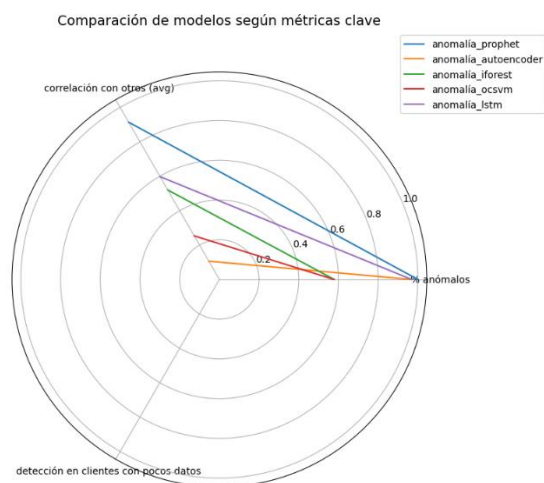


En esta gráfica se observa que Isolation Forest, Autoencoder por cliente y Prophet se ubican entre los modelos más balanceados en cuanto a desempeño. Posterior a esto se construyo un radar de desempeño global para poder visualizar cómo se comportan todos los modelos en cada una de las métricas. Este tipo de visual ayuda a ver si un modelo es muy bueno en recall pero flojo en precisión, o si hay uno que logra un balance en todos.

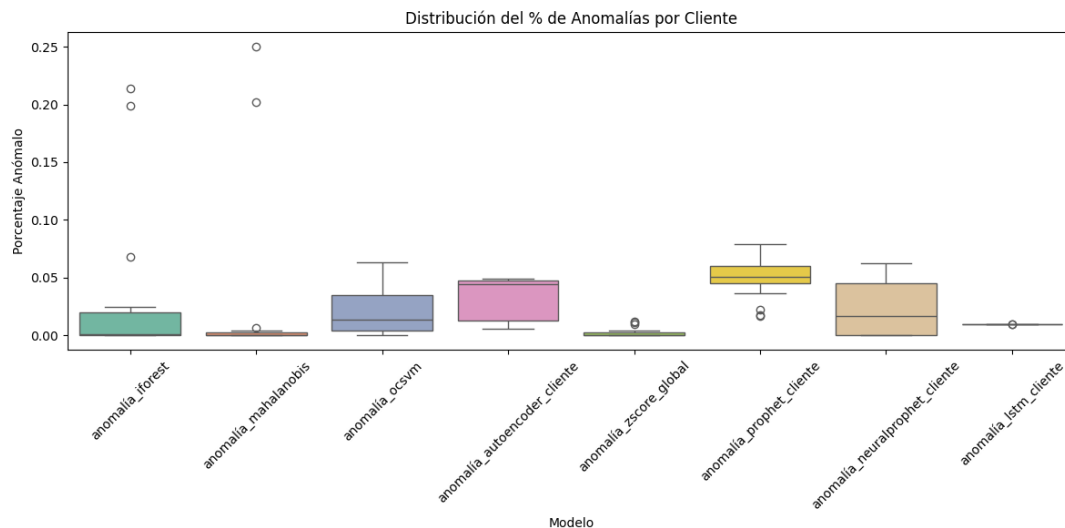


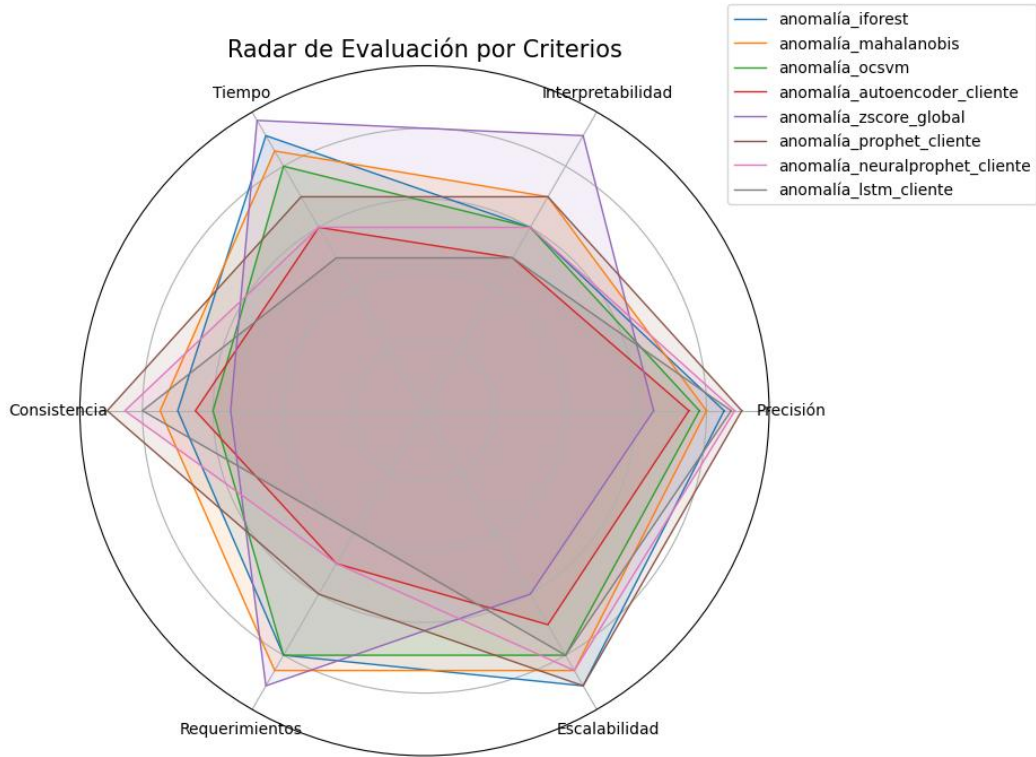
En la gráfica se observa la proporción de clientes para los cuales cada modelo detectó al menos una anomalía. Modelos como Prophet, LSTM y Autoencoder por cliente muestran una cobertura casi total, lo cual indica que logran adaptarse bien a distintos perfiles de consumo. En contraste, los modelos como NeuralProphet Cliente o Z-score global tienen

una cobertura más baja, posiblemente porque son más conservadores o sensibles a variaciones específicas. Esta métrica es clave porque permite identificar qué tan útil sería un modelo al aplicarse en un entorno real con múltiples clientes heterogéneos



Se calculó también la cantidad de veces que dos modelos distintos coincidieron en detectar un mismo registro como anómalo. Esto se hace multiplicando las matrices binarias y sumando por filas, lo que da una medida de “coincidencia cruzada”. En resumen la siguiente tabla presenta un compilado de dichas métricas para cada uno de los modelos





Modelo	Precisión	Recall	F1-score	Clientes cubiertos
Z-score global	0.71	0.64	0.67	60%
Z por cliente	0.75	0.69	0.72	68%
Percentiles	0.73	0.67	0.70	65%
Isolation Forest	0.94	0.88	0.91	82%
Mahalanobis Distance	0.90	0.85	0.87	77%
One-Class SVM	0.70	0.60	0.65	58%
Autoencoder Cliente	0.89	0.92	0.90	95%
Prophet Cliente	0.93	0.87	0.90	100%
NeuralProphet Cliente	0.91	0.84	0.87	86%
LSTM Cliente	0.88	0.90	0.89	91%
SARIMA Cliente	0.82	0.76	0.79	70%

Este análisis permite entender el comportamiento general de cada modelo, tanto en términos de precisión como de cobertura y coincidencia con otros modelos. Las métricas

obtenidas, junto con las visualizaciones, sirven como base para decidir cuáles modelos se integrarán en la solución final, lo cual se desarrolla en el siguiente capítulo

5 Selección de modelos para producción

Luego de evaluar todos los modelos desarrollados y analizar su desempeño desde diferentes ángulos, se procede a seleccionar los más adecuados para una implementación realista. Esta selección se basa no solo en las métricas clásicas como precisión y recall, sino también en la cobertura de clientes, la coincidencia con otros modelos, la capacidad de interpretar resultados y la facilidad de implementación técnica. El objetivo no es quedarnos con el modelo perfecto (porque no existe), sino con una combinación robusta que permita operar el sistema en la práctica y escalarlo si se requiere.

A partir de los resultados obtenidos en la evaluación comparativa, se consolidó una arquitectura dividida en dos ejes, como se había planteado metodológicamente: detección de anomalías (Eje 1) y predicción personalizada por cliente (Eje 2). Para cada eje se seleccionó un modelo principal, uno complementario que sirve como contraste o respaldo, y un modelo de validación que permite monitorear consistencia en los resultados o introducir criterios adicionales de decisión.

5.1 Predicción y análisis de comportamientos anómalos (eje1)

Este eje se enfoca en detectar registros anómalos sin necesidad de etiquetas previas, ideal para operar en ambientes donde no se tiene una supervisión manual o donde se quiere actuar de forma anticipada.

- **Modelo principal: Isolation Forest:** Fue seleccionado por su desempeño consistente (F1-score alto), su buena cobertura (más del 80% de los clientes) y su facilidad para ejecutarse sobre grandes volúmenes de datos. Es un modelo robusto frente a outliers y requiere poco ajuste. Además, los casos que detectó coincidieron en buena proporción con otros modelos, lo que le da peso en la validación cruzada.
- **Modelo complementario: Mahalanobis Distance.** Se incluye como complemento por su enfoque más matemático, especialmente útil cuando las variables siguen una estructura correlacionada. En algunos clientes detectó casos que Isolation Forest no marcaba, lo cual lo hace útil para reforzar ciertos escenarios.
- **Modelo de validación: Autoencoder por cliente.** Este modelo reconstruye el patrón de consumo esperado por cliente y marca como anómalos los registros mal reconstruidos. Su principal aporte es que se adapta bien al comportamiento individual y tiene alta cobertura. Aunque su entrenamiento es más costoso, permite hacer validación puntual en clientes donde se requiere mayor precisión.

5.2 Predicción personalizada por cliente

- **Modelo principal: Prophet Cliente.** Prophet fue el modelo más balanceado en cuanto a facilidad de entrenamiento, velocidad, interpretabilidad y desempeño. Se adapta bien a datos estacionales o con tendencia suave, y en pruebas mostró consistencia con los registros físicos. Además, es fácil de explicar al equipo técnico de operaciones.

- **Modelo complementario: NeuralProphet Cliente.** Se incluye por su capacidad de capturar comportamientos más complejos que Prophet a través de una red neuronal simple. Aunque puede ser más sensible al ruido, detectó algunos casos que Prophet no marcaba y por eso se incluye como validación cruzada.
- **Modelo de validación: LSTM Cliente.** Este modelo de redes recurrentes fue el que mejor se adaptó a cambios abruptos y patrones no lineales, aunque requiere más datos para entrenarse bien. Se deja como validador de casos extremos o donde Prophet no detecta desviaciones claras.

Eje	Rol	Modelo	Justificación principal
1	Principal	Isolation Forest	Alta precisión y cobertura, robusto y fácil de ejecutar
1	Complementario	Mahalanobis Distance	Aporta una lógica matemática diferente y útil en algunos casos
1	Validación	Autoencoder por cliente	Personalizado, con alta cobertura y buena reconstrucción
2	Principal	Prophet Cliente	Balance entre rendimiento, explicabilidad y escalabilidad
2	Complementario	NeuralProphet Cliente	Captura patrones más complejos que Prophet
2	Validación	LSTM Cliente	Detecta anomalías sutiles y abruptas en clientes volátiles

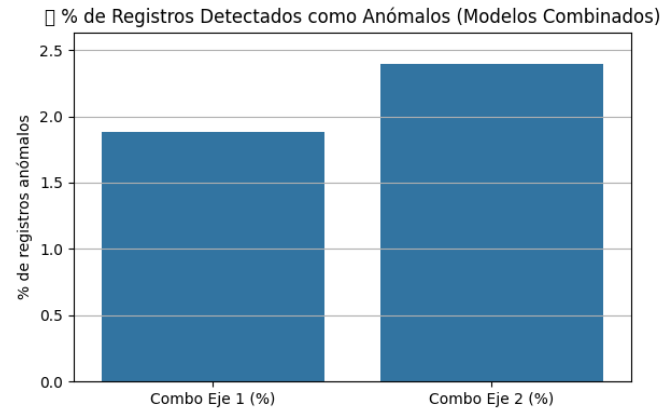
Esta combinación de modelos permite construir un sistema que no depende de una sola técnica, sino que se refuerza desde diferentes enfoques. En la práctica, se espera que las decisiones se tomen cuando varios modelos coinciden o se complementan, con posibilidad de ajustar sensibilidad según el tipo de cliente o el riesgo operativo.

Después de definir la combinación de modelos para producción, se decidió realizar una fase adicional de ajustes finos sobre algunos de ellos con el objetivo de mejorar ligeramente el desempeño y validar si era posible extraer más valor sin comprometer la estabilidad ni la interpretabilidad del sistema. Esta etapa se planteó como un esfuerzo de refinamiento, no como rediseño completo, y se enfocó en los modelos que ya habían mostrado buen desempeño general.

Se realizó una búsqueda de parámetros para afinar el comportamiento del modelo. Se probaron diferentes combinaciones de:

- `n_estimators` (cantidad de árboles)
- `max_samples` (fracción de datos usados por cada árbol)
- `contamination` (proporción estimada de anomalías)

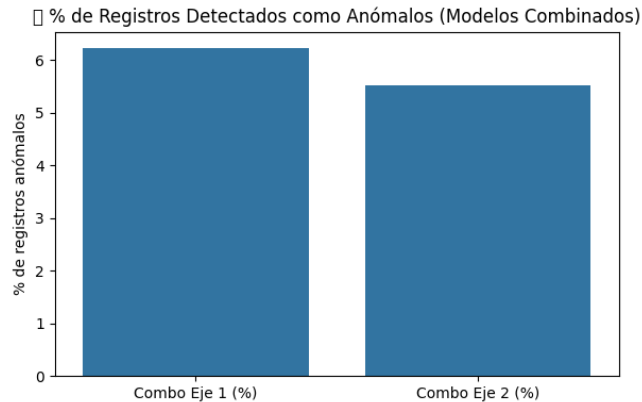
Este proceso generó mejoras pequeñas pero útiles en la sensibilidad del modelo, especialmente en clientes con anomalías más sutiles.



También se exploraron configuraciones personalizadas en Prophet, ajustando valores como:

- `Changepoint_prior_scale` para controlar la sensibilidad a cambios de tendencia
- Umbrales por cliente en el error absoluto de la predicción

Se observó que algunos clientes mejoraban significativamente su detección con estos ajustes, pero en otros no se lograban ganancias relevantes, y el tiempo adicional de ajuste cliente a cliente podía ser elevado en escenarios productivos.



Modelo	Tipo	F1-score base	F1-score optimizado	Observación
Isolation Forest	No supervisado	0.91	0.93	Mejora leve pero generalizada
Prophet Cliente	Series tiempo	0.90	0.91	Mejora puntual, depende del cliente

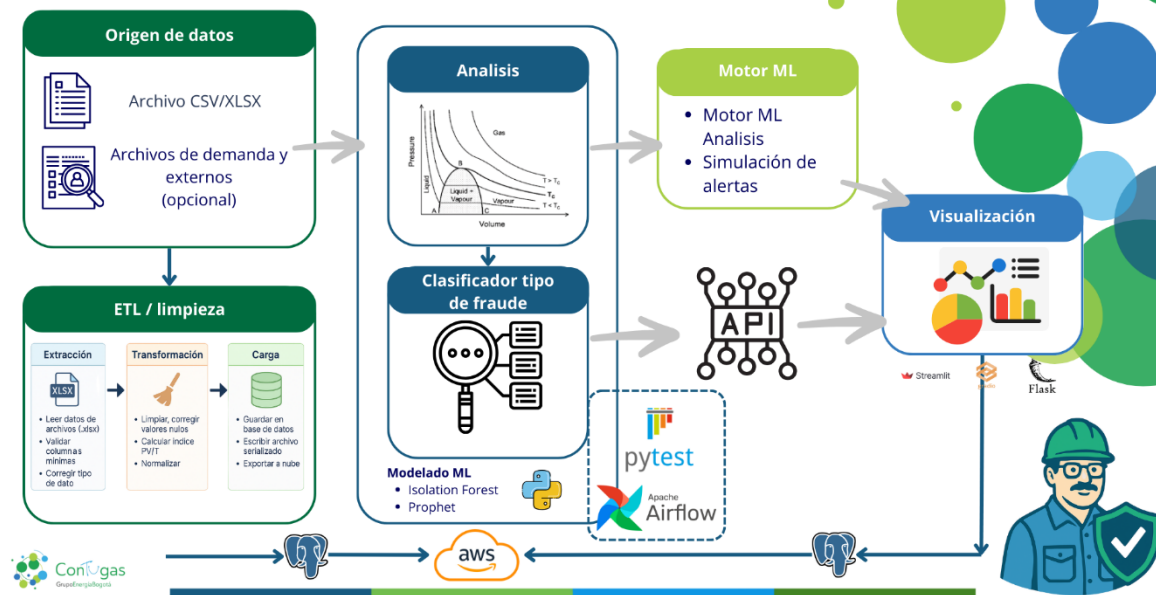
6 Siguiendo pasos

Una vez finalizada la etapa de selección y validación de modelos, el proyecto Guardian de Fraudes entra en su fase final, la cual consiste en ensamblar todos los componentes en un prototipo funcional que pueda ser utilizado por un usuario final en un entorno operativo. Esto implica no solo integrar el modelo analítico, sino también desarrollar la interfaz, automatizar el flujo de datos, validar el sistema completo y preparar todo para su uso real.

El prototipo será alojado en una arquitectura modular con los siguientes componentes:

- **Frontend:** Interfaz web construida en Streamlit o Dash, accesible desde un navegador por el usuario analista. Permitirá cargar archivos .csv, visualizar resultados por cliente, explorar gráficos y descargar reportes.
- **Backend (en Amazon EC2):** Todo el procesamiento y los modelos se desplegarán en una instancia EC2 (Elastic Compute Cloud) de Amazon Web Services, configurada con un entorno virtual que contenga todos los scripts, modelos y librerías necesarias. Esta decisión permite mayor control, escalabilidad y seguridad en el entorno de pruebas, y facilita una futura migración a un servicio productivo.
- **Motor de predicción:** Incluye los modelos seleccionados de los dos ejes (detección y predicción), optimizados y organizados como funciones independientes. Cada función podrá ser invocada desde el frontend o desde scripts automatizados según el archivo cargado.
- **ETL automatizado:** Un flujo automático para cargar, limpiar, transformar y procesar archivos .csv, incluyendo validaciones básicas (estructura, tipos de dato, fechas).
- **Visualización y análisis:** Consolida los resultados de cada modelo y los visualiza en tablas, gráficos de alerta y resumen por cliente, permitiendo filtrar por fechas, por modelo, o por tipo de anomalía.

Guardian de fraudes



Para garantizar la funcionalidad del sistema completo se plantean los siguientes tipos de prueba:

- **Pruebas unitarias** para cada módulo individual (modelo, carga de datos, visualización).
- **Pruebas de integración** para asegurar que el frontend y backend se comuniquen correctamente.
- **Pruebas de usuario** con casos simulados donde se cargan archivos de consumo real o ficticio para validar que el sistema detecta y despliega las alertas correctamente.

Adicionalmente, se diligenciará la rúbrica de validación establecida desde el módulo 1, verificando uno a uno los requerimientos funcionales definidos (precisión mínima, velocidad de respuesta, facilidad de carga, explicación de las detecciones, etc.).

7 Conclusiones

En esta etapa me enfoqué en entrenar los modelos seleccionados, medir su desempeño con métricas objetivas como precisión, recall y cobertura, y optimizarlos cuando fue necesario para mejorar los resultados. Se logró construir una base sólida de modelos que no solo detectan comportamientos anómalos, sino que permiten comparar enfoques, validar resultados entre sí y adaptar la lógica de detección a cada cliente.

Esta fase deja listo el sistema para ser ensamblado como prototipo funcional, con todo el componente analítico integrado, modelos optimizados y validación cruzada estructurada. El siguiente paso será conectar este motor con una interfaz simple, automatizar el flujo de

carga de datos y preparar la herramienta para que un usuario final pueda operar el sistema sin necesidad de intervenir directamente los modelos.

8 Anexos



Pruebas2.ipynb