UNIVERSIDAD PONTIFICIA COMILLAS
Escuela Técnica Superior de Ingeniería (ICAI)
Máster Universitario en Big Data

TRABAJO FIN DE MÁSTER

ARQUITECTURA DE OBSERVABILIDAD Y ALERTA TEMPRANA PARA ECOSISTEMAS ADTECH: FRAMEWORK ESCALABLE DE MONITORIZACIÓN DE ALTA GRANULARIDAD

Autor: Navarro Tudury, Rubén
Director: de Prada Martínez, Álvaro
Entidad Colaboradora: Seedtag

Madrid, 2026

Declaro, bajo mi responsabilidad, que el Proyecto presentado con el título

Arquitectura de Observabilidad y Alerta Temprana para Ecosistemas AdTech: Framework Escalable de Monitorización de Alta Granularidad

en la ETS de Ingeniería - ICAI de la Universidad Pontificia Comillas en el curso académico 2025/26 es de mi autoría, original e inédito y no ha sido presentado con anterioridad a otros efectos.

El Proyecto no es plagio de otro, ni total ni parcialmente y la información que ha sido tomada de otros documentos está debidamente referenciada.

Fdo.: Rubén Navarro Tudury Fecha: 13/ 05/ 2003

Autorizada la entrega del proyecto

EL DIRECTOR DEL PROYECTO

Fdo.: Álvaro de Prada Martínez Fecha: 18/ 05/ 2026

ARQUITECTURA DE OBSERVABILIDAD Y ALERTA TEMPRANA PARA ECOSISTEMAS ADTECH: FRAMEWORK ESCALABLE DE MONITORIZACIÓN DE ALTA GRANULARIDAD

Autor: Navarro Tudury, Rubén.
Director: De Prada Martínez, Álvaro.
Entidad Colaboradora: Seedtag.

RESUMEN DEL PROYECTO

Este TFM consiste en un sistema de detección de anomalías para Seedtag. El sistema procesa millones de combinaciones dimensionales en tiempo real sobre cinco tablas de métricas, pero centrándose en la horaria comparando un día y hora de la semana concreto con el mismo día de la semana y hora de hace 1, 2, 3 y 4 semanas. Gracias a un sistema de alertas inteligentes en Slack, se ha conseguido reducir el tiempo de respuesta de días a horas, minimizando las pérdidas económicas.

Palabras clave: detección, observabilidad, alertas.

Introducción

En el ecosistema AdTech se generan millones de transacciones por segundo mediante la compra-venta automatizada de espacios publicitarios en tiempo real, lo que genera un volumen masivo de datos. En este entorno, cambios en métricas como coste, bid floors o bid inputs pueden producir pérdidas económicas en cuestión de minutos.

Las herramientas de Business Intelligence tradicionales, basadas en revisión manual de dashboards, no consiguen detectar anomalías en combinaciones dimensionales granulares ya que quedan ocultas tras sus agregados.

Asimismo, los datos del sector publicitario presentan una alta estacionalidad (anual, mensual, semanal e intradía). Por ello, se ha implementado la técnica de Same-Hour Historical Matching para el análisis horario y Same-WeekDay Historical Matching para el diario.

Definición del proyecto

El proyecto construye un framework analítico que observa el tráfico en miles de combinaciones dimensionales, comparando el rendimiento actual de cada una frente a su histórico de las últimas semanas. El sistema opera directamente en la capa de transformación del Data Lakehouse, con ejecuciones horarias y diarias, gracias al uso de dbt. Además es completamente dinámico, permitiendo a cada stakeholder ajustar los thresholds de alerta sin necesidad de un conocimiento técnico.

Descripción del sistema

La arquitectura se divide en tres capas fundamentales:

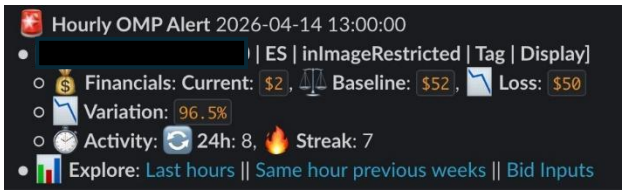
1. **Modelos dbt:** Donde se utiliza toda la lógica SQL para cálculos de los baselines, división en múltiples combinaciones dimensionales, comparación con Same-Hour/WeekDay Historical Matching y deduplicación jerárquica.
2. **Orquestación en Apache Airflow:** Implementa sensores de disponibilidad y ejecución para ejecutar los modelos sobre datos completos y actualizados, garantizando la fiabilidad de las alertas
3. **Notificación y gestión dinámica:** Con el uso de Slack, Google Sheets y Python se permite una configuración personalizada de thresholds y mecanismos de exclusión temporal que permite enviar las alertas relevantes al canal correspondiente

Resultados

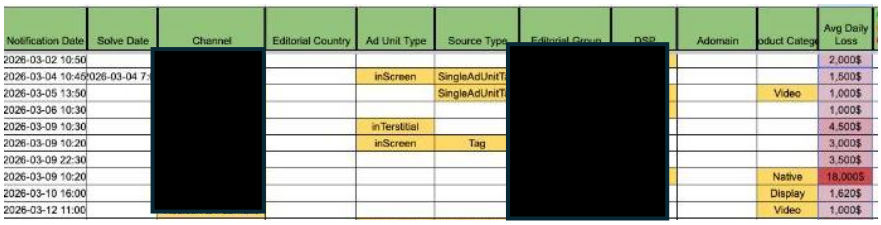
El sistema identifica caídas y desconexiones en las primeras horas de ocurrencia, detectando incluso aquellas anomalías que no son detectados en los dashboards agregados, posibilita la configuración de thresholds por país y tipo de alerta para adaptarse a diferentes escalas de negocio y reducir el ruido considerablemente.

A su vez, el mecanismo de exclusión temporal y la implementación de botones interactivos en Slack evitan la saturación de los canales de comunicación.

En el mes de prueba consiguió detectar anomalías que reducían un total de $70000 por día de tardanza en su detección.



1. Resultados 1



2. Resultados 2

Conclusiones

Este TFM demuestra que es posible construir un sistema de detección de anomalías robusto, transparente y de alta granularidad utilizando SQL, dbt y herramientas de orquestación estándar, sin recurrir a algoritmos de Machine Learning complejos de 'caja negra'. La arquitectura desarrollada es replicable a cualquier empresa del sector AdTech y extensible a cualquier métrica o fuente de datos que se desee.

Referencias

Onetag. (s. f.). *Smart observability: Building one of the industry's most efficient RTB infrastructures*. https://www.onetag.com/smart-observability-building-one-of-the-industrys-most-efficient-rtb-infrastructures/

Junta de Andalucía – Agencia Digital. (s. f.). *Arquitectura de referencia observabilidad*. https://desarrollo.juntadeandalucia.es/recursos/reglas-pautas/arquitectura-referencia-observabilidad

Digiday. (s. f.). *Inside real-time bidding: How programmatic actually works*. https://digiday.com/media/what-is-real-time-bidding/

Amazon Advertising. (s. f.). *What is AdTech?* https://advertising.amazon.com/es-es/library/guides/what-is-adtech#2d

DataCamp. (s. f.). *What is a data lakehouse?* https://www.datacamp.com/blog/what-is-a-data-lakehouse

Monte Carlo Data. (s. f.). How to build your own data anomaly detectors using SQL. https://www.montecarlodata.com/blog-how-to-build-your-own-data-anomaly-detectors-using-sql/

Great Expectations. (s. f.). Data quality monitoring. https://greatexpectations.io/data-quality-monitoring/

OBSERVABILITY ARCHITECTURE AND EARLY WARNING SYSTEM FOR ADTECH ECOSYSTEMS: A SCALABLE HIGH-GRANULARITY MONITORING FRAMEWORK

Author: Navarro Tudury, Rubén.
Supervisor: De Prada Martínez, Álvaro.
Collaborating Entity: Seedtag.

ABSTRACT

This Master's Thesis presents an anomaly detection system developed for Seedtag. The system processes millions of dimensional combinations in real time across five metrics tables, with a particular focus on hourly analysis by comparing a specific day of the week and hour against the same day and hour from the previous one, two, three, and four weeks. Through an intelligent Slack alerting system, the solution has reduced response times from days to hours, significantly minimizing economic losses.

Keywords: detección, observability, alerting.

Introduction

In the AdTech ecosystem, millions of transactions per second are generated through the automated real-time buying and selling of advertising inventory, producing a massive volume of data. In this environment, changes in metrics such as cost, bid floors, or bid inputs can lead to significant economic losses within minutes.

Traditional Business Intelligence tools, which rely on manual dashboard reviews, are unable to effectively detect anomalies in granular dimensional combinations, as these remain hidden behind aggregated metrics.

Moreover, advertising industry data exhibits strong seasonality patterns, including yearly, monthly, weekly, and intraday trends. To address this, the system implements the Same-Hour Historical Matching technique for hourly analysis and Same-Weekday Historical Matching for daily analysis.

Project Definition

The project builds an analytical framework that monitors traffic across thousands of dimensional combinations, comparing the current performance of each one against its historical behavior over the previous weeks. The system operates directly within the transformation layer of the Data Lakehouse, with hourly and daily executions enabled through the use of dbt. It is also fully dynamic, allowing each stakeholder to adjust alert thresholds without requiring technical expertise.

System Description

The architecture is divided into three fundamental layers:

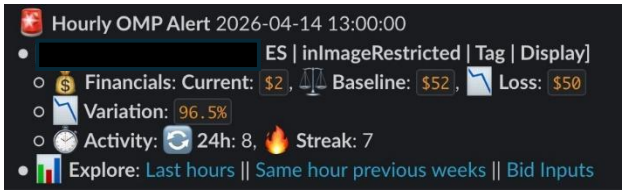
1. **dbt Models:** where all the SQL logic is implemented for baseline calculations, splitting into multiple dimensional combinations, Same-Hour / Weekday Historical Matching comparisons, and hierarchical deduplication.
2. **Orchestration in Apache Airflow:** it implements availability and execution sensors to run models on complete and up-to-date data, ensuring the reliability of alerts.
3. **Notification and dynamic management:** using Slack, Google Sheets, and Python, the system enables custom threshold configuration and temporary exclusion mechanisms, allowing relevant alerts to be sent to the appropriate channel.

Results

The system identifies drops and disconnections within the first hours of occurrence, even detecting anomalies that are not visible in aggregated dashboards. It also enables the configuration of thresholds by country and alert type, allowing adaptation to different business scales and significantly reducing noise.

In addition, the temporary exclusion mechanism and the implementation of interactive buttons in Slack help prevent the saturation of communication channels.

During the one-month testing period, the system was able to detect anomalies that were responsible for a total estimated loss reduction of $70,000 per day of detection delay.



3. Results 1

Notification Date	Solve Date	Channel	Editorial Country	Ad Unit Type	Source Type	Editorial Group	DSP	Adomain	oduct Categ	Avg Daily Loss
2026-03-02 10:50		[redacted]				[redacted]				2,000$
2026-03-04 10:45	2026-03-04 7:0			inScreen	SingleAdUnitTa					1,500$
2026-03-05 13:50					SingleAdUnitTa				Video	1,000$
2026-03-06 10:30										1,000$
2026-03-09 10:30				inTerstitial						4,500$
2026-03-09 10:20				inScreen	Tag					3,000$
2026-03-09 22:30										3,500$
2026-03-09 10:20									Native	18,000$
2026-03-10 16:00									Display	1,620$
2026-03-12 11:00									Video	1,000$

4. Results 2

Conclusions

This Master's Thesis demonstrates that it is possible to build a robust, transparent, and highly granular anomaly detection system using SQL, dbt, and standard orchestration tools, without relying on complex "black-box" machine learning algorithms. The developed architecture is replicable across any company in the AdTech sector and can be extended to any metric or data source as needed.

References

Onetag. (s. f.). *Smart observability: Building one of the industry's most efficient RTB infrastructures*. https://www.onetag.com/smart-observability-building-one-of-the-industrys-most-efficient-rtb-infrastructures/

Junta de Andalucía – Agencia Digital. (s. f.). *Arquitectura de referencia observabilidad*. https://desarrollo.juntadeandalucia.es/recursos/reglas-pautas/arquitectura-referencia-observabilidad

Digiday. (s. f.). *Inside real-time bidding: How programmatic actually works*. https://digiday.com/media/what-is-real-time-bidding/

Amazon Advertising. (s. f.). *What is AdTech?* https://advertising.amazon.com/es-es/library/guides/what-is-adtech#2d

DataCamp. (s. f.). *What is a data lakehouse?* https://www.datacamp.com/blog/what-is-a-data-lakehouse

Monte Carlo Data. (s. f.). How to build your own data anomaly detectors using SQL. https://www.montecarlodata.com/blog-how-to-build-your-own-data-anomaly-detectors-using-sql/

Great Expectations. (s. f.). Data quality monitoring. https://greatexpectations.io/data-quality-monitoring/

Índice de la Memoria

Índice de Tablas

Índice de Ilustraciones

Capítulo 1. Introducción

Cada vez que un usuario abre una página web, antes de que termine de cargar, ya se ha celebrado una subasta. En menos de 100 milisegundos, múltiples plataformas han pujado por mostrarle un anuncio, una ha ganado y el anuncio se ha puesto en el espacio publicitario correspondiente. Multiplicado por millones de usuarios simultáneos, el ecosistema AdTech genera millones de estas transacciones por segundo, produciendo un volumen masivo de datos en tiempo real.

En este entorno, cualquier cambio en métricas como el coste, los bid floor o los bid inputs puede producir pérdidas económicas instantáneas sin que nadie lo detecte, el cual es un problema de suma importancia ya que la pérdida total puede ser mayor que problemas económicamente más grandes, al tardar más en detectarse. Algunos de estos problemas son: la desconexión de un formato concreto, la caída de pujas en una región específica, o un error de configuración que afecta a un grupo editorial concreto. Estos incidentes no interrumpen el flujo global de datos, sino que quedan ocultos tras los totales agregados, y pueden permanecer durante días hasta que son detectados por un analista, pero para entonces ya ha habido una perdida significativa.

Las herramientas tradicionales de Business Intelligence no están diseñadas para este tipo de detección. Detectar ciertos problemas manualmente utilizando dashboards agregados es como intentar encontrar una aguja en un pajar. A esto se suma que el tráfico publicitario no se comporta igual un lunes que un viernes, ni en enero que en noviembre, ni en la mañana que en la noche. Esta alta estacionalidad hace que comparar un valor actual con un promedio genérico sea una fuente constante de falsas alarmas.

Este proyecto nace para resolver exactamente estos problemas, construyendo un sistema automatizado que supervise miles de combinaciones dimensionales simultáneamente, detecte cualquier anomalía en sus primeras horas de ocurrencia y notifique al equipo responsable con el problema y la pérdida económica estimada.

Capítulo 2. Descripción de las Tecnologías

En este capítulo se describen las principales tecnologías utilizadas para el desarrollo del sistema.

2.1 AdTech, RTB, SSP y DSP

El término AdTech engloba el conjunto de plataformas y herramientas tecnológicas que automatizan la compra-venta de espacios publicitarios digitales. El mecanismo central de este ecosistema es el Real-Time Bidding (RTB), el cual es un sistema de subastas en tiempo real en el que, cada vez que un usuario carga una página web, se celebra una puja automatizada en milisegundos para decidir qué anuncio se le muestra en el espacio publicitario.

En este proceso intervienen dos tipos de plataformas. Por un lado, los SSP (Supply-Side Platforms) representan a los publishers, es decir, a los propietarios de los espacios publicitarios que se ponen a la venta para poder monetizarlos. Por otro, los DSP (Demand-Side Platforms) representan a los anunciantes, que pujan por esos espacios publicitarios para mostrar sus anuncios.

2.2 Data Lakehouse

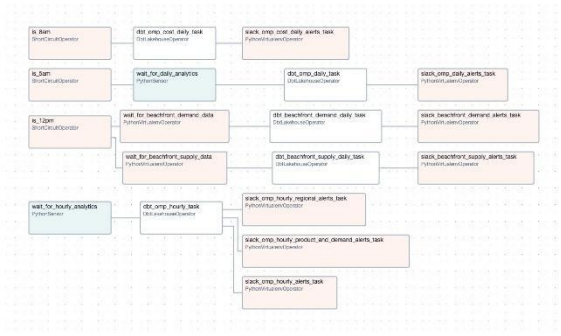
El Data Lakehouse es una arquitectura de almacenamiento y análisis de datos que soluciona los problemas del Data Lake y Data Warehouse, integrando el beneficio de ambos, escalabilidad y bajo coste de almacenamiento del Data Lake junto con el rendimiento, fiabilidad y gobernanza de los datos del Data Warehouse.

2.3 dbt (Data Build Tool)

dbt es una herramienta de transformación de datos que permite construir pipelines analíticos en SQL mediante un control de versiones, tests automatizados y documentación para la perfecta interpretación de las tablas y sus variables. Opera exclusivamente en la capa de transformación del Data Lakehouse, ejecutando las consultas directamente sobre él sin mover los datos.

2.4 Apache Airflow

Apache Airflow es una plataforma de orquestación de workflows o tasks que permite definir, programar y monitorizar pipelines de datos mediante grafos dirigidos con dependencias y pasos definidos. Un DAG define el conjunto de tasks que componen un pipeline y las dependencias entre ellos, de esta manera cada task se ejecuta en un orden especificado y cuando sus dependencias han finalizado para poder tener fiabilidad en los datos, además realiza reintentos automáticos ante fallos y ofrece la capacidad de reprocesar periodos pasados en cualquier momento. El DAG junto con sus tasks, funcionalidades y dependencias son escritos en Python.

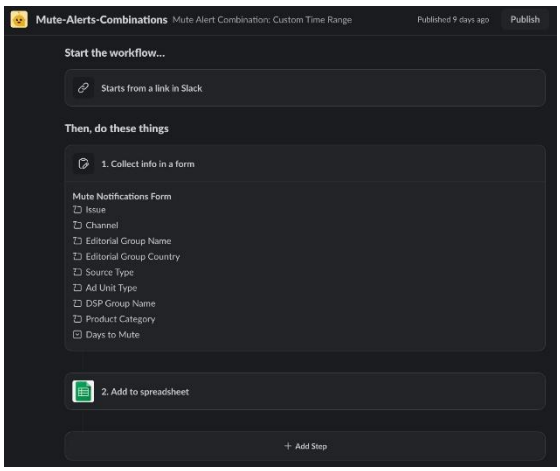


5. Grafo Airflow

6. Ejecuciones Airflow

2.5 Slack Webhooks y Bots Interactivos

Slack es la plataforma de comunicación utilizada para la entrega de notificaciones, es una herramienta semejante a Teams. Los Incoming Webhooks son URLs que permiten a aplicaciones externas, como archivos escritos en Python, publicar mensajes en un canal de Slack de forma automática y programática. Los Bots de Slack son workflows interactivos que permiten añadir formularios y botones directamente en los grupos y mensajes de Slack, de esta forma el usuario puede interactuar y modificar el sistema sin necesidad de tener el conocimiento técnico de otras herramientas.



7. Bot Slack

Capítulo 3. Estado de la Cuestión

La monitorización y la detección de anomalías en series temporales con seasonality es ampliamente utilizado en el ámbito del Machine Learning, los modelos ARIMA y la ingeniería de datos. Sin embargo, son soluciones bastante complejas, que además, necesitan de un historial grande de datos de cada una de las combinaciones para funcionar adecuadamente

3.1 Limitaciones del Business Intelligence Tradicional

Históricamente, la supervisión del rendimiento de cada una de las secciones de negocio y de la compañía en general ha recaído en analistas que revisan visualmente dashboards en herramientas como Tableau, Power BI o Superset. Este enfoque reactivo tiene una escalabilidad muy limitada al depender de una persona para revisar manualmente los datos, a su vez, es humanamente imposible detectar cada una de las caídas en combinaciones muy granulares ya que para ellos sería necesario revisar manualmente el rendimiento de miles de combinaciones dimensionales entre canales, DSPs, SSPs, formatos, ….

Cuando una métrica cae en una combinación dimensional muy específica, por ejemplo, el formato de vídeo en un dominio particular en España, el problema suele quedar oculto por los totales agregados, pasando desapercibido durante días. La ausencia de automatización en la detección convierte al analista en un cuello de botella que frena la escalabilidad.

3.2 El Problema de la Estacionalidad y los Umbrales Estáticos

Las primeras aproximaciones hacia la automatización consistieron en establecer umbrales fijos (por ejemplo, enviar alerta si el ingreso baja de 100$). Esta técnica resulta inútil en el ecosistema AdTech debido a su alta estacionalidad. El tráfico web y el gasto publicitario varían drásticamente no solo a lo largo del año, sino según el día de la semana y la hora del día: los lunes a las 8:00 tienen un patrón radicalmente distinto al de los viernes a las 20:00, lo mismo pasa en Diciembre frente a Enero.

Los umbrales estáticos generan dos falsas, falsos positivos y falsos negativos. Las medias móviles simples tampoco son tan efectivas, ya que no discriminan entre días laborables y fines de semana.

3.3 Plataformas de Data Observability

En los últimos años han ganado fuerza herramientas de observabilidad de datos como Monte Carlo Data y Great Expectations. Ambas se enfocan en asegurar la calidad de los datos mediante la detección de problemas en distintas dimensiones como el volumen, la fiabilidad, la completitud o los cambios en la estructura y distribución de los datos.

Se utilizan sobre todo para identificar fallos en pipelines o transformaciones antes de que impacten en dashboards o modelos, por lo que no está diseñado para detectar anomalías sobre combinaciones dimensionales granulares de datos como ocurre en las AdTechs, por ende, sería muy complejo su adaptación para usarlas en el proyecto.

3.4 Algoritmos de Machine Learning para Detección de Anomalías

La detección de anomalías en series temporales suele ser muy común en algoritmos como ARIMA, Isolation Forests o LSTM. Aunque funcionan perfectamente frente a la estacionalidad, presentan desventajas significativas en el contexto de este proyecto:

- **Coste computacional:** entrenar y ejecutar modelos de ML para cada hora y para cada una de las miles de combinaciones dimensionales resulta computacionalmente muy costoso.
- **Opacidad:** para los equipos a los que van dirigidas las alertas, un algoritmo complejo actúa como una caja negra, y cuando se lanza una alerta es difícil comprobar matemáticamente por qué el modelo ha considerado un valor como anómalo, lo que retrasa la toma de decisiones y pone en duda la fiabilidad del algoritmo.
- **Falta de datos en combinaciones nuevas:** las combinaciones dimensionales recientes o de bajo volumen no disponen de suficiente histórico para entrenar un modelo predictivo con garantías y necesitarían de mucho tiempo para tener los datos necesarios para una predicción fiable, en cambio, el proyecto solo precisaría de unas pocas semanas de datos.

Capítulo 4. Definición del Trabajo

4.1 Justificación

El mercado de la publicidad programática mueve cientos de miles de millones de euros anuales a nivel global, y cualquier empresa que opere en él compite con muchas otras mediante pujas que se realizan en cuestión de milisegundos. Una caída silenciosa de una combinación puede acumular pérdidas durante días sin que nadie lo sepa.

Las soluciones existentes no resuelven este problema. Las plataformas de observabilidad como Monte Carlo o Great Expectations monitorizan la calidad del dato pero no las métricas de negocio, los algoritmos de Machine Learning son costosos computacionalmente además de inviables para combinaciones de bajo volumen, los umbrales estáticos generan un ruido constante de falsas alertas que hace que los equipos las acaben ignorando y los dashboards tradicionales pueden no detectar una caída en una combinación concreta, camuflandola por su agregado.

Este proyecto cubre esos inconvenientes con un sistema que monitoriza miles de combinaciones simultáneamente, que neutraliza la estacionalidad sin modelos de caja negra, y que entrega alertas a los equipos correspondientes encargados de actuar. Su arquitectura sobre dbt y Airflow lo hace mantenible de forma automática, y su capa de configuración en Google Sheets lo hace viable para la modificación de los thresholds por los equipos de negocio, sin necesidad de habilidades técnicas.

En términos de balance total, el coste de implantación es mínimo frente al valor de detectar en horas lo que hoy tarda días en identificarse. Una incidencia no detectada durante 24 horas en una combinación de alto tráfico puede suponer pérdidas de incluso 11000 euros, cifra que supera por sí sola el coste del proyecto.

4.2 Objetivos

El proyecto se articula en torno a cuatro objetivos principales:

- El diseño de modelos dbt documentados, que procesen las tablas, enriquezcan los datos con metadatos dimensionales y generen subtotales para todas las combinaciones.
- La implementación de una lógica SQL que calcule un baseline dinámico para cada combinación mediante Same-Hour Historical Matching y Same-WeekDay Historical Matching, con thresholds configurables que permitan adaptar la sensibilidad a cada una de las reglas de negocio sin modificar el código.
- El desarrollo de un módulo de Smart Filtering y deduplicación que construya una lógica que evalúe el número de alertas por nivel dimensional para determinar la jerarquía de la alerta y detectar si es una caída es una dimensión concreta o una caída a nivel más granular, a su vez, debe eliminar las alertas relacionadas entre ellas que tienen una misma causa raíz.
- La integración de todos los modelos en un framework de orquestación con ejecución horaria y diaria, que tenga un mecanismo que: envíe las anomalías a los canales de Slack correspondientes y gestione los thresholds y exclusiones de forma dinámica en Google Sheets.

4.3 Metodología

La ejecución del proyecto se guía por una metodología de desarrollo iterativo e incremental, con validaciones continuas en cada etapa para el perfecto desarrollo del proyecto. Este enfoque permite ajustar tanto la lógica como la sensibilidad de los algoritmos antes de avanzar a la siguiente fase, incorporando el feedback de los stakeholders de forma continua.

El plan de trabajo se estructura en cuatro fases que cubren el período de noviembre de 2025 a mayo de 2026.

La Fase 1, de análisis exploratorio y diseño del modelo base, parte de un enfoque analítico antes que técnico. Antes de escribir una sola línea de código, es necesario entender en profundidad la estructura y el comportamiento de los datos disponibles en cada una de las tablas. Esta fase se dedica al entendimiento del negocio en una AdTech, la identificación de las métricas clave a utilizar en el proyecto y el significado de cada una de las variables que van a ser utilizadas en el análisis.

La Fase 2, de desarrollo iterativo del motor analítico y la lógica de detección, es la fase central y más exigente del proyecto. En esta fase se construye la consulta SQL que concentra toda la lógica de detección necesaria como la división por todas las posibles combinaciones dimensionales, el cálculo del baseline histórico, comparación con valores anteriores, la deduplicación jerárquica y la aplicación de los filtros de sensibilidad configurables. Esta fase es necesariamente iterativa para validar los resultados frente a datos históricos reales y comprobar que funciona a la perfección.

La Fase 3, de modelado en dbt, orquestación en Airflow y notificaciones, toma la consulta SQL desarrollada en la fase anterior y la replica en los distintos modelos dbt, uno por cada tabla de origen. La orquestación se implementa mediante DAGs en Apache Airflow con sensores de disponibilidad de datos, se desarrolla el sistema de alertas en Slack con mediante sus webhooks para poder enviar cada alerta al canal correspondiente y se implementa la capa de configuración dinámica en Google Sheets.

La Fase 4, de histórico, mecanismo de exclusión e iteración con stakeholders, dota al proyecto de una funcionalidad operativa y una trazabilidad necesarias para la evaluación del proyecto y su escalabilidad a largo plazo. Se crea el histórico de alertas y el mecanismo de exclusión temporal almacenado en Google Sheets, así como el botón interactivo en Slack. El feedback de los stakeholders permite ajustar thresholds y realizar los cambios pertinentes para la correcta adopción del proyecto por los equipos de la empresa, generando el mayor valor posible.

4.4 Planificación

El proyecto se desarrolló entre noviembre de 2025 y mayo de 2026, con una dedicación aproximada de 25 horas semanales, lo que supone un total estimado de unas 600 horas de desarrollo. La siguiente tabla refleja la distribución temporal de las actividades principales agrupadas por mes:

Tarea	Nov	Dic	Ene	Feb	Mar	Abr	May
Análisis exploratorio y diseño del modelo base	X	X					
Desarrollo iterativo del motor analítico y la lógica de detección		X	X	X			
Modelado en dbt, orquestación en Airflow y notificaciones				X	X	X	
Histórico, mecanismo de exclusión e iteración con stakeholders						X	X

1. Planificación del Proyecto

Capítulo 5. Arquitectura del Sistema Desarrollado

Este capítulo describe en detalle cada uno de los componentes que conforman el sistema de detección de anomalías, desde el análisis de los datos de entrada hasta la notificación de las alertas.

5.1 Análisis del Sistema

El núcleo del sistema es la tabla Revenue horario, que registra los ingresos publicitarios hora a hora en la plataforma. Sobre ella se construye la lógica de detección principal, ya que es la tabla de mayor granularidad temporal y la que permite una respuesta más rápida ante incidencias. Las dimensiones que la definen son: Channel, Editorial Group, DSP, Region, Ad Unit Type, Source Type y Product Type.

El resto de las tablas son OMP Revenue diario, BeachFront Supply Revenue diario, BeachFront Demand Revenue diario y OMP Coste diario. Cada una se ejecuta a nivel diario y tiene sus propias dimensiones y tabla fuente, aunque el patrón de detección es parecido al de la tabla horaria.

Para cada una de estas tablas, el sistema sigue el mismo flujo:

- Se enriquecen los datos mediante otras tablas auxiliares.
- Se generan los subtotales para todas las combinaciones dimensionales posibles.
- Se calcula el baseline histórico de cada combinación en su franja horaria/diaria.
- Se detectan las alertas mediante la comparativa actual vs baseline.
- Se hace un deduplicado y filtrado jerárquico de alertas.
- Se notifican las alertas relevantes por Slack.

Las tablas auxiliares sirven para hacer las distintas operaciones necesarias para poder estandarizar los procesos, como la conversión de divisas a euros o la estandarización de nombres de canales y DSPs.

5.2 Diseño: El Motor de Alta Granularidad

5.2.1 Generación de Combinaciones Dimensionales

La consulta base de cada modelo dbt aplica una función CUBE sobre las dimensiones de la tabla fuente, generando automáticamente todas las posibles combinaciones. Para las siete dimensiones de OMP, esto produce hasta $2^7 = 128$ combinaciones por fila de datos. Cada fila del resultado representa una combinación dimensional concreta donde puede aparecer el valor agregado en alguna de las variables, el valor NULL representa el agregado de todos los posibles valores en esa variable.

5.2.2 Estrategia Incremental en dbt

Cada modelo dbt está configurado con estrategia incremental por lo que en cada ejecución, el modelo procesa únicamente los datos de la última hora o día disponibles, añadiendo los resultados a la tabla sin necesidad de reprocesar el más datos de los necesarios. Esto reduce drásticamente el tiempo de ejecución y el coste computacional.

5.3 Implementación: La Lógica de Detección de Anomalías

5.3.1 Same-Hour Historical Matching y Same-WeekDay Historical Matching

Para la tabla horaria se aplican ambas técnicas de forma combinada. El Same-Hour Historical Matching compara el valor actual con el promedio de esa misma hora en las cuatro semanas anteriores, mientras que el Same-WeekDay Historical Matching restringe además esa comparación al mismo día de la semana, eliminando la estacionalidad intradia y semanal. Para las tablas diarias, donde no hay granularidad horaria, se utiliza únicamente el Same-WeekDay Historical Matching, comparando cada día con el promedio de ese mismo día de la semana en las cuatro semanas anteriores.

5.3.2 Umbrales Configurables y Lectura desde Google Sheets

El sistema no aplica un único threshold, sino que lee dinámicamente los thresholds configurados en Google Sheets en cada ejecución de Airflow que se dedica a enviar las alertas a los stakeholders. La sheet de Google Sheets que contiene los thresholds está dividida por país y tipo de alerta (con o sin Editorial Group), pudiendo tener varios thresholds para un mismo país o tipo de alerta, lo único que habría que hacer es añadirlos verticalmente en la misma columna. El código lee todos los valores de la columna correspondiente y los aplica en la cláusula WHERE de la consulta como condiciones OR:

```
WHERE (1er_Threshold) OR (2do_Threshold) OR (3er_Threshold) OR ...
```

Este diseño permite a los equipos de negocio añadir, modificar o eliminar thresholds sin necesidad de modificar el código ni tener mucho conocimiento técnico de alguna herramienta compleja. Cada threshold puede combinar condiciones sobre el ratio de pérdida porcentual (por ejemplo, caída superior al 40% respecto al baseline), sobre el valor de pérdida en dolares (por ejemplo, caída superior a $40 respecto al baseline) y concurrencia de la alerta en las últimas 24 horas (por ejemplo, que la combinación haya sido detectada en más de 3 ejecuciones consecutivas), evitando así la mayor cantidad de ruido posible.

5.3.3 Smart Filtering: Deduplicación Jerárquica de Alertas

El Smart Filtering se implementa mediante funciones de ventana SQL. Para cada alerta detectada, el sistema calcula el número de combinaciones siblings (hermanos) que tiene, esto se entiende como las alertas con la misma combinación pero con otro valor en alguna dimensión (Por ejemplo, Mesa + Madera + Azul es hermano de Mesa + Madera + Rojo ya que ambos están bajo la combinación Mesa + Madera). Si ese número supera un valor definido en el código, el sistema entiende que el problema es de un nivel superior y suprime todas las alertas de nivel inferior, con la finalidad de evitar alertas duplicadas. El sistema intenta detectar el problema con la mayor granularidad posible para notificar únicamente la raíz del problema.

5.3.4 DAG de Airflow

El sistema se orquesta mediante un único DAG en Airflow que centraliza la lógica de detección y notificación. Para cada una de las tablas, el flujo de trabajo se divide en tasks que tienen tres bloques de finalidad distintos:

1. **Paso previo a la ejecución del modelo dbt:** Verifica que las tablas que va a utilizar el modelo dbt contengan los datos del timestamp que se va a procesar. Si los datos no están disponibles, el sensor reintenta de forma automática hasta alcanzar un timeout. En algunas tablas no hay una verificación de los datos ya que no es posible, en cambio se crea un task que comprueba la hora del sistema y deja continuar el flujo cuando es la hora objetivo.
2. **Ejecución del modelo dbt:** Ejecuta la lógica de detección completa, generando una nueva tabla en la que se usa el cálculo de baseline, aplicación de umbrales y deduplicación jerárquica.
3. **Paso posterior a la ejecución del modelo dbt:** Lee las alertas generadas, aplica las exclusiones y los umbrales dinámicos consultando Google Sheets, y publica los mensajes formateados en los canales de Slack correspondientes.

Esto permite una gestión independiente de cada tabla pero gestionadas dentro de un mismo DAG, ya que cada tabla tiene unas dependencias distintas.

5.3.5 Enrutamiento Inteligente de Alertas

Las alertas se dirigen a distintos canales de Slack en función de las características de la combinación afectada. El criterio principal de enrutamiento es la presencia o ausencia de Editorial Group. Las combinaciones sin Editorial Group se dirigen al canal de producto/demanda, mientras que las que tienen Editorial Group se enrutan al canal específico del país correspondiente, ya que cada Editorial Group pertenece a un país. Con este diseño se consigue que cada alerta llegue al equipo con la responsabilidad operativa adecuada para actuar sobre ella.

5.3.6 Formato del Mensaje de Alerta

Cada mensaje de alerta incluye la información necesaria para que el equipo pueda evaluar la gravedad del incidente y actuar sin necesidad de consultar herramientas adicionales:

- **Combinación dimensional:** Identificación de las dimensiones afectados (canal, país, DSP, editorial group, etc.).
- **Métrica actual:** Revenue o coste en dólares observado en el periodo analizado (hora o día).
- **Baseline histórico:** Promedio del revenue o coste de las cuatro semanas anteriores en la misma franja temporal.
- **Desviación porcentual:** Variación porcentual del valor actual respecto al baseline.
- **Impacto económico:** Estimación de la pérdida financiera en dólares en esa hora o día.
- **Acceso directo:** Enlace a gráficos prefiltrados para la visualización detallada del incidente en superset.

5.3.7 Filtro de Frecuencia de Alertas

El sistema implementa un filtro automático de frecuencia que limita el número de alertas enviadas por una combinación con el mismo Editorial Group o Channel + DSP en un período de 24 horas. En los canales regionales, el sistema verifica antes de cada envío si ya se ha notificado una alerta del mismo Editorial Group en las últimas 24 horas, en caso afirmativo, la alerta no se envía. En los canales de producto y demanda, la verificación se realiza sobre la combinación Channel + DSP. Si ya existe una alerta enviada para esa combinación en las últimas 24 horas, no se vuelve a notificar.

Esta lógica se implementa consultando la hoja de histórico de alertas de Google Sheets antes de cada publicación para no saturar los canales de comunicación con la misma alerta.

Esto solo funciona para la tabla horaria.

5.3.8 Mecanismo de Exclusión Temporal

Para evitar la saturación del canal de comunicación cuando un incidente ya ha sido identificado y está siendo gestionado, se creó un mecanismo de exclusión temporal. Un botón interactivo en Slack permite a cualquier stakeholder silenciar una combinación durante un máximo de siete días completando un formulario directamente en el canal, gracias a los bots de Slack.

Al enviar el formulario, un webhook escribe automáticamente una nueva fila en la hoja de exclusiones de Google Sheets con la combinación dimensional a excluir, el número de días de silenciado (entre 1 y 7), persona que ha aplicado la exclusión y fecha de envío del formulario, para poder hacer el cálculo con el número de días silenciado. En cada ejecución del task de notificación, el sistema consulta esta hoja y evita enviar cualquier combinacion excluida.

Esto solo funciona para la tabla horaria.

5.3.9 Histórico de Alertas

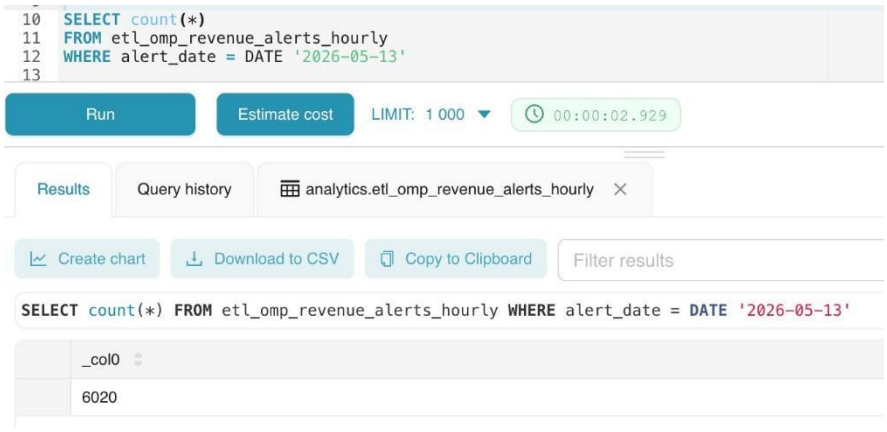
La hoja de histórico de alertas en Google Sheets registra automáticamente cada notificación que se ha enviado a los canales de Slack, junto con la pérdida económica horaria, el canal al que fue enviada y un campo de comentario que el equipo responsable puede completar una vez resuelto el incidente para tener un registro de las alertas y su correcto incidente, de esta manera se puede evaluar la efectividad del sistema.

Capítulo 6. Análisis de Resultados

En este capítulo se presentan los principales resultados obtenidos del proyecto desarrollado, evaluando su comportamiento sobre datos reales.

6.1 Cobertura y Granularidad

El sistema cubre de forma simultánea las cinco tablas de monitorización definidas en el alcance del proyecto. Durante las ejecuciones horarias, el modelo evaluó cientos de miles de combinaciones dimensionales, consolidando los resultados en aproximadamente 6.000 registros de datos. El análisis abarcó desde el nivel más granular (canal, país, DSP, editorial group, source type, formato y categoría) hasta las agregaciones de nivel superior. Este volumen masivo de supervisión automatizada resulta completamente inasumible mediante los procesos de revisión manual empleados antiguamente.



8. Count Datos por Día

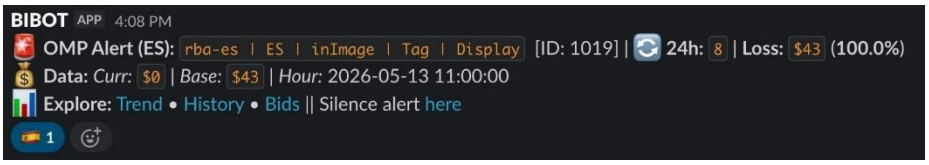
9. Fila de Datos pt1

10. FIla de Datos pt2

6.2 Detección Temprana de Incidencias

La evaluación sobre datos reales confirmó la eficacia del sistema para identificar las anomalías que los dashboards agregados no lograban reflejar. En la mayoría de los casos, la alerta fue generada la primera hora tras la ocurrencia del incidente.

A lo largo de uno de los meses, el sistema emitió 25 alertas críticas de negocio que resultaron ser verdaderas. La detección temprana de estas anomalías como desconexiones o caídas de tráfico, representó una mitigación de pérdidas estimada en 70.000 dólares por cada 24 horas de retraso que hubiera supuesto la detección manual.



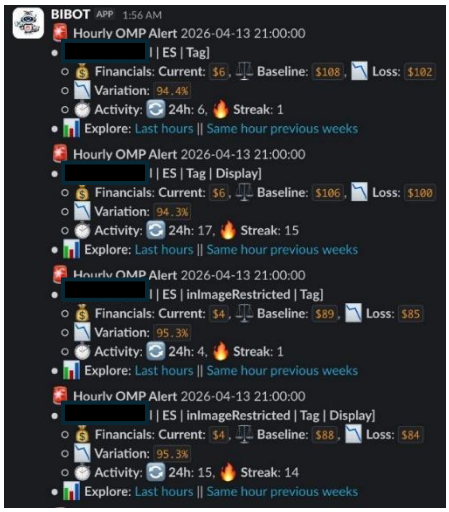
11. Alerta Real

Notification Date	Solve Date	Channel	Editorial Country	Ad Unit Type	Source Type	Editorial Group	DSP	Adomain	oduct Categ	Avg Daily Loss
2026-03-02 10:50		[redacted]				[redacted]				2,000$
2026-03-04 10:45	2026-03-04 7:0			inScreen	SingleAdUnitTag					1,500$
2026-03-05 13:50					SingleAdUnitTag				Video	1,000$
2026-03-06 10:30										1,000$
2026-03-09 10:30				inTerstitial						4,500$
2026-03-09 10:20				inScreen	Tag					3,000$
2026-03-09 22:30										3,500$
2026-03-09 10:20									Native	18,000$
2026-03-10 16:00									Display	1,620$
2026-03-12 11:00									Video	1,000$

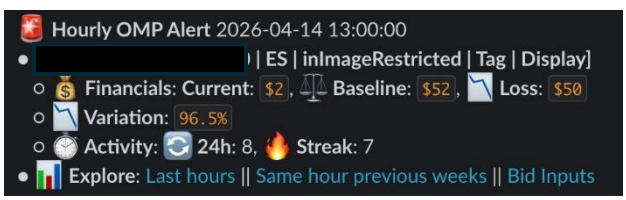
12. Historial de Alertas

6.3 Efectividad del Smart Filtering y Deduplicación Jerárquica

El rendimiento del Smart Filtering y deduplicación redució drásticamente el volumen de notificaciones redundantes que representaban la misma alerta. El sistema suprimió con éxito las alertas duplicadas, quedándose solamente con la raíz del problema. Como resultado, las caídas que habrían generado una cantidad considerable de registros duplicados fueron comprimidas a 1 o un máximo de 2 mensajes raíz. Todo ello consiguió una reducción de alertas que no aportaban información útil, evitando la saturación de los canales de comunicación.



13. Alertas Duplicadas



14. Alertas Deduplicadas Jerárquicamente

6.4 Configuración a la Escala de Negocio y Relevancia de Alertas

La puesta en producción dejó claro que una anomalía en una de las métricas no siempre es un problema relevante ya que hay mercados que trabajan con valores mucho más grandes, lo que puede suponer una pérdida insignificante frente a otros mercados que trabajan con valores mucho más pequeños.

Durante las pruebas quedó claro que la sensibilidad del sistema debía ajustarse al tamaño de cada mercado. Por ejemplo, una pérdida de $200 por hora apenas se nota en un país de alto volumen como Estados Unidos, pero supone una caída crítica para un mercado más pequeño como México.

La posibilidad de apilar múltiples umbrales en Google Sheets permitió ajustar estas reglas de negocio para cada mercado. Como resultado, se consiguió que el sistema solo enviase los avisos más relevantes, reduciendo el volumen de mensajes en los canales de comunicación para que el equipo pudiera centrarse en los problemas realmente importantes.

B	C	D	E	F
ES				US
Query clause		Variables to filter		Query clause
percentage > 90 AND occurrences_24h > 4		revenue_loss		evenue_loss > 60 AND percentage > 90 AND occurrences_24h > 4
revenue_loss > 120 AND percentage > 60		percentage		revenue_loss > 400 AND percentage > 60
		occurrences_24h		
		consecutive_streak		

15. Thresholds por Region

6.5 Adopción Operativa y el Desafío del Feedback Loop

Las alertas llegan diariamente a los canales correspondientes y los equipos de operaciones reciben la información de las caídas en tiempo real.

Sin embargo, hemos detectado un área clara de mejora. Aunque el envío de la notificación funciona correctamente, el uso de funcionalidades interactivas como el botón de exclusión temporal en Slack ha sido mínima, ya que la gente suele tardar en acostumbrarse a los cambios y al uso de las nuevas herramientas.

Actualmente, el principal desafío del sistema radica en el feedback loop. Una vez que la alerta aterriza en el canal, resulta difícil trazar de forma sistemática si el equipo responsable la ha revisado, qué acciones correctivas han tomado o cuál ha sido la causa raíz del incidente, ya que muchas veces no completan el apartado dedicado a 'Comentarios' en cada alerta. Esta falta de trazabilidad dificulta medir algunos parámetros interesantes como el tiempo real de resolución y documentar soluciones estandarizadas.

channel_id	editorial_group_name	Publisher_country	source_type	ad_unit_type	dsp_group_name	Product Category
	Zo[redacted]					

16. Alerta Excluida por Botón pt1

From date	Expiration days	To date (Do Not Edit)	Comments	Perso
2026-04-30	7	2026-05-0[redacted]	Zoomer Mute - We know why the drops are happening	@Zac

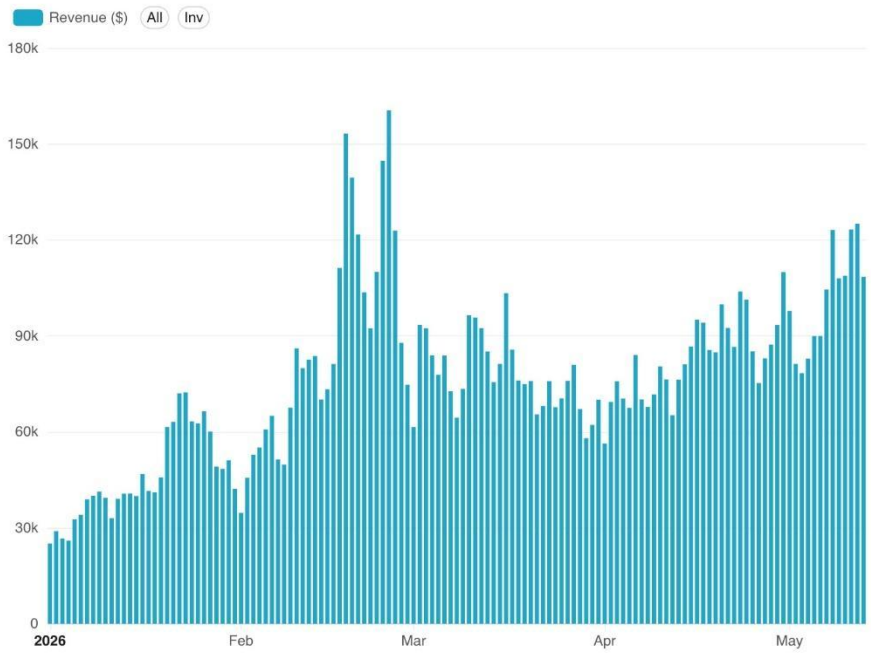
17. Alerta Excluida por Botón pt2

18. Alerta Dirigida a Stakeholder

6.6 Limitaciones del Sistema

A pesar de los buenos resultados obtenidos, hay varias limitaciones que se deben tenerse en cuenta:

- **Anomalias previas en el histórico:** Al utilizar la técnica de Same-Hour Historical Matching, el sistema asume que las últimas cuatro semanas tienen un comportamiento “normal”. Si en estas semanas o en la hora objetivo, ocurrió un evento excepcional, como Black Friday o una gran campaña previa, el promedio histórico estará sesgado. Esto provoca que el sistema genere falsos positivos o falsos negativos.
- **Cold Start:** Si aparece un nuevo DSP, formato, Channel, …, el modelo será incapaz de detectar una caída o anomalía de forma fiable hasta que acumulen datos suficientes para generar un promedio histórico representativo. Se requiere de mínimo de 1 semana de datos para su funcionamiento, aunque con poca fiabilidad, y de 4 semanas para su total fiabilidad.



19. Ejemplo Estacionalidad Alerta

Capítulo 7. Conclusiones y Trabajos Futuros

7.1 Conclusiones

Este Trabajo Fin de Máster ha demostrado que es posible construir un sistema de detección de anomalías robusto, transparente, escalable y de alta granularidad utilizando exclusivamente herramientas estándar del ecosistema de datos moderno: SQL, dbt, Apache Airflow, Google Sheets y Slack. La solución no requiere modelos de Machine Learning complejos ni infraestructura adicional, y opera directamente sobre el Data Lakehouse existente.

La técnica de Same-Hour Historical Matching se ha mostrado como un mecanismo eficaz para neutralizar la estacionalidad intradiaria y semanal del tráfico publicitario, que constituye el principal obstáculo para la detección de anomalías en el ecosistema AdTech. La comparación estricta con el mismo día de la semana y la misma hora garantiza que las alertas generadas responden a desviaciones reales y no a variaciones naturales del tráfico.

El módulo de Smart Filtering ha resuelto el problema de la tormenta de alertas, que es el principal motivo por el que los sistemas de alta granularidad suelen ser abandonados operativamente. La deduplicación jerárquica garantiza que cada alerta recibida en Slack sea responda a un problema raíz único.

La arquitectura de configuración dinámica en Google Sheets ha demostrado ser un elemento diferencial para la viabilidad operativa del proyecto: permite que los equipos de negocio sean autónomos en la gestión de umbrales y exclusiones, sin depender del equipo técnico para cada ajuste.

Más allá de la viabilidad técnica, la implementación del sistema ha demostrado un impacto financiero directo. Al reducir el tiempo de detección de anomalías drásticamente frente a la revisión manual, se ha mitigado de forma proactiva la pérdida de ingresos, validando el retorno de inversión de la arquitectura analítica diseñada. Si bien el modelo presenta áreas de mejora frente a escenarios estadísticos complejos como el arranque en frío (Cold Start), la robustez del diseño base ha superado con éxito las expectativas operativas.

En definitiva, el sistema desarrollado cumple los cuatro objetivos planteados: cubre con alta granularidad las cinco tablas de monitorización definidas, detecta anomalías en las primeras horas de ocurrencia, elimina el ruido mediante filtrado jerárquico y proporciona una capa de gestión operativa sostenible a largo plazo.

7.2 Trabajos Futuros

El sistema desarrollado sienta las bases de una plataforma de observabilidad más amplia que puede evolucionar en varias direcciones:

- **Framework de gestión de alertas:** Actualmente el sistema notifica la alerta pero no dispone de un mecanismo para registrar la respuesta del equipo. Se propone desarrollar un framework de gestión donde los stakeholders puedan visualizar todas las alertas activas, categorizar la causa raíz de cada incidente y marcarlas como resueltas. Este registro permitiría medir el tiempo medio de recuperación (MTTR), identificar patrones de fallo recurrentes y evaluar la efectividad operativa del equipo a lo largo del tiempo.
- **Extensión a nuevas tablas y métricas:** La arquitectura modular de dbt permite replicar el sistema de alertas a cualquier nueva tabla fuente o métrica de negocio con

un esfuerzo mínimo. Como primer candidato natural se identifican métricas como las Bid Inputs, que presentan patrones de estacionalidad similares al revenue.

- **Agente de IA para diagnóstico y visualización:** Como evolución natural del sistema, se propone el desarrollo de un agente de IA al que los stakeholders puedan consultar directamente sobre cualquier combinación dimensional afectada. El agente sería capaz de analizar el contexto histórico de la alerta, proporcionar una explicación de la posible causa raíz y ofrecer información adicional relevante para la toma de decisiones. Adicionalmente, ante situaciones que requieran una exploración visual más detallada, el agente podría generar dinámicamente dashboards con los filtros necesarios para analizar la combinación afectada sin necesidad de configuración manual.

Capítulo 8. Bibliografía

[1] Onetag. (s. f.). *Smart observability: Building one of the industry's most efficient RTB infrastructures*. https://www.onetag.com/smart-observability-building-one-of-the-industrys-most-efficient-rtb-infrastructures/

[2] Junta de Andalucía – Agencia Digital. (s. f.). *Arquitectura de referencia observabilidad*. https://desarrollo.juntadeandalucia.es/recursos/reglas-pautas/arquitectura-referencia-observabilidad

[3] Digiday. (s. f.). *Inside real-time bidding: How programmatic actually works*. https://digiday.com/media/what-is-real-time-bidding/

[4] Amazon Advertising. (s. f.). *What is AdTech?* https://advertising.amazon.com/es-es/library/guides/what-is-adtech#2d

[5] DataCamp. (s. f.). *What is a data lakehouse?* https://www.datacamp.com/blog/what-is-a-data-lakehouse

[6] Monte Carlo Data. (s. f.). How to build your own data anomaly detectors using SQL. https://www.montecarlodata.com/blog-how-to-build-your-own-data-anomaly-detectors-using-sql/

[7] Great Expectations. (s. f.). Data quality monitoring. https://greatexpectations.io/data-quality-monitoring/

Capítulo 9. ODS (Objetivos de Desarrollo Sostenible)

El proyecto se alinea con varios Objetivos de Desarrollo Sostenible:

ODS 8. TRABAJO DECENTE Y CRECIMIENTO ECONÓMICO

El sistema protege los ingresos de Seedtag, reduce perdidas económicas irrecuperables y mejora la eficiencia de los equipos.

ODS 9. INDUSTRIA, INNOVACIÓN E INFRAESTRUCTURA

El proyecto consiste en un sistema de alta granularidad replicable a cualquier empresa del sector.