



# MÁSTER UNIVERSITARIO BIG DATA Y ANÁLITICA AVANZADA.

PROYECTO FIN DE MASTER

“Gestión y optimización de la información de  
incidencias de suministro eléctrico en UFD Distribución  
de Electricidad”

Autor: Ana María Suárez Jiménez.

Director: Aitor Guinea Unzalu

Co-Director: Nombre Apellido1 Apellido2

Madrid



Declaro, bajo mi responsabilidad, que el Proyecto presentado con el título  
“Gestión y optimización de la información de incidencias de suministro eléctrico en UFD  
Distribución de Electricidad”

en la ETS de Ingeniería - ICAI de la Universidad Pontificia Comillas en el  
curso académico 2024/25 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.: Ana María Suárez Jiménez      Fecha: 26 de junio de 2025

Autorizada la entrega del proyecto

EL DIRECTOR DEL PROYECTO



Fdo.: Aitor Guinea Unzalu      Fecha: 26 de junio de 2025





# MÁSTER UNIVERSITARIO BIG DATA Y ANÁLITICA AVANZADA.

PROYECTO FIN DE MÁSTER

“Gestión y optimización de la información de  
incidencias de suministro eléctrico en UFD Distribución  
de Electricidad”

Autor: Ana María Suárez Jiménez

Director: Aitor Guinea Unzalu

Madrid



*“Convertir datos en información y ésta en conocimiento útil es el verdadero desafío en la era digital.”*

*Thomas Davenport*



# Agradecimientos

Quiero expresar mi más sincero agradecimiento a UFD Distribución de Electricidad por brindarme la oportunidad de realizar mis prácticas y facilitarme los medios necesarios para el desarrollo de este proyecto.

Agradezco especialmente a mi tutor en UFD, Aitor Guinea, por confiar en mí, por estar siempre dispuesto a ayudarme y por resolver cada una de mis dudas e inquietudes. Extiendo también mi gratitud al resto del equipo de Programación, quienes me han acompañado y hecho sentir parte de su equipo desde el primer día.

Finalmente, quiero agradecer a mi familia y mis amigos, esos que siempre están ahí y saben estar.



# GESTIÓN Y OPTIMIZACIÓN DE LA INFORMACION DE INCIDENCIAS DE SUMINISTRO ELECTRICO EN UFD Distribución de Electricidad.

**Autor: Suárez Jiménez, Ana María.**

Director: Guinea Unzalu, Aitor.

Entidad Colaboradora: UFD Distribución de Electricidad.

## RESUMEN DEL PROYECTO

Este trabajo analiza los datos de incidencias eléctricas registrados por UFD en el periodo 2022–2025, con el objetivo de evaluar la calidad de su tipificación y proponer mejoras aplicables tanto a incidencias programadas como imprevistas. Además, se estudia el impacto que tendría incorporar las redes de BT en los indicadores de calidad SAIDI y SAIFI, tradicionalmente calculados solo para MT. Para ello, se ha desarrollado un sistema de explotación de datos utilizando SQL, Python y Power BI.

**Palabras clave:** Análisis de incidencias, Calidad de suministro eléctrico, SAIDI, Data Lake, SQL, Power BI, Visualización de datos.

### 1. Introducción

La normativa vigente (RD 1955/2000 y Orden ECO/797/2002) obliga a las empresas distribuidoras a mantener un registro exhaustivo de las incidencias de suministro eléctrico. UFD gestiona este registro a través del Sistema de Gestión de Incidencias (SGI), donde se almacena información sobre tipo, motivo, material averiado, instalación origen, duración y clientes afectados, entre otros datos clave.

Este trabajo tiene un doble objetivo:

- Evaluar la calidad de la tipificación de las incidencias y proponer mejoras que faciliten su integración en el futuro sistema ADMS.
- Analizar cómo se distribuyen los indicadores SAIDI y SAIFI por niveles de tensión, considerando especialmente el peso creciente de las interrupciones en BT.

Para ello, se ha diseñado un sistema de explotación de datos con SQL, Python y Power BI que permite analizar la evolución de las incidencias, su tipificación y su impacto en los indicadores de calidad.

### 2. Definición del proyecto

El objetivo principal de este trabajo es analizar y optimizar la información registrada sobre incidencias eléctricas en UFD, con el fin de mejorar su calidad y facilitar su explotación operativa. Para ello, se plantean los siguientes objetivos específicos:

1. **Analizar la evolución de las incidencias** programadas e imprevistas, entre 2022 y 2025, con el fin de detectar patrones repetitivos en la tipificación, códigos más empleados, redundancias y otras limitaciones que afecten a su utilidad.
2. **Evaluar la calidad de tipificación de las incidencias** (motivo/causa, instalación origen, material averiado) y proponer líneas de mejora que aseguren mayor coherencia y capacidad de explotación de los datos.

3. **Evaluar la aportación de las incidencias de MT y BT en el SAIDI/SAIFI**, analizando en qué medida un cambio metodológico alteraría la evaluación de la calidad de suministro.
4. **Desarrollar herramientas de análisis y visualización interactiva** que faciliten la explotación de los datos

### 3. Descripción del sistema de explotación de datos

El sistema diseñado permite integrar, transformar y analizar los datos procedentes del SGI y otras fuentes de UFD, facilitando su explotación operativa a través de herramientas de análisis y visualización. Las fases son:

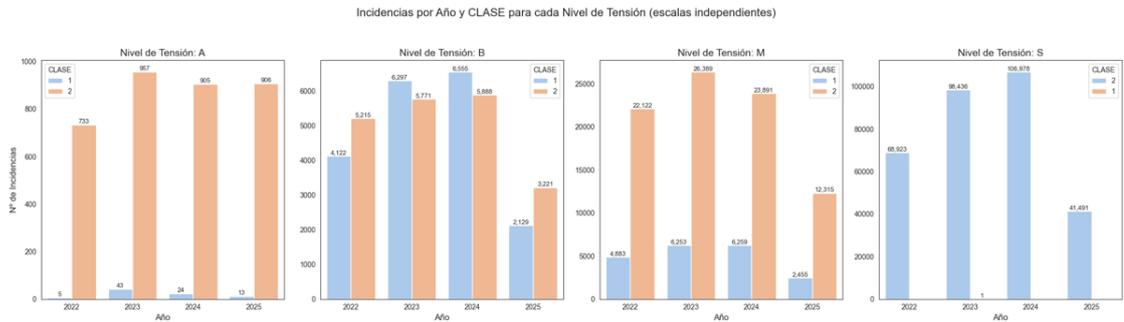
1. **Comprensión del contexto técnico y de los sistemas de información:** como paso previo, ha sido necesario familiarizarse con el funcionamiento del suministro eléctrico y con los sistemas de información utilizados por UFD. Esto ha incluido el estudio del ciclo de vida de una incidencia, la estructura del Sistema de Gestión de Incidencias (SGI), las distintas clasificaciones y tipologías disponibles, así como los indicadores de calidad aplicados en el sector eléctrico (TIEPI, NIEPI.SAIDI y SAIFI)
2. **Extracción de datos:** acceso a las tablas del Data Lake corporativo mediante consultas SQL, accediendo a información sobre incidencias, clientes afectados y tiempos de interrupción.
3. **Procesado y transformación:** se han limpiado y homogeneizado formatos, generando nuevas variables auxiliares y estandarizado los datos.
4. **Análisis exploratorio de las tipologías y cálculo de indicadores:** con SQL y Python se ha llevado a cabo un análisis detallado de las tipologías empleadas tanto para incidencias programadas como imprevistas desglosadas por nivel de tensión. Asimismo, se han calculado los indicadores SAIDI y SAIFI agregados por nivel de tensión para el año 2024.
5. **Visualización y explotación final:** se han desarrollado dos paneles en Power BI conectados directamente al Data Lake. El primero está orientado a la revisión y análisis de las tipologías, permitiendo filtrar y explorar por clase de incidencia, nivel de tensión, zona geográfica y tipo. El segundo, muestra la aportación de las incidencias programadas e imprevistas, por niveles de tensión, al SAIDI y al SAIFI.



Figura 1: Sistema de Explotación de Datos. Elaboración propia

## 4. Resultados

- **Evolución de las incidencias**

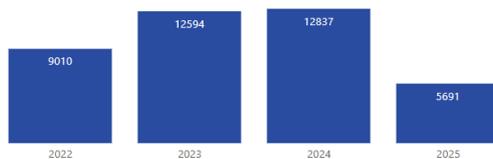


\* Clase 1 Trabajos Programados y clase 2 Incidencias Imprevistas

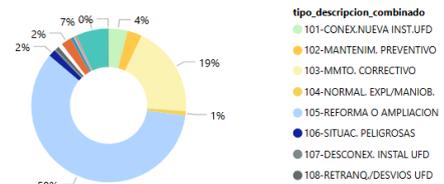
Del análisis de los datos de incidencias de los últimos 3 años y parte del año actual, se puede apreciar que se ha dado un incremento de incidencias en todos los niveles de tensión, destacando las SU y las de BT programadas.

- **Calidad de la tipificación de incidencias.**

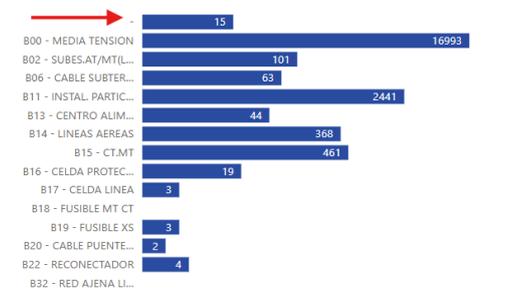
Numero de incidencias por año y clase



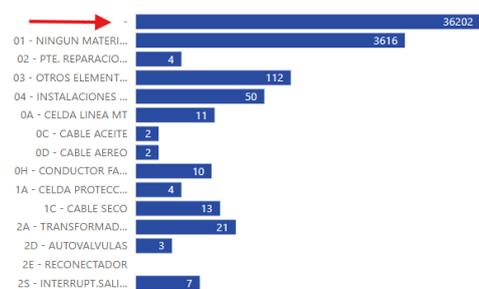
Número de Incidencias por Tipo



Numero de incidencias por Instalación Origen



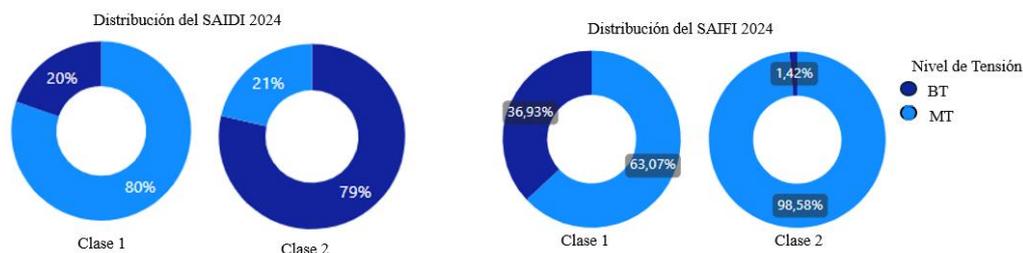
Numero de Incidencias por Material Averiado



Tipificación poco precisa, con abuso de códigos genéricos, campos vacíos o redundantes. Especialmente crítico en BT y SU.

- **Impacto de indicadores de calidad de suministro**

- El SAIDI y SAIFI imprevisto sigue dominado por la MT (>80%).
- En el SAIDI programado, BT aporta cerca del 80%, reflejando el impacto creciente de las actuaciones programadas en redes de BT (digitalización y renovación).
- Este cambio muestra que la inclusión de BT en indicadores como SAIDI puede tener un efecto significativo sobre la percepción de la calidad de suministro.



## 5. Conclusiones

- La calidad de la tipificación actual no es suficiente para garantizar una gestión eficiente ni para su futura integración en sistemas inteligentes.
- Para incidencias programadas, se propone utilizar la codificación de los trabajos del sistema GDA, mejorando así la trazabilidad y precisión de los datos.
- Para las incidencias imprevistas, es necesario revisar y simplificar la codificación, eliminando códigos redundantes, restringiendo las tipologías según el nivel de tensión y añadiendo mejoras como:
  - Un campo para identificar reparaciones provisionales.
  - Tipologías específicas para reenganches automáticos (<3 minutos).
  - Conservación del tipo original cuando se reclasifica como Fuerza Mayor.
- El análisis de SAIDI y SAIFI demuestra que la MT sigue siendo el nivel que más impacta en la calidad del suministro eléctrico (duración y frecuencia de interrupciones imprevistas), pero la BT tiene ya un peso muy relevante en las interrupciones programadas, con una aportación creciente que debe ser tomada en cuenta en el diseño futuro de los indicadores.

## 6. Referencias

- [1] Gobierno de España, “Real Decreto 1955/2000, de 1 de diciembre por el que se regulan las actividades de transporte, distribución, comercialización, suministro y procedimientos de autorización de instalaciones de energía eléctrica,” 2000. [En línea]. Disponible en: <https://www.boe.es/eli/es/rd/2000/12/01/1955/con> [Fecha de consulta 15 de marzo de 2025]
- [2] Gobierno de España, “Orden ECO/797/2002, de 22 de marzo, por la que se establecen los procedimientos de medida y control de la continuidad del suministro eléctrico,” 2002. [En línea]. Disponible en: <https://www.boe.es/eli/es/o/2002/03/22/eco797> [Fecha de consulta 15 de marzo de 2025]

# INFORMATION MANAGEMENT AND OPTIMIZATION OF POWER SUPPLY INCIDENTS IN UFD Electricity Distribution.

**Author: Suárez Jiménez, Ana María.**

Supervisor: Guinea Unzalu, Aitor.

Collaborating Entity: UFD Electricity Distribution.

## ABSTRACT

This project analyzes the data on electrical incidents recorded by UFD during the period 2022–2025, with the aim of evaluating the quality of their classification and proposing improvements applicable to both planned and unplanned incidents. In addition, it examines the potential impact of including low-voltage (LV) networks in the SAIDI and SAIFI quality indicators, which have traditionally been calculated only for medium voltage (MV). To achieve this, a data exploitation system has been developed using SQL, Python, and Power BI.

**Keywords:** Incident analysis, Power supply quality, SAIDI, Data Lake, SQL, Power BI, Data visualization

## 1. Introduction

The current regulations (Royal Decree 1955/2000 and Order ECO/797/2002) require electricity distribution companies to maintain a detailed record of power supply incidents. UFD manages this record through the Incident Management System (SGI), where information such as the type, cause, damaged material, originating facility, duration, and affected customers, among other key data, is stored.

This project has a dual objective:

- To evaluate the quality of incident classification and propose improvements that facilitate its integration into the future ADMS (Advanced Distribution Management System).
- To analyze how the SAIDI and SAIFI indicators are distributed across voltage levels, with special consideration of the increasing impact of interruptions in low-voltage (LV) networks.

To achieve this, a data exploitation system has been developed using SQL, Python, and Power BI, allowing for analysis of incident trends, classification quality, and their impact on quality indicators.

## 2. Project Definition

The main objective of this project is to analyze and optimize the information recorded on electrical incidents in UFD, with the aim of improving its quality and facilitating its operational use. To this end, the following specific objectives are proposed:

1. **Analyze the evolution of planned and unplanned incidents** between 2022 and 2025, to identify recurring patterns in classification, most frequently used codes, redundancies, and other limitations affecting data usefulness.

2. **Evaluate the quality of incident classification** (reason/cause, originating facility, faulty equipment) and propose improvements to ensure greater consistency and data usability.
3. **Assess the contribution of MV and LV incidents to SAIDI/SAIFI**, analyzing to what extent a methodological change would affect the evaluation of power supply quality.
4. **Develop analysis and interactive visualization tools** to facilitate the exploitation of data.

### 3. Description of the Data Exploitation System

The designed system enables integration, transformation, and analysis of data from the SGI and other UFD sources, facilitating its operational use through analysis and visualization tools. The process consists of the following phases:

1. **Understanding the technical context and information systems:** As a preliminary step, it was necessary to become familiar with the functioning of the power supply system and the information systems used by UFD. This included studying the life cycle of an incident, the structure of the Incident Management System (SGI), the various classifications and typologies available, as well as the quality indicators applied in the electricity sector (TIEPI, NIEPI, SAIDI, and SAIFI).
2. **Data extraction:** Accessing tables from the corporate Data Lake via SQL queries, retrieving information on incidents, affected customers, and outage durations.
3. **Processing and transformation:** Data formats were cleaned and standardized, new auxiliary variables were generated, and data were harmonized.
4. **Exploratory analysis of typologies and indicator calculation:** A detailed analysis was conducted using SQL and Python on the typologies used for both planned and unplanned incidents, broken down by voltage level. Additionally, SAIDI and SAIFI indicators were calculated and aggregated by voltage level for the year 2024.
5. **Visualization and final exploitation:** Two Power BI dashboards were developed, connected directly to the Data Lake. The first focuses on the review and analysis of typologies, allowing filtering and exploration by incident class, voltage level, geographic area, and type. The second displays the contribution of planned and unplanned incidents, by voltage level, to SAIDI and SAIFI.

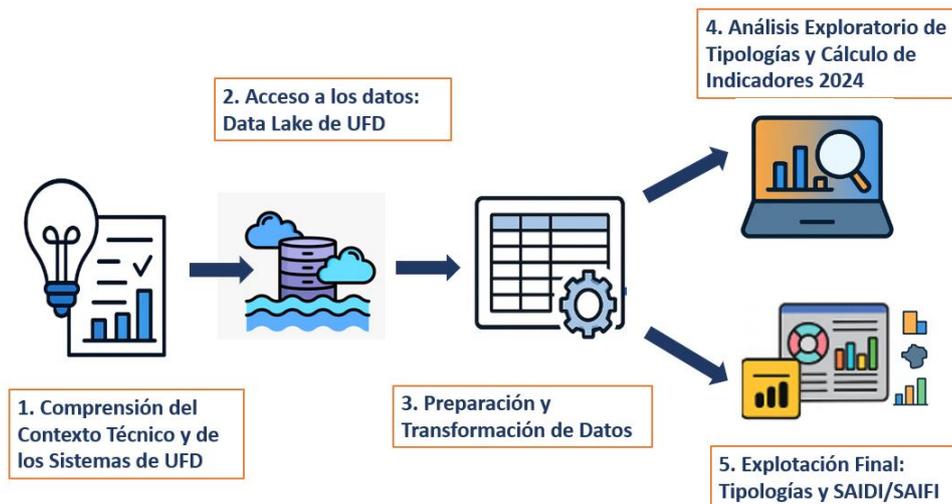
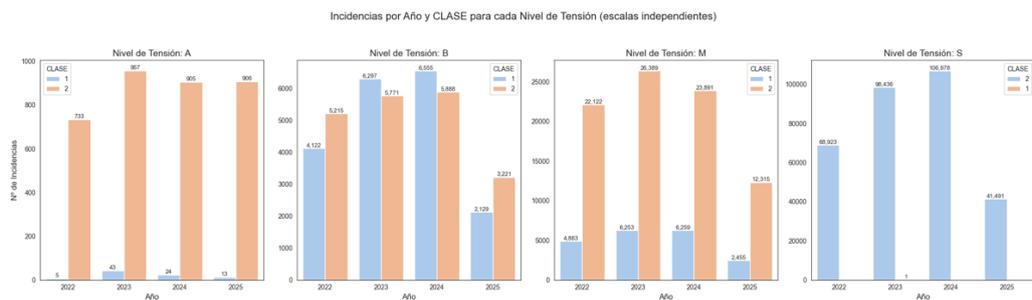


Figura 2: Data Exploitation System. Author's own work

## 4. Results

### • Evolution of Incidents

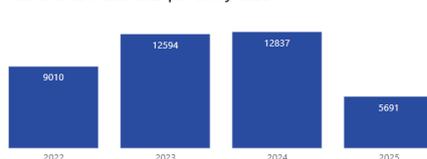


\* Clase 1 Trabajos Programados y clase 2 Incidencias Imprevistas

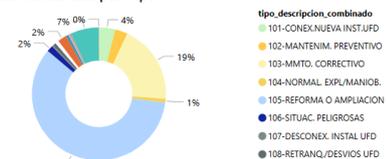
The analysis of incident data from the past three years reveals an increase in incidents across all voltage levels, with a notable rise in SU incidents and scheduled low-voltage (LV) outages.

### • Quality of Incident Classification.

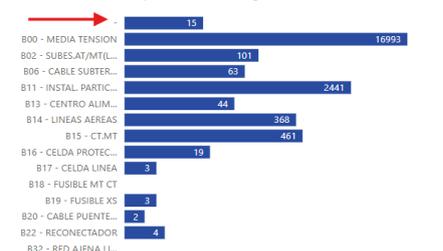
Numero de incidencias por año y clase



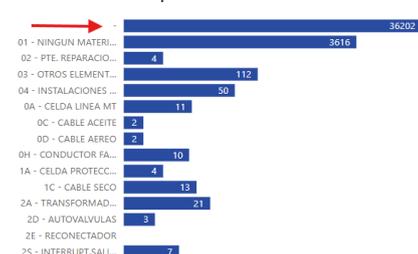
Numero de Incidencias por Tipo



Numero de incidencias por Instalación Origen

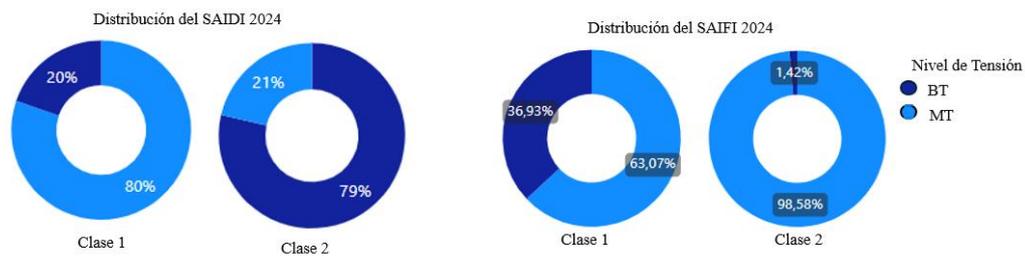


Numero de Incidencias por Material Averiado



Classification is imprecise, with excessive use of generic codes, empty or redundant fields. This issue is particularly critical in LV and SU incidents.

- **Impacto n Power Supply Quality Indicators.**
  - **Unplanned SAIDI and SAIFI** remain dominated by **MV** (>80%).
  - In **scheduled SAIDI**, **LV contributes nearly 80%**, reflecting the growing impact of scheduled actions on LV networks (digitalization and renewal).
  - This shift shows that including LV in indicators such as SAIDI can have a significant effect on the perception of power supply quality.



## 5. Conclusions

- The current quality of incident classification is not sufficient to ensure efficient management nor its future integration into intelligent systems.
- For planned incidents, it is proposed to use the work coding from the GDA system, thereby improving data traceability and accuracy.
- For unplanned incidents, it is necessary to revise and simplify the classification system by eliminating redundant codes, restricting typologies according to voltage level, and introducing improvements such as:
  - A field to identify provisional repairs.
  - Specific typologies for automatic reclosures (<3 minutes).
  - Preservation of the original type when reclassified as Force Majeure.
- The analysis of SAIDI and SAIFI shows that medium voltage (MV) continues to be the level with the greatest impact on power supply quality (in terms of duration and frequency of unplanned outages). However, low voltage (LV) now has a significant share of planned interruptions, with a growing contribution that should be considered in the future design of quality indicators.

## 6. References

- [1] Government of Spain, Royal Decree 1955/2000, of December 1, regulating the activities of transmission, distribution, commercialization, supply, and authorization procedures for electricity facilities, 2000. [Online]. Available at: <https://www.boe.es/eli/es/rd/2000/12/01/1955/con> [Accessed: March 15, 2025].
- [2] Government of Spain, Order ECO/797/2002, of March 22, establishing the procedures for measurement and control of power supply continuity, 2002. [Online]. Available at: <https://www.boe.es/eli/es/o/2002/03/22/eco797> [Accessed: March 15, 2025].



## *Índice de la memoria*

<b>Capítulo 1. Introducción.....</b>	<b>10</b>
1.1 Motivación del proyecto .....	11
<b>Capítulo 2. Investigación Previa. Estado de la Cuestión en UFD.....</b>	<b>14</b>
2.1 Investigación Previa.....	14
Calidad de Servicio.....	14
Definición de Incidencia.....	15
Indicadores de Calidad Individual y Calidad Zonal .....	16
2.2 Estado de la Cuestión en UFD.....	22
El Sistema de Gestión de Incidencias (SGI) y sus Fuentes de Información .....	23
Estructura y contenido de la información registrada en el SGI.....	25
Integración y Almacenamiento de la Información. ....	31
Uso de la Información y Aplicaciones Prácticas. ....	32
<b>Capítulo 3. Definición del Trabajo .....</b>	<b>37</b>
3.1 Justificación .....	37
3.2 Objetivos.....	37
3.3 Metodología.....	39
3.4 Herramientas y entorno tecnológico .....	41
3.5 Estimación Económica y Planificación .....	42
<b>Capítulo 4. Desarrollo del Proyecto.....</b>	<b>45</b>
4.1 Preparación y compresión del entorno del trabajo .....	45
4.2 Acceso, limpieza y estructuración de los datos .....	46
4.3 Análisis del comportamiento y evolución del tipo de incidencias.....	48
4.4 Análisis de la tipificación de las Incidencias .....	49
Tipificación de las Actuaciones Programadas.....	50
Tipificación de las incidencias imprevistas .....	50
4.5 Comparación de indicadores de calidad de suministro.....	51
4.6 Desarrollo de herramientas de visualización de datos. ....	53
Visualización de la evolución de la tipificación de las incidencias.....	53
Visualización de indicadores SAIDI y SAIFI .....	54
<b>Capítulo 5. Análisis de Resultados.....</b>	<b>57</b>

5.1	Evolución del Número de Incidencias por Nivel de Tensión (2022-2024) y Tipo.	57
5.2	Análisis de la Tipificación de Incidencias.	58
5.2.1	Tipificación de las Incidencias Programadas	58
5.2.2	Tipificación de las Incidencias Imprevistas	62
5.3	Comparación de indicadores de calidad	67
5.4	Desarrollo de Herramientas de Visualización	68
	Visualización de la evolución de la tipificación de incidencias	68
	Visualización de indicadores SAIDI y SAIFI	74
<b>Capítulo 6.</b>	<b>Conclusiones y Trabajos Futuros</b>	<b>75</b>
6.1.	Evolución del número de incidencias	75
6.2.	Tipificación de incidencias programadas	75
6.2.1	Principales limitaciones en la codificación actual	75
6.2.2	Propuesta de mejora	76
6.3	Tipificación de incidencias Imprevistas.	76
6.3.1	Principales limitaciones en la codificación actual de incidencias imprevistas...	76
6.3.2	Propuestas de Mejora	76
6.4	Conclusiones sobre los Indicadores de Calidad (SAIDI y SAIFI)	77
<b>Capítulo 7.</b>	<b>Bibliografía</b>	<b>79</b>
<b>ANEXO 1:</b>	<b>Estructura y Contenido de las Bases de Datos</b>	<b>1</b>
ANEXO 1.1	Base de datos de Incidencias	1
ANEXO 1.2	Descripción de Variable “TIPO”	4
ANEXO 1.3	Descripción de Variable “Ins_Origen”	5
<b>ANEXO II:</b>	<b>Graficas y Resultados Parciales.</b>	<b>1</b>
	Evolución de las Incidencias por Tipología y Nivel de Tensión (2022-2024).	2
	Evolución de las Incidencias de AT por Tipología (2022-2024).	3
	Evolución de las Incidencias de MT por Tipología (2022-2024).	3
	Evolución de las Incidencias de BT por Tipología (2022-2024).	4
	Evolución de las Incidencias SU (2022-2024).	5
	Tipificación de las Incidencias Programadas	5
	Evolución del Número de incidencias Programadas (2022-2024 y primer trimestre de 2025).	5

Evolución de las Incidencias Programadas de MT por Tipo de Incidencia.(2022-2024 y primer trimestre de 2025). .....	6
Evolución del Número de Incidencias Programadas de BT por tipo de Incidencia.(2022-2024 y primer trimestre de 2025). .....	7
Evolución Incidencias Programadas de MT y BT por Instalación Origen.(2022-2024 y primer trimestre de 2025). .....	7
Análisis de los campos de texto libre en las incidencias de MT y BT del año 2024 y del primer trimestre de 2025. ....	8
Análisis de la codificación de trabajos en el sistema GDA (2022–2024) .....	9
Distribución de trabajos por Tipo de Actuación.....	9
Distribución de Trabajos por Clase de Actividad.....	11
Tipificación de las Incidencias Imprevistas.....	13
Evolución del Número de incidencias Imprevistas tipo SU (Suministro Único) (2022-2024 y primer trimestre de 2025). .....	14
Evolución del Número de incidencias Imprevistas de BT (2022-2024 y primer trimestre de 2025). .....	21
Evolución del Número de incidencias Imprevistas de MT (2022-2024 y primer trimestre de 2025). .....	28
Distribución del SAIDI y SAIFI por nivel de Tensión (MT Y BT) (2022-2025) .....	37
SAIDI .....	39
SAIFI .....	41
<b>ANEXO III: Código Empleado para el Desarrollo del TFM .....</b>	<b>1</b>
Análisis Explotario de los Datos de Incidencias .....	1
Cálculo de SAIDI y SAIFI 2024-2025 por clase (programada e imprevista) .....	127

## *Índice de figuras*

Figura 1: Sistema de Explotación de Datos. Elaboración propia .....	12
Figura 2: Data Exploitation System. Author's own work .....	17
Figura 3. Mapa de Sistemas de UFD relacionados con SGI. Elaboración propia.....	23
Figura 4. SGI Vista General de las incidencias ocurridas en un periodo de tiempo. ....	26
Figura 5. SGI: Detalle de incidencia. ....	29
Figura 6. SGI Detalle de información asociada a trabajos programados. ....	30
Figura 7. SGI: Campo Observaciones .....	31
Figura 8. Seguimiento mensual de calidad de suministro: TIEPI y NIEPI mensual y acumulado.....	34
Figura 9. Seguimiento mensual de calidad de suministro: evolución del TIEPI diario. ....	34
Figura 10: Seguimiento mensual de calidad de suministro: desglose del TIEPI total por CC.AA.....	35
Figura 11: Seguimiento mensual de calidad de suministro:evolución del TIEPI total acumulado por Área Energética .....	36
Figura 12: Panel de evolución de la tipificación de incidencias clase 1 (programadas) .....	70
Figura 13: Panel de evolución de la tipificación de incidencias clase 2 (imprevistas). ....	71
Figura 14: Panel de evolución de la tipificación de incidencias clase 3 (régimen especial, sin interrupción). ....	72
Figura 15: Panel de Navegación Jerárquica de Tipificación .....	73

## *Índice de tablas*

Tabla 1. Límites calidad suministro individual para puntos de suministro conectados en MT (de 1 a 36kV)[1] .....	17
Tabla 2. Límites calidad de suministro individual para puntos de suministro conectados en BT ( $\leq 1\text{kV}$ ) [1].....	17
Tabla 3:Límites calidad de suministro zonal. [2] .....	18
Tabla 4. Correspondencia entre Causa e Incidencia según OREN ECO797/2002 y Campo Motivo SGI.....	30
Tabla 5: Mapa de calor: Instalación Origen / Tipo de Incidencias Programadas MT. Año 2022 .....	61
Tabla 6: Mapa de calor: Instalación Origen / Tipo de Incidencias Programadas BT. Año 2022 .....	61
Tabla 7: Resumen Tipificación Incidencias Imprevistas Nivel SU.....	64
Tabla 8: Resumen Tipificación Incendias Imprevistas Nivel BT.....	65
Tabla 9: Resumen Tipificación Incendias Imprevistas Nivel MT.....	67
Tabla 10: Evolución del Número de Incidencias. Elaboración Propia.....	A2-1

## Índice de gráficos

Gráfico 1: Electricidad: SAIDI no planificado, sin eventos excepcionales (min/cliente)[4]. .....	21
Gráfico 2: Electricidad: SAIFI no planificados sin eventos excepcionales (interrupciones/cliente) [4] .....	21
Gráfico 3: Diagrama de Gantt de la Planificación del TFM.....	44
Gráfico 4: Número de Incidencias por Año (2022), Nivel de Tensión (A: AT, B:BT, M:MT y S:SU) y Clase 1: Programadas. 2: Imprevistas). Elaboración propia .....	57
Gráfico 5: Tipología de Incidencias Programadas en BT.....	59
Gráfico 6: Tipología de Incidencias Programadas en MT.....	59
Gráfico 7: Instalación Origen de Incidencias Programadas en BT .....	60
Gráfico 8: Instalación Origen de Incidencias Programadas en MT.....	60
Gráfico 9:Número de Incidencias por Nivel de Tensión (2022-2024). (A: AT, B:BT, M:MT y S:SU). Elaboración propia.....	A2-1
Gráfico 10: Incidencias por Tipología (1: Programadas 2: Imprevistas) y Niveles de Tensión (A: AT, B:BT, M:MT y S:SU) Elaboración propia.....	A2-2
Gráfico 11: Incidencias de AT por Tipología (Programadas e Imprevistas). Elaboración propia.....	A2-3
Gráfico 12: Incidencias de MT por Tipología (Programadas e Imprevistas) Elaboración propia.....	A2-4
Gráfico 13: Incidencias en BT por Tipología (Programadas e Imprevistas) Elaboración propia.....	A2-4
Gráfico 14: Incidencias SU por Tipología (Programadas e Imprevistas) Elaboración propia. .....	A2-5
Gráfico 15: Evolución del número de incidencias programadas por nivel de tensión y año. Elaboración propia.....	A2-6
Gráfico 16: Evolución del número de incidencias programadas de MT por tipo de incidencia. Elaboración propia.....	A2-7
Gráfico 17: Evolución del número de incidencias programadas de BT por tipo de incidencia. Elaboración propia.....	A2-7
Gráfico 18: Evolución del número de incidencias programadas de MT y BT por instalación de origen en el periodo estudiado. Elaboración propia. ....	A2-8
Gráfico 19: Palabras con mayor frecuencia en el campo “DESCRIPCIÓN y OBSERVACIONES” para incidencias programadas en MT y BT, durante 2024 y primer trimestre de 2025. ....	A2-9
Gráfico 20: Distribución de trabajos por Tipo de Actuación en GDA (2022–2024). Elaboración propia.....	A2-10
Gráfico 21: Distribución de trabajos en AT por Tipo de Actuación en GDA (2022–2024). Elaboración propia.....	A2-10
Gráfico 22: Distribución de trabajos en MT por Tipo de Actuación en GDA (2022–2024). Elaboración propia.....	A2-11
Gráfico 23: Distribución de trabajos en BT por Tipo de Actuación en GDA (2022–2024). Elaboración propia.....	A2-11

Gráfico 24: Distribución trabajos por Clase Actividad en GDA (2022–2024). Elaboración propia.....	A2-12
Gráfico 25: Distribución trabajos AT por Clase Actividad en GDA (2022–2024).....	A2-12
Gráfico 26: Distribución trabajos MT por Clase Actividad en GDA (2022–2024).....	A2-13
Gráfico 27: Distribución trabajos BT por Clase Actividad en GDA (2022–2024).....	A2-13
Gráfico 28: Incidencias Imprevistas para Suministros Únicos por Año y Tipo.....	A2-14
Gráfico 29: Distribución del campo INS_ORIGEN para incidencias Tipo 303, Nivel SU, por año. ....	A2-15
Gráfico 30: Distribución del campo MAT_AVERIADO para incidencias Tipo 303, Nivel SU, por año. ....	A2-16
Gráfico 31: Distribución del campo INS_ORIGEN para incidencias Tipo 317, Nivel SU, por año .....	A2-16
Gráfico 32: Distribución del campo INS_ORIGEN para incidencias Tipo 317, Nivel SU (2022- marzo 2025) .....	A2-17
Gráfico 33: Distribución del campo MAT_AVERIADO para incidencias Tipo 317, Nivel SU, por año. ....	A2-17
Gráfico 34: Distribución del campo INS_ORIGEN para incidencias Tipo 416, Nivel SU, por año. ....	A2-18
Gráfico 35: Distribución del campo INS_ORIGEN para incidencias Tipo 317, Nivel SU (2022- marzo 2025) .....	A2-18
Gráfico 36: Distribución del campo MAT_AVERIADO para incidencias Tipo 416, Nivel SU, por año. ....	A2-19
Gráfico 37: Distribución del campo INS_ORIGEN para incidencias Tipo 417, Nivel SU (2022- marzo 2025) .....	A2-19
Gráfico 38: Distribución del campo MAT_AVERIADO para incidencias Tipo 417, Nivel SU, periodo (2022-marzo 2025).....	A2-20
Gráfico 39: Incidencias Imprevistas de BT por Año y Tipo .....	A2-21
Gráfico 40: Distribución del campo Ins_Origen para incidencias Tipo 303 de BT, por año.....	A2-22
Gráfico 41: Distribución del campo MAT_AVERIADO para incidencias Tipo 303 de BT, por año. ....	A2-23
Gráfico 42: Distribución del campo Ins_Origen para incidencias Tipo 301 de BT por año. ....	A2-24
Gráfico 43: Distribución del campo MAT_AVERIADO para incidencias Tipo 301 de BT, por año. ....	A2-24
Gráfico 44: Distribución del campo Ins_Origen para incidencias Tipo 416 de BT, por año. ....	A2-25
Gráfico 45: Distribución del campo MAT_AVERIADO para incidencias Tipo 416 de BT, por año. ....	A2-26
Gráfico 46: Distribución del campo Ins_Origen para incidencias Tipo 202 de BT, por año. ....	A2-26
Gráfico 47: Distribución del campo MAT_AVERIADO para incidencias Tipo 202 de BT, por año .....	A2-27
Gráfico 48: Incidencias Imprevistas de MT por Año y por Tipo .....	A2-28
Gráfico 49: Incidencias Imprevistas con interrupciones de mas de 3 minutos en MT por Tipo y Año .....	A2-29

Gráfico 50: Distribución del campo Ins_Origen para incidencias Tipo 301 de MT, por año .....	A2-30
Gráfico 51: Distribución del campo MAT_AVERIADO para incidencias Tipo 301 de MT, por año. ....	A2-31
Gráfico 52: Distribución del campo Ins_Origen para incidencias Tipo 303 de MT, por año .....	A2-32
Gráfico 53: Distribución del campo MAT_AVERIADO para incidencias Tipo 303 de MT, por año. ....	A2-32
Gráfico 54: Distribución del campo Ins_Origen para incidencias Tipo 304 de MT, por año .....	A2-33
Gráfico 55: Distribución del campo MAT_AVERIADO para incidencias Tipo 304 de MT, por año .....	A2-33
Gráfico 56: Distribución del campo Ins_Origen para incidencias Tipo 406 de MT, por año .....	A2-34
Gráfico 57: Distribución del campo MAT_AVERIADO para incidencias Tipo 406 de MT, por año .....	A2-35
Gráfico 58: Distribución del campo Ins_Origen para incidencias Tipo 202 de MT, por año .....	35
Gráfico 59: Distribución del campo MAT_AVERIADO para incidencias Tipo 202 de MT, por año. ....	A2-36
Gráfico 60: Frecuencia de palabras repetidas en Observaciones en las Incidencias Imprevistas de MT Tipo 202-Árboles .....	A2-37
Gráfico 61: Evolución del SAIDI en minutos, por clase 1: Incidencias Programadas, 2: Incidencias Imprevistas .....	A2-38
Gráfico 62: Distribución SAIDI por Clase y Nivel de Tensión. AÑO 2022 .....	A2-39
Gráfico 63: Distribución SAIDI por Clase y Nivel de Tensión. AÑO 2023 .....	A2-39
Gráfico 64: Distribución SAIDI por Clase y Nivel de Tensión. AÑO 2024 .....	A2-40
Gráfico 65: Distribución SAIDI por Clase y Nivel de Tensión. AÑO 2025 .....	A2-40
Gráfico 66: Distribución SAIFI por Clase y Nivel de Tensión. AÑO 2022 .....	A2-42
Gráfico 67: Distribución SAIFI por Clase y Nivel de Tensión. AÑO 2023 .....	A2-42
Gráfico 68: Distribución SAIFI por Clase y Nivel de Tensión. AÑO 2024 .....	A2-42
Gráfico 69: Distribución SAIFI por Clase y Nivel de Tensión. AÑO 2025 .....	A2-43

## *Lista de Acrónimos*

<b>Acrónimo</b>	<b>Significado</b>
ADMS	Advanced Distribution Management System
ASIDI	Average System Interruption Duration Index
ASIFI	Average System Interruption Frequency Index
AT	Alta Tensión (mayor o igual que 36kV)
AWS	Amazon Web Services
BT	Baja Tensión (menor o igual a 1kV)
CEER	Council of European Energy Regulators
COR	Centro de Operación de Red
GDA	Sistema de Gestión de Activos
GIS	Geographic Information System
IGEA	Geographic Enterprise Application
MT	Media Tensión (entre 1kV y 36kV)
NIEPI	Número de Interrupciones Equivalente a la Potencia Instalada
RD	Real Decreto
SAIDI	System Average Interruption Duration Index
SAIFI	System Average Interruption Frequency Index
SCADA	Supervisory Control and Data Acquisition
SIG	Sistema de Gestión de Incidencias
SU	Suministro Único
TFM	Trabajo Fin de Máster
TIEPI	Tiempo de Interrupción Equivalente a la Potencia Instalada
UFD	UFD Distribución de Electricidad

## Capítulo 1. INTRODUCCIÓN

La distribución de energía eléctrica en España es una actividad regulada en la que diferentes normativas exigen a las empresas distribuidoras registrar todas las incidencias que afectan al suministro de los consumidores conectados a sus redes. En particular, el Real Decreto 1955/2000 establece en su artículo 104 que «el distribuidor deberá disponer de un sistema de registro de incidencias de acuerdo con el procedimiento de medida y control de la continuidad del suministro y de la calidad del producto» [1], de modo que pueda registrar las incidencias en la continuidad de suministro para cada uno de los consumidores conectados a sus redes. Asimismo, en su artículo 108, el mismo Real Decreto dispone la obligatoriedad de «elaborar anualmente información detallada de los valores de los índices de calidad para el ámbito de las distintas zonas de cada provincia donde ejercen su actividad» [1]. Estos índices deben desglosarse por provincia, zona (urbana, semiurbana, rural concentrada y rural dispersa), nivel de tensión y tipología de interrupción (programada o imprevista), distinguiendo, en las imprevistas entre causas de terceros, fuerza mayor y propias de la distribución.

Por su parte, la Orden ECO/797/2002 establece «los criterios y la metodología a seguir para la recogida y tratamiento de los datos de la continuidad del suministro, incluyendo los necesarios para la elaboración de los índices de calidad zonal TIEPI, percentil 80 del TIEPI y NIEPI.» [2]

La información sobre las incidencias que afectan a los consumidores conectados a la red de distribución se obtiene fundamentalmente de las siguientes fuentes:

- **Sistemas de telecontrol (SCADA):** *Supervisory Control and Data Acquisition*, que capturan de forma automática las maniobras de apertura y cierre de los elementos de corte instalados en la red y telecontrolados desde el Centro de Operación de Red (COR). Si alguna de estas maniobras implica una pérdida de suministro, genera una interrupción que debe quedar registrada en el Sistema de Gestión de Incidencias (SGI).

- **Maniobras manuales:** Para los elementos de corte de la red que no están telecontrolados desde el COR, el personal de campo comunica la maniobra al operador del COR en tiempo real y este a su vez debe introducirla en el sistema. Tanto la maniobra como cualquier interrupción derivada de dicha maniobra queda igualmente reflejada en el SGI.
- **Avisos de falta de suministro receptionados por el Centro de Atención al Cliente:** Cuando un cliente llama comunicando que no tiene suministro eléctrico, tras comprobar que no se trata de un corte por impago, una avería en la instalación del consumidor o una interrupción asociada a instalaciones de tensión superior, se registra la incidencia en el SGI contabilizándose la duración de la interrupción desde el momento en que esta se comunica.

Toda esta información se integra en el SGI, desarrollado por UFD para almacenar, de forma auditable y homogénea, tanto los datos de continuidad de suministro (intervalos de tiempo de las interrupciones, potencia afectada y número de clientes afectados) como los datos complementarios (instalación afectada, nivel de tensión, tipo de incidencia, motivo, material averiado, zona geográfica, descripción y observaciones de la incidencia).

## **1.1 MOTIVACIÓN DEL PROYECTO**

La gran cantidad de información registrada en el SGI constituye un activo estratégico para UFD Distribución de Electricidad. No solo garantiza el cumplimiento normativo, el SGI es una aplicación informática auditable que «permite a un tercero independiente verificar todo el proceso de captación, modificación, transmisión, soporte y tratamiento de la información» [2], sino que también ofrece datos de calidad que una vez organizados y analizados, facilitan la toma de decisiones basadas en datos.

El presente Trabajo de Fin de Máster, titulado “Gestión y optimización de la información de incidencias de suministro eléctrico en UFD Distribución de Electricidad”, tiene como objetivo principal organizar y analizar los datos recogidos y almacenados en el SGI para maximizar su utilidad operativa. El análisis se estructura en torno a los siguientes ejes:

- **Evolución de las interrupciones:** estudiar la evolución de las incidencias en los últimos años, considerando tanto la potencia afectada como el número de clientes, distinguiendo entre interrupciones programadas e imprevistas, y buscando posibles relaciones con la tipificación utilizada actualmente.
- **Evaluación de la tipificación:** revisar y analizar la clasificación actual de las incidencias registradas en el SGI (motivos, causas y materiales averiados) y proponer líneas de mejora que permitan una mayor coherencia, trazabilidad y capacidad de explotación.
- **Comparación de indicadores:** comparar los indicadores tradicionales basados en potencia (TIEPI y NIEPI) con aquellos centrados en el número de clientes (SAIDI y SAIFI), con el objetivo de evaluar el impacto que tendría un cambio de enfoque en la priorización de inversiones y en los planes de mejora de la red.
- **Desarrollo de herramientas interactivas:** diseñar una solución de explotación dinámica que permita analizar la evolución y calidad de la tipificación mediante consultas y visualizaciones interactivas, utilizando herramientas como SQL, Python y Power BI, adaptadas a las necesidades de las distintas áreas de la compañía.

Al perfeccionar la calidad de la información, la accesibilidad de los reportes y la explotación analítica de las incidencias, este proyecto contribuirá al proceso de transformación digital de UFD, fortaleciendo la eficiencia operativa y en última instancia contribuyendo a una mejora de la calidad de suministro.



## **Capítulo 2. INVESTIGACIÓN PREVIA. ESTADO DE LA CUESTIÓN EN UFD**

### **2.1 INVESTIGACIÓN PREVIA**

#### **CALIDAD DE SERVICIO**

«La calidad de servicio es el conjunto de características, técnicas y comerciales, inherentes al suministro eléctrico exigibles por los sujetos, consumidores y por los órganos competentes de la Administración»[1]

La calidad de servicio en el ámbito eléctrico se compone de tres aspectos fundamentales:

1. **Continuidad de Suministro:** Se refiere a la capacidad de la red para mantener el suministro eléctrico sin interrupciones. En este apartado se contabilizan tanto el número de interrupciones como la duración de las mismas.
2. **Calidad del Producto:** Se refiere a las características de la onda de tensión que llega al consumidor final. El valor de la tensión debe mantenerse dentro de los límites reglamentarios ( $V_{nominal} \pm 7\%$ ), evitando tanto las subtensiones como las sobretensiones, y garantizando una forma de onda senoidal y una frecuencia estable de 50 Hz. En este apartado se incluyen las interrupciones de servicio y huecos de tensión de duración inferior a 3 minutos. (Art.102 del RD 1955/2000).
3. **Calidad en la Atención a los Consumidores:** Se refiere al asesoramiento al consumidor en la contratación, medición de la energía consumida y facturación a la Comercializadora, así como la conexión en plazo de nuevos puntos de suministro, la gestión de reclamación y todos los servicios relacionados con la atención al cliente. (Art. 103 del RD 1955/2000).

En el presente TFM, se analiza la información recogida en el SGI que está asociada a la continuidad de suministro eléctrico.

## **DEFINICIÓN DE INCIDENCIA.**

«**Incidencia:** Es todo evento, y sus consecuencias asociadas, originado en los sistemas de Generación, Transporte o Distribución, que sea causa de una o varias interrupciones imprevistas de suministro con instalaciones afectadas relacionadas temporal y eléctricamente» [2]

Asimismo, en el art. 100 del RD 1955/2000 se define como **interrupción de alimentación**, «la condición en que la tensión del punto de suministro no supera el 10 por 100 de la tensión declarada» [1]

Con base a estas definiciones, las empresas distribuidoras deben registrar todos los eventos que se producen en sus redes, independientemente del nivel de tensión o de la instalación de origen, siempre que estos generen interrupciones en el suministro a los consumidores conectados a la red. Dichos eventos se agrupan en una única incidencia asociada a la instalación eléctrica de mayor nivel, que puede ser:

1. Subestación
2. Circuito de Media Tensión (MT):  $1\text{ kV} < V \leq 36\text{ kV}$
3. Centro de Transformación (CT)
4. Transformador MT/BT
5. Salida de BT
6. Suministro único (SU)

A modo de ejemplo, si se produce una interrupción debido a la desconexión de un transformador de una subestación, la incidencia afecta a todos los clientes conectados a los circuitos de MT que se alimentan desde ese transformador. En este caso, se asigna a cada cliente el tiempo exacto durante el cual estuvo sin suministro, teniendo en cuenta que, durante las maniobras de reposición del servicio, es posible que cada circuito se restablezca en momentos distintos o que, incluso, cada CT de un mismo circuito de MT, se vea afectado durante intervalos de tiempo distintos.

Así, el sistema de registro y almacenamiento debe cumplir con todo lo dispuesto en el artículo 5 de la Orden ECO797 en tanto que «deberá reflejar exactamente las instalaciones afectadas por la incidencia, asociadas a los intervalos horarios de interrupción, y las maniobras que modifiquen el estado de afectación de las instalaciones»[2]

### ***2.1.1.1 Tipificación de las Interrupciones de Suministro.***

Para realizar determinados trabajos de conservación y de mantenimiento sobre las instalaciones eléctricas, así como para la conexión de nuevas instalaciones, puede ser necesario cortar el suministro eléctrico. Consecuentemente, todos los consumidores conectados a estas redes sufrirán una **Interrupción Programada** siempre que se cumpla lo dispuesto en el apartado 3 del artículo 101 del RD 1955/2000:

- «Autorización del órgano competente de Energía de la Comunidad Autónoma (CCAA) con una antelación mínima de 72 horas.
- Comunicación a los consumidores afectados con una antelación mínima de 24 horas, diferenciando:
  - Consumidores conectados en redes de tensión superior a 1kV: comunicación individualizada.
  - Resto de Consumidores: Mediante carteles anunciadores situados en lugares visibles y mediante dos medios de comunicación de mayor difusión de la provincia del suministro» [1].

El resto de las interrupciones y aquellas que, aun siendo programadas, no cumplan los criterios citados previamente, se tipifican como **Interrupciones Imprevistas**.

## **INDICADORES DE CALIDAD INDIVIDUAL Y CALIDAD ZONAL**

### ***2.1.1.2 Indicadores de la calidad de suministro individual en España.***

Según lo establecido en el artículo 104 del RD 1955/2000, para cada consumidor conectado a la red de distribución, no se puede superar ni el número de **interrupciones imprevistas** mayores de tres minutos ni el tiempo de estas para cada año natural. Los límites se establecen teniendo en cuenta tanto la tensión de la alimentación del consumidor como el municipio en

el que se encuentren ubicados. En la Tabla 1 y Tabla 2 se muestran los límites establecidos para los consumidores conectados a la red de MT y de BT, respectivamente.

	<b>Número de horas</b>	<b>Número de interrupciones</b>
<b>Zona urbana</b>	3,5	7
<b>Zona semiurbana</b>	7	11
<b>Zona rural concentrada</b>	11	14
<b>Zona rural dispersa</b>	15	19

Tabla 1. Límites calidad suministro individual para puntos de suministro conectados en MT (de 1 a 36kV) [1]

	<b>Número de horas</b>	<b>Número de interrupciones</b>
<b>Zona urbana</b>	5	10
<b>Zona semiurbana</b>	9	13
<b>Zona rural concentrada</b>	14	16
<b>Zona rural dispersa</b>	19	22

Tabla 2. Límites calidad de suministro individual para puntos de suministro conectados en BT ( $\leq 1kV$ ) [1]  
La clasificación de las zonas de distribución se establece en la Orden ECO/797/2002, y son:

- Zona Urbana: Municipios con más de 20.000 puntos de suministro y las capitales de provincia con independencia del número de puntos de suministro.
- Zona Semiurbana: Municipios con más de 2.000 y menos de 20.000 puntos de suministro.
- Zona Rural Concentrada: Municipios con más de 200 y menos de 2.000 puntos de suministro.
- Zona Rural Dispersa: Municipios con menos de 200 puntos de suministro y aquellos ubicados fuera de los núcleos de población.

El incumplimiento de alguno de los valores fijado en las tablas anteriores da lugar a descuentos en la facturación de los consumidores, según lo establecido en el artículo 105 del RD 1955/2000.

### ***2.1.1.3 Indicadores de la calidad de suministro zonal en España.***

Con objeto de medir la calidad de suministro de un distribuidor en una zona, solo se tienen en cuenta las interrupciones imprevistas de una duración superior a 3 minutos. Las interrupciones programadas que estén debidamente documentadas conforme a la normativa

(véase Apratdo 3.1.2.1) quedan excluidas del cómputo. Se definen los siguientes indicadores:

- TIEPI: Es el tiempo de interrupción equivalente a la potencia instalada en MT.
- NIEPI: Es el número de interrupciones equivalente a la potencia instalada en MT.
- PERCENTIL 80% del TIEPI: Es el valor del TIEPI que no es superado por el 80% de los municipios de una provincia.

Para el cálculo del TIEPI y el NIEPI se utilizan las siguientes fórmulas:

$$\llcorner TIEPI = \frac{\sum_{i=1}^k (PI_i \times H_i)}{\sum PI} \text{ y } NIEPI = \frac{\sum_{i=1}^k (PI_i)}{\sum PI} \text{ siendo:}$$

$\sum PI$  = la suma de la potencia instalada en los CTs de UFD más la potencia contratada en los CTs particulares.

$PI_i$  = potencia instalada en los CT de UFD y la potencia contratada en los CTS particulares afectados por la interrupción “i”. Se mide en kVA.

$H_i$  = tiempo de interrupción del suministro en la interrupción “i”. Se mide en horas.

$k$  = número total de interrupciones imprevistas de duración superior a 3 minutos.»[2]

Las empresas distribuidoras, debe mantener los niveles de calidad zonal establecidos en el artículo 106 del RD 1955/2000, cuyos límites se recogen en la Tabla 3

	<b>Tiepi (Horas)</b>	<b>Percentil 80 del TIEPI (Horas)</b>	<b>NIEPI (Número)</b>
<b>Zona urbana</b>	1,5	2,5	3
<b>Zona semiurbana</b>	3,5	5	5
<b>Zona rural concentrada</b>	6	10	8
<b>Zona rural dispersa</b>	9	15	12

*Tabla 3: Límites calidad de suministro zonal. [2]*

Cuando alguno de los valores límite establecidos en el artículo 106 del RD 1955/2000 [1] se supera de forma reiterada en un municipio, la Administración competente puede requerir a

la empresa distribuidora la presentación de un plan de mejora de la calidad del suministro, destinado a corregir las deficiencias detectadas en la red, según lo dispuesto en el artículo 107 del mismo Real Decreto. La no presentación o no ejecución de este plan puede ser considerada como una infracción grave, de acuerdo con el artículo 61.4 de la Ley 54/1997, de 27 de noviembre, del Sector Eléctrico [5]. Este tipo de infracción puede conllevar sanciones económicas de hasta 6.000.000 de euros.

Aunque actualmente en España la calidad zonal de suministro eléctrico se mide utilizando el TIEPI y al NIEPI, el apartado 2 del artículo 98 del RD 1955/2000 establece que «el Ministerio de Economía podrá establecer otros indicadores adicionales y podrá revisar los límites de los valores establecidos» [1].

#### ***2.1.1.1 Otros indicadores de la calidad de suministro.***

Como se ha visto, para medir la calidad de suministro eléctrico es necesario disponer de indicadores específicos. Aunque cada país cuenta con su propia normativa, la mayoría de los sistemas eléctricos utilizan indicadores de calidad de suministro zonal comunes, como el SAIDI y SAIFI.

#### **SAIDI (System Average Interruption Duration Index)**

El SAIDI, mide el tiempo promedio anual durante el cual un consumidor se encuentra sin suministro eléctrico debido a interrupciones.[3] Se mide en minutos o horas por cliente y año y se calcula según la siguiente fórmula:

$$SAIDI = \frac{\sum_{i=1}^k (H_i \times N^{\circ}Consumidores_i)}{\sum N^{\circ}Consumidores}$$

Donde:

- $H_i$  = tiempo de interrupción del suministro en la interrupción “i”. En horas.
- $N^{\circ}Consumidores_i$  = número de consumidores afectados por la interrupción “i”

- $\sum N^{\circ}Consumidores$  = número total de consumidores conectados a la red de distribución.
- $k$  = número total de interrupciones.

### **SAIFI (System Average Interruption Frequency Index):**

El SAIFI, mide el promedio anual de número de interrupciones que sufre un consumidor conectado a la red de distribución.[3] Se mide en número de interrupciones por cliente y año y se calcula según la siguiente fórmula:

$$SAIFI = \frac{\sum_{i=1}^k (N^{\circ}Consumidores_i)}{\sum N^{\circ}Consumidores}$$

Donde:

- $N^{\circ}Consumidores_i$  = número de consumidores afectados por la interrupción “i”
- $\sum N^{\circ}Consumidores$  = número total de consumidores conectados a la red de distribución.
- $k$  = número total de interrupciones.

### **CAIDI (Customer Average Interruption Duration Index)**

El CAIDI mide la duración media de las interrupciones que realmente se han producido, es decir, el tiempo promedio que dura cada interrupción sufrida por los consumidores afectados. Se mide en minutos o horas por interrupción y se calcula como:

$$CAIDI = \frac{SAIDI}{SAIFI}$$

El SAIDI mide el tiempo total de interrupción de suministro, repartido entre todos los clientes, hayan sufrido la interrupción o no. El SAIFI mide cuántas interrupciones de suministro sufre de media un cliente. El CAIDI mide, cuando hay una interrupción de suministro, cuánto dura de media.

El uso de estos indicadores, nos permiten comparar el desempeño de las redes de distribución de distintos países como se puede comprobar en los siguientes figuras extraídas del Informe 6.1 sobre la Continuidad del Suministro de Electricidad y Gas, Consejo de Reguladores Europeos de la Energía [4].

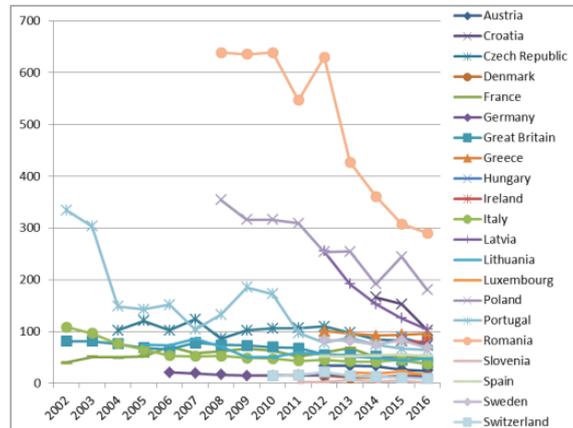


Gráfico 1: Electricidad: SAIDI no planificado, sin eventos excepcionales (min/cliente)[4].

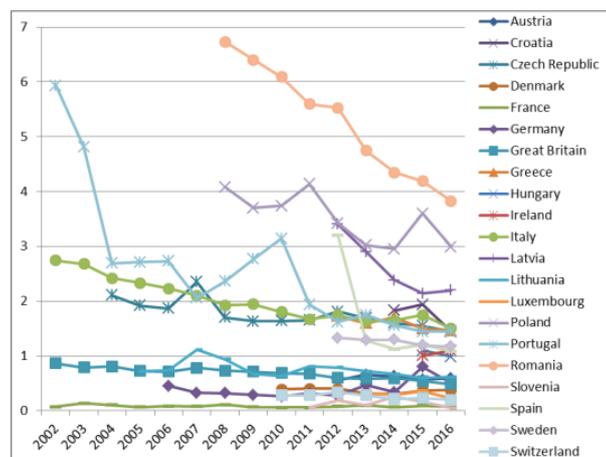


Gráfico 2: Electricidad: SAIFI no planificados sin eventos excepcionales (interrupciones/cliente) [4]

Los indicadores TIEPI y NIEPI, tienen su equivalente funcional en los indicadores internacionales ASIDI (Average System Interruption Duration Index) y ASIFI (Average System Interruption Frequency Index). Ambos pares de indicadores evalúan la calidad del suministro desde una perspectiva técnica, ponderando las interrupciones en función de la potencia afectada y no del número de consumidores. Por el contrario, SAIDI y SAIFI evalúan la calidad del suministro desde una perspectiva comercial o de experiencia del

cliente, ya que ponderan las interrupciones en función del número de consumidores afectados, sin tener en cuenta la potencia instalada asociada.

## **2.2 ESTADO DE LA CUESTIÓN EN UFD**

Como paso previo al análisis, se ha estudiado el mapa de sistemas que están interrelacionados con el SGI y que intervienen en la recogida de datos de interrupción de suministro, tanto imprevistas como programadas, y en todos los niveles de tensión.

Estos sistemas se controlan y gestionan desde el Centro de Operación de Red (COR) de UFD, responsable de la operación de 116.000 kilómetros de líneas de Alta, Media y Baja Tensión, que dan servicio a 3,8 millones de puntos de suministro en las comunidades autónomas de Galicia, Castilla-La Mancha, Castilla y León y Madrid [6].

El COR opera las 24 horas los 365 días al año y dispone de un centro de respaldo preparado para activarse en caso de emergencia o contingencia operativa del COR principal.

Con el objetivo de comprender cómo se gestiona y consolida la información relativa a las incidencias de suministro eléctrico en UFD, se ha elaborado un esquema que representa los principales sistemas que intervienen en la captura, transmisión y procesamiento de dicha información. Estos sistemas, gestionados de forma centralizada desde el COR, permiten coordinar las actuaciones en la red, registrar las maniobras realizadas tanto en campo como de forma remota mediante telecontrol, y canalizar las llamadas y avisos hacia el Sistema de Gestión de Incidencias (SGI), que actúa como núcleo central del proceso. A continuación, se presenta un mapa conceptual que refleja esta interconexión y el flujo de datos entre los sistemas implicados.

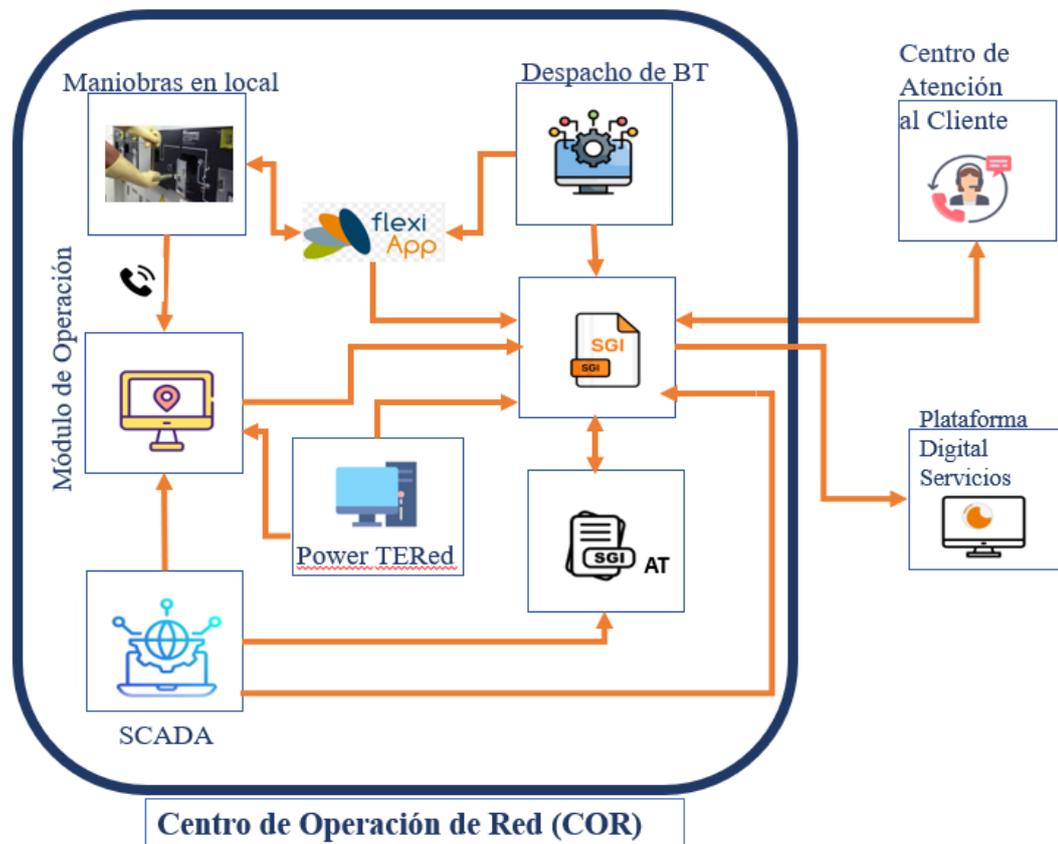


Figura 3. Mapa de Sistemas de UFD relacionados con SGI. Elaboración propia.

## EL SISTEMA DE GESTIÓN DE INCIDENCIAS (SGI) Y SUS FUENTES DE INFORMACIÓN

El Sistema de Gestión de Incidencias, SGI, es la herramienta utilizada por UFD para recoger y registrar la información relativa a las incidencias que se produzcan en su red de distribución, en cumplimiento de lo establecido en la Orden ECO/797/2002, concretamente en el Anexo “Procedimiento de medida y control de la continuidad del suministro eléctrico”, apartado 4, donde se indica que “las empresas distribuidoras deben disponer de sistemas que permitan recoger información sobre las incidencias que se produzcan en su red de distribución” [1].

El COR de UFD es el encargado de gestionar el SGI, así como de verificar y completar la información que este recibe desde otros sistemas, entre los que destacan:

- **SCADA (Supervisory Control and Data Acquisition)**

Envía automáticamente al SGI los registros de apertura y cierre de los elementos telecontrolados instalados en la red de distribución, siempre que estas maniobras estén asociadas a una interrupción o recuperación del suministro. Además de la maniobra, se transmite el instante de tiempo en el que esta se produce, lo que garantiza la precisión de los tiempos de interrupción registrados en la incidencia.

- **Módulo de Operación**

Esta herramienta, integrada en un sistema GIS (Geographic Information System), que permite representar gráficamente la red eléctrica sobre un mapa y gestionar su topología. Está basado en la base de datos de instalaciones, cuya actualización y mantenimiento siguen un procedimiento riguroso que garantiza que el modelo refleja fielmente la red real.

El módulo de operación permite conocer en todo momento el estado de los elementos de maniobra, tanto telecontrolados como no telecontrolados. La operativa habitual consiste en que el COR se comunica telefónicamente con los equipos de operación local en campo para ordenar la ejecución de la maniobra. Mientras el operador local realiza la apertura o cierre del elemento en campo, el técnico del COR registra la acción en el módulo de operación, lo que permite reflejar en el sistema tanto el cambio de estado como el tiempo exacto de ejecución. Así se garantiza que los tiempos registrados en el SGI coinciden con los reales.

- **Centro de Atención al cliente.**

Cuando el cliente contacta con el Centro de Atención al Cliente de UFD para notificar una falta de suministro eléctrico, el operador comprueba que dicha falta no esté asociada a una incidencia en curso, ni se deba a un corte por impago. Además, mediante cuestionario, se determina si podría tratarse de una avería en la instalación interior del cliente, en cuyo caso deberá resolverla su instalador particular. Si no cumple ninguna de estas condiciones, el aviso de falta de suministro se convierte en una incidencia en el SGI, tomando como hora de inicio la del primer aviso recibido.

- **Despacho de BT**

La digitalización de la red de BT mediante la instalación de Supervisores Avanzados de Baja Tensión (SABT) así como los equipos de medida inteligentes (Smart Meters), permiten disponer de información en tiempo real sobre el estado de la red de BT. Esta información se gestiona y supervisa en el COR y se utiliza para la detección de interrupciones de suministro que darán origen a una incidencia en el SGI.

- **FlexiApp**

Es una aplicación auxiliar que facilita el intercambio de información entre el COR y los equipos de operación local. A través de FlexiApp, el COR activa los recursos necesarios para que se desplacen a las instalaciones. Una vez allí, los equipos técnicos reparan la instalación, resuelven la incidencia y completan la información en la FlexiApp. Esta información retorna al COR, donde es validada y registrada finalmente en el SGI.

- **.PowerTERed**

Aplicación para la gestión de trabajos e intervenciones programadas en la red de distribución de UFD, ya sea para trabajos de mantenimiento o para conectar nuevas instalaciones. A través de esta aplicación se planifican y gestionan los trabajos programados, que dan lugar a incidencias programadas que también quedan registradas en el SGI.

El diseño resumido cumple con lo dispuesto en el apartado 5 del Anexo I de la Orden ECO/797/2002, en tanto que “permite a un tercero independiente verificar todo el proceso de captación, modificación, transmisión, soporte y tratamiento de la información, para afirmar que la información elaborada por la empresa corresponde a la realidad” [2].

## **ESTRUCTURA Y CONTENIDO DE LA INFORMACIÓN REGISTRADA EN EL SGI**

Cada incidencia registrada en el SGI contiene una estructura de datos que permite identificar con precisión las instalaciones afectadas, los clientes que han sufrido la interrupción del suministro, la duración de la interrupción y las características técnicas del evento, como se muestra en la Figura 4.

CENTRO RESPONSABLE: 9400 LEON Pag.: 1 de 2  
 DEL: 1 - 4 - 2025 AL: 14 - 4 - 2025 TIPO TRABAJO: \_\_ TODOS  
 HORA DESDE: \_\_ : \_\_ HASTA: \_\_ : \_\_ MOTIVO: \_\_  
 ESTADO: \_\_ NIV. TENSION: \_\_

l	Instalacion afectada	Es.	F. Detec.	NT	Tr	Pot afect.	Tiepi	Clien	Durac
	24CQ75 POLIGONO INDUSTRIAL	EM	02-04 12:17	BT		95		1	114
	24CE21 CEROSAN	1 EM	03-04 15:53	BT		379		78	273
	C.H. ESPINOSA	6	EM 04-04 16:55	MT		1300	0,19	663	178
	MILLA DE TERA	5	EM 05-04 03:35	MT		2827	0,32	1479	149
	ASTORGA	3	EM 05-04 07:05	MT		246	0,10	29	656
	ALBARES	6	EM 05-04 13:29	MT		3360	0,02	1682	68
	24CFH3 FOLGOSO ESCUELAS	EM	10-04 11:13	BT		880		239	107
	24AQ28 VILLASINDE	EM	13-04 12:20	BT		198		68	123
	24A825 CABANAS DE LA DORNIL	EM	13-04 21:22	BT		489		125	41
	LA BAÑEZA	15	RS 01-04 08:33	MT		6296	0,23	1179	64
	SANTA MARIA DEL PARAM	3	RS 01-04 12:19	MT		1911	0,01	118	5
	ASTORGA	6	RS 02-04 11:09	MT		1622	0,01	892	7

Figura 4. SGI Vista General de las incidencias ocurridas en un periodo de tiempo.

Entre los campos más relevantes se encuentran:

### 1. Instalación afectada por la pérdida de suministro:

- **Subestación:** la interrupción afecta a todos los circuitos de MT conectados a la misma barra de la subestación.
- **Circuito de MT:** la incidencia puede afectar a todo el circuito o únicamente a un tramo de este.
- **Centro de Transformación (CT):** si dispone de varios transformadores MT/BT, la afectación puede limitarse a uno de ellos o abarcar varios.
- **Salidas de baja tensión (BT):** desde cada transformador se alimentan varias salidas de BT; la interrupción puede afectar a una o varias de ellas, o a tramos determinados.
- **Suministro único (SU):** la interrupción afecta exclusivamente a un único punto de suministro.

### 2. Fecha y hora de detección de la incidencia:

Se registra el instante exacto en que se inicia la primera interrupción y con ella empieza a computar la duración de la incidencia.

### 3. Nivel de tensión:

Se indica el nivel de tensión en el que se produce la incidencia, según la siguiente clasificación:

- Alta Tensión (**AT**):  $V > 66 \text{ kV}$
- Red de Reparto:  $66 \text{ kV} \geq V > 20 \text{ kV}$
- Media Tensión (**MT**):  $20 \text{ kV} \geq V > 1 \text{ kV}$
- Baja Tensión (**BT**):  $V \leq 1 \text{ kV}$
- Suministro único (SU)

#### **4. Potencia afectada por la interrupción:**

De acuerdo con el apartado 9.1 de la Orden ECO/797/2002, se considera:

- En los CT de UFD, se registra la **potencia instalada** (en kVA).
- En los CT particulares, se registra la **potencia contratada** (en kW).

También debe conocerse el municipio en el que se encuentra cada CT para la segmentación exigida en el RD 1955/2000.

#### **5. Número de clientes afectados:**

Corresponde al número de puntos de suministro que dependen de la instalación afectada y que han sufrido la interrupción.

#### **6. Duración de la incidencia:**

Tiempo transcurrido desde el inicio de la incidencia hasta que todos los puntos de suministro afectados han recuperado el servicio eléctrico.

Cabe destacar que en una misma incidencia, la duración de la interrupción para cada una de las instalaciones afectadas puede ser distinta en función de la secuencia de reposición, como se muestra en la Figura 5.

CONSULTA DE INTERRUPCIONES

Pag.: 1 de 4

I. AFECTADA: SS MILLA DE TERA 5

Instalaciones	Pr	Clien.	Pot (KVA)	F.Detec.	F.Resol.	Imp	Dur
49A182 VALLELUENGO	UF	59	50	05-04/03:42	05-04/05:29	X	106
49A205 SANTA EULALIA	UF	96	160	05-04/03:38	05-04/05:29	X	110
49A207 GASOLINERA MOMBU	UF	11	100	05-04/03:42	05-04/06:05	X	143
49A208 MOMBUEY N 1	UF	57	100	05-04/03:42	05-04/06:05	X	143
49A209 MOMBUEY N 2	UF	117	160	05-04/03:42	05-04/06:05	X	143
49A210 OTERO CENTENOS	UF	77	100	05-04/03:42	05-04/06:05	X	143
49A211 ANTA DE TERA	UF	1	50	05-04/03:36	05-04/03:40		4
49A212 VALDEMERILLA	UF	24	50	05-04/03:36	05-04/03:40	X	4
49A213 FRESNO DE LA CAR	UF	78	160	05-04/03:36	05-04/03:40	X	4
49A214 SAN SALVADOR DE	UF	68	160	05-04/03:36	05-04/03:40	X	4

PF20: Matricula ant. PF9: Volver PF12: Resumen interrupciones

Figura 4: SGI. Detalle de interrupciones de una incidencia.

## 7. Tipo de interrupción:

Según el apartado 8 de la orden ECO 797/2002, cada interrupción se clasifica como programada o imprevista.

## 8. Motivo de la incidencia:

Las causas posibles de las incidencias, se deben agrupar según la siguiente clasificación establecida en la ORDEN ECO/797/2002:

- a. Generación. Acreditada mediante informe del Operador del Sistema..
- b. Transporte. Acreditada mediante informe del Operador del Sistema
- c. Terceros:
  - i. Otra empresa distribuidora
  - ii. Instalaciones particulares. (Ej. una línea de MT que alimenta a un solo cliente.
  - iii. Acciones intencionadas o accidentales de terceros, conocidos o no. (Por ejemplo un camión que derriba un poste o una pala excavadora que pica un cable)
  - iv. Huelgas legales.
- d. Fuerza Mayor:

Estas incidencias deben ser aceptadas como tales por la Administración competente e incluyen las que se producen por fenómenos atmosféricos extraordinarios (Por ejemplo, DANA, temporal con vientos huracanados).
- e. Propias de la red de distribución.
  - i. Atmosféricas

- ii. Agentes externos: Animales, arbolado, etc.
- iii. Internas: Fallos de equipos, fallos de materiales, mantenimiento no programado, cambios de explotación, etc.
- iv. Causas desconocidas.

En el SGI, se utiliza el campo “Motivo de la Incidencia” para registrar la causa de cada incidencia mediante un código predefinido.

```

CENTRO: 9410 OP PONFERRADA          ESTADO: EM ENVIADO MNTO.          Pag.: 1 de 1
DESCRIPCION: APOYO NAAS4TGY 18-8 RAMAS
MOTIVO : 201 ANIMALES                I.AFECTADA: SS ALBARES          6
I.ORIGEN: B14 LINEAS AEREAS          LOCALIZ. :
RETEN (FEC.-HORA): 00 / 00 / 00 00 : 00 MAT. AVER.: 03 OTROS ELEMENTOS
TIPO TRABAJO:
FECHA DET: 05-04-25 13:29 FEC.PREV.RES.: - - : Z.TRAB.:
TIEPI: 0,02 PxT: 16541 TIEBT: 0,08 NIEPI: COD.: 941015137
NUM.CC.TT.      POTENCIA (KVA)      CLTES. AFECTADOS      DURACION
uf part        cctt uf cctt part        cctt uf cctt part        (min)
-----
2              260                    267                    63

Totales:
-----
2              260                    267

```

Figura 5. SGI: Detalle de incidencia.

Estos códigos permiten tener un mayor nivel de detalle y se agrupan en bloques según su naturaleza:

- Códigos 1XX: Asociados a **interrupciones programadas**
- Códigos 2XX, 3XX, 4XX: Corresponden a **interrupciones imprevistas**

Las incidencias asociadas a trabajos programados incluyen campos adicionales como: “Pérdida de Mercado”, “Notificación en plazo” y “Tipo de Trabajo”, tal como se muestra en la Figura 6. Esta información permite determinar si las interrupciones asociadas se consideran como programadas o, por el contrario, considerarse como imprevista. *Figura 6. SGI Detalle de información asociada a trabajos programados.*

```

CENTRO: 9420 DP LA BANEZA ESTADO: RS RESUELTA Pag.: 1 de 1
DESCRIPCION: .2513-702 DESCARGO NECESARIO PARA AISLAR LMT AST704
MOTIVO : 105 REFORMA O AMPLIACION I.AFECTADA: SS ASTORGA 4
I. ORIGEN: B00 MEDIA TENSION LOCALIZ. :
RETEN(FEC.-HORA): 00 / 00 / 00 00 : 00 MAT. AVER.:
PERDIDA DE MERCADO: S NOTIF.PLAZO: N TIPO TRABAJO: D DESCARGO
FECHA DET: 17-02-25 08:00 FEC.PREV.RES.: 23-02-25 20:00 Z.TRAB.: NORMALIZADA ZD
TIEPI: 0,03 PxT: 24311 TIEBT: NIEPI: COD.: 942021263
NUM.CC.TT. POTENCIA (KVA) CLTES. AFECTADOS DURACION
uf part cctt uf cctt part cctt uf cctt part (min)
-----
10 4 4480 40 71 3 5
1 110 1 7

Totales:
-----
10 5 4480 150 71 4
PF11: instalaciones afectadas PF12: obser.COR PF15 obser.Sol. PF20: EOL/AZT
PF22: G.E. PF13: clientes afect. PF14: seguimiento PF23: Ct's causante averia
PF21: Mod. Notif. PF18: SEGUI. MAN PF16: LOCAL. INSTAL. PF8 : ZONA DESCARGO

```

Figura 6. SGI Detalle de información asociada a trabajos programados.

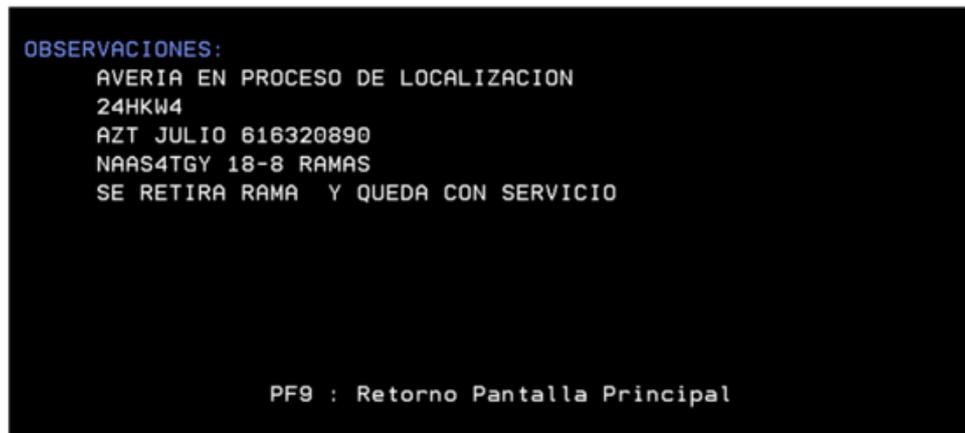
Además, existe una correspondencia entre los códigos del campo “Motivo” y la causa de incidencia definida en la normativa, como se muestra en la Tabla 4:

Causa Incidencia (Orden ECO 797/2002)	Motivos SGI
Terceros	4XX
Propias de la red de distribución	2XX
Causa desconocida	3XX

Tabla 4. Correspondencia entre Causa e Incidencia según OREN ECO797/2002 y Campo Motivo SGI

Además, SGI registra se registra información complementaria como:

- 9. Instalación de origen:** aquella que ha originado la interrupción del suministro (puede diferir de la instalación afectada). Ver detalle en *Figura 5*
- 10. Material averiado:** equipo o componente defectuoso relacionado con la incidencia. Ver detalle en *Figura 5*
- 11. Campos de texto libre:** *Observaciones* y *Descripción*, donde se pueden introducir detalles complementarios no tipificados.



*Figura 7. SGI: Campo Observaciones*

Como se ha expuesto, el SGI almacena una gran cantidad de información estructurada sobre la red de distribución. Esta información, gestionada y explotada adecuadamente, se convierte en un recurso valioso para la toma de decisiones operativas, la planificación de inversiones y la mejora de la calidad del servicio. El objetivo de este TFM es precisamente, analizar y optimizar el uso de esta información.

## **INTEGRACIÓN Y ALMACENAMIENTO DE LA INFORMACIÓN.**

La información registrada en el SGI se almacena en el Data Lake corporativo, un entorno centralizado concebido como un sistema de almacenamiento masivo que permite guardar grandes volúmenes de datos, tanto estructurados como no estructurados, bien en su formato original o en versiones tratadas para distintos usos analíticos.

En el Data Lake toda la información asociada al SGI reside en una única base de datos llamada *Master\_SGI*. Esta base de datos contiene 41 tablas que contienen la información que gestiona el COR y que procede de los distintos sistemas corporativos interrelacionado con el SGI:

- SGI
- Módulo de Operación
- SCADA.

- Power TEREed
- Despacho de Baja Tensión (DBT)
- Plataforma Digital de Servicios
- FlexiApp
- GMO (antes Saturne)

La integración de todos estos datos en el SGI y su almacenamiento en el Data Lake permite su acceso por parte de distintas áreas de la compañía como se verá a continuación.

## **USO DE LA INFORMACIÓN Y APLICACIONES PRÁCTICAS.**

EL SGI responde a la necesidad de clara de disponer de un sistema de registro de incidencias, tal y como establece el apartado 5 de la ORDEN ECO 797/2002[2]. Su primera aplicación práctica es posibilitar el cálculo de los indicadores de calidad de suministro, tanto individual como zonal, de forma trazable, auditable y conforme a los requisitos regulatorios.

### ***2.2.1.1 Evaluación de la calidad de suministro individual***

Anualmente, es necesario evaluar la calidad de suministro individual para cada uno de los puntos de suministro conectados a la red de distribución. Para ello, se consideran únicamente las interrupciones imprevistas, y dentro de estas, solo aquellas que sean imputables a la empresa distribuidora. Quedan, por tanto, excluidas las interrupciones debidas a terceros, al transporte, a la generación o a causas de fuerza mayor, según lo establecido en la normativa vigente.

A partir del número de interrupciones, la duración total de las mismas y la tipología del municipio en el que se encuentra el punto de suministro (urbano, semiurbano, rural concentrado o rural disperso), se determina si el suministro ha incumplido los umbrales establecidos para ese año.

En caso de que se superen dichos límites, el consumidor tiene derecho a un descuento automático en su facturación, que debe ser aplicado por la empresa distribuidora en los tres primeros meses del año siguiente, según lo dispuesto en el artículo 105 del Real Decreto 1955/2000.

#### ***2.2.1.1 Evaluación de la calidad de suministro zonal***

De la misma forma que se calculan los índices de calidad individual con las incidencias registradas en el SGI, también se calculan anualmente los indicadores de calidad zonal definidos en el artículo 106 del RD 1955/2000: TIEPI, NIEPI y el percentil 80 del TIEPI, tomando como base las interrupciones imprevistas de duración superior a tres minutos. Estos valores se evalúan por tipo de zona (urbana, semiurbana, rural concentrada o dispersa), con el objetivo de verificar si se cumple el nivel de calidad exigido.

Como se ha indicado, cuando alguno de estos umbrales se supera de forma reiterada en un municipio, la empresa distribuidora debe presentar un plan de mejora de la calidad del suministro.

Adicionalmente, cada año se elabora un informe que debe enviarse a la CNMC, incluyendo los resultados de los indicadores zonales. “La información obtenida será sometida a las correspondientes auditorías, a fin de obtener un examen sistemático e independiente. Para ello, deberán disponer de un registro de todas las incidencias detectadas durante los últimos cuatro años”[1].

#### ***2.2.1.2 Elaboración del Informe Mensual de seguimiento de la Calidad de Suministro.***

UFD elabora mensualmente un “Informe de Seguimiento de la Calidad de Suministro”, con objeto de que toda la organización sea conocedora de la evolución de los principales indicadores (TIEPI, NIEPI y Número de Incidencias), tanto a nivel global como segmentado por las distintas Áreas Energéticas de la Compañía.

#### **Seguimiento TIEPI y NIEPI Total de Empresa.**

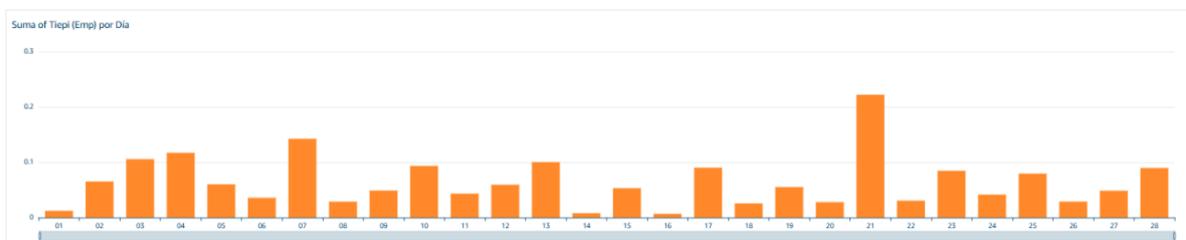
A continuación, se presenta el seguimiento del TIEPI y NIEPI (Figura 8), donde se detallan los valores mensuales y acumulados de TIEPI y NIEPI en comparación con los objetivos fijados por la empresa. Esta información se elabora a mes cerrado, excluyendo las incidencias clasificadas como fuerza mayor.



*Figura 8. Seguimiento mensual de calidad de suministro: TIEPI y NIEPI mensual y acumulado.*  
**Seguimiento diario del TIEPI Total de Empresa.**

Dada la importancia que tiene la calidad de suministro en una empresa distribuidora de electricidad, se realiza un seguimiento diario del TIEPI,

permite identificar de forma rápida aquellos días en los que se han producido un mayor número de incidencias en la red de distribución o en los que estas han tenido una duración. Esta información facilita la detección de incidencias más relevantes, que pueden precisar de un análisis más exhaustivo. Ver la Figura 9.



*Figura 9. Seguimiento mensual de calidad de suministro: evolución del TIEPI diario.*

**Desglose del TIEPI total por zonas de distribución autonómicas.**

En el informe también se representa la aportación al TIEPI total de UFD de las distintas zonas de distribución correspondientes a cada comunidad autónoma. En la



se observa que, para el mes seleccionado, Galicia aporta más de la mitad, 50% del TIEPI total, con 0.922 minutos sobre un total de 1.8 minutos, seguida por la Comunidad de Madrid con un 20% (0,365 minutos), Castilla La Mancha con un 20% (0,350 minutos) y Castilla y León un 10% (0,167minutos).



Figura 10: Seguimiento mensual de calidad de suministro: desglose del TIEPI total por CC.AA.  
**Seguimiento del TIEPI Total de Empresa acumulado por Áreas Energéticas.**

Por último, el Informe Mensual de Seguimiento de la Calidad de Suministro permite que cada una de las Áreas Energéticas de la empresa disponga, mes a mes, de la evolución del TIEPI acumulado en comparación con su objetivo anual. Esta información permite detectar zonas con valores peores de calidad del suministro.



*Figura 11: Seguimiento mensual de calidad de suministro: evolución del TIEPI total acumulado por Área Energética*

## **Capítulo 3. DEFINICIÓN DEL TRABAJO**

Tras analizar el Estado de la Cuestión en UFD y comprender en detalle el funcionamiento del Sistema de Gestión de Incidencias (SGI), así como los sistemas que lo alimentan y la información que contiene, este capítulo tiene como objetivo definir el trabajo a desarrollar en el marco del presente Trabajo de Fin de Máster. Para ello, se presenta la justificación técnica del proyecto, sus objetivos concretos, la metodología adoptada para su desarrollo y la planificación temporal de las actividades previstas.

### **3.1 JUSTIFICACIÓN**

Como se ha expuesto, UFD Distribución de Electricidad cuenta con un sistema consolidado para la gestión de incidencias, el SGI, que garantiza el cumplimiento normativo en materia de calidad de suministro y permite disponer de una gran cantidad de información asociada tanto a trabajos programados como a incidencias imprevistas.

Asimismo, a través del Informe Mensual de Seguimiento de la Calidad de Suministro, se puede visualizar la evolución de los principales indicadores, tanto de forma global como por áreas energéticas, así como el número de incidencias registradas y las causas más frecuentes que las originan.

El presente Trabajo de Fin de Máster se plantea como una oportunidad para impulsar el aprovechamiento de esta información mediante herramientas avanzadas de análisis y visualización, facilitando la toma de decisiones basadas en datos y alineando este proceso con la transformación digital que UFD está experimentando.

### **3.2 OBJETIVOS**

El presente Trabajo de Fin de Máster, titulado “*Gestión y optimización de la información de incidencias de suministro eléctrico en UFD Distribución de Electricidad*”, se plantea como una oportunidad para seguir extrayendo valor de una fuente de datos muy completa, contribuyendo a su evolución, explotación técnica y adaptación a los nuevos retos del sector.

**Objetivo general:**

Optimizar la gestión y el aprovechamiento de la información registrada en el SGI, con el fin de fortalecer el uso analítico de los datos en UFD.

### **Objetivos específicos:**

Las principales líneas de trabajo del proyecto son:

#### **1. Analizar el comportamiento y evolución del tipo de incidencias:**

Se estudiará la evolución temporal de las incidencias, tanto programadas como imprevistas, con el objetivo de detectar patrones, analizar su distribución por zonas y evaluar si la tipificación actual permite identificar el origen técnico o la causa raíz.

#### **2. Revisión fundamentada de las tipificaciones en el SGI:**

Con la próxima implantación del sistema ADMS (Advanced Distribution Management System), resulta clave disponer de una estructura de tipificación clara y coherente que permita automatizar la gestión de interrupciones.

Este objetivo se centra en evaluar la calidad y utilidad real de los campos de tipificación del SGI (motivo, material averiado, instalación origen, etc.), tanto para incidencias imprevistas como para trabajos programados, detectando redundancias y limitaciones.

Se estudiarán los datos históricos registrados en el sistema entre 2022 y 2025, evaluando:

- La frecuencia de uso de los distintos códigos estructurados.
- La complementariedad o redundancia entre campos.
- La información contenida en campos de texto libre como *Observaciones* y *Descripción*.

El objetivo no es definir una nueva propuesta técnica de clasificación, sino identificar las limitaciones actuales y plantear líneas de mejora que sirvan como base para una futura revisión, facilitando así la migración al nuevo entorno ADMS.

### **3. Comparación de modelos de indicadores de calidad de suministro:**

Actualmente, los indicadores de calidad de suministro zonal, TIEPI y NIEPI, se basan en la potencia afectada en las interrupciones de más de tres minutos y solo tienen en cuenta las incidencias en redes de MT, mientras que SAIDI y SAIFI se basan en el número de clientes afectados e incluyen también las interrupciones en BT. Esta diferencia puede suponer cambios importantes en la evaluación de la calidad. Además, el artículo 98.2 del RD 1955/2000 contempla la posibilidad de establecer nuevos indicadores. Por ello, este proyecto evaluará cómo afectaría un cambio metodológico a la realidad operativa de UFD, contrastando ambos enfoques con datos reales para anticipar las consecuencias de posibles cambios regulatorios.

### **4. Explotación técnica de la información contenida en el SGI:**

A partir de los datos registrados en el SGI y almacenados en el Data Lake corporativo, se desarrollará un sistema de análisis y visualización interactiva mediante el uso combinado de SQL, Python y Power BI.

El objetivo es construir paneles dinámicos que permitan consultar, filtrar y explorar la información sobre incidencias de forma estructurada, flexible y adaptada a distintos perfiles técnicos de la organización.

Estos paneles ofrecerán una visión complementaria de la tipificación de incidencias y de los indicadores de calidad de suministro, facilitando la explotación operativa de los datos y sirviendo como base para futuras decisiones técnicas o estratégicas.

## **3.3 METODOLOGÍA**

El presente Trabajo de Fin de Máster se ha desarrollado siguiendo una metodología estructurada, comenzando por un análisis técnico y normativo del tratamiento de las incidencias de suministro eléctrico, con el objetivo de comprender en profundidad qué información se registra en el SGI y qué sistemas interactúan con él. A partir de ahí, se ha

trabajado en la organización, depuración y explotación de estos datos, aplicando herramientas de análisis y visualización.

El trabajo se ha articulado en las siguientes fases:

### **1. Estudio técnico-normativo del contexto de las incidencias**

Como punto de partida, se ha realizado un análisis detallado del concepto de incidencia eléctrica y de su tratamiento normativo, de acuerdo con el Real Decreto 1955/2000 y la Orden ECO/797/2002. Asimismo, se ha estudiado el ecosistema de sistemas operativos de UFD que interactúan con el SGI, incluyendo el COR, SCADA, FlexiApp, PowerTERed y el despacho de BT. Esta fase ha sido esencial para entender el flujo real de información y la estructura de los datos recogidos.

### **2. Acceso a los datos**

Los datos recogidos por el SGI se almacenan de forma segura en el Data Lake corporativo basado en Amazon S3, la plataforma de almacenamiento de Amazon Web Server (AWS). A través de consultas SQL ejecutadas mediante Amazon Athena, se han extraído los registros necesarios para llevar a cabo el análisis, incluyendo información sobre incidencias, tiempo de interrupción y clientes afectados.

### **3. Identificación de tendencias**

Utilizando datos del periodo 2022-2025, se ha analizado la evolución del número de incidencias por clase (programadas e imprevistas) y nivel de tensión (AT, MT, BT, SU). A partir de esta evolución se ha realizado un análisis exploratorio de las tipologías más frecuentes, evaluando para cada una de ellas la información registrada en los campos *Instalación Origen* y *Material Averiado*.

### **4. Análisis de la calidad y coherencia de la tipificación**

Se ha llevado a cabo una revisión crítica del uso de los códigos de tipificación en el SGI por clase (motivo e instalación origen.), contrastando su frecuencia de uso con el contenido de

los campos de texto libre (observaciones y descripción). Esta fase tiene como objetivo formular propuestas de mejora en la tipificación que permitan avanzar hacia una clasificación más operativa, que garantice la coherencia, claridad y utilidad de los registros, especialmente de cara a su integración en el futuro sistema ADMS.

### **5. Evaluación comparativa de indicadores de calidad de suministro**

A partir de los registros históricos, se han calculado y comparado los indicadores TIEPI/NIEPI frente a SAIDI/SAIFI, teniendo en cuenta sus diferencias metodológicas y su impacto potencial en la evaluación regulatoria y la priorización de inversiones. Este análisis permite anticipar el efecto que tendría una eventual evolución de los indicadores de referencia, tal como contempla el artículo 98.2 del RD 1955/2000.

### **6. Construcción de visualizaciones dinámicas y cuadros de mando**

Finalmente, se desarrollan paneles interactivos en Power BI que permiten explorar la evolución de las incidencias y los indicadores de calidad desde distintos enfoques (territorial, temporal, técnico), facilitando el análisis tanto desde el área operativa como desde una perspectiva estratégica.

## **3.4 HERRAMIENTAS Y ENTORNO TECNOLÓGICO**

Para la realización del presente Trabajo de Fin de Máster, se han empleado diversas herramientas tecnológicas que han permitido el acceso, tratamiento y análisis de la información contenida en el sistema SGI, así como su posterior visualización. A continuación, se detallan las principales:

- **Data Lake corporativo de UFD:** repositorio centralizado de datos basado en Amazon S3. A través de este entorno se accede a grandes volúmenes de información operativa, entre ellos los datos del sistema SGI, concretamente desde la base de datos *Master\_SGI*.
- **Amazon Athena y SQL:** se ha utilizado SQL para ejecutar consultas sobre el Data Lake mediante Amazon Athena, lo que ha permitido una extracción eficiente de los

registros de interés y la aplicación de filtros por fechas, clases de incidencia o nivel de tensión.

- **Python:** herramienta principal para el tratamiento y análisis de los datos extraídos. Se ha utilizado para depurar registros, realizar clasificaciones automáticas (por tipo de incidencia, nivel de tensión, etc.), calcular distribuciones estadísticas y generar los gráficos incluidos en la memoria y anexos.
- **Excel:** empleado de manera complementaria para la validación rápida de resultados, la organización preliminar de tablas y como apoyo en tareas puntuales de análisis.
- **Power BI:** plataforma utilizada para construir cuadros de mando interactivos, facilitando la exploración visual de la evolución de incidencias y el análisis comparativo de indicadores. Estas visualizaciones han servido de apoyo a las conclusiones del trabajo y como propuesta para su uso interno en UFD para continuar profundizando en el análisis de la tipificación de incidencias y mejorar su explotación operativa.
- **Azure Virtual Desktop (AVD):** entorno corporativo remoto utilizado para acceder a todas las herramientas y sistemas necesarios desde el entorno digital seguro de UFD.

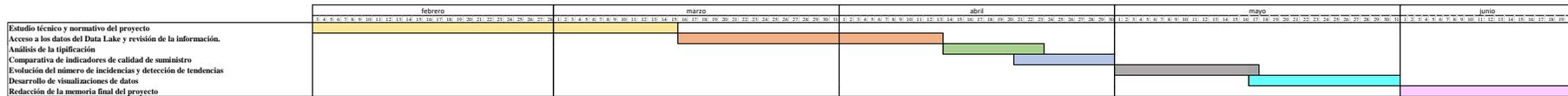
Los scripts desarrollados específicamente para este trabajo se incluyen en el **Anexo II**, junto con capturas de algunas consultas y funciones utilizadas.

### **3.5 ESTIMACIÓN ECONÓMICA Y PLANIFICACIÓN**

El presente Trabajo de Fin de Máster no ha implicado costes económicos adicionales, ya que se ha desarrollado en el marco de una beca de colaboración en UFD, utilizando herramientas (SQL, Power BI, Python...) y datos accesibles a través del Data Lake de la compañía.

Cabe destacar la implicación del tutor de prácticas y la colaboración de distintos departamentos, cuyo apoyo técnico ha sido clave para el avance del proyecto

La planificación se ha estructurado en fases sucesivas, combinando el aprendizaje de herramientas con el análisis progresivo de datos. A continuación, se presenta el diagrama de Gantt con la distribución temporal estimada de las actividades.



*Gráfico 3: Diagrama de Gantt de la Planificación del TFM*

## **Capítulo 4. DESARROLLO DEL PROYECTO**

En este capítulo se describe el trabajo realizado durante el desarrollo del presente Trabajo de Fin de Máster. A partir del análisis normativo y técnico expuesto previamente, se ha llevado a cabo un estudio detallado de la información registrada en UFD, accediendo al Data Lake corporativo donde se almacenan la información procedente de los distintos sistemas interconectados con el SGI.

El objetivo ha sido optimizar el uso de esta información para mejorar su utilidad operativa y estratégica en UFD. El desarrollo del proyecto se ha estructurado en distintas fases, que se detallan a continuación.

### **4.1 PREPARACIÓN Y COMPRESIÓN DEL ENTORNO DEL TRABAJO**

Tras el análisis teórico realizado en el capítulo de “*Investigación previa. Estado de la cuestión*”, se abordó esta primera fase práctica con el objetivo de consolidar la comprensión del funcionamiento del SGI y del ecosistema de sistemas operativos que se integran en el COR, responsable de la gestión de incidencias en UFD.

Esta etapa ha permitido comprender en detalle cómo se genera, transmite y consolida la información sobre las interrupciones de suministro en la red de distribución y ha sido clave para orientar adecuadamente las fases posteriores del proyecto. Además, se revisaron las definiciones normativas de los diferentes tipos de incidencias, los criterios de clasificación de interrupciones y los campos requeridos por la legislación vigente, con el fin de asegurar que el análisis estuviera alineado con los requisitos regulatorios.

También se identificaron los sistemas y herramientas necesarios para el acceso a la información (acceso al SGI y al Data Lake, uso de SQL en el entorno corporativo, etc.), así como los tipos de tablas y estructuras disponibles para el tratamiento de datos.

## 4.2 ACCESO, LIMPIEZA Y ESTRUCTURACIÓN DE LOS DATOS

La primera fase técnica del proyecto ha consistido en el análisis de la información contenida en el Data Lake corporativo, en el que se centralizan todos los registros procedentes tanto del SGI como de otros sistemas y aplicaciones operados desde el COR para la gestión de las incidencias imprevistas y de los trabajos programados, como la de otros sistemas corporativos.

A partir de los conocimientos adquiridos en la fase previa de investigación, se diseñaron consultas SQL orientadas a la extracción de los registros de la base de datos *Master\_SGI*, ubicada en el Data Lake. Esta base de datos está compuesta por 41 tablas, cada una con estructuras distintas, por lo que fue necesario:

- Identificar las tablas con información relevante para cada caso de estudio.
- Comprender la estructura y los campos clave.
- Establecer relaciones entre tablas mediante campos comunes para concatenar y unir los datos correctamente.

La extracción incluyó todas las incidencias correspondientes al trienio 2022-2024 y parte del 2025, sin aplicar restricciones iniciales en cuanto a la duración, motivo o causa de la incidencia y abarcando todos los niveles de tensión (AT, MT, BT, SU), así como las interrupciones etiquetadas como Sin TIEPI (ST), es decir, aquellas cuya duración es inferior a 3 minutos.

Cada registro contiene múltiples campos, entre ellos los relativos a tiempo de interrupción, número de clientes afectados, potencia afectada, instalación origen, material averiado, clase y tipo de incidencia. La relación completa de campos utilizados se recoge en el Anexo I.

Las tareas principales de depuración y estandarización incluyeron:

- **Homogeneización** de formatos de fecha y hora, códigos de instalación y valores categóricos.

- **Depuración** de duplicados y registros incompletos.
- **Clasificación** de incidencias en programadas e imprevistas según la variable *Clase*.
- **Creación de campos combinados** que unifican el código y su descripción correspondiente en los campos *Tipo*, *Instalación Origen* y *Material Averiado*, para facilitar la interpretación y explotación visual.
- **Revisión de la coherencia entre tipo y clase**

Una parte importante del análisis consistió en verificar la coherencia entre los valores de los campos “Tipo de incidencia” y “Clase”, aplicando los siguientes criterios:

- Las incidencias Clase 1 (programadas) deberían tener un código de tipo que comience por 1.
- Las incidencias Clase 2 (imprevistas) pueden tener códigos que comienzan por 2 (propias de red), 3 (causa desconocida) o 4 (terceros).

Durante esta revisión se detectaron algunas incoherencias, como incidencias registradas como Clase 1 cuyo tipo no comenzaba por 1. Consultados los técnicos responsables, indicaron que en algunos casos estas inconsistencias se deben a procesos de retipificación manual realizados durante la revisión de las incidencias.

Para el cálculo de indicadores como SAIDI y SAIFI, fue necesario trabajar a nivel de interrupción, ya que una misma incidencia puede generar varias interrupciones en distintos momentos o afectar a varios puntos. Se tuvieron en cuenta:

- El número de clientes afectados en cada interrupción.
- La duración de cada interrupción, medida en minutos.

Posteriormente, para cada incidencia, se agregaron los valores correspondientes a todas las interrupciones asociadas, con el objetivo de reflejar de forma precisa el impacto total en los clientes. Este enfoque permite cumplir con la metodología de cálculo definida para los

indicadores SAIDI (System Average Interruption Duration Index) y SAIFI (System Average Interruption Frequency Index), que se expresan en función del número total de interrupciones por cliente y la duración media experimentada.

Para llevar a cabo estas tareas se han utilizado consultas SQL sobre el Data Lake y scripts en Python desarrollados específicamente para este proyecto (recogidos en el Anexo II), que permitieron limpiar, preparar y analizar los datos, así como calcular indicadores como SAIDI y SAIFI a partir de las interrupciones registradas.

### **4.3 ANÁLISIS DEL COMPORTAMIENTO Y EVOLUCIÓN DEL TIPO DE INCIDENCIAS.**

Para obtener una visión general de la evolución de las incidencias registradas en la red, se analizaron los registros de la base de datos *Master\_SGI* cuya variable *Clase* toma el valor 1 (trabajos programados) o 2 (averías imprevistas), es decir, aquellas que implican una interrupción del suministro eléctrico. El análisis se realizó atendiendo a tres ejes:

- El tipo de incidencia (programada vs imprevista),
- El nivel de tensión afectado (AT, MT, BT, SU),
- El año natural de registro.

Este enfoque permitió detectar un aumento significativo de trabajos programados entre 2022 y 2023, especialmente en los niveles de media y baja tensión, manteniendo la tendencia en 2024. Para investigar las posibles causas de este incremento, se realizó un análisis adicional, descrito en el Capítulo 5, sobre los trabajos registrados en el sistema GDA.

Los resultados gráficos asociados a este análisis se presentan el **Anexo III**, donde se representan, para cada nivel de tensión, las incidencias clasificadas por tipología (programadas e imprevistas). Los gráficos permiten visualizar patrones temporales, diferencias por nivel de tensión y servirán de base para el análisis de la tipificación de las incidencias.

## **4.4 ANÁLISIS DE LA TIPIFICACIÓN DE LAS INCIDENCIAS**

Una de las principales líneas de trabajo del presente proyecto consiste en la revisión y optimización de la tipificación del campo “Motivo” del SGI, con el objetivo de proponer una estructura más operativa y coherente, que permita identificar con claridad el tipo de actuación que origina el descargo, así como el material averiado o la instalación que da origen a la incidencia. Esta mejora resulta especialmente relevante de cara a su futura implantación en el sistema ADMS

Además, la tipificación debe garantizar la adecuada correspondencia con la clasificación establecida en la ORDEN ECO/797/2002, recogida en el apartado 3.2.2, epígrafe 8, y que agrupa las causas de las incidencias del siguiente modo:

- a. Generación.
- b. Transporte.
- c. Terceros:
  - i. Otra empresa distribuidora
  - ii. Instalaciones particulares.
  - iii. Acciones intencionadas o accidentales de terceros
  - iv. Huelgas legales.
- d. Fuerza Mayor:
- e. Propias de la red de distribución.
  - i. Atmosféricas
  - ii. Agentes externos: Animales, arbolado, etc.
  - iii. Internas
  - iv. Causas desconocidas.

Este apartado se divide en dos bloques diferenciados, correspondientes al análisis de la tipificación de las incidencias programadas y de las incidencias imprevistas, con el fin de identificar códigos redundantes, inconsistentes o en desuso y proponer una simplificación y reorganización alineada con los requerimientos normativos y operativos.

## **TIPIFICACIÓN DE LAS ACTUACIONES PROGRAMADAS.**

Se ha realizado un análisis detallado de la tipificación utilizada en las incidencias asociadas a trabajos programados, centrándose en el campo *Motivo* del SGI, que en la base de datos *Master\_SGI* se denomina *TIPO*, con el objetivo de identificar cuáles son los códigos más utilizados en cada nivel de tensión.

Paralelamente, se han revisado los códigos más empleados en el campo *Ins\_Origen* para valorar la información contenida en el mismo. Asimismo, se ha llevado a cabo una revisión del contenido de los campos de texto libre *Observaciones* y *Descripción*, que frecuentemente contienen información adicional no estructurada, con el fin de evaluar si esta información puede integrarse dentro una tipificación estructurada

El análisis se centró en las incidencias Clase 1 registradas en los niveles de MT y BT, ya que, como se observó en el análisis de la evolución de incidencias, estos niveles mostraron un incremento importante en el periodo estudiado.

Además, se analizaron los trabajos registrados en el sistema GDA, evaluando los campos *Tipo de actuación* (18 valores) y *Clase de actividad* (más de 200 valores), con el objetivo de estudiar su distribución y utilidad como base común para la tipificación de incidencias. Este análisis se segmentó por niveles de tensión (AT, MT y BT) con el fin de comprobar si esta codificación podría emplearse en las incidencias.

Los resultados obtenidos en este análisis pueden consultarse en los Gráficos 11 al 23 del Anexo III, y servirán como base para formular una propuesta de simplificación y clarificación de los códigos de tipificación.

## **TIPIFICACIÓN DE LAS INCIDENCIAS IMPREVISTAS**

Una vez analizada la tipificación en los trabajos programados, se ha procedido a realizar el mismo ejercicio sobre las incidencias imprevistas, centrándose nuevamente en el campo *Motivo* del SGI que en la base de datos *Master\_SGI* se denomina *TIPO*. Además, se ha analizado el campo *Inst. Origen*, y posteriormente *Mat\_Averiado* con el objetivo de evaluar

si aporta información complementaria o más concisa que pueda reforzar la correcta clasificación de las incidencias.

Este análisis se ha considerado especialmente relevante debido a que:

- Las incidencias imprevistas representan la mayoría de las interrupciones registradas en la red, especialmente en los niveles de MT, BT y SU, como se ha puesto de manifiesto en el capítulo de análisis de resultados (apartado 6.2).
- En el contexto actual de transformación digital y de despliegue del futuro sistema ADMS, contar con una tipificación robusta y homogénea resulta esencial para mejorar la calidad de la información, facilitar el tratamiento automatizado de las incidencias y permitir una respuesta operativa más eficiente.
- Las *Observaciones* en texto libre son actualmente una fuente rica de información operativa, pero presentan limitaciones para su explotación automática y dificultan el análisis masivo de datos.

Por ello, los resultados obtenidos en este apartado permitirán establecer una propuesta de optimización de la tipificación de las incidencias imprevistas, orientada a definir un modelo más estructurado, claro y eficiente.

## **4.5 COMPARACIÓN DE INDICADORES DE CALIDAD DE SUMINISTRO**

El objetivo de este análisis es evaluar cómo varía la aportación de los niveles de Media Tensión (MT) y Baja Tensión (BT) en el cálculo de SAIDI y SAIFI, teniendo en cuenta que los indicadores TIEPI y NIEPI únicamente contemplan las incidencias de MT.

A diferencia de estos últimos, que ponderan las interrupciones en función de la potencia afectada (kW), los indicadores SAIDI y SAIFI se calculan en función del número de clientes afectados, lo que permite obtener una visión más orientada al impacto directo sobre los usuarios.

Para ello, se ha trabajado a partir de los registros de la base Master\_SGI, que ya contiene un campo específico con los valores de SAIDI y SAIFI asignados a cada interrupción. El proceso ha consistido en:

- Extracción de datos desde el Data Lake corporativo, concretamente de la tabla Master\_SGI, para el periodo 2022-2025.
- Aplicación de un filtro para considerar únicamente las interrupciones con duración superior a 180 segundos, en coherencia con el criterio aplicado en el cálculo de TIEPI y NIEPI.
- No se han considerado las interrupciones clasificadas como Suministro Único (SU), manteniendo el foco en los niveles Media Tensión (MT) y Baja Tensión (BT).
- Segmentación de los datos por:
  - Año natural de registro.
  - Clase de incidencia (Programadas o Imprevistas).

A partir de los datos obtenidos, se ha calculado, para cada combinación de año y clase, el porcentaje de SAIDI y SAIFI que corresponde a las interrupciones de BT frente a MT, con el objetivo de analizar cómo varía la aportación relativa de cada nivel de tensión.

Este análisis permite evaluar cómo afectaría a la percepción de la calidad del suministro la inclusión de las incidencias de Baja Tensión, actualmente no consideradas en los indicadores TIEPI y NIEPI.

En particular, se busca entender si el incremento observado en los últimos años en el número de trabajos programados en BT, tal como se ha descrito en el apartado 4.3, tendría un impacto relevante sobre los indicadores de calidad cuando se consideran desde la perspectiva de los clientes (SAIDI/SAIFI), en lugar de la potencia instalada (TIEPI/NIEPI).

Los resultados gráficos de este análisis se encuentran recogidos en el Anexo III.5, donde se representa, para cada año y clase de incidencia, el peso relativo que aportan los niveles de MT y BT sobre los valores totales de SAIDI y SAIFI.

Por último, la revisión detallada de los valores absolutos de SAIDI y SAIFI, desglosada por clase de incidencia y nivel de tensión para el periodo 2022-2025, se desarrolla en el apartado 4.6, dentro del panel de visualización diseñado en Power BI.

#### **4.6 DESARROLLO DE HERRAMIENTAS DE VISUALIZACIÓN DE DATOS.**

Como parte del proyecto, se ha desarrollado un panel interactivo en Power BI, orientado a facilitar el análisis y la explotación operativa de la información registrada en el SGI. Esta herramienta permite consultar, filtrar y explorar los datos de forma estructurada y flexible, adaptándose a distintos perfiles técnicos de la organización.

La creación de este panel responde al objetivo de proporcionar una herramienta ágil que permita visualizar de manera clara tanto la evolución de las incidencias como el comportamiento de los indicadores de calidad de suministro, facilitando la detección de patrones, posibles inconsistencias en la tipificación y la toma de decisiones operativas.

#### **VISUALIZACIÓN DE LA EVOLUCIÓN DE LA TIPIFICACIÓN DE LAS INCIDENCIAS**

Como apoyo al análisis previo, se ha desarrollado un panel en Power BI que permite visualizar de forma dinámica la evolución de la tipificación de las incidencias registradas en el SGI durante el periodo 2022–2025.

El panel está compuesto por cuatro pantallas principales:

- Tres pantallas enfocadas en la evolución anual del número de incidencias, diferenciadas según su clase:
  - Programadas (Clase 1)

- Imprevistas (Clase 2)
- Régimen especial o sin interrupción  
Cada una de ellas incluye filtros por nivel de tensión y centro de operación, y permite analizar la distribución de las incidencias según los principales campos de tipificación: Tipo de Incidencia, Instalación Origen y Material Averiado.
- Una cuarta pantalla con navegación jerárquica, que permite explorar la información desde una vista agregada hasta el detalle más técnico. Para cada tipo de incidencia, se muestra el número total de registros, el TIEPI acumulado y el número de clientes afectados, con posibilidad de desglosar por instalación origen y material averiado.

Este desarrollo constituye una herramienta práctica para la revisión de la calidad de la tipificación, la detección de patrones y la identificación de posibles incoherencias o áreas de mejora en la codificación. Además, aporta una base visual sólida para orientar las acciones futuras de limpieza, reorganización y estandarización de la información de incidencias en el marco de la transición hacia el sistema ADMS.

## **VISUALIZACIÓN DE INDICADORES SAIDI Y SAIFI**

De forma complementaria, se ha desarrollado un segundo panel específico para el análisis de los indicadores SAIDI y SAIFI, con el objetivo de representar de manera visual y operativa cómo se distribuyen estos indicadores según distintos criterios y dimensiones.

Para su elaboración se ha trabajado a partir de los registros de interrupciones, incluidos en la base *Master\_SGI*, considerando que una misma incidencia puede generar más de una interrupción. Se han aplicado los siguientes filtros y transformaciones:

- Interrupciones con duración superior a 180 segundos (3 minutos), en coherencia con el criterio utilizado para el cálculo de los indicadores regulados TIEPI y NIEPI.

- Exclusión de las incidencias clasificadas como Suministro Único (SU), centrandolo el análisis en los niveles de AT, MT y BT.
- Los valores de SAIDI en origen se encontraban expresados en segundos, por lo que, para facilitar la interpretación de los datos, se ha creado una variable denominada SAIDI\_horas que es la que se muestra en las visualizaciones desarrolladas.

El panel consta de una pantalla diferenciando el SAIDI y el SAIFI, que incluyen las siguientes visualizaciones:

- Un gráfico circular, que muestra la distribución porcentual del SAIDI y del SAIFI según el nivel de tensión (MT y BT), permitiendo analizar cómo se reparte el indicador entre estos dos niveles.
- Un gráfico de barras, que representa los valores absolutos de SAIDI y del SAIFI, desglosados por clase de incidencia (programadas e imprevistas). Este gráfico permite observar el valor en horas (para SAIDI) o en número de interrupciones por cliente (para SAIFI) que aportan las incidencias programadas o las imprevistas, pudiendo aplicar filtros por nivel de tensión para conocer el valor absoluto que aporta cada uno de ellos.

Además, el panel dispone de filtros interactivos que permiten ajustar la visualización según:

- Año de análisis.
- Clase de incidencia (Programadas o Imprevistas).
- Provincia.

También se ha incorporado un filtro que permite aislar las incidencias tipificadas como fuerza mayor, para analizar su aportación al SAIDI y SAIFI, o bien excluirlas del análisis si se desea.

Este desarrollo permite analizar de forma detallada la distribución del SAIDI y SAIFI, tanto en términos relativos (porcentaje por nivel de tensión) como en términos absolutos

(valor total por clase de incidencia), proporcionando una herramienta operativa para evaluar cómo se comportan estos indicadores bajo distintos criterios de análisis.

## Capítulo 5. ANÁLISIS DE RESULTADOS

Tras la preparación y estructuración de los datos descrita en el capítulo anterior, se procede en este capítulo al análisis de los resultados obtenidos.

### 5.1 EVOLUCIÓN DEL NÚMERO DE INCIDENCIAS POR NIVEL DE TENSIÓN (2022-2024) Y TIPO.

Como primer paso en el análisis de resultados, se ha estudiado la evolución del número total de incidencias registradas entre 2022 y 2024, clasificadas por niveles de tensión y por tipo de incidencia, distinguiendo entre incidencias programadas e incidencias imprevistas, originadas por averías en la red. Los resultados se resumen en la siguiente gráfica.

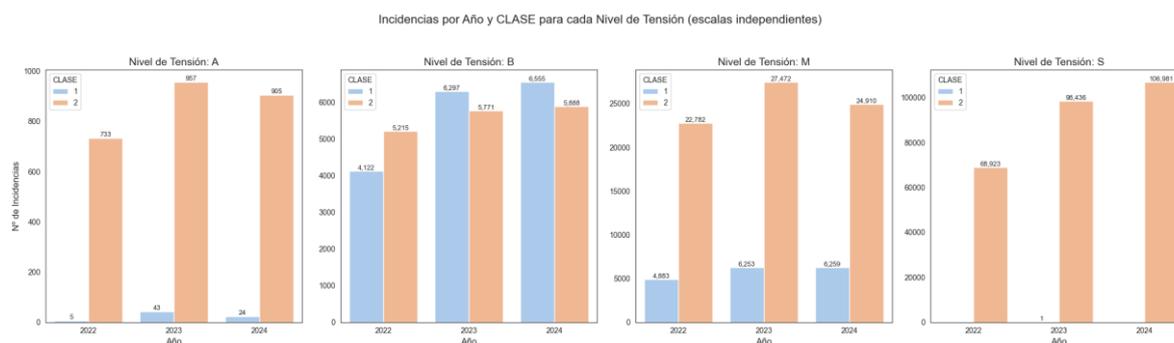


Gráfico 4: Número de Incidencias por Año (2022), Nivel de Tensión (A: AT, B:BT, M:MT y S:SU) y Clase 1: Programadas. 2: Imprevistas). Elaboración propia.

En AT, el número de incidencias es significativamente menor que en el resto de los niveles, lo cual es coherente con el diseño mallado y redundante de estas redes. Se observa que el número de trabajos programados que provocan interrupciones de suministro es muy reducido comparándolo con los otros niveles.

En MT, las incidencias imprevistas se mantuvieron relativamente constantes en los años 2022 y 2023, con una reducción cercana al 10% en 2024. Por el contrario, las programadas se incrementaron casi un 30% entre el 2022 y 2023, manteniéndose en valores similares en 2024. Esta evolución puede estar asociada a un incremento de inversión en la construcción y mejora de las instalaciones.

En BT, el número de incidencias programadas ha crecido casi un 60%, en el periodo estudiado. Este incremento refleja un aumento significativo de las actuaciones planificadas sobre la red, para realizar actuaciones de mejora y digitalización de las redes de baja tensión. Las incidencias imprevistas también aumentaron, aunque de forma más moderada, con un crecimiento aproximado del 13%.

Respecto a las incidencias clasificadas como SU, la totalidad de estas incidencias corresponde a interrupciones imprevistas. Se aprecia un incremento de más del 70% en el trienio analizado, que será objeto de análisis en apartados posteriores.

## **5.2 ANÁLISIS DE LA TIPIFICACIÓN DE INCIDENCIAS.**

Uno de los objetivos principales de este trabajo es evaluar si la tipificación actual en el SGI permite identificar y clasificar adecuadamente las distintas actuaciones que generan interrupciones programadas e imprevistas. En este sentido, se ha realizado un análisis exhaustivo de los campos estructurados *Inst\_Origen* y *Tipo*, así como de los campos de texto libre *Observaciones* y *Descripción*.

Adicionalmente, para las incidencias programadas que derivan de trabajos planificados, se ha revisado la tipificación de los trabajos en el sistema de Gestión de Activos (GDA), donde se dan de alta las actuaciones. Aquellas que necesitan dejar sin servicio la red para ejecutarse, posteriormente dan lugar a incidencias en el SGI.

Los resultados de esta exploración se presentan a continuación, diferenciando entre trabajos programados y averías imprevistas.

### **5.2.1 TIPIFICACIÓN DE LAS INCIDENCIAS PROGRAMADAS**

Se ha llevado a cabo un análisis detallado de los registros de Clase 1 (programadas) correspondientes a los niveles de media tensión (MT) y baja tensión (BT), que son los que mostraron un mayor incremento en el número de incidencias en el periodo 2022–2024.

El análisis se ha centrado en los principales campos de tipificación utilizados en el SGI:

- **Campo estructurado *Tipo*** (denominado "Motivo" en el SGI): identifica la causa asociada al trabajo programado. Se observó un uso mayoritario de pocos códigos tanto para MT como para BT —principalmente *105 Reforma o ampliación*, *103 Mantenimiento correctivo* y *199 Solicitado por el cliente*—, sin que esta tipificación permita diferenciar claramente el tipo de actuación concreta realizada, como se puede ver en el Gráfico 5 para BT y en el Gráfico 6 para MT.

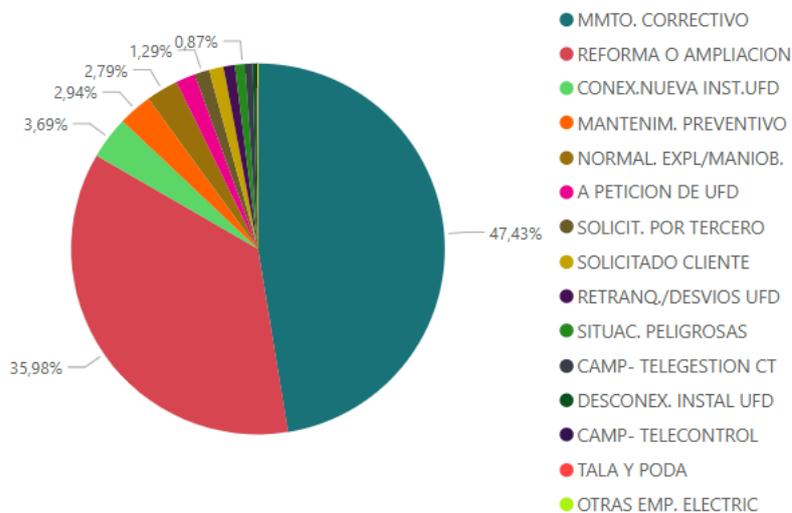


Gráfico 5: Tipología de Incidencias Programadas en BT

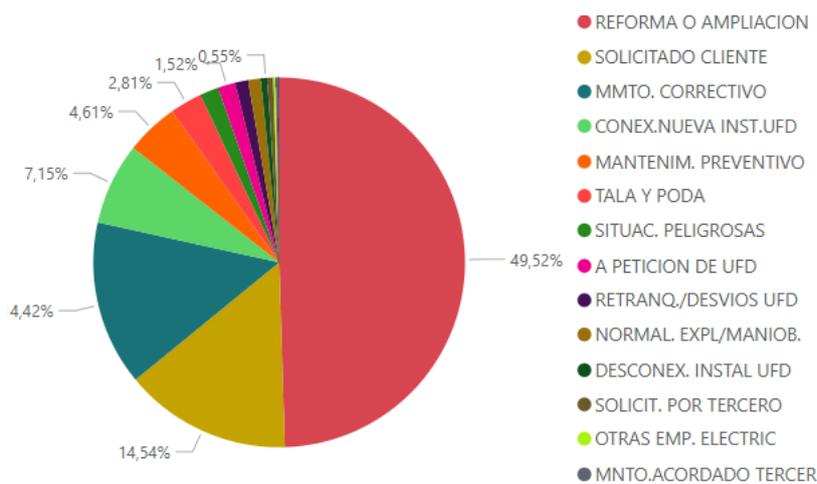


Gráfico 6: Tipología de Incidencias Programadas en MT

- **Campo *Inst\_Origen*:** que debería permitir identificar el activo o instalación origen de la incidencia. Sin embargo, en la práctica, las incidencias de MT y BT se asocian casi exclusivamente a los códigos genéricos B00 *Media Tensión* y C00 *Baja Tensión*, sin aportar información específica sobre el activo implicado ni información adicional que no quede reflejada ya en el campo *Nivel de Tensión*.

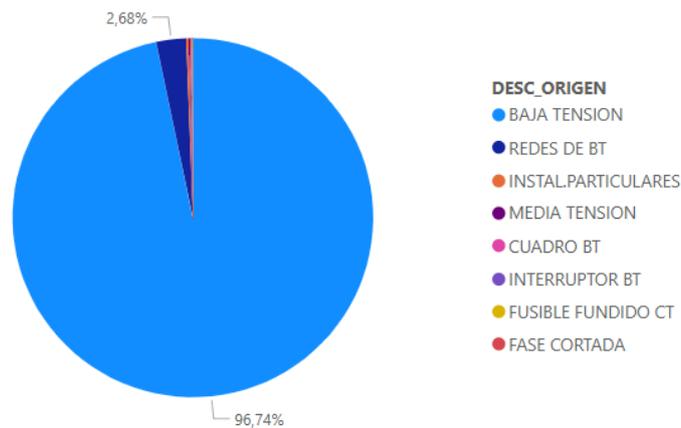


Gráfico 7: Instalación Origen de Incidencias Programadas en BT

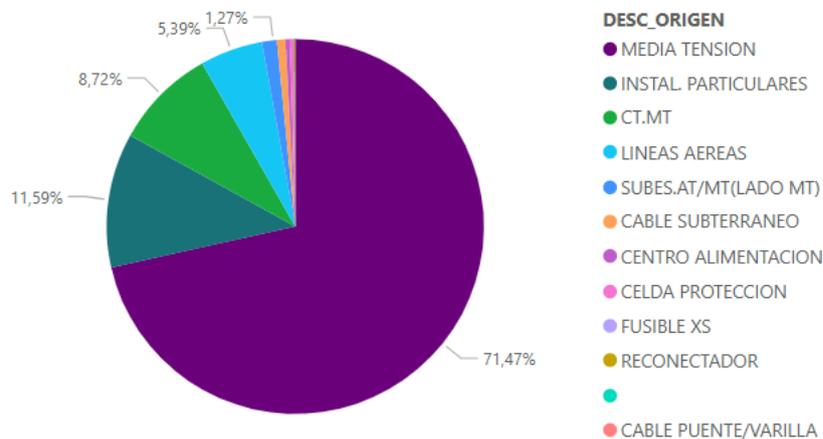


Gráfico 8: Instalación Origen de Incidencias Programadas en MT

Como complemento al análisis anterior, se ha generado un mapa de calor que muestra la frecuencia de las combinaciones entre los campos *Tipo* e *Inst\_Origen* en las incidencias programadas de media y baja tensión. Estas visualizaciones permiten identificar qué asociaciones son más frecuentes y valorar su utilidad como base de clasificación.

En el caso de la media tensión (MT), el resultado evidencia que la mayoría de las incidencias se concentran en las combinaciones [105:Reforma o ampliación, B00: Media Tensión], [103:Mantenimiento correctivo, B00: Media Tensión] lo que indica un uso muy limitado de las posibilidades de codificación existentes.

DES_TIPO	BAJA TENSION	CUADRO BT	FASE CORTADA	FUSIBLE FUNDIDO CT	INSTAL.PARTICULARES	INTERRUPTOR BT	MEDIA TENSION	REDES DE BT
A PETICION DE UFD	58			2	5	1		3
CAMP- TELECONTROL	3							
CAMP- TELEGESTION CT	26							1
CONEX.NUEVA INST.UFD	147	1						4
DESCONEX. INSTAL UFD	13							
MANTENIM. PREVENTIVO	116							5
MMTO. CORRECTIVO	1908	2	2				2	41
NORMAL EXPL/MANIOB.	110					1		4
OTRAS EMP. ELECTRIC	1							
REFORMA O AMPLIACION	1438	1					2	40
RETRANQ./DESVIOS UFD	39							1
SITUAC. PELIGROSAS	32					1		3
SOLICIT. POR TERCERO	48					1		4
SOLICITADO CLIENTE	45					2		5
TALA Y PODA	2							

Tabla 5: Mapa de calor: Instalación Origen / Tipo de Incidencias Programadas MT. Año 2022

De forma similar, en la baja tensión (BT) destaca la combinación [105: Reforma o ampliación, C00: Baja Tensión], que representa la mayoría de los registros.

DES_TIPO	CABLE SUBTERRANEO	CELDA PROTECCION	CENTRO ALIMENTACION	CT.MT	FUSIBLE XS	INSTAL. PARTICULARES	LINEAS AEREAS	MEDIA TENSION	RECONECTADOR	SUBES.AT/MT (LADO MT)
A PETICION DE UFD	2	1			16	1	14	39		1
CAMP- TELECONTROL								1		
CONEX.NUEVA INST.UFD	15			1	25	2	27	278		1
DESCONEX. INSTAL UFD					2			24		1
FALLO MATERIAL-EQUIP					1					
INSTAL. PARTICULARES						1				
LOCALIZ/NORMAL EXPL								3		
MANIOBRA REPOS.SERVI								1		
MANTENIM. PREVENTIVO	1	1	1		16	2	12	164		28
MMTO. CORRECTIVO	5	6	4		84	1	37	539		28
MINTO.ACORDADO TERCER								8		
NORMAL EXPL/MANIOB.	2			1	1	2	17	30		1
OTRAS EMP. ELECTRIC							1	7		
REFORMA O AMPLIACION	6	4	9		268	2	97	2027		2
RETRANQ./DESVIOS UFD	2				3			49		
SITUAC. PELIGROSAS		2	1		9			59		
SOLICIT. POR TERCERO						1		17		
SOLICITADO CLIENTE	3		1		2		6	143		1
TALA Y PODA							36	101		

Tabla 6: Mapa de calor: Instalación Origen / Tipo de Incidencias Programadas BT. Año 2022

- **Campos de texto libre *Observaciones y Descripción***: contienen información adicional no estructurada. En el análisis de frecuencia (Anexo III), se identificaron palabras recurrentes como *descargo, cambio, trafo, celdas, CBTI*, entre otras. No obstante, la falta de estandarización y coherencia en el vocabulario utilizado impide su explotación automática como herramienta de clasificación.
- De forma complementaria, se analizaron **los trabajos registrados en el sistema GDA**, origen de estas actuaciones programadas. Este sistema permite categorizar los trabajos mediante los campos *tipo de actuación* (18 categorías) y *clase de actividad* (más de 280 categorías). Se constató que resulta muy difícil inferir el trabajo concreto realizado a partir de la codificación actual y que la mayoría de los trabajos se concentran en un conjunto muy reducido de cada uno de estos campos, lo que refuerza la necesidad de racionalizar la codificación actual, reduciendo su complejidad.

Por todo ello, se propone una revisión integral de los esquemas de tipificación actuales, planteando como mejora la adopción de los *tipos de actuación* definidos en el GDA como base unificada, lo que garantizará una codificación coherente y alineada de las actividades programadas.

## **5.2.2 TIPIFICACIÓN DE LAS INCIDENCIAS IMPREVISTAS**

Se ha llevado a cabo un análisis detallado de los registros clasificados como Clase 2 (imprevistas) en el SGI, correspondientes a los niveles de tensión MT, BT y SU durante el periodo 2022–2024.

Para cada uno de estos niveles, se identificaron los valores más empleados en el campo *Tipo*, y se analizó para cada uno de ellos la instalación origen registrada (*Inst\_Origen*) y el elemento afectado (*Mat\_Averiado*). Este análisis permite evaluar la consistencia y utilidad de la tipificación existente, así como detectar posibles patrones o limitaciones en el registro de información.

A continuación, se presentan los resultados obtenidos para cada nivel de tensión, diferenciando las particularidades observadas en cada caso.

### 5.2.1.1 Nivel SU (Suministro Único)

En el caso de las incidencias que afectan a un único suministro (nivel SU), se han analizado las tipologías más empleadas, descartando aquellas que no aportan información útil en los campos relacionados con la instalación que origina la avería. Esto incluye, por ejemplo, incidencias anuladas tras su registro inicial o aquellas que se resuelven automáticamente, sin dejar información adicional relevante. Teniendo en cuenta estos criterios, el análisis se ha centrado en las siguientes tipologías del campo *Tipo*:

416	Suministro Correcto.
417	Equipo de Medida.
303	Fallo de Material
317	Otra actuación sin corte.

En la Tabla 7 se recogen, para cada uno de los tipos de incidencia analizados, los códigos más frecuentemente utilizados en los campos *Inst\_Origen* y *Mat\_Averiado*, considerando únicamente los registros en los que están cumplimentados. Aunque en muchos casos esta información no se registra, se han identificado los códigos más representativos entre los registros que sí disponen de dicha información.

Tipo Incidencia	Código de Instalación Origen		Código de Material Averiado
303: Fallo de Material	C00	Baja Tensión	3 Otros Elementos
	C60	Redes de BT	
	C74	Contador	
	C84	Acometida Averiada	
	C90	Fase Cortada	
	C91	Neutro Cortado	
317: Otras Actuaciones / Sin Corte	C00	Baja Tensión	3 Otros Elementos 1 Ningún Material
	C50	Instal. Particulares	
	C60	Redes de BT	
	C84	Acometida Averiada	
	B00	Media Tensión	
	C98	Fase Tierra	
416: Suministro Correcto	C00	Baja Tensión	1 Ningún Material
	C50	Instal. Particulares	4 Instal. Particulares

417: Equipo de Medida	C72	ICP Contador	1 Ningún Material
	C73	ICP Vivienda	6H ICP
	C74	Contador	5H Contador

Tabla 7: Resumen Tipificación Incidencias Imprevistas Nivel SU.

Los resultados parciales y el desarrollo detallado del análisis pueden consultarse en el Anexo III: Tipificación de las incidencias imprevistas (SU).

Este análisis sugiere que, aunque la tipificación en el nivel SU refleja ciertos patrones, se revise debe revisar, simplificar y mejorar.

### 5.2.1.2 Nivel de Tensión Baja Tensión

Al igual que en el caso anterior, se han analizado las tipologías más comunes que para las incidencias imprevistas en redes de BT. Estas, en orden decreciente de frecuencia, son:

303	FALLO MATERIAL-EQUIP
301	CAUSA DESCONOCIDA
416	SUMINI. CORRECTO RED
202	ARBOLES

En la Tabla 8 se recogen, para cada uno de los tipos de incidencia analizados, los códigos más frecuentemente utilizados en los campos *Inst\_Origen* y *Mat\_Averiado*, considerando únicamente los registros en los que sí están cumplimentados. Aunque esta información no siempre se registra, se han identificado los códigos más representativos entre los registros que sí disponen de dicha información y se presentan en orden de frecuencia decreciente.

Tipo Incidencia	Código de Instalación Origen		Código de Material Averiado	
303 : Fallo de Material	C90	Fase Cortada	3	Otros Elementos
	C60	Redes de BT	0H	Conductor Fase BT
	C91	Neutro Cortado	2	Pte. Reparación Definitiva
	C00	Baja Tensión	6A	Fusible de BT en CT
	C94	Fusible Fundido en CT	1H	Conductor Neutro en BT
301: Causa Desconocida	C94	Fusible Fundido en CT	6A	Fusible de BT en CT
	C97	Interruptor BT	1	Ningún Material
	C00	Baja Tensión	7A	Interruptor BT
	C60	Redes de BT	3	Otros Elementos
	C90	Fase Cortada	0H	Conductor Fase BT
416: Suministro Correcto	C50	Instal.Particulares	1	Ningún Material
	C00	Baja Tensión	4	Instal.Particulares

Tipo Incidencia	Código de Instalación Origen		Código de Material Averiado	
	C60	Redes de BT	3	Otros Elementos
	C98	Fase Tierra	2H	CGP
	C74	Contador	2	Pdte. Reparación Definitiva
202 Árboles	C60	Redes de BT	3	Otros Elementos
	C90	Fase Cortada	0H	Conductor Fase BT
	C91	Neutro Cortado	2	Pdte. Reparación Definitiva
	C00	Baja Tensión	1H	Conductor Neutro BT
	C84	Acometida Cortada	0D	Cable Aéreo

Tabla 8: Resumen Tipificación Incendias Imprevistas Nivel BT

Los resultados parciales y el desarrollo detallado del análisis pueden consultarse en el Anexo III: Tipificación de las incidencias imprevistas de BT.

Este análisis sugiere que la tipificación actual debería revisarse en profundidad, tanto para simplificar los códigos utilizados como para aclarar el uso específico de aquellos que resultan similares y se emplean de forma indistinta (por ejemplo, C00 vs. C60, o Mat\_Averiado 03 vs. 02).

Además, no parece necesario permitir el uso de todos los códigos en todas las tipologías de incidencia, por lo que sería recomendable limitar los valores posibles en función del tipo de incidencia, facilitando así una codificación más clara, coherente y automatizable.

### 5.2.1.3 Nivel de Tensión Media Tensión

Al igual que en el caso anterior, se han analizado las tipologías más comunes para las incidencias imprevistas en redes de media tensión (MT). Tras filtrar todas las incidencias sin TIEPI —es decir, aquellas en las que la duración de la interrupción es inferior a tres minutos—, dado que se generan de forma automática y no contienen información relevante, se han seleccionado las tipologías más representativas.

Según lo descrito en el Anexo III – Tipificación de las Incidencias Imprevistas, las siguientes tipologías, ordenadas por frecuencia decreciente, han sido objeto de análisis:

301	CAUSA DESCONOCIDA
303	FALLO MATERIAL-EQUIP
304	LOCALIZ/NORMAL EXPLO
202	ARBOLES
406	INSTALACIONES PARTICULARES

En la se recogen, para cada uno de los tipos de incidencia analizados, los códigos más frecuentemente utilizados en los campos *Inst\_Origen* y *Mat\_Averiado*. Aunque esta información no siempre se registra, se han identificado los códigos más representativos entre los registros que sí disponen de dicha información y se presentan en orden de frecuencia decreciente.

Tipo Incidencia	Código de Instalación Origen		Código de Material Averiado	
301: Causa Desconocida	B00	Media Tensión	1	Ningún Material
	B14	Líneas Aéreas	4D	Fusible XS
	B19	Fusible XS	3A	Fusible de MT
	B22	Reconectador	3	Otros Elementos
	B16	Celda De Protección	2	Pte. Reparación Def
303 : Fallo de Material	B06	Cable Subterráneo		Vacío
	B14	Líneas Aéreas	1C	Cable Seco
	B00	Media Tensión	2	Pte. Reparación Def
	B15	CT MT	3	Otros Elementos
	B19	Fusible XS	4D	Fusible XS
			2A	Transformador MT
304: Localización / Normalización de la explotación	B00	Media Tensión	1	Ningún Material
	B14	Líneas Aéreas		Vacío
	B15	CT MT	3	Otros Elementos
	B22	Reconectador	2A	Transformador MT
	B16	Celda De Protección	4	Instalaciones Partic
			4D	Fusible XS
406: Instalaciones Particulares	B11	Instal. Particulares	4	Instalaciones Particulares
	B00	Media Tensión		Vacío
	B14	Líneas Aéreas	1	Ningún Material
	B06	Cable Subterráneo	4D	Fusible Fundido
	B19	Fusible XS	3D	Base XS/SXS
			3	Otros Elementos
202: Árboles	B14	Líneas Aéreas	1	Ningún Material
	B00	Media Tensión		Vacío
	B19	Fusible Xs	0D	Cable Aéreo

	B15	CT MT	3	Otros Elementos
	B22	Reconectador	4D	Fusible XS
			2	Pte. Reparación Defi

Tabla 9: Resumen Tipificación Incendias Imprevistas Nivel MT

Los resultados parciales y el desarrollo detallado del análisis pueden consultarse en el Anexo III: Tipificación de las incidencias imprevistas de BT.

Este análisis sugiere que la tipificación actual debería revisarse en profundidad, tanto para simplificar los códigos utilizados como para aclarar el uso específico de aquellos que resultan similares y se emplean de forma indistinta.

Además, no parece necesario permitir el uso de todos los códigos en todas las tipologías de incidencia, por lo que sería recomendable limitar los valores posibles en función del tipo de incidencia, facilitando así una codificación más clara, coherente y automatizable.

### 5.3 COMPARACIÓN DE INDICADORES DE CALIDAD

#### *SAIDI (horas)*

A lo largo del periodo estudiado 2022-2024 se observa que:

El SAIDI total está repartido en un 72% a MT y un 28% a BT.

El SAIDI programado (Clase 1) está repartido en 20 % MT y un 80% BT

El SAIDI imprevisto (Clase 2) está repartido en casi 80% MT y un 20% BT

En valores absolutos:

SAIDI imprevisto en MT: entre 1,6 y 1,7 horas/año.

SAIDI programado en BT: entre 0,21 y 0,36 horas/año.

Este comportamiento destaca cómo la baja tensión, tradicionalmente excluida de los indicadores oficiales (TIEPI y NIEPI), empieza a tener un peso considerable en el ámbito de las interrupciones programadas, derivado del incremento de actuaciones de mejora, renovación y digitalización de la red.

### ***SAIFI (interrupciones)***

- El SAIFI total está repartido en un 96,7% en MT y un 3,3% en BT.
- El SAIFI programado (Clase 1) se reparte entre un 61–74% en MT y un 26–39% en BT.
- El SAIFI imprevisto (Clase 2) está claramente concentrado en MT, que aporta más del 98%, mientras que BT tiene un peso residual, siempre inferior al 2%.

En valores absolutos:

- SAIFI imprevisto en MT: entre 6,3 y 7,6 interrupciones/año.
- SAIFI programado en BT: entre 0,4 y 0,5 interrupciones/año.

La frecuencia de interrupciones sigue dominada claramente por la MT, especialmente en el ámbito de las incidencias imprevistas. Sin embargo, las actuaciones programadas en BT adquieren una relevancia creciente.

## **5.4 DESARROLLO DE HERRAMIENTAS DE VISUALIZACIÓN**

### **VISUALIZACIÓN DE LA EVOLUCIÓN DE LA TIPIFICACIÓN DE INCIDENCIAS**

Se ha desarrollado un panel en Power BI que permite analizar la evolución de la tipificación de las incidencias registradas en el SGI en el periodo 2022–2025.

Este panel se ha diseñado con una estructura común para las tres clases de incidencias consideradas:

- Clase 1 – Programadas (Descargos)
- Clase 2 – Imprevistas (Averías)
- Clase 3 – Régimen especial (Sin interrupción)

Cada uno de los paneles permite:

- Visualizar la evolución del número de incidencias por año y clase.
- Analizar la distribución de incidencias por:

- Tipo de incidencia (campo TIPO) mediante un gráfico circular interactivo.
- Instalación origen (INST\_ORIGEN) y Material averiado (MAT\_AVERIADO) mediante gráficos de barras.
- Aplicar filtros por:
  - Año
  - Nivel de tensión (AT, MT, BT, SU)
  - Centro de operaciones

Una característica clave del panel es la interactividad. La selección de cualquier categoría en el gráfico circular de Tipos actualiza automáticamente los gráficos de *Instalación Origen* y *Material Averiado*, permitiendo un análisis detallado y contextualizado para cada tipología.

La estructura de los tres paneles es idéntica; únicamente varía el conjunto de incidencias que se visualiza, en función de si corresponden a programadas, imprevistas o régimen especial.

### ***Panel Incidencias Programadas***

A continuación, se presenta un ejemplo de uno de los paneles, correspondiente a las incidencias programadas (Clase 1).

En este panel, Figura 12, se comprueba que a lo largo del periodo analizado, los *Tipos de Incidencia* más frecuentes son:

- 101 – Conexión de nuevas instalaciones
- 102 – Mantenimiento preventivo
- 103 – Mantenimiento correctivo

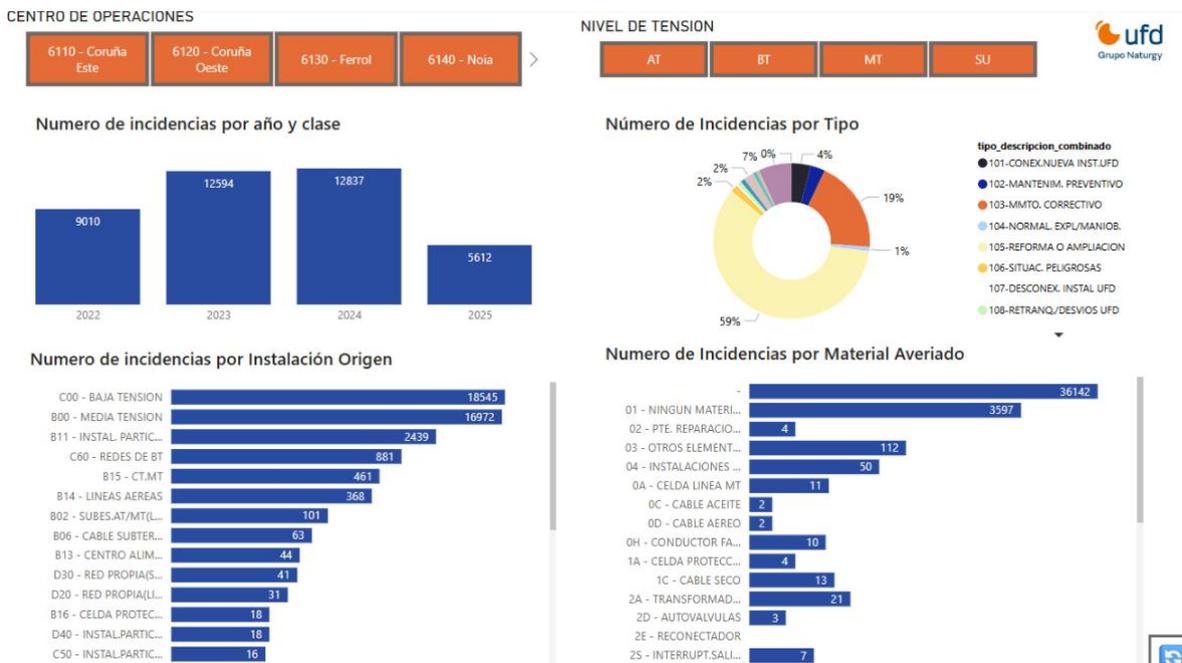


Figura 12: Panel de evolución de la tipificación de incidencias clase 1 (programadas)

Dado que en la imagen capturada del panel no se ha aplicado filtro por nivel de tensión, se observa que la frecuencia de las instalaciones origen B00 (Media Tensión) y C00 (Baja Tensión) es muy similar.

En cuanto al campo Material Averiado, en la mayoría de los registros aparece vacío o sin información relevante, lo cual es coherente con la naturaleza de este tipo de incidencias, ya que, al tratarse de trabajos programados, generalmente no están asociados a fallos de elementos concretos de la red.

Este análisis pone de manifiesto que, en el caso de las incidencias programadas, sería altamente recomendable que la tipificación quedase definida ya en el momento de la creación del trabajo en el sistema GDA.

Dado que estos trabajos son planificados previamente y su ejecución es la que posteriormente da lugar a la incidencia de clase 1 en el SGI, tendría pleno sentido que la información relativa al tipo de actuación, la instalación origen e incluso, cuando sea aplicable, el material afectado, se heredara de forma automática desde el GDA al generar la incidencia.

Este enfoque garantizaría una codificación mucho más precisa y coherente, reduciría la carga administrativa asociada a la cumplimentación manual en el SGI por parte del COR, y, además, facilitaría el posterior análisis y explotación de los datos, tanto para fines operativos como regulatorios.

### Panel Incidencias Imprevistas

A continuación, se muestra el panel correspondiente a las incidencias imprevistas (Clase 2), en el que se representa su evolución y distribución para el periodo 2022–2025.

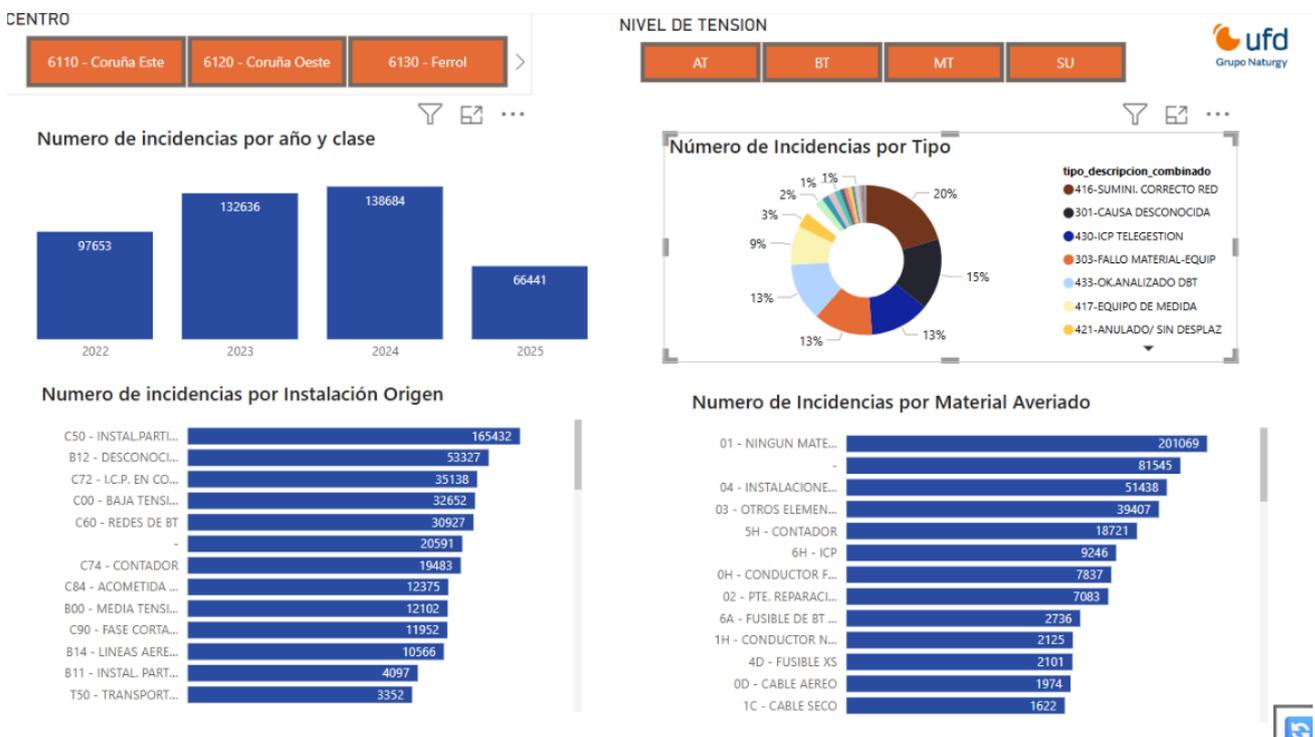


Figura 13: Panel de evolución de la tipificación de incidencias clase 2 (imprevistas).

En este caso, los tipos más frecuentes son:

- 416 – Suministro correcto red
- 301 – Causa desconocida
- 430 – ICP Telegestión
- 303 – Fallo de material/equipamiento

### Panel de Régimen Especial

El tercer panel desarrollado corresponde a las incidencias clasificadas como Régimen Especial (Clase 3), es decir, aquellas que no generan interrupción de suministro.

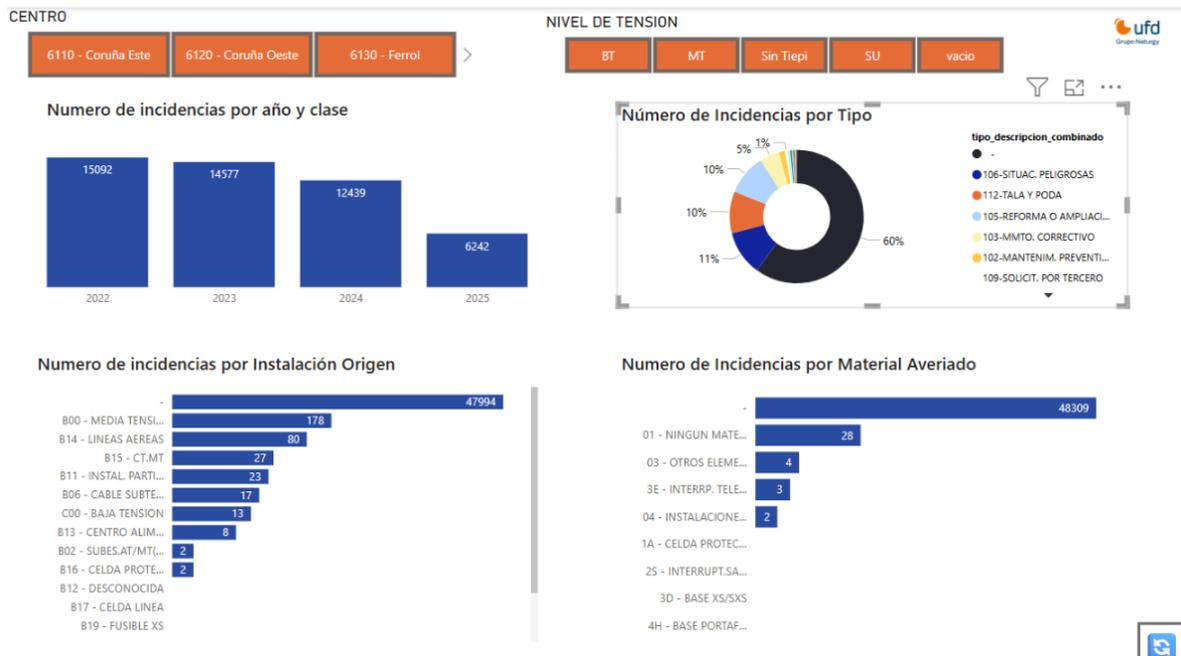


Figura 14: Panel de evolución de la tipificación de incidencias clase 3 (régimen especial, sin interrupción).

### Panel del Navegación Jerárquica de Tipificación

El cuarto panel desarrollado incorpora una navegación jerárquica, que permite explorar la información desde una vista global hasta el detalle más técnico de cada incidencia.

Este panel ofrece una visión agregada del conjunto de incidencias, mostrando para cada Tipo de incidencia:

- El número total de registros.
- El TIEPI acumulado asociado.
- El número total de clientes afectados.

Adicionalmente, permite desglosar esta información por:

- Instalación Origen.
- Material Averiado.

La navegación jerárquica facilita seleccionar un tipo de incidencia concreto y, a partir de ahí, analizar cómo se distribuyen las incidencias según los activos implicados y los materiales asociados, proporcionando una visión detallada y contextualizada.

Este desarrollo constituye una herramienta práctica y eficaz para:

- Revisar la calidad y coherencia de la tipificación actual.
- Detectar patrones operativos o de registro.
- Identificar posibles incoherencias, anomalías o errores en la codificación.
- Orientar acciones de mejora en la limpieza, reorganización y estandarización de la información.

CLASE: 1 - Descargos, 2 - Incidencias, 3 - Régimen Especial

NIVEL DE TENSION: AT, BT, MT, SU

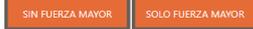
CENTRO: Todas

Año	tipo_descripcion_combinado	N	2023		2024		2025		Total			
			Suma de Pxt_horas	Recuento de CODIGO	Suma de N_CLIEN	Suma de Pxt_horas	Recuento de CODIGO	Suma de N_CLIEN	Suma de Pxt_horas	Recuento de CODIGO	Suma de N_CLIEN	Suma de Pxt_horas
	900-FUERZA MAYOR	964	52.511,90	1212	1406222	48.283,06	945	1018914	57.613,75	4315	4694373	184.976,74
	499-MANIOBRA REPOS.SERVI	740	28,13	138	67577	622,67	86	30477	94,22	242	104794	745,01
	800 - MEDIA TENSION	983	6,72	52	25963	403,52	20	5791	33,36	77	33737	443,60
	01 - NINGUN MATERIAL	522	0,00	12	5892	189,65	6	2596	17,09	19	8810	206,73
	04 - INSTALACIONES PARTIC	559	6,72	39	19776	213,88	14	3195	16,28	56	23530	236,87
	B06 - CABLE SUBTERRANEO	102	0,00	1	295	0,00				2	1397	0,00
	B11 - INSTAL. PARTICULARES	579	21,42	64	34874	195,59	44	16977	33,48	119	56530	250,48
	B14 - LINEAS AEREAS	78	0,00	16	6070	21,97	14	5447	26,68	32	11595	48,65
	B15 - CT.MT			1	1	0,00				1	1	0,00
	B19 - FUSIBLE XS						2	915	0,00	2	915	0,00
	B22 - RECONECTADOR			5	669	1,59	4	593	0,70	9	1262	2,29
	D40 -						1	8	0,00	1	8	0,00
	426-REPOS. FUERA PLAZO	338	0,00	1	13	0,00				8	2270	1,39
	423-NORMALIZ.INST. PART	057	36,13	35	10750	1,87	21	5253	12,85	231	84335	74,13
	422-A PETICION AUTORIDAD	449	141,48	6	1473	285,52	9	12576	679,44	20	17498	1.106,44
	417-EQUIPO DE MEDIDA	284	0,00							46	162284	0,00
	416-SUMINI. CORRECTO RED	389	0,00	6	2536	0,00	5	1171	99,24	22	13846	125,53
	409-OTRA EMPRESA ELECTR.	146	628,81	30	23886	159,48	2833	3920213	3.839.590,30	2954	4036136	3.842.384,38
	407-TERCEROS IDENTIF.	513	2.837,32	57	72509	3.997,69	28	27977	1.824,35	209	290435	17.826,21
	406-INSTAL. PARTICULARES	927	10.545,26	417	206545	10.518,36	203	104019	5.415,07	1723	839637	37.589,03
	405-TERCEROS NO IDENTIF.	725	1.240,29	26	26897	2.116,83	18	8792	660,48	118	77015	7.091,24
	399-MANIOBRA REPOS.SERVI	321	16.070,89	1030	709115	16.397,21	423	352705	4.197,34	2562	1955987	36.795,68
	331-INC.COMUN. UFD	556	93,69	13	3709	24,67	41	6339	161,24	82	13891	339,36
	318-PERSON.TRAB.PARA UFD	231	1.224,24	49	67905	1.047,93	15	21414	192,21	177	245309	3.232,94
	317-OTRA ACTUAC/SIN CORT	331	2,09	9	4663	40,33	6	723	17,88	21	6128	60,30
	Total	147	200.193,95	24910	21709292	205.131,82	15375	15003323	4.064.800,66	90539	83754390	4.660.811,74

Figura 15: Panel de Navegación Jerárquica de Tipificación

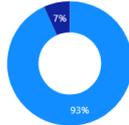
## VISUALIZACIÓN DE INDICADORES SAIDI Y SAIFI

AÑO



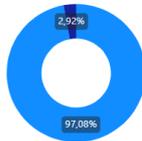
Suma de SAIDI por Nivel de Tensión

NIV\_TENS  
 ● M  
 ● B  
 ● A



Suma de SAIFI por Nivel de Tensión

NIV\_TENS  
 ● M  
 ● B  
 ● A

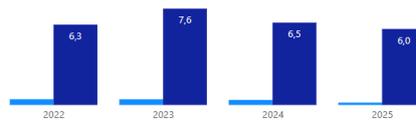


NIVEL DE TENSION



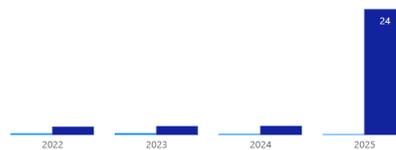
Suma de SAIFI por Año y Clase

CLASE  
 ● 1  
 ● 2



Suma de SAIDI por Año y Clase

CLASE  
 ● 1  
 ● 2



## **Capítulo 6. CONCLUSIONES Y TRABAJOS FUTUROS**

Este proyecto ha permitido analizar en profundidad la información contenida en las incidencias registradas en el SGI, evaluando la calidad de su tipificación y proponiendo mejoras para su futura integración en el sistema ADMS.

Se han cumplido los principales objetivos planteados: acceso, análisis y explotación de los datos del Data Lake, identificación de limitaciones en la tipificación actual, y desarrollo de herramientas visuales que facilitan su revisión y seguimiento.

A continuación, se detallan las principales conclusiones alcanzadas:

### **6.1. EVOLUCIÓN DEL NÚMERO DE INCIDENCIAS**

El análisis del periodo 2022–2024 refleja un incremento significativo en el número de incidencias programadas en los niveles de media y baja tensión, mientras que las incidencias imprevistas muestran una evolución más estable.

Sin embargo, no ha sido posible determinar con claridad qué tipos de actuaciones están detrás de este crecimiento, debido a la falta de trazabilidad y a las limitaciones en la tipificación actual. Esto evidencia la necesidad urgente de mejorar la conexión entre los trabajos ejecutados y las incidencias que generan.

### **6.2. TIPIFICACIÓN DE INCIDENCIAS PROGRAMADAS**

#### **6.2.1 PRINCIPALES LIMITACIONES EN LA CODIFICACIÓN ACTUAL**

- Los campos **TIPO** e **INST\_ORIGEN** son excesivamente genéricos y no permiten identificar ni el tipo de actuación ni el activo afectado.
- Los campos de texto libre (**DESCRIPCIÓN** y **OBSERVACIONES**) contienen información poco estructurada, no explotable de forma automática.

- Partiendo de la codificación de una incidencia programada, no se puede determinar con suficiente precisión que tipo de trabajo se ha llevado a cabo. No hay trazabilidad entre SGI y GDA, sistema en el que se dan de alta los trabajos.

## **6.2.2 PROPUESTA DE MEJORA**

Se propone utilizar la tipificación de GDA como referencia en toda la cadena de gestión de trabajos, desde su planificación hasta el registro de incidencias. Este enfoque permitiría identificar de forma clara el tipo de actuación, el activo implicado y la naturaleza de la incidencia generada, contribuyendo así a mejorar la calidad de los datos operativos y a facilitar su explotación futura.

## **6.3 TIPIFICACIÓN DE INCIDENCIAS IMPREVISTAS.**

Los análisis realizados en este trabajo evidencian que la tipificación actual no garantiza la coherencia ni la homogeneidad en el registro de las incidencias. Es habitual encontrar casos donde dos incidencias asociadas a la misma instalación origen y al mismo material averiado terminan codificadas con tipologías distintas, lo que compromete la calidad y la explotación de la información.

### **6.3.1 PRINCIPALES LIMITACIONES EN LA CODIFICACIÓN ACTUAL DE INCIDENCIAS IMPREVISTAS**

- La estructura actual es compleja, poco homogénea y presenta una elevada redundancia.
- Existen códigos muy genéricos que se aplican de forma ambigua y otros que apenas se utilizan o que no importan información relevante.

### **6.3.2 PROPUESTAS DE MEJORA**

- **Revisión y simplificación de la codificación**  
Reducir la lista de códigos de *Tipo, Instalación Origen y Material Averiado*, manteniendo únicamente aquellos que aporten valor real desde el punto de vista técnico y operativo.

- **Codificación específica según nivel de tensión**

Definir listas acotadas para cada nivel de tensión (SU, BT y MT), eliminando la lógica actual de codificación genérica que permite el uso de cualquier código en cualquier nivel.

- **Fusión de campos redundantes**

Evaluar la conveniencia de fusionar los campos *Instalación Origen* y *Material Averiado* cuando ambos representan información redundante o de escaso valor diferencial.

- **Tipología específica para reenganches automáticos**

Crear un código específico para las incidencias automáticas (interrupciones de menos de 3 minutos), que actualmente se registran incorrectamente como *Tipo 301 – Causa desconocida*, distorsionando los análisis.

- **Incorporación de un campo binario para reparaciones provisionales**

Añadir un campo específico que indique si la incidencia se ha resuelto de forma provisional (pendiente de reparación definitiva). Esto evitaría el uso inadecuado del valor 2 – *Pendiente de reparación definitiva* en el campo *Material Averiado*, que en realidad no representa un material.

- **Evitar pérdida de información en reclasificaciones**

Cuando una incidencia se reclasifica como *Fuerza Mayor* (Tipo 900), se propone conservar siempre el tipo original mediante un campo adicional o una marca específica, garantizando así la trazabilidad completa del evento.

## **6.4 CONCLUSIONES SOBRE LOS INDICADORES DE CALIDAD (SAIDI Y SAIFI)**

Los resultados obtenidos confirman que:

- Las averías en la red de Media Tensión (MT) continúan siendo el factor que mayor impacto tiene en la calidad del suministro eléctrico, tanto en la duración (SAIDI imprevisto) como en la frecuencia de las interrupciones (SAIFI imprevisto), manteniéndose la tendencia histórica reflejada en los indicadores regulados TIEPI y NIEPI.

- Las interrupciones programadas en Baja Tensión (BT) adquieren un peso cada vez más relevante en el SAIDI programado, consecuencia directa del incremento de las actuaciones de mejora, renovación y digitalización de la red de BT llevadas a cabo en los últimos años.

Este análisis pone de manifiesto que, aunque la BT ha estado tradicionalmente excluida de los indicadores oficiales (TIEPI y NIEPI), su peso en las interrupciones programadas es ya significativo. Por su parte, la MT sigue siendo el nivel que concentra de forma clara la duración y frecuencia de las incidencias imprevistas.

## Capítulo 7. BIBLIOGRAFÍA

- [1] Gobierno de España, “Real Decreto 1955/2000, de 1 de diciembre por el que se regulan las actividades de transporte, distribución, comercialización, suministro y procedimientos de autorización de instalaciones de energía eléctrica,” 2000. [En línea]. Disponible en: <https://www.boe.es/eli/es/rd/2000/12/01/1955/con> [Fecha de consulta 15 de marzo de 2025].
- [2] Gobierno de España, “Orden ECO/797/2002, de 22 de marzo, por la que se establecen los procedimientos de medida y control de la continuidad del suministro eléctrico,” 2002. [En línea]. Disponible en: <https://www.boe.es/eli/es/o/2002/03/22/eco797> [Fecha de consulta 15 de marzo de 2025].
- [3] D. Campa Cervero, “*Valoración de la fiabilidad de una red eléctrica de distribución*, Proyecto Fin de Carrera, Escuela Técnica Superior de Ingenieros de Minas, Universidad Politécnica de Madrid,” Sep. 2012. . [En línea]. Disponible en: [https://oa.upm.es/14985/1/PFC\\_Diego\\_Campa\\_Cervero.pdf](https://oa.upm.es/14985/1/PFC_Diego_Campa_Cervero.pdf) [Fecha de consulta 13 de abril de 2025].
- [4] CEER, “CEER Benchmarking Report 6.1 on the Continuity of Electricity and Gas Supply, Council of European Energy Regulators, Ref: C18-EQS-86-03, Bruselas, Bélgica, Jul. :018,”2016. [En línea]. Disponible en: [https://www.ceer.eu/wp-content/uploads/2024/04/C18-EQS-86-03\\_Benchmarking\\_Report\\_6.1.pdf](https://www.ceer.eu/wp-content/uploads/2024/04/C18-EQS-86-03_Benchmarking_Report_6.1.pdf) [Fecha de consulta 13 de abril de 2025].
- [5] España, *Ley 54/1997, de 27 de noviembre, del Sector Eléctrico*. 1997. [En línea]. Disponible en: <https://www.boe.es/eli/es/l/1997/11/27/54/con> [Fecha de consulta 13 de abril de 2025].
- [6] UFD, “Quienes Somos.” (2024), [En línea]. Disponible en: <https://www.ufd.es/quienes-somos/> [Fecha de consulta 14 de abril de 2025].
- [7] UFD-Programación, “Informe Mensual de seguimiento de la Calidad de Suministro”, Documento interno, marzo 2025.
- [8]
- [9] Bendezu, K.. “DIAGRAMA UML Y ARQUITECTURA DEL SISTEMA“. *Sistemas Distribuidos* 2013. Febrero, 2013. <http://comparape.blogspot.com.es/2013/02/diagrama-uml-y-arquitectura-del-sistema.html>.
- [10] Herrero Alcántara, T. “Big Data: ¿Moda u oportunidad de negocio para el emprendedor?”, Think Big, Octubre 2014. <http://blogthinkbig.com/big-data-emprendedor/>.
- [11] Loeffler, B. “Cloud Computing: What is Infrastructure as a Service”, Microsoft Technet Magazine, October 211. <https://technet.microsoft.com/en-us/magazine/hh509051.aspx>
- [12] Vlassis, N.A.; Papakonstantinou, G.; Tsanakas, P. *Dynamic sensory probabilistic maps for mobile robot localization*. Source: Proceedings. 1998 IEEE / RSJ International Conference on Intelligent Robots and Systems. Innovations in Theory, Practice and Applications (Cat. No.98CH36190) New York, NY, USA: IEEE, 1998.p.718-23 vol.2 of 3 vol. xlv+2010 pp. 11.

# ANEXO 1: ESTRUCTURA Y CONTENIDO DE LAS BASES DE DATOS

En este Anexo se describen los campos incluidos en la extracción del Data Lake, indicando el nombre del campo, el tipo de dato correspondiente y una breve descripción de su contenido.

## ANEXO 1.1 Base de datos de Incidencias

A continuación se muestran los campos disponibles en la base de datos de incidencias.

Nombre del campo	Tipo de dato	Descripción
CODIGO	Número entero	Código numérico para identificar la incidencia
Tipo	Categorico (numérico)	Código numérico que identifica el motivo de la incidencia.
Desc. Tipo	Texto	Nombre descriptivo del motivo de la incidencia.
Ins.Origen	Categorico (texto)	Código compuesto por una letra (B, C, D, G y T) y una serie de números, identificador de la instalación origen de la incidencia.
Desc_Origen	Texto	Nombre descriptivo de la Instalación Origen
Niv_Tens	Categorico (texto)	Identificador del nivel de tensión. A AT B BT M MT S SU T ST Vacías
Ins_Afectada_SGD	Numérico	Código de la instalación afectada con el que se identifica la instalación en la aplicación Empresarial Geográfica (IGEA)
Nombre_Instalación	Texto	Nombre de la instalación (línea de MT, centro de Transformación...)
Mat. Averiado	Categorico (texto)	Código que permite identificar el material averiado.
Desc. Mat Averiado	Texto	Nombre del material averiado.

Nombre del campo	Tipo de dato	Descripción
Estado	Catagórico (texto)	Código que permite la identificación del estado en el que se encuentra la incidencia: PT Pendiente EB Enviada Brigada CL Causa Localizada ER En Resolución SR Servicio Repuesto EM Enviada a Mantenimiento
Clase	Catagórico (numérico)	1 : Asociados a trabajos programados 2: Asociados a incidencias imprevistas 3: Sin interrupción
Potencia	Numérico (entero)	Potencia afectada en kVA para CT's de UFD y en KW contratados para CT's particulares.
N_Clien	Numérico (entero)	Número de clientes afectados por la incidencia, esto es el número de puntos de suministro afectados.
PxT	Numérico (entero)	Sumatorio del producto de la potencia de cada CT por el tiempo de interrupción.
Duración	Numérico (entero)	Duración de la incidencia en minutos
FECHA-DET	Fecha	Fecha de detección de la incidencia
HORA-DET	Hora	Hora de detección de la incidencia
COD_GRUPO	Número entero	Código numérico para identificar la incidencia padre.
Centro	Catagórico (numérico)	Código numérico que identifica el centro de trabajo.
NOT_PLAZO	Binario (SI/NO)	Indica si han cumplido los requisitos de notificación en plazo para considerar la interrupción como programada.
TIPO_TRABAJO	Catagórico	D= Descargo IT= Intervención TB= Trabajo BT RE= Reg Especial CA= Trabajo CAMP ID= Intervención desc. IR= Intervención reg. esp
Mercado	Binario	Si= se produce perdida de mercado y por tanto hay interrupción de suministro (para trabajos programados)
Sin_Tiepi	Binario (SI/NO)	Si hay interrupción que suponga TIEPI
Documentados	Binario (S/N)	Campo para indicar si se ha localizado al causante de la avería en el caso de que sea un tercero.
CODIGO_DESCARGO	Texto	Código del trabajo en otro sistema (SGM, GDA,SGT, etc.)

<b>Nombre del campo</b>	<b>Tipo de dato</b>	<b>Descripción</b>
Observaciones	Texto	Texto introducido por el COR durante la resolución de la incidencia o del trabajo programado
Localización	Categorico (numérico)	Matrícula del CT causante de la avería.
Descripción	Texto	Breve descripción de la incidencia.
CTS_PARTICULARES	Número	Identifica la instalación particular que ha dado origen a la avería o al trabajo programado.

## ANEXO 1.2 Descripción de Variable “TIPO”

Código de Tipo	Descripción Tipo
101	CONEX.NUEVA INST.UFD
102	MANTENIM. PREVENTIVO
103	MMTO. CORRECTIVO
104	NORMAL. EXPL/MANIOB.
105	REFORMA O AMPLIACION
106	SITUAC. PELIGROSAS
107	DESCONEX. INSTAL UFD
108	RETRANQ./DESVIOS UFD
109	SOLICIT. POR TERCERO
112	TALA Y PODA
114	A PETICION DE UFD
120	CAMP- PROTECCION
121	CAMP- TELECONTROL
122	CAMP- TELEGESTION CT
123	CAMP-CONTROL Y MEDID
181	MNTO.ACORDADO TERCER
198	OTRAS EMP. ELECTRIC
199	SOLICITADO CLIENTE
201	ANIMALES
202	ARBOLES
208	INUNDACION/MOV.TIERR
209	ATMOSFÉRICAS
301	CAUSA DESCONOCIDA
302	SOBRECARGAS
303	FALLO MATERIAL-EQUIP
304	LOCALIZ/NORMAL EXPLO
305	REPARACION URGENTE
317	OTRA ACTUAC/SIN CORT
318	PERSON.TRAB.PARA UFD
328	REPOS. CORTE ERRONEO
331	INC.COMUN. UFD
399	MANIOBRA REPOS.SERVI
405	TERCEROS NO IDENTIF.
406	INSTAL. PARTICULARES
407	TERCEROS IDENTIF.
409	OTRA EMPRESA ELECTR.
410	HUELGAS LEGALES
416	SUMINI. CORRECTO RED
417	EQUIPO DE MEDIDA
419	DESLASTRE FREC/CARGA
420	ANULADO/ CON DESPLAZ
421	ANULADO/ SIN DESPLAZ
422	A PETICION AUTORIDAD
423	NORMALIZ.INST. PART
426	REPOS. FUERA PLAZO
427	REPOS. MAQ OXIG/ESEN
430	ICP TELEGESTION
433	OK.ANALIZADO DBT
499	MANIOBRA REPOS.SERVI
900	FUERZA MAYOR

### ANEXO 1.3 Descripción de Variable “Ins Origen”

Código de Ins_Origen	Instalación Origen
B00	MEDIA TENSION
B02	SUBES.AT/MT(LADO MT)
B06	CABLE SUBTERRANEO
B11	INSTAL. PARTICULARES
B12	DESCONOCIDA
B13	CENTRO ALIMENTACION
B14	LINEAS AEREAS
B15	CT.MT
B16	CELDA PROTECCION
B17	CELDA LINEA
B18	FUSIBLE MT CT
B19	FUSIBLE XS
B20	CABLE PUENTE/VARILLA
B21	AUTOSECCIONADOR
B22	RECONECTADOR
B31	RED AJENA SSEE
B32	RED AJENA LINEAS/CAB
C00	BAJA TENSION
C11	CUADRO BT
C21	CABLES PUENTE BT
C50	INSTAL.PARTICULARES
C60	REDES DE BT
C70	DESCONOCIDA
C71	FBLE. FUNDIDO ACOMET
C72	I.C.P. EN CONTADOR
C73	I.C.P. EN VIVIENDA
C74	CONTADOR
C84	ACOMETIDA AVERIADA
C85	C.G.P QUEMADA
C88	FASES COMUNICADAS
C89	FASE/NEUTRO COMUNIC.
C90	FASE CORTADA
C91	NEUTRO CORTADO
C94	FUSIBLE FUNDIDO CT
C95	CONTACTOR
C96	RELOJ T/N
C97	INTERRUPTOR BT
C98	FASE TIERRA
C99	CG DERIV.QUEMADA
D20	RED PROPIA(LIN/CAB)

D30	RED PROPIA(SSEE)
D31	RED AJENA (SSEE)
D32	RED AJENA (LIN/CAB)
D40	INSTAL.PARTICULARES
D50	OTRAS INSTALACIONES
G00	GENERACION
T50	TRANSPORTE AJENA
T60	TRANSPORTE PROPIA

## ANEXO II: GRÁFICAS Y RESULTADOS PARCIALES.

Como primer paso en el análisis de resultados, se ha estudiado la evolución del número total de incidencias registradas entre 2022 y 2024, clasificadas por niveles de tensión. Este análisis permite identificar tendencias generales y valorar la evolución de la frecuencia de incidencias en cada nivel de tensión (AT, MT, BT y SU). Los resultados se recogen en el Gráfico 9 y en la **¡Error! No se encuentra el origen de la referencia..**

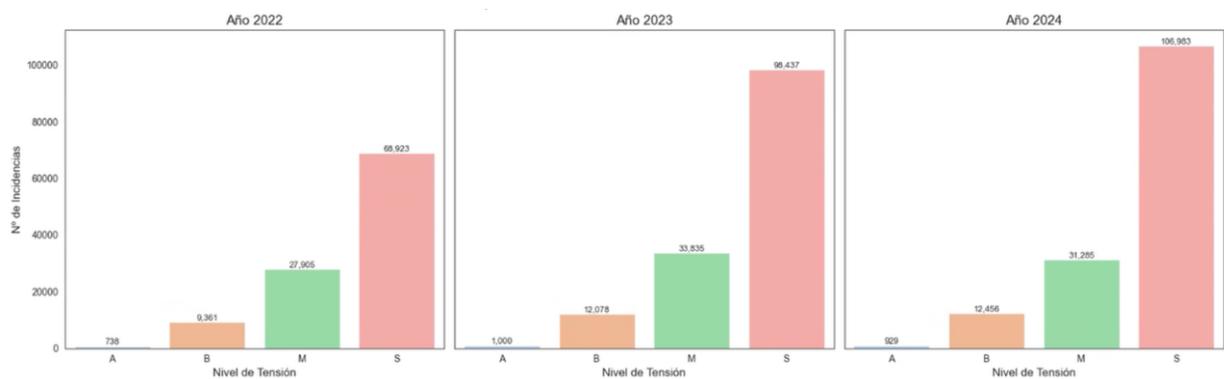


Gráfico 9: Número de Incidencias por Nivel de Tensión (2022-2024). (A: AT, B: BT, M: MT y S: SU).  
Elaboración propia

	Año 2022	Año 2023	Año 2024
<b>AT</b>	738	1.000	929
<b>MT</b>	27.905	33.835	31.285
<b>BT</b>	9.361	12.078	12.456
<b>SU</b>	68.923	98.437	106.983

Tabla 10: Evolución del Número de Incidencias. Elaboración Propia

Análisis por nivel de tensión:

- **AT:**

El número de incidencias es significativamente menor que en el resto de niveles, lo cual es coherente con el diseño de estas redes, que suelen estar malladas y contar con equipos redundantes, lo que reduce la probabilidad de que el fallo de un equipo produzca interrupciones de suministro.

- **MT**

Se observa un aumento de incidencias en 2023 respecto al 2022, seguido de un ligero descenso en 2024.

- **BT:**  
El número de incidencias muestra una tendencia creciente constante durante el periodo analizado.
- **SU:**  
Las interrupciones afectan a un único punto de suministro: Se aprecia un incremento notable en el periodo analizado. Para profundizar en este comportamiento, se propone analizar el campo *Ins\_Origen* en registros SU, con el fin de determinar si este aumento se relaciona con un tratamiento diferente de las averías particulares.

A continuación se realizará una segmentación por tipo de incidencias (programada o imprevista) para seguir avanzando en el análisis causal.

## EVOLUCIÓN DE LAS INCIDENCIAS POR TIPOLOGÍA Y NIVEL DE TENSIÓN (2022-2024).

Tras el análisis global, se realiza una segmentación de las incidencias por tipología, diferenciando entre programadas (Clase 1) e imprevistas (Clase 2). Esta clasificación permite valorar si los aumentos observados responden a trabajos planificados o a averías inesperadas. Los resultados consolidados se muestran en el Gráfico 10.

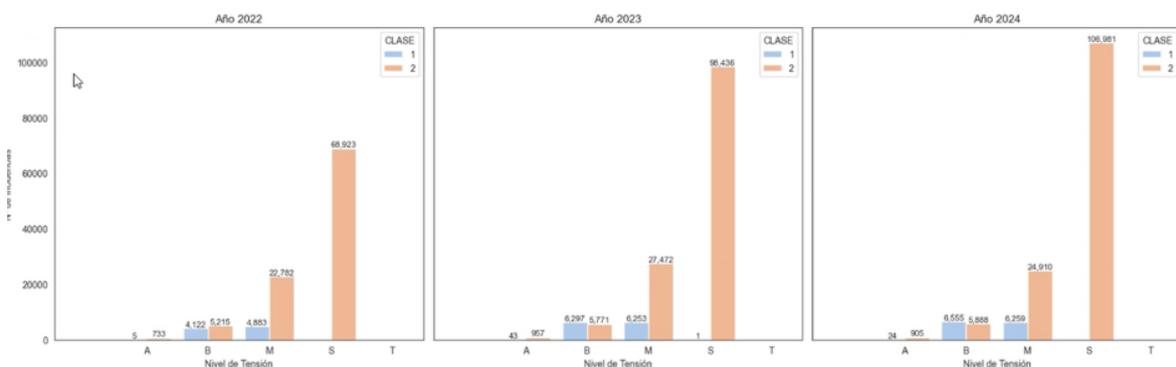
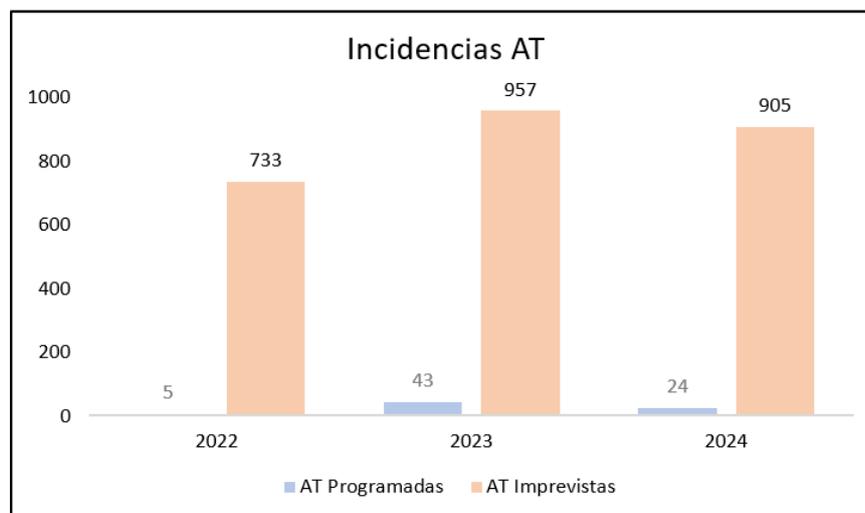


Gráfico 10: Incidencias por Tipología (1: Programadas 2: Imprevistas) y Niveles de Tensión (A: AT, B:BT, M:MT y S:SU) Elaboración propia.

Dado que los valores entre niveles presentan diferencias significativas, se procede a analizar cada uno de ellos por separado a continuación.

### EVOLUCIÓN DE LAS INCIDENCIAS DE AT POR TIPOLOGÍA (2022-2024).

En el Gráfico 11 se observa que el número de **incidencias programadas** en AT es muy reducido, aunque experimentó un pico en 2023 (de 5 a 43), seguido de una disminución en 2024. Las **imprevistas** se mantienen estables y en valores bajos, lo que es coherente con la configuración mallada y la capacidad de maniobra de estas redes.



*Gráfico 11: Incidencias de AT por Tipología (Programadas e Imprevistas). Elaboración propia.*

### EVOLUCIÓN DE LAS INCIDENCIAS DE MT POR TIPOLOGÍA (2022-2024).

El **¡Error! No se encuentra el origen de la referencia.** muestra que las incidencias imprevistas en MT se mantuvieron estables entre 2022 y 2023, con una reducción cercana al 10% en 2024. Las programadas, en cambio, aumentaron cerca del 30% entre 2022 y 2023, manteniéndose en valores similares en 2024. Este comportamiento puede reflejar un incremento de la actividad planificada asociada a la mejora de instalaciones..

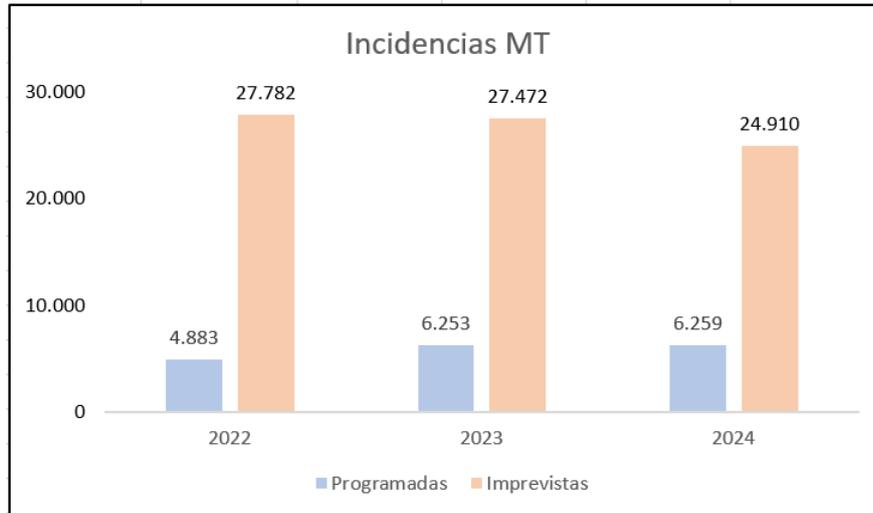


Gráfico 12: Incidencias de MT por Tipología (Programadas e Imprevistas) Elaboración propia.

## EVOLUCIÓN DE LAS INCIDENCIAS DE BT POR TIPOLOGÍA (2022-2024).

En el

Gráfico 13: Incidencias en BT por Tipología (Programadas e Imprevistas) Elaboración propia. se aprecia un crecimiento sostenido en ambos tipos de incidencias. Las **programadas** pasaron de 4.122 en 2022 a 6.555 en 2024, lo que representa un aumento del 59% y las **imprevistas** también aumentaron, aunque en menor proporción (de 5.215 a 5.888), un 13% acumulado.

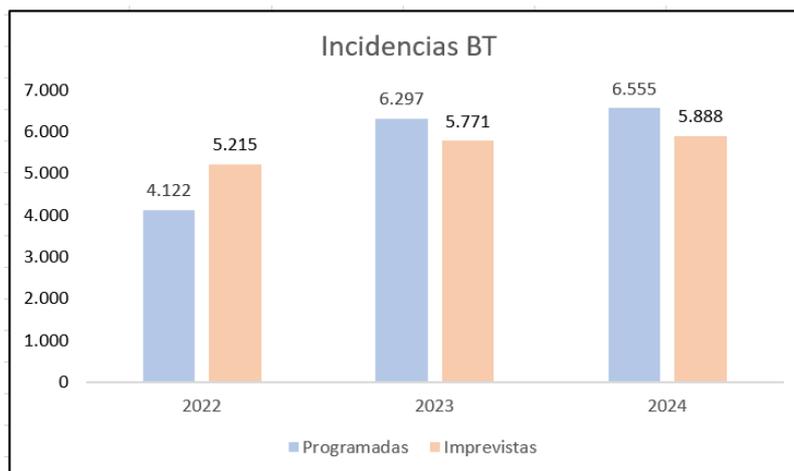


Gráfico 13: Incidencias en BT por Tipología (Programadas e Imprevistas) Elaboración propia.

## EVOLUCIÓN DE LAS INCIDENCIAS SU (2022-2024).

En el **¡Error! No se encuentra el origen de la referencia.** se representa la evolución de incidencias clasificadas como SU. Como se explicó anteriormente, estas interrupciones afectan a un único punto de suministro y se corresponden exclusivamente con incidencias imprevistas. El número casi se duplica entre 2022 y 2024, lo que refuerza la necesidad de revisar la forma en que se registran.

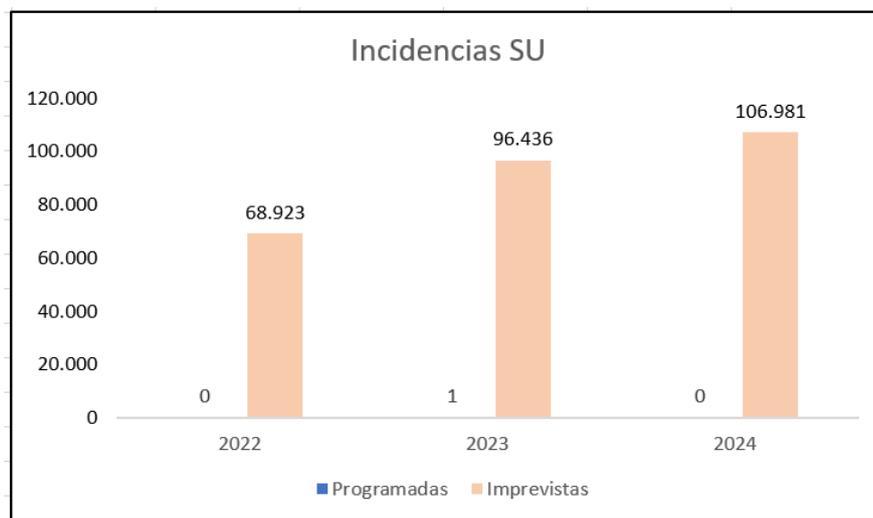


Gráfico 14: Incidencias SU por Tipología (Programadas e Imprevistas) Elaboración propia.

## TIPIFICACIÓN DE LAS INCIDENCIAS PROGRAMADAS

### EVOLUCIÓN DEL NÚMERO DE INCIDENCIAS PROGRAMADAS (2022-2024 Y PRIMER TRIMESTRE DE 2025).

En el Gráfico 15 se muestra la evolución de las incidencias programadas (Clase 1) entre 2022 y 2024, incluyendo el primer trimestre de 2025.

Se aprecia un incremento significativo en 2023 respecto a 2022, tanto en media tensión (MT) como en baja tensión (BT). Esta tendencia se mantiene durante 2024, lo que sugiere un posible cambio en los tipos de trabajos ejecutados o en la forma en que estos se tipifican en el SGI..

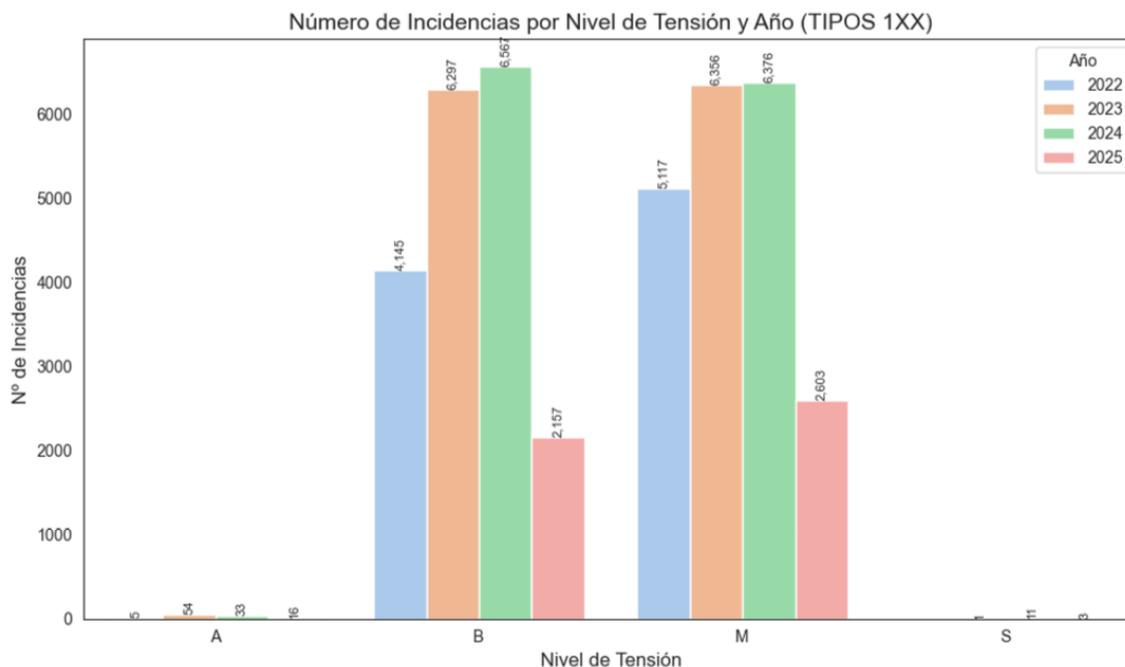


Gráfico 15: Evolución del número de incidencias programadas por nivel de tensión y año. Elaboración propia.

## **EVOLUCIÓN DE LAS INCIDENCIAS PROGRAMADAS DE MT POR TIPO DE INCIDENCIA.(2022-2024 Y PRIMER TRIMESTRE DE 2025).**

Para entender qué trabajos han provocado el aumento observado, se analiza la evolución de las incidencias programadas en MT por tipo de incidencia (*TIPO* en *Master\_SGI*).

En el Gráfico 16 se muestra que los tres los tipos más empleados a lo largo de estos años son:

- 105 Reforma o ampliación
- 199 Solicitado por el cliente
- 103 Mantenimiento correctivo

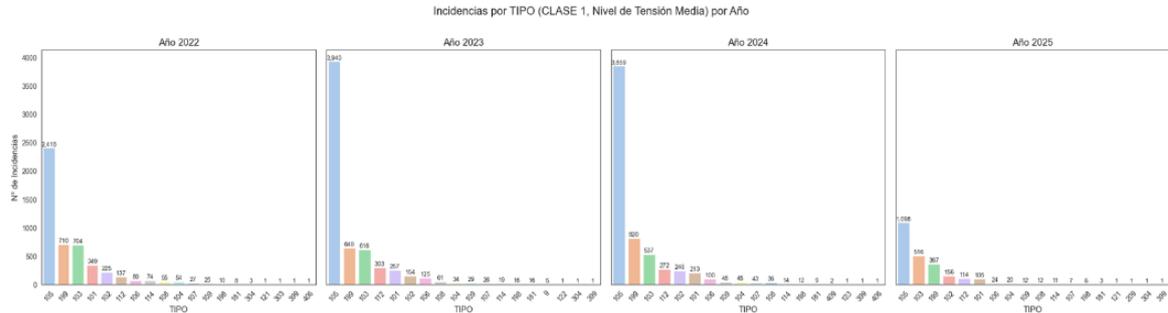


Gráfico 16: Evolución del número de incidencias programadas de MT por tipo de incidencia. Elaboración propia.

### EVOLUCIÓN DEL NÚMERO DE INCIDENCIAS PROGRAMADAS DE BT POR TIPO DE INCIDENCIA.(2022-2024 Y PRIMER TRIMESTRE DE 2025).

De forma análoga, el Gráfico 17 recoge los datos correspondientes a BT. También aquí se repiten como más frecuentes los mismos tres códigos:

- 105: Reforma o ampliación
- 199: Solicitado por el cliente
- 103: Mantenimiento correctivo

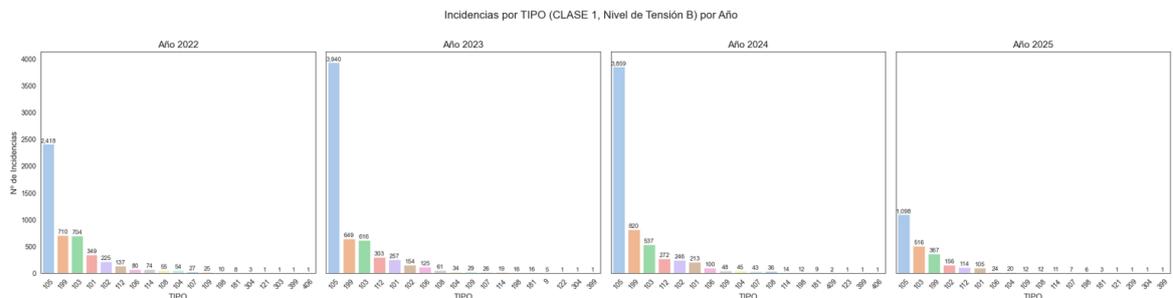


Gráfico 17: Evolución del número de incidencias programadas de BT por tipo de incidencia. Elaboración propia.

### EVOLUCIÓN INCIDENCIAS PROGRAMADAS DE MT Y BT POR INSTALACIÓN ORIGEN.(2022-2024 Y PRIMER TRIMESTRE DE 2025).

Finalmente, se analiza la variable *Ins\_Origen* en los registros de MT y BT con incidencias programadas. El objetivo es evaluar si este campo aporta información relevante sobre el tipo de instalación afectada.

Como se muestra en el Gráfico 18, la práctica totalidad de incidencias programadas en BT están asociadas al código C00 (red BT), y en MT al código B00 (red MT). Esta codificación genérica no ofrece detalles adicionales que permitan una clasificación más específica, lo que limita su utilidad para análisis operativos o estratégicos.

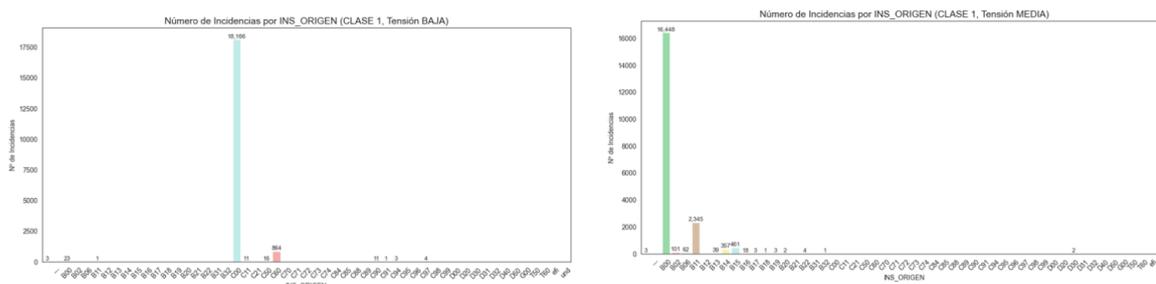


Gráfico 18: Evolución del número de incidencias programadas de MT y BT por instalación de origen en el periodo estudiado. Elaboración propia.

## **ANÁLISIS DE LOS CAMPOS DE TEXTO LIBRE EN LAS INCIDENCIAS DE MT Y BT DEL AÑO 2024 Y DEL PRIMER TRIMESTRE DE 2025.**

Como complemento al análisis de los códigos estructurados (*TIPO* e *Ins\_Origen*), se llevó a cabo una revisión de los campos de texto libre *Observaciones* y *Descripción*, presentes en los registros de incidencias programadas.

La exploración se ha centrado en las incidencias de MT y BT correspondientes a los años 2024 y primer trimestre de 2025 seleccionando las incidencias de Clase 1.

Los resultados, representados en el Gráfico 19, muestran las palabras más frecuentes utilizadas en los campos *Descripción* y *Observaciones*. Se observa que términos como *descargo*, *trabajo*, *cambio solicitud o sustitución* aparecen de forma reiterada tanto en MT como en BT si bien no aportan información relevante para determinar el tipo de trabajo. Aparecen algunos términos asociados al activo sobre el que se trabaja, como *trafo* (*transformador*), *celdas*, *CBTI* (*cuadro de baja tensión*), *celdas*, entre otros. No obstante, la variedad de términos y la falta de estandarización en el vocabulario dificultan su explotación automática.

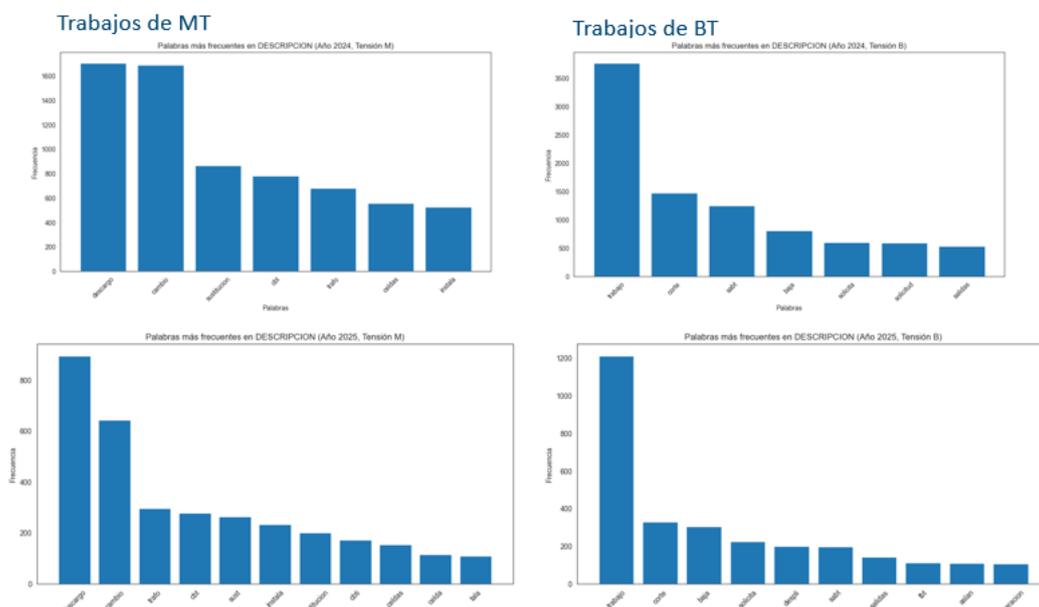


Gráfico 19: Palabras con mayor frecuencia en el campo “DESCRIPCIÓN y OBSERVACIONES” para incidencias programadas en MT y BT, durante 2024 y primer trimestre de 2025.

## ANÁLISIS DE LA CODIFICACIÓN DE TRABAJOS EN EL SISTEMA GDA (2022–2024)

El objetivo de este análisis es estudiar cómo se codifican los trabajos en el sistema GDA a través de los campos *Tipo de actuación* y *Clase de actividad*, para evaluar si se puede aplicar esta tipificación a las incidencias que se van a generar para ejecutar estos trabajos posteriormente.

### DISTRIBUCIÓN DE TRABAJOS POR TIPO DE ACTUACIÓN

En primer lugar, se ha analizado la frecuencia de aparición de los distintos valores del campo *Tipo de actuación* del sistema GDA, con el objetivo de evaluar si esta codificación podría utilizarse como base para clasificar las incidencias programadas en los sistemas de operación.

Este campo admite 18 valores distintos, que agrupan los trabajos en categorías como: *Correctivo inmediato*, *Correctivo programable*, *Reglamentación*, *Extensión de red*, *Tala y poda*, *Adecuación*, entre otros.



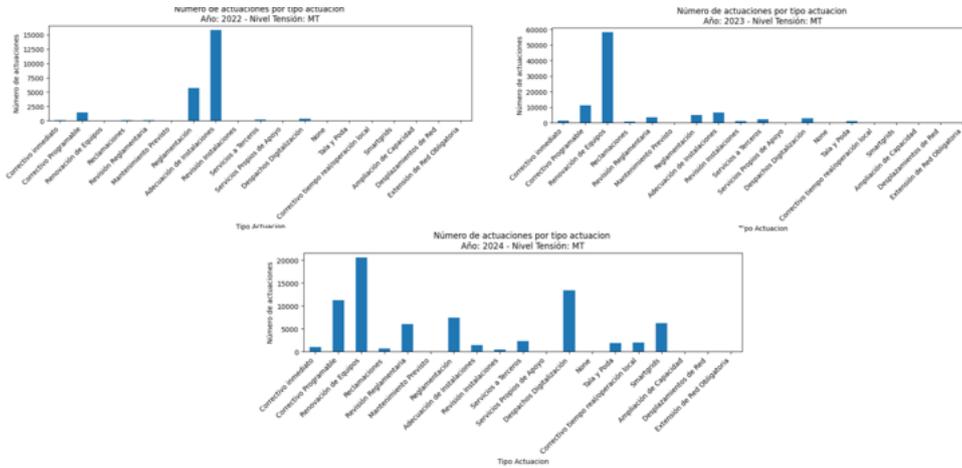


Gráfico 22: Distribución de trabajos en MT por Tipo de Actuación en GDA (2022–2024). Elaboración propia.

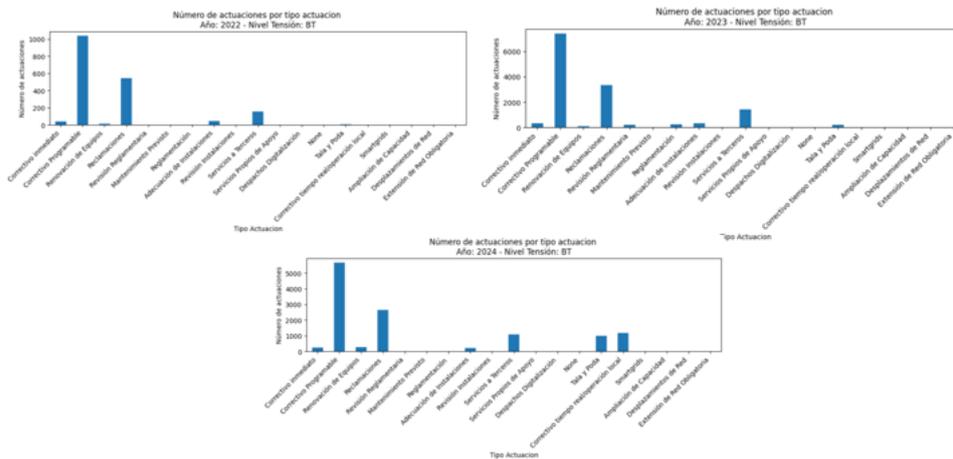


Gráfico 23: Distribución de trabajos en BT por Tipo de Actuación en GDA (2022–2024). Elaboración propia.

## DISTRIBUCIÓN DE TRABAJOS POR CLASE DE ACTIVIDAD

A continuación, se ha analizado el campo *Clase de actividad* del sistema GDA, con el objetivo de evaluar su utilidad como posible elemento de clasificación para las incidencias derivadas de trabajos. Este campo admite más de 280 valores distintos.

El análisis se ha realizado para el periodo 2022–2024. En primer lugar, se estudió la distribución global de todas las clases de actividad registradas en GDA durante este trienio, independientemente del nivel de tensión. Los resultados se representan en el Gráfico 24: Distribución trabajos por Clase Actividad en GDA (2022–2024). Elaboración propia.

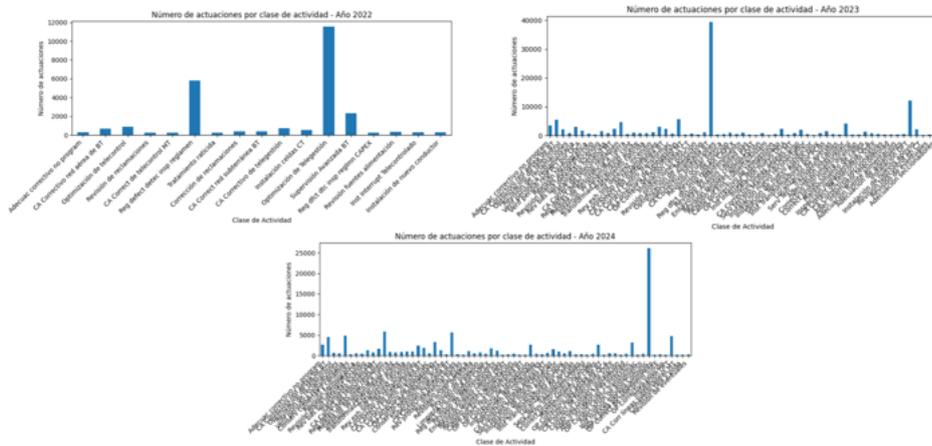


Gráfico 24: Distribución trabajos por Clase Actividad en GDA (2022–2024). Elaboración propia.

Posteriormente, se desagregaron los datos por nivel de tensión, con el objetivo de identificar patrones específicos. Se puede observar que en 2022 se emplearon en todos los niveles de tensión un número más reducido de clases, mientras que en los años posteriores se amplía la variedad de clases utilizadas. En AT, a partir del 2023 se mantiene una distribución bastante homogénea (Gráfico 25).

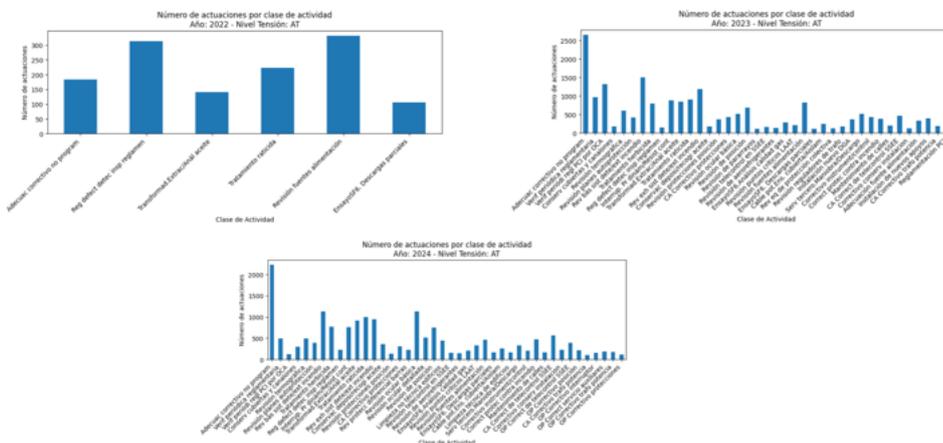


Gráfico 25: Distribución trabajos AT por Clase Actividad en GDA (2022–2024).

En MT y BT, se observa que la mayoría de los trabajos se concentran en un conjunto muy reducido de clases de actividad, lo que sugiere que se podría simplificar la actual clasificación. Los resultados se recogen en el Gráfico 26 (MT) y Gráfico 27 (BT).

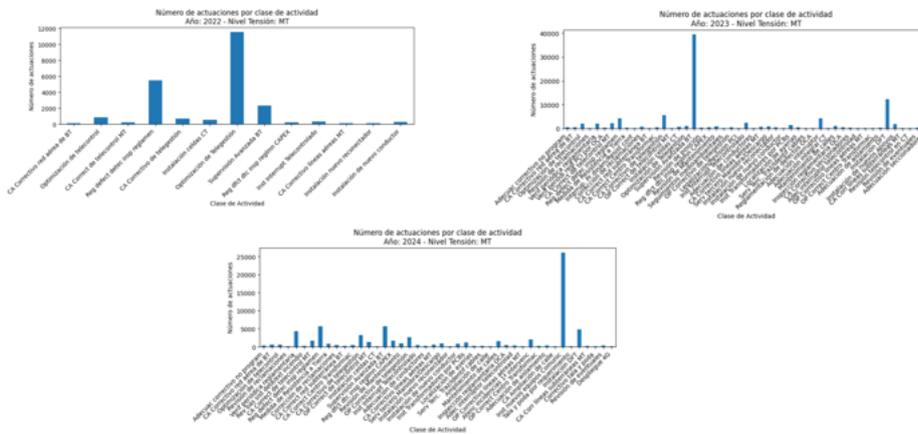


Gráfico 26: Distribución trabajos MT por Clase Actividad en GDA (2022–2024)

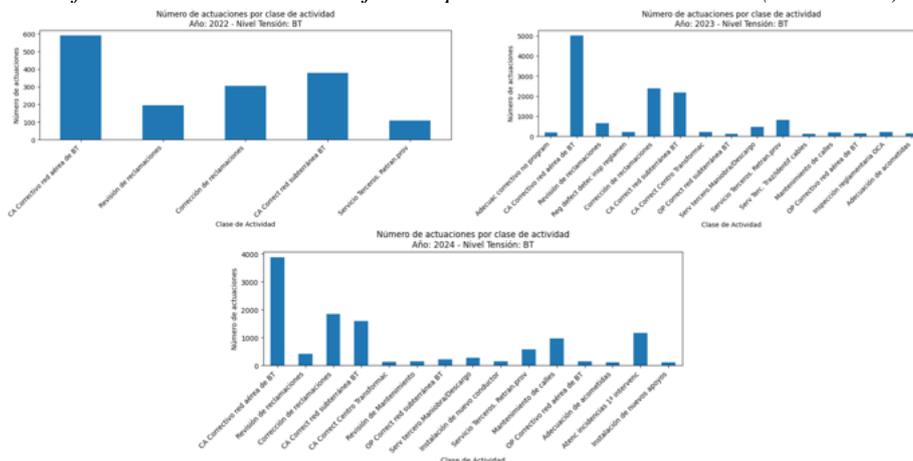


Gráfico 27: Distribución trabajos BT por Clase Actividad en GDA (2022–2024)

Por otro lado, es importante señalar que no todos los trabajos codificados en GDA generan incidencias de Clase 1 (programadas). De hecho, una proporción significativa corresponde a incidencias de Clase 3. Este aspecto deberá tenerse en cuenta si se plantea una integración de esta codificación con el sistema de gestión de incidencia.

## TIPIFICACIÓN DE LAS INCIDENCIAS IMPREVISTAS.

Dada la diversidad de casuísticas que presentan las incidencias imprevistas, se ha optado por realizar un análisis diferenciado por nivel de tensión: SU, BT y MT.

Las incidencias imprevistas de AT no se incluyen en este primer estudio, ya que actualmente no están plenamente integradas en el SGI. Además, su volumen es muy reducido en

comparación con el resto de niveles. No obstante, si en el futuro se considera necesario, podría abordarse un análisis específico utilizando los datos del sistema SGI de AT.

## EVOLUCIÓN DEL NÚMERO DE INCIDENCIAS IMPREVISTAS TIPO SU (SUMINISTRO ÚNICO) (2022-2024 Y PRIMER TRIMESTRE DE 2025).

En el Gráfico 28 se muestra la evolución de las incidencias imprevistas que afectan a un único punto de suministro (SU, Clase 2) registradas entre los años 2022 y 2024, incluyendo el primer trimestre de 2025. Las incidencias se han clasificado según el valor del campo *Tipo*.

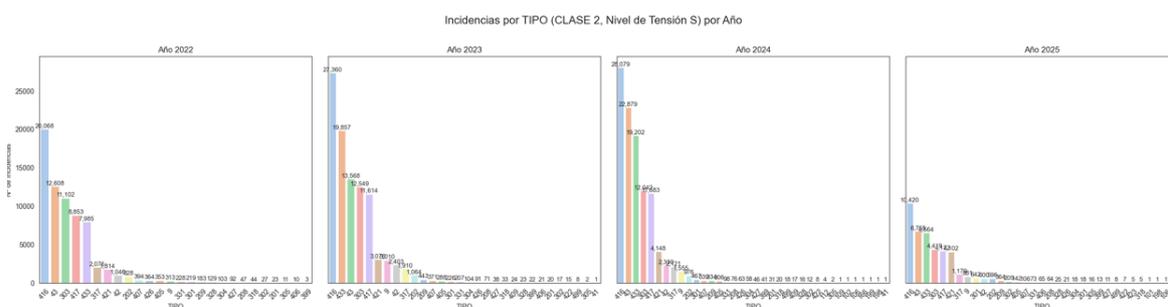


Gráfico 28: Incidencias Imprevistas para Suministros Únicos por Año y Tipo

Durante el periodo analizado, los códigos más frecuentes han sido los siguientes:

416	SUMINI. CORRECTO RED
417	EQUIPO DE MEDIDA
420	ANULADO/ CON DESPLAZ
421	ANULADO/ SIN DESPLAZ
430	ICP TELEGESTION
433	OK.ANALIZADO DBT
303	FALLO MATERIAL-EQUIP
317	OTRA ACTUAC/SIN CORT

En el caso de los códigos 420 y 421, su frecuencia se incrementa significativamente, pasando de aproximadamente 2.660 incidencias en 2022 a cerca de 6.500 en 2024. Dado que estas incidencias corresponden con avisos anulados, no se considera necesario un análisis adicional del campo *Inst\_Origen*.

También se observa un aumento significativo en los tipos 430 y 433, que prácticamente duplican su volumen entre 2022 y 2024 (de 20.593 a 42.081 incidencias). Estas incidencias suelen resolverse de forma automática, por lo que tampoco requieren un análisis en profundidad.

Se ha considerado necesario profundizar en tres de las tipologías más frecuentes de incidencias SU:

Tipo 303 (Fallo material – Equipo),

Tipo 317 (Otra actuación sin corte), y

Tipo 416 (Suministro correcto – Red).

Para cada una de ellas, se ha revisado la información registrada en los campos *Ins\_Origen* y *Mat\_Averiado*, con el objetivo de determinar si estos campos, junto con el campo Tipo, permiten identificar con suficiente precisión el origen y la naturaleza de la incidencia.

### Análisis del Tipo 303 – Fallo Material (Nivel SU)

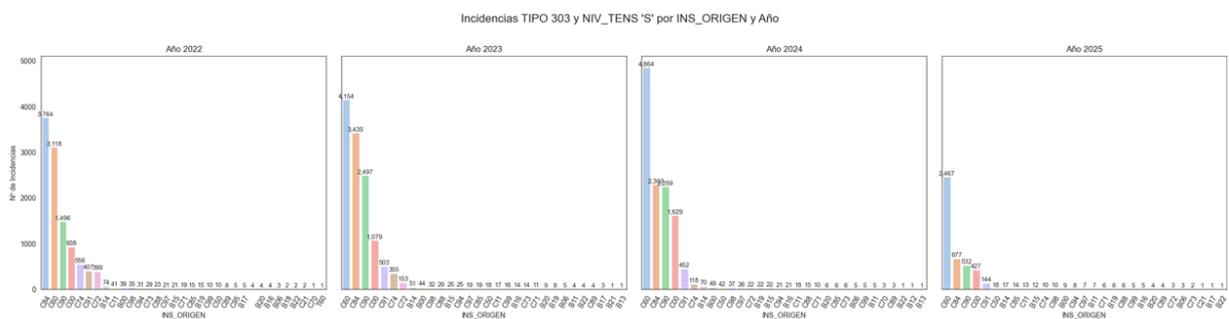
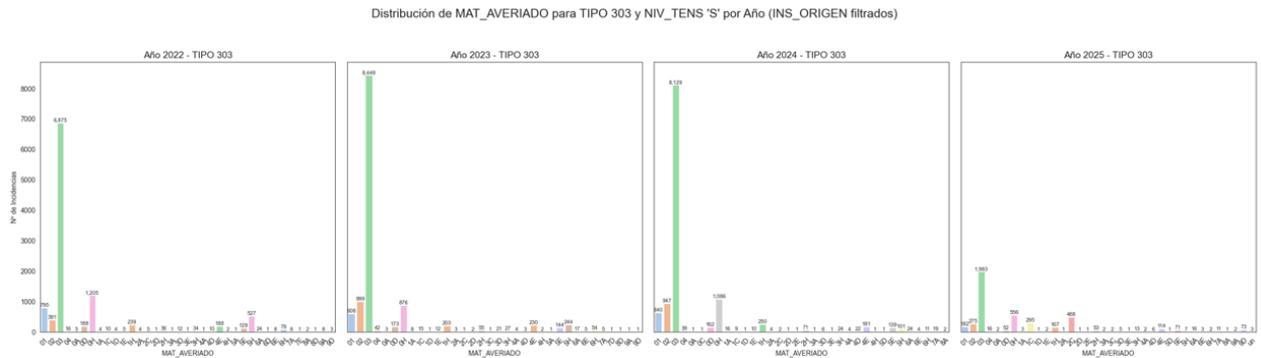


Gráfico 29: Distribución del campo *INS\_ORIGEN* para incidencias Tipo 303, Nivel SU, por año.

En el Gráfico 29 se representa la frecuencia de valores registrados en el campo *Ins\_Origen*. Se observa que, en todos los años analizados, la mayoría de las incidencias se concentran en un número muy reducido de códigos, concretamente en

C00	BAJA TENSION
C60	REDES DE BT
C74	CONTADOR
C84	ACOMETIDA AVERIADA
C90	FASE CORTADA
C91	NEUTRO CORTADO

Aunque existen otros códigos registrados, su frecuencia es muy baja y dispersa, lo que dificulta extraer conclusiones relevantes a partir de este campo.



A2-Gráfico 30: Distribución del campo MAT\_AVERIADO para incidencias Tipo 303, Nivel SU, por año.

El Gráfico 30 muestra la distribución del campo *Mat\_Averiado*. Se puede apreciar que el código más empleado es el *03:OTROS ELEMENTOS*, lo cual no aporta información adicional, al no especificarse el material afectado.

En conclusión, para las incidencias *Tipo 303 - Fallo Material*, el campo *Ins\_Origen* podría simplificarse reduciendo el número de categorías utilizadas, mientras que el campo *Mat\_Averiado* no aporta información adicional.

### Análisis del Tipo 317 Otras Actuaciones / Sin Corte (Nivel SU)

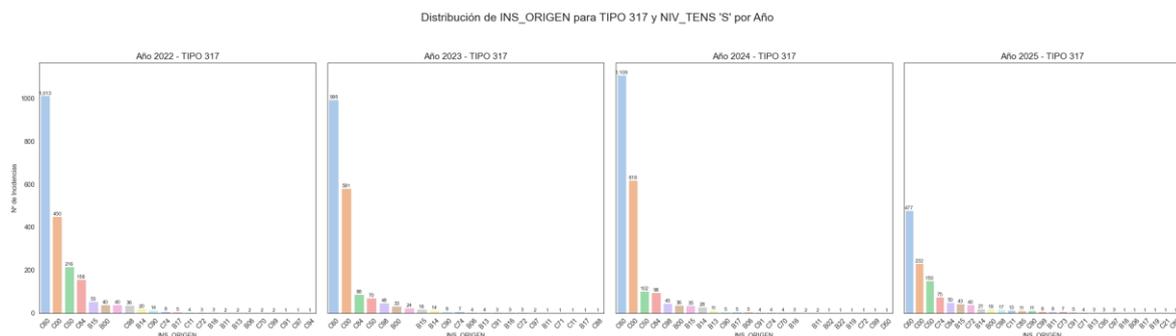


Gráfico 31: Distribución del campo INS\_ORIGEN para incidencias Tipo 317, Nivel SU, por año

En el *Gráfico 31* se representa la frecuencia de valores registrados en el campo *Ins\_Origen*. Se observa que, en todos los años analizados, la mayoría de las incidencias se concentran en un número muy reducido de códigos, concretamente en

C00	BAJA TENSION
C50	INSTAL.PARTICULARES



**Análisis del Tipo 416 – Suministro Correcto (Nivel SU).**

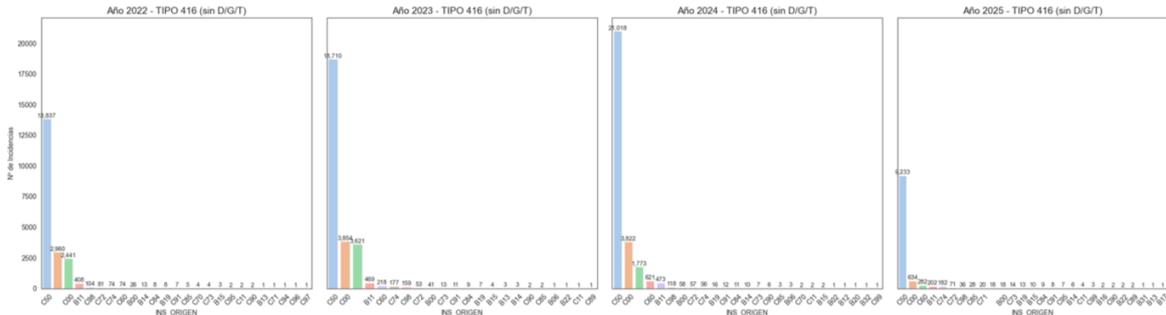


Gráfico 34: Distribución del campo *INS\_ORIGEN* para incidencias Tipo 416, Nivel SU, por año.

En el Gráfico 34 se representa la frecuencia de valores registrados en el campo *Ins\_Origen*. Se observa que, en todos los años analizados, la mayoría de las incidencias se concentran en dos códigos, quedando un buen número sin tipificar (barra color verde)

Los códigos más empleados son:

C00	BAJA TENSION
C50	INSTAL.PARTICULARES

A modo de resumen se han añadido las incidencias de todo el periodo analizado:

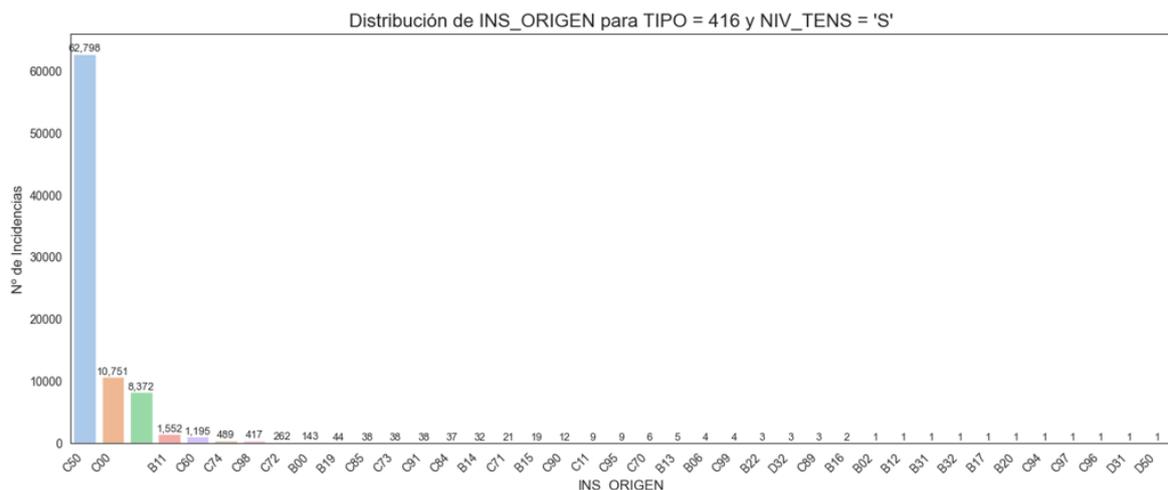


Gráfico 35: Distribución del campo *INS\_ORIGEN* para incidencias Tipo 317, Nivel SU (2022- marzo 2025)

Aunque existen otros códigos registrados, su frecuencia es muy baja y dispersa, lo que dificulta extraer conclusiones relevantes a partir de este campo.

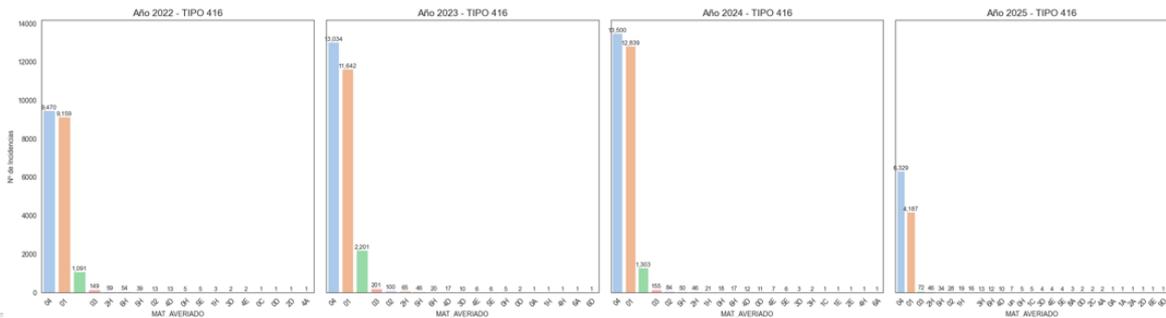


Gráfico 36: Distribución del campo MAT\_AVERIADO para incidencias Tipo 416, Nivel SU, por año.

El Gráfico 36 muestra la distribución del campo *Mat\_Averiado*. Se puede apreciar que los dos códigos más empleados son el 04 y 01. Se debe revisar las diferencias entre la tipología 317 *Otras Actuaciones / Sin Corte* y las 416 *Suministro Correcto*, pues en ambas tiene los mismos orígenes y pocas variaciones con respecto al material averiado. Asimismo se debería explicitar si este tipo de SU requieren de desplazamiento a campo de los equipos.

### Análisis del Tipo 417 – Equipo de Medida (Nivel SU)

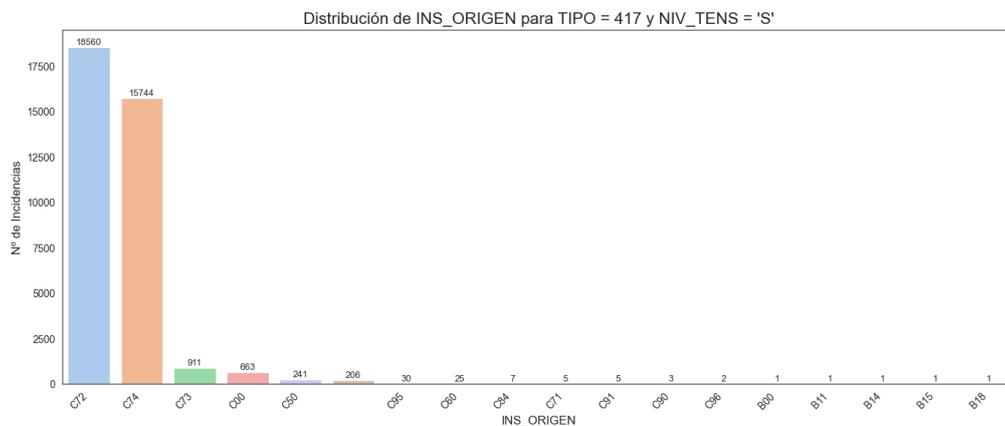


Gráfico 37: Distribución del campo INS\_ORIGEN para incidencias Tipo 417, Nivel SU (2022- marzo 2025)

En el Gráfico 37 se representa la frecuencia de valores registrados en el campo *Ins\_Origen*. Se observa que, en todos los años analizados, la mayoría de las incidencias se concentran en dos códigos, mientras que un número considerable queda sin tipificar (barra color verde)

Los códigos más empleados son:

C72	ICP Contador (Interruptor Control Potencia)
C73	ICP Vivienda
C74	CONTADOR

A continuación, se revisa el campo *Mat\_Averiado*, representando en el Gráfico 38, donde se observa que, el material averiado registrado es mayoritariamente

- 01: Ningún material.
- 5H Contador
- 6H ICP,

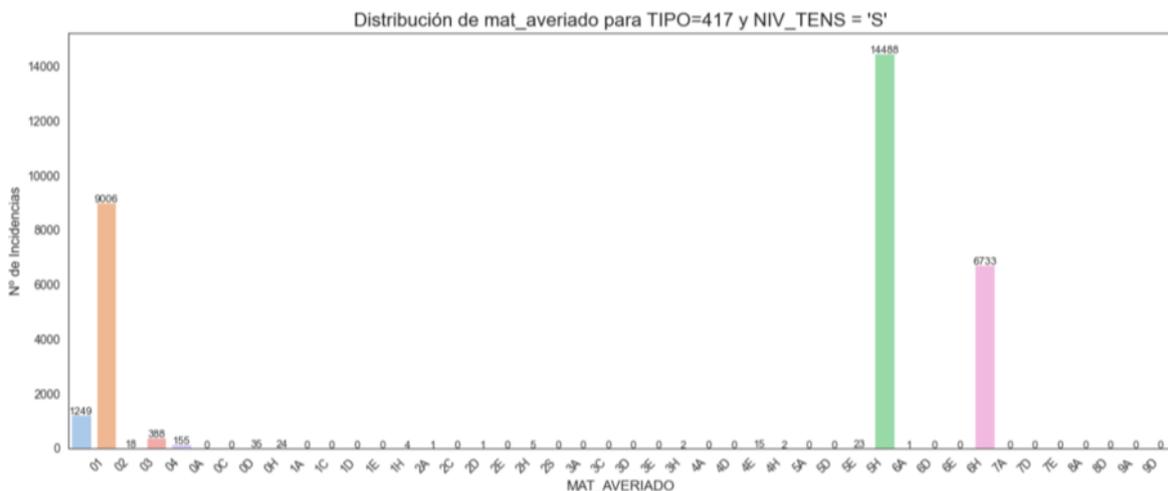


Gráfico 38: Distribución del campo MAT\_AVERIADO para incidencias Tipo 417, Nivel SU, periodo (2022-marzo 2025).

En esta tipología, se constata que la instalación origen está correctamente tipificada. Sin embargo, el campo *Mat\_Averiado* no aporta información adicional relevante, ya que los materiales identificados coinciden de forma directa con lo que ya se deduce del campo *Inst\_Origen*.

## **RESUMEN**

En las incidencias tipo 303 (Fallo de Material) y 317 (Otras Actuaciones sin Corte), el campo *Ins\_Origen* presenta ambigüedades. No resulta clara la diferencia práctica entre los códigos C00 (Baja Tensión) y C60 (Redes de BT), lo que limita su utilidad como indicador del origen de la incidencia. Además, en ambos tipos, el campo *Mat\_Averiado* suele cumplimentarse con los valores 01 (Ningún Material) o 03 (Otros Elementos), que no aportan información adicional.

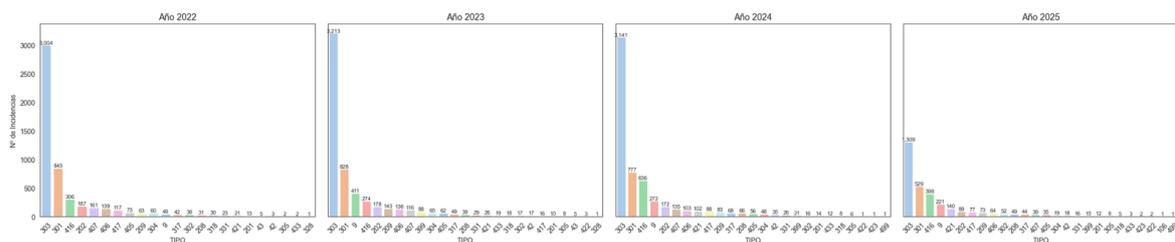
En el caso concreto del tipo 317 (Otras Actuaciones sin Corte), no es posible identificar con los campos estructurados disponibles a qué tipo de actuación responde la incidencia ni en

qué se diferencia funcionalmente de las del tipo 416 (Suministro Correcto), que indica que no ha habido interrupción imputable a la distribuidora. Esta confusión se agrava cuando ambas se tipifican con el código C00 (Baja Tensión) como instalación origen. Sería necesario, por tanto, revisar y aclarar conceptualmente las diferencias entre estas dos tipologías y establecer criterios claros sobre cómo deben cumplimentarse los campos asociados.

Respecto a las incidencias tipo 417 (Equipo de Medida), aunque los campos *Inst\_Origen* y *Mat\_Averiado* están correctamente informados (p. ej., C72/C74 y códigos específicos como 5H Contador o 6H ICP), el campo de material averiado no añade información sustancial más allá de lo que ya indica la instalación origen. En consecuencia, también en este caso sería recomendable revisar la utilidad de este campo y valorar posibles mejoras en su definición o uso.

## **EVOLUCIÓN DEL NÚMERO DE INCIDENCIAS IMPREVISTAS DE BT (2022-2024 Y PRIMER TRIMESTRE DE 2025).**

En el Gráfico 39 se muestra la evolución de las incidencias imprevistas de BT registradas entre los años 2022 y 2024, incluyendo el primer trimestre de 2025. Las incidencias se han clasificado según el valor del campo *Tipo*.



*Gráfico 39: Incidencias Imprevistas de BT por Año y Tipo*

Durante el periodo analizado, los códigos más frecuentes han sido los siguientes:

303	FALLO MATERIAL-EQUIP
301	CAUSA DESCONOCIDA
416	SUMINI. CORRECTO RED
202	ARBOLES

En el caso de las incidencias tipo 900: Fuerza Mayor, no considera oportuno analizar su tipificación, puesto que esta se determina a posteriori, una vez que la administración competente así lo clasifica.

Posteriormente, para cada una de estas tipologías identificadas, se ha revisado la información registrada en los campos *Ins\_Origen* y *Mat\_Averiado*, con el objetivo de determinar si estos campos, junto con el campo *Tipo*, permiten identificar con suficiente precisión el origen y la naturaleza de la incidencia.

### Análisis del Tipo 303 – Fallo Material (Nivel BT)

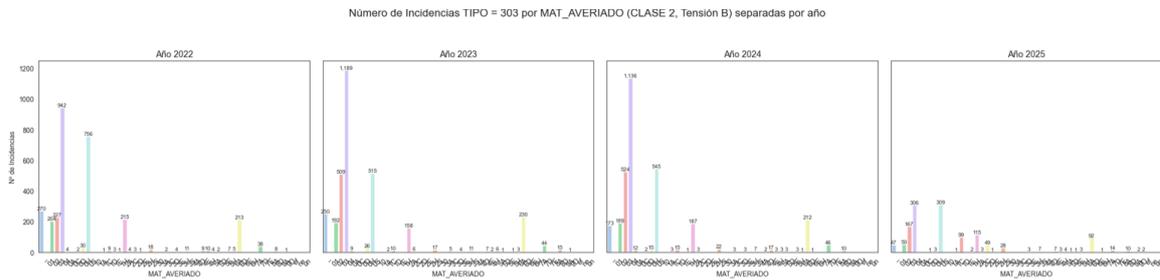


Gráfico 40: Distribución del campo *Ins\_Origen* para incidencias Tipo 303 de BT, por año

En el Gráfico 40 se representa la frecuencia de valores registrados en el campo *Ins\_Origen*. Se observa que, en todos los años analizados, la mayoría de las incidencias se concentran en un número muy reducido de códigos, concretamente en los siguientes, ordenados de mayor a menor frecuencia:

C90	FASE CORTADA
C60	REDES DE BT
C91	NEUTRO CORTADO
C00	BAJA TENSION
C94	FUSIBLE FUNDIDO EN CT

Aunque existen otros códigos registrados, su frecuencia es muy baja y dispersa, lo que dificulta extraer conclusiones relevantes a partir de este campo y sugiere que los códigos se pueden simplificar.



*Gráfico 41: Distribución del campo MAT\_AVERIADO para incidencias Tipo 303 de BT, por año.*

El Gráfico 41 muestra la distribución de valores registrados en el campo *Mat\_Averiado*. Los códigos más empleado son, ordenados de mayor a menor frecuencia:

- 03 Otros Elementos.
- 0H Conductor Fase BT.
- 02 Pendiente de Reparación Definitiva.
- 6A Fusible de BT en CT.
- 1H Conductor neutro en BT.

Para las incidencias *Tipo 303 - Fallo Material*, el campo *Ins\_Origen* podría simplificarse reduciendo el número de categorías utilizadas o aclarando mejor su significado (por ejemplo, distinguiendo entre C00 Baja Tensión y C60 Redes de Baja Tensión).

Asimismo, sería conveniente revisar existe redundancia entre los campos *Ins\_Origen* y *Mat\_Averiado* (por ejemplo C94 Fusible Fundido en CT con el *Mat\_Averiado* 6A Fusible de BT en CT) ya que aparentemente algunas categorías de ambos parecen referirse a lo mismo.

Por otro lado, el código *02 Pendiente de Reparación Definitiva* no parece corresponder a un material averiado, sino más bien a un estado de resolución de la incidencia.

### Análisis del Tipo 301 – Causa Desconocida (Nivel BT)

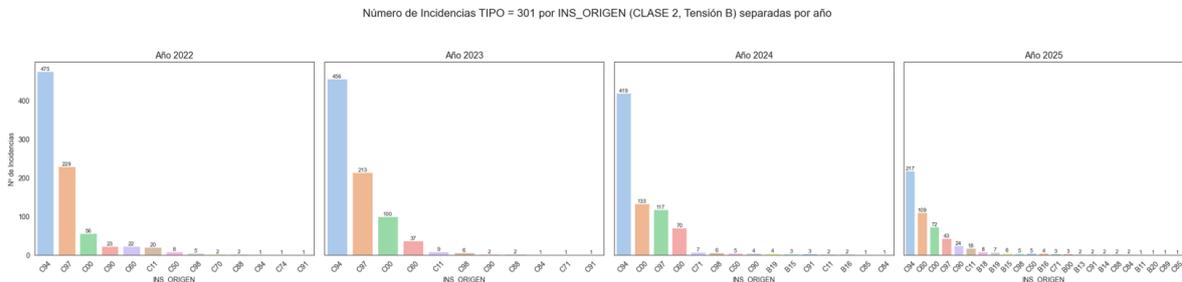


Gráfico 42: Distribución del campo *Ins\_Origen* para incidencias Tipo 301 de BT por año.

En el Gráfico 42 se representa la frecuencia de valores registrados en el campo *Ins\_Origen*. Se observa que, en todos los años analizados, la mayoría de las incidencias se concentran en un número muy reducido de códigos, concretamente en los que se muestran a continuación ordenados de mayor a menor frecuencia:

C94	FUSIBLE FUNDIDO EN CT
C97	INTERRUPTOR BT
C00	BAJA TENSION
C60	REDES DE BT
C90	FASE CORTADA

Aunque existen otros códigos registrados, su frecuencia es menor, por lo que habiendo algunos que no se han usado más de una o dos veces durante todo el periodo analizado y sugiere que los códigos se pueden simplificar.

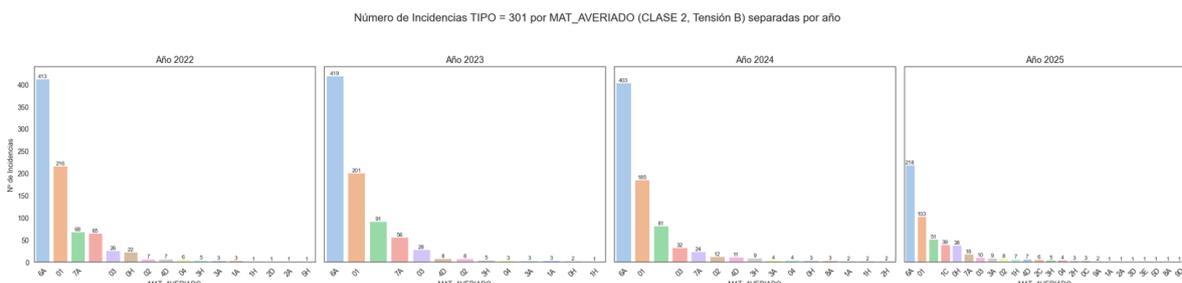


Gráfico 43: Distribución del campo *MAT\_AVERIADO* para incidencias Tipo 301 de BT, por año.

El Gráfico 43 muestra la distribución de valores registrados en el campo *Mat\_Averiado*. Los códigos más empleado son, ordenados de mayor a menor frecuencia:

- 6A Fusible de BT en CT
- 01 Ningún Material

- 7A Interruptor BT
- 03 Otros Elementos.
- 0H Conductor Fase BT.

Para las incidencias *Tipo 301 – Causa Desconocida*, el campo *Ins\_Origen* podría referirse a aquel elemento de la red que interrumpe el paso de energía, pero sorprende que el campo material averiado haga referencia a equipos, pues en ese caso no debería tipificarse como Causa Desconocida.

### Análisis del Tipo 416 – Suministro Correcto Red (Nivel BT)

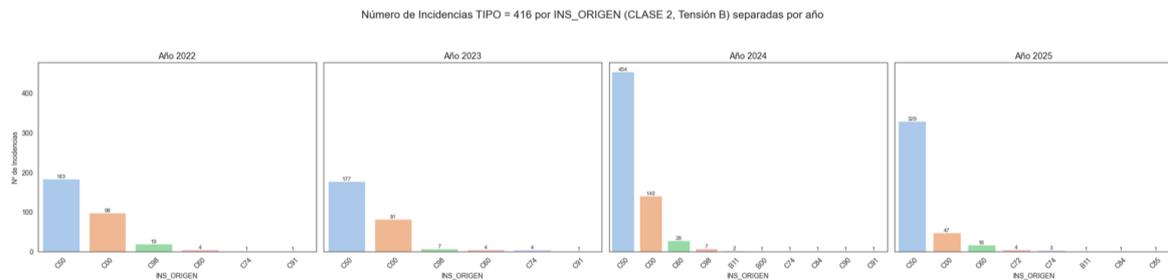


Gráfico 44: Distribución del campo *Ins\_Origen* para incidencias Tipo 416 de BT, por año.

En el Gráfico 44 se representa la frecuencia de valores registrados en el campo *Ins\_Origen*. La mayoría de las incidencias tipificadas como 416- Suministro Correcto se concentran en un número muy reducido de códigos de *Ins\_Origen*, que se detallan a continuación en orden descendente de frecuencia:

C50	INSTALACIONES PARTICULARES.
C00	BAJA TENSION
C60	REDES DE BT
C98	FASE TIERRA
C74	CONTADOR

Los códigos resultan propios de este tipo de incidencia, aunque debería aclararse la diferencia funcional entre C00 y C60, ya que sin instrucciones concisas pueden emplearse de forma indistinta. Además, existen otros códigos que se emplean con muy poca frecuencia y que, al menos para esta tipología, podrían revisarse o incluso deshabilitarse.

Número de Incidencias TIPO = 416 por MAT\_AVERIADO (CLASE 2, Tensión B) separadas por año

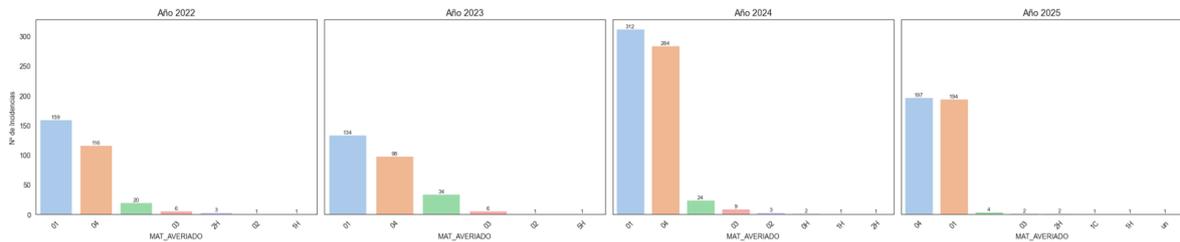


Gráfico 45: Distribución del campo MAT\_AVERIADO para incidencias Tipo 416 de BT, por año.

El Gráfico 45 muestra la distribución de valores registrados en el campo *Mat\_Averiado*. Los códigos más empleado son, ordenados de mayor a menor frecuencia:

- 01 Ningún Material
- 04 Instalaciones Particulares
- 03 Otros Elementos
- 2H CGP
- 02 Pendiente de Reparación Definitiva

También se observa un número relevante de incidencias en las que este campo no ha sido cumplimentado. Cabe señalar que el uso del código 02 (Pendiente de Reparación Definitiva) puede resultar confuso, ya que no representa un material averiado como tal, sino más bien una condición o estado del trabajo.

### Análisis del Tipo 202 – Arboles (Nivel BT)

Número de Incidencias TIPO = 202 por INS\_ORIGEN (CLASE 2, Tensión B) separadas por año

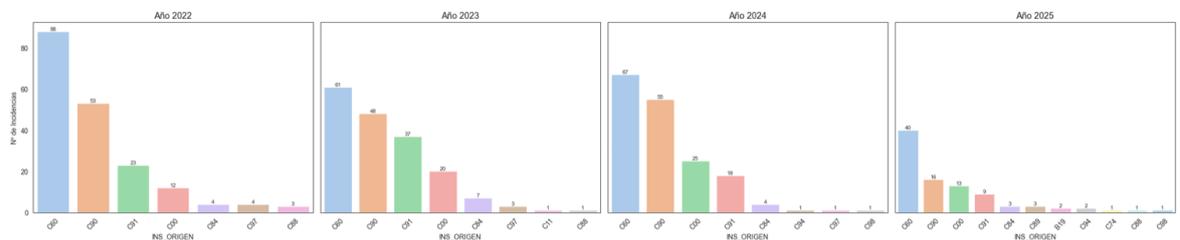


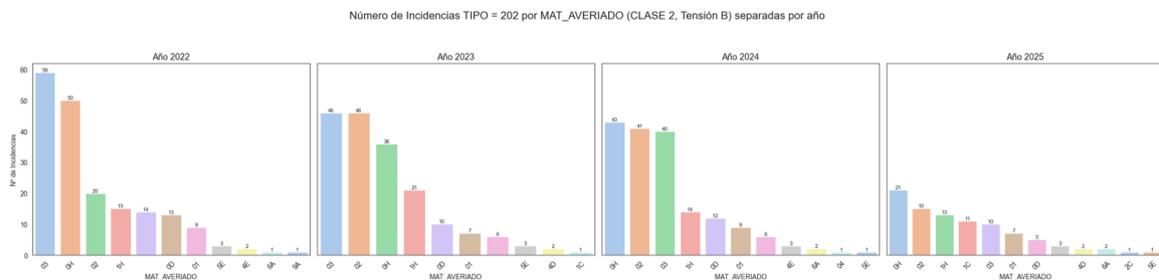
Gráfico 46: Distribución del campo Ins\_Origen para incidencias Tipo 202 de BT, por año.

En el Gráfico 46 se representa la frecuencia de valores registrados en el campo *Ins\_Origen*. La mayoría de las incidencias tipificadas como 202- Arboles se concentran en un número

muy reducido de códigos de *Ins\_Origen*, que se detallan a continuación en orden descendente de frecuencia:

C60	REDES DE BT
C90	FASE CORTADA
C91	NEUTRO CORTADO
C00	BAJA TENSION
C84	ACOMETIDA CORTADA

Los códigos resultan ser propios de este tipo de incidencia. Hay que insistir en el uso indistinto de C00 y C60. Además existen otros códigos que se emplean con muy poca frecuencia y que, al menos para esta tipología, podrían revisarse o incluso deshabilitarse.



*Gráfico 47: Distribución del campo MAT\_AVERIADO para incidencias Tipo 202 de BT, por año*

El Gráfico 47 muestra la distribución de valores registrados en el campo *Mat\_Averiado*. Los códigos más empleados son, ordenados de mayor a menor frecuencia:

- 03 Otros Elementos
- 0H Conductor Fase BT
- 02 Pendiente de Reparación Definitiva
- 1H Conductor Neutro BT
- 0D Cable Aéreo

El uso del código 02 (*Pendiente de Reparación Definitiva*) puede resultar confuso, ya que no representa un material averiado como tal, sino más bien una condición o estado del trabajo, siendo importante que en estos casos se identifique adecuadamente el material averiado para posteriormente hacer la reparación.

## EVOLUCIÓN DEL NÚMERO DE INCIDENCIAS IMPREVISTAS DE MT (2022-2024 Y PRIMER TRIMESTRE DE 2025).

En el Gráfico 48 se muestra la evolución de las incidencias imprevistas de MT registradas entre los años 2022 y 2024, incluyendo el primer trimestre de 2025. Las incidencias se han clasificado según el valor del campo *Tipo*.

Se puede observar que el número de incidencias imprevistas de MT Tipo 301-Causa Desconocida es muy superior al resto de tipologías. Este volumen se debe a que el SGI asigna automáticamente el tipo 301 a las incidencias asociadas a reenganches en las líneas de MT: interrupciones de menos de tres minutos, en las que el sistema de protección del circuito abre por una falta pasajera y se reconecta en forma automática tras unos segundos, sin intervención por parte del COR ni de equipos en campo.

Dado que en la mayoría de estos casos no se llega a conocer el origen exacto de la desconexión y los campos *Inst\_Origen* y *Mat\_Averiado* no se encuentran cumplimentados, es recomendable filtrar y excluir estas incidencias del análisis detallado, ya que su inclusión podría distorsionar los resultados.

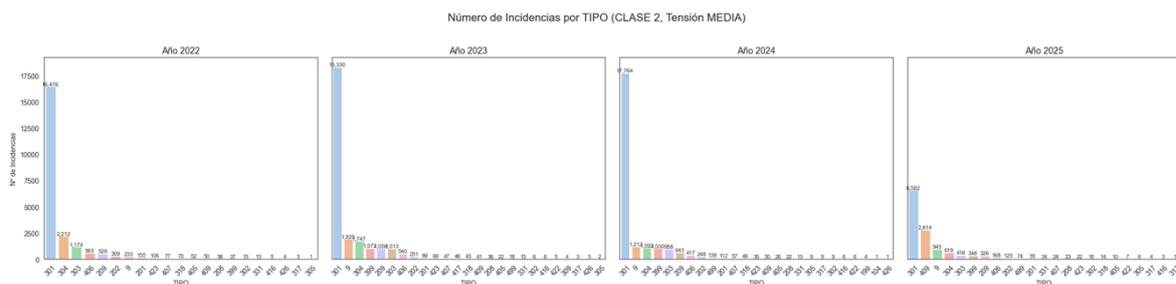


Gráfico 48: Incidencias Imprevistas de MT por Año y por Tipo

En el Gráfico 49 se muestra la evolución de las incidencias imprevistas tras eliminar los reenganches y aquellas incidencias con interrupciones de duración inferior a 3 minutos, con el fin de centrar el análisis en los casos que sí aportan información técnica útil.

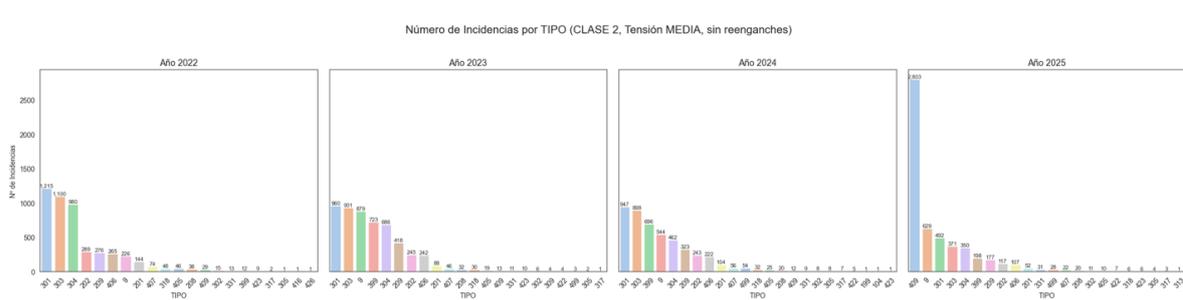


Gráfico 49: Incidencias Imprevistas con interrupciones de mas de 3 minutos en MT porTipo y Año

El Gráfico 49 muestra la distribución de valores registrados en el campo *Mat\_Averiado*. Los códigos más empleado son, ordenados de mayor a menor frecuencia:

301	CAUSA DESCONOCIDA
303	FALLO MATERIAL-EQUIP
304	LOCALIZ/NORMAL EXPLO
900	FUERZA MAYOR
399	MANIOBRA REPOS.SERVI
209	ATMOSFÉRICAS
202	ARBOLES
406	INSTAL. PARTICULARES
201	ANIMALES
407	TERCEROS IDENTIF.
318	PERSON.TRAB.PARA UFD
208	INUNDACION/MOV.TIERR

Para iniciar el análisis de las incidencias de MT, se seleccionan las tipologías más frecuentes y que pueden aportar más información técnica.

En el caso del Tipo 900 - Fuerza Mayor, no considera oportuno analizar su tipificación, ya que se asigna a posteriori una vez que la administración competente así lo clasifica. No obstante, se recomienda conservar la tipificación original antes de su reclasificación como fuerza mayor, para no perder información potencialmente útil sobre la causa inicial.

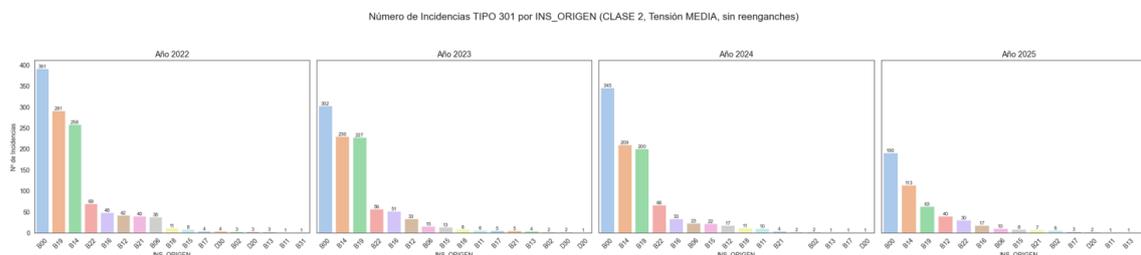
Tampoco se analizan, en esta fase, las incidencias codificadas como 209 – Atmosféricas, ya que este código se emplea con frecuencia como comodín para cualquier incidencia con condiciones meteorológicas adversas. Se sugiere sustituirlo por valores más específicos que aporten mayor detalle.

Se seleccionan finalmente para iniciar el estudio

301	CAUSA DESCONOCIDA
303	FALLO MATERIAL-EQUIP
304	LOCALIZ/NORMAL EXPLO
202	ARBOLES
406	INSTALACIONES PARTICULARES

Posteriormente, para cada una de estas tipologías identificadas, se ha revisado la información registrada en los campos *Ins\_Origen* y *Mat\_Averiado*, con el objetivo de determinar si estos campos, junto con el campo *Tipo*, permiten identificar con suficiente precisión el origen y la naturaleza de la incidencia.

### **Análisis del Tipo 301 – Causa Desconocida (Nivel MT)**



*Gráfico 50: Distribución del campo *Ins\_Origen* para incidencias Tipo 301 de MT, por año*

En el Gráfico 50 se representa la frecuencia de valores registrados en el campo *Ins\_Origen*. Se observa que, en todos los años analizados, la mayoría de las incidencias se concentran en un número muy reducido de códigos, concretamente en los siguientes, ordenados de mayor a menor frecuencia:

B00	MEDIA TENSIÓN
B14	LÍNEAS AÉREAS
B19	FUSIBLE XS
B22	RECONNECTADOR
B16	CELDA DE PROTECCIÓN

Aunque existen otros códigos registrados, su frecuencia es muy baja, lo que sugiere que podrían simplificarse los valores disponibles o restringirse los aplicables a este tipo de incidencia.

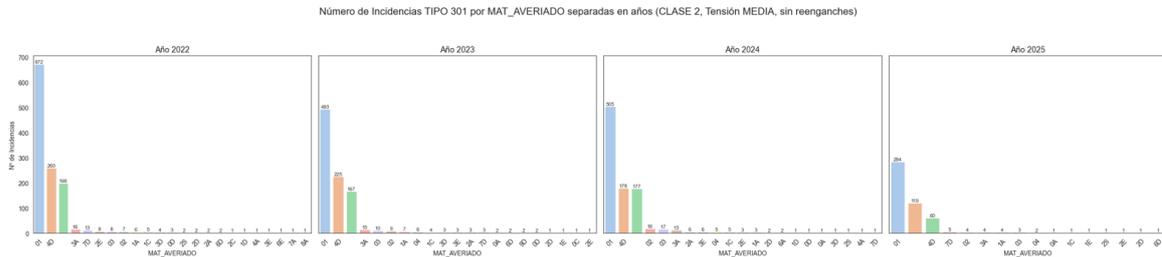


Gráfico 51: Distribución del campo MAT\_AVERIADO para incidencias Tipo 301 de MT, por año.

En el Gráfico 51 se representa la frecuencia de valores registrados en el campo *Mat\_Averiado*. Los códigos más empleados son los siguientes en orden descendente de frecuencia:

01	NINGÚN MATERIAL
4D	FUSIBLE XS
3A	FUSIBLE de MT
03	OTROS ELEMENTOS
02	PTE. REPARACIÓN DEF

Para las incidencias *Tipo 301 – Causa Desconocida*, el campo *Ins\_Origen* podría simplificarse reduciendo el número de categorías utilizadas clarificando en qué casos debe aplicarse cada una.

Asimismo, sería conveniente revisar existe redundancia entre los campos *Ins\_Origen* y *Mat\_Averiado* (por ejemplo, *B19 Fusible XS* con *4D Fusible XS*) ya que aparentemente algunas categorías de ambos parecen referirse a lo mismo.

Por otro lado, el código *02 Pendiente de Reparación Definitiva* no parece corresponder a un material averiado, sino más bien a un estado de resolución de la incidencia, por lo que convendría revisar su uso en este contexto.

Finalmente, resulta llamativo que el tipo “Causa Desconocida” sea el más utilizado para tipificar las incidencias imprevistas de MT, cuando en la mayoría de los casos se dispone de información relevante en los campos Observaciones, Descripción y, en bastantes registros, también en *Inst\_Origen* y *Mat\_Averiado*. Esto sugiere que el criterio de asignación de esta tipología debería revisarse, fomentando una tipificación más precisa y útil para el análisis posterior.

### Análisis del Tipo 303 – Fallo Material (Nivel MT)

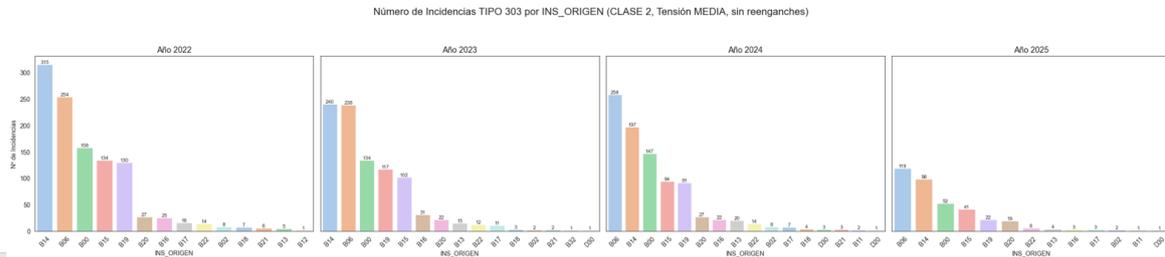


Gráfico 52: Distribución del campo *Ins\_Origen* para incidencias Tipo 303 de MT, por año

En el Gráfico 52 se representa la frecuencia de valores registrados en el campo *Ins\_Origen*. Se observa que, en todos los años analizados, la mayoría de las incidencias se concentran en un número muy reducido de códigos, concretamente en los siguientes, ordenados de mayor a menor frecuencia:

B06	CABLE SUBTERRÁNEO
B14	LÍNEAS AÉREAS
B00	MEDIA TENSIÓN
B15	CT MT
B19	FUSIBLE XS

Aunque existen otros códigos registrados, su frecuencia es muy baja, lo que sugiere que los códigos se pueden simplificar y normalizar.

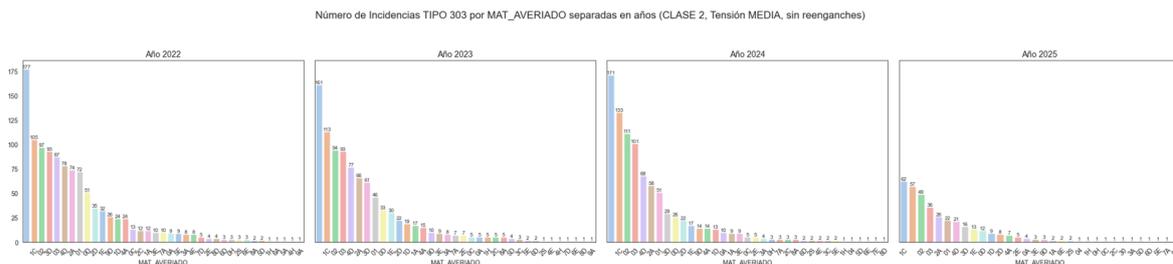


Gráfico 53: Distribución del campo *MAT\_AVERIADO* para incidencias Tipo 303 de MT, por año.

El Gráfico 53 se representa la frecuencia de valores registrados en el campo *Mat\_Averiado*. Los códigos más empleado son, ordenados de mayor a menor frecuencia:

	VACÍO
1C	CABLE SECO
02	PTE. REPARACIÓN DEF
03	OTROS ELEMENTOS
4D	FUSIBLE XS
2A	TRANSFORMADOR MT

Para las incidencias *Tipo 303 - Fallo Material*, destaca que el valor más empleado para el campo *Mat\_Averiado* es el campo vacío, lo que refleja una falta de cumplimentación sistemática (columna azul inicial). Por otra parte el Código *02 Pendiente de Reparación Definitiva* no parece corresponder a un material averiado, sino más bien a un estado de resolución de la incidencia, por lo que se recomienda revisar su uso.

### Análisis del Tipo 304 – Localización / Normalización explotación Origen (Nivel MT)

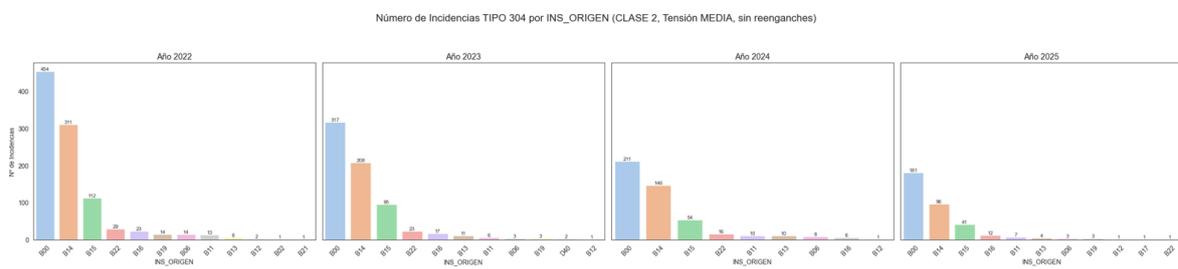


Gráfico 54: Distribución del campo *Ins\_Origen* para incidencias Tipo 304 de MT, por año

En el **¡Error! No se encuentra el origen de la referencia.** se representa la frecuencia de valores registrados en el campo *Ins\_Origen*. Se observa que, en todos los años analizados, la mayoría de las incidencias se concentran en un número muy reducido de códigos, concretamente en los siguientes, ordenados de mayor a menor frecuencia:

B00	MEDIA TENSIÓN
B14	LÍNEAS AÉREAS
B15	CT MT
B22	RECONNECTADOR
B16	CELDA DE PROTECCIÓN

El uso reiterado exclusivo de estos códigos, sugiere que es necesario simplificar y normalizar los mismos, manteniendo solo aquellos que aportan información.

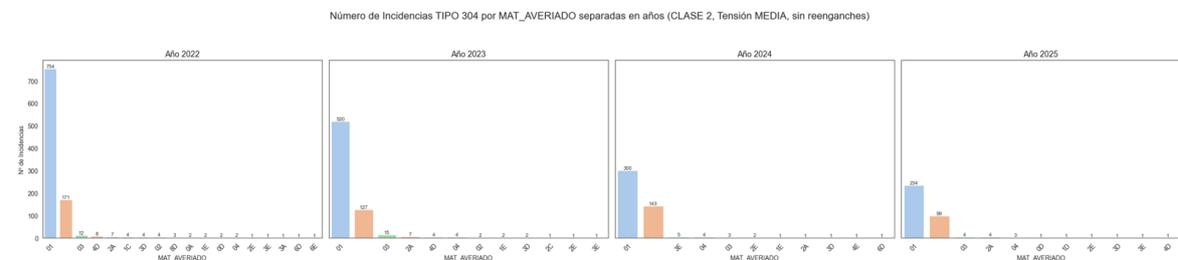


Gráfico 55: Distribución del campo *MAT\_AVERIADO* para incidencias Tipo 304 de MT, por año

*El*

Número de Incidencias TIPO 304 por MAT\_AVERIADO separadas en años (CLASE 2, Tensión MEDIA, sin reenganches)

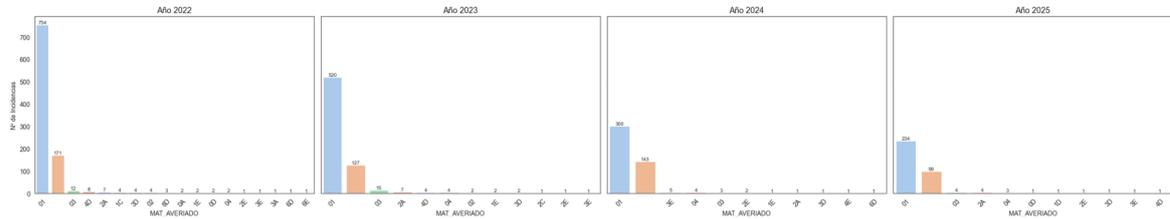


Gráfico 55 se representa la frecuencia de valores registrados en el campo *Mat\_Averiado*. Los códigos más empleado son, ordenados de mayor a menor frecuencia:

01	NINGUN MATERIAL
	VACIO
03	OTROS ELEMENTOS
2A	TRANSFORMADOR MT
04	INSTALACIONES PARTIC
4D	FUSIBLE XS

Para las incidencias *Tipo 304 – Normalización de la Explotación*, parece coherente que el valor más frecuente sea *01 Ningún Material*, seguido de registros con el campo vacío, ya que esta tipología se emplea para devolver la red a su estado normal tras haber resuelto la avería.

### Análisis del Tipo 406 – Instalaciones Particulares (Nivel MT)

Número de Incidencias TIPO 406 por INS\_ORIGEN (CLASE 2, Tensión MEDIA, sin reenganches)

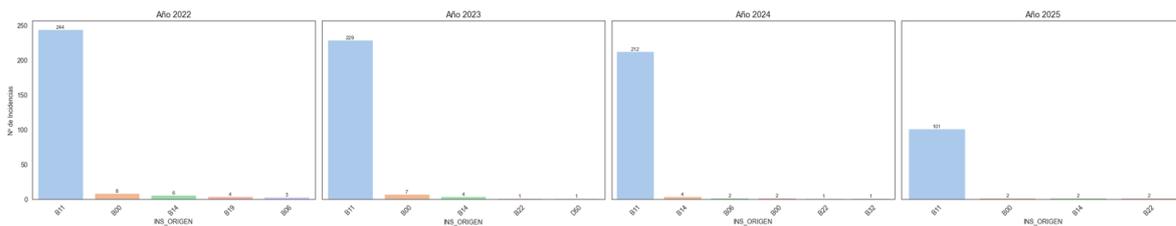


Gráfico 56: Distribución del campo *Ins\_Origen* para incidencias *Tipo 406* de *MT*, por año

En el Gráfico 56 se representa la frecuencia de valores registrados en el campo *Ins\_Origen*. Se observa que, en todos los años analizados, la mayoría de las incidencias se concentran prácticamente en un único código. A continuación se muestran los códigos más empleados, ordenados de mayor a menor frecuencia:

B11	INSTAL. PARTICULARES
B00	MEDIA TENSIÓN

B14	LÍNEAS AÉREAS
B06	CABLE SUBTERRÁNEO
B19	FUSIBLE XS

Que el código más empleado sea B11 – Instalaciones Particulares, precisamente para incidencias del tipo 406 – Instalaciones Particulares, indica que cumplimentar este campo no añade información adicional. Por tanto, sería conveniente simplificar y normalizar los valores del campo *Ins\_Origen*, manteniendo solo aquellos que realmente aporten valor informativo o que permitan diferenciar entre situaciones relevantes.

Número de Incidencias TIPO 406 por MAT\_AVERIADO separadas en años (CLASE 2, Tensión MEDIA, sin reenganches)

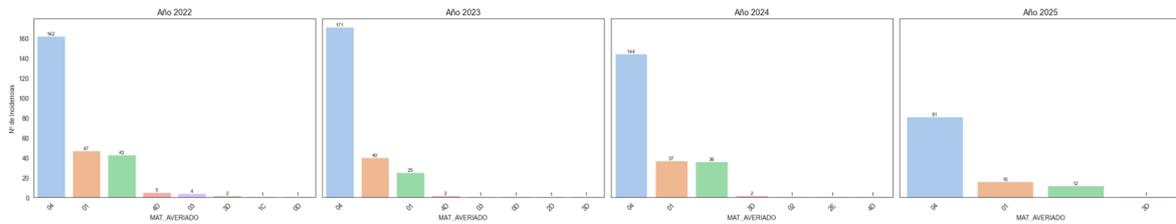


Gráfico 57: Distribución del campo *MAT\_AVERIADO* para incidencias Tipo 406 de MT, por año

El Gráfico 57 representa la frecuencia de valores registrados en el campo *Mat\_Averiado*. Los códigos más empleado son, ordenados de mayor a menor frecuencia:

04	INSTALACIONES PARTIC
	VACIO
01	NINGUN MATERIAL
4D	FUSIBLE FUNDIDO
3D	BASE XS/SXS
03	OTROS ELEMENTOS

Para las incidencias Tipo 406 – Instalaciones Particulares, el hecho de que el valor más frecuente sea 04 – Instalaciones Particulares (coincidente con la tipología del incidente) implica una redundancia informativa. Le siguen los valores vacío y 01 – Ningún Material, que, aunque coherentes en contexto, no añaden información relevante. En conjunto, esto sugiere que el campo *Mat\_Averiado* podría optimizarse limitando los códigos disponibles o estableciendo criterios más claros sobre cuándo debe cumplimentarse y con qué propósito.

### **Análisis del Tipo 202 – Árboles (Nivel MT)**

Número de Incidencias TIPO 202 por INS\_ORIGEN (CLASE 2, Tensión MEDIA, sin reenganches)

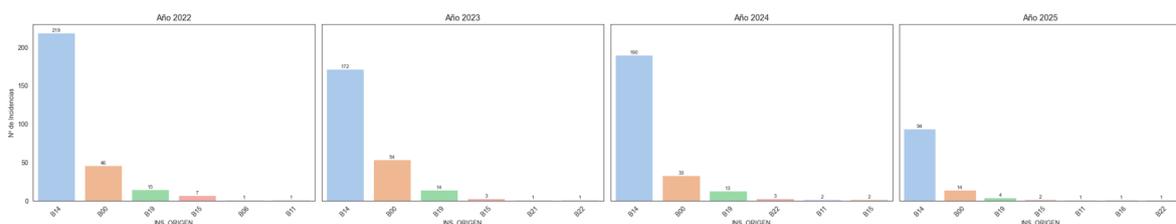


Gráfico 58: Distribución del campo *Ins\_Origen* para incidencias Tipo 202 de MT, por año

En el Gráfico 58 se representa la frecuencia de valores registrados en el campo *Ins\_Origen*. Se observa que, en todos los años analizados, la mayoría de las incidencias se concentran en un número muy reducido de códigos, concretamente en los siguientes, ordenados de mayor a menor frecuencia:

B14	LÍNEAS AÉREAS
B00	MEDIA TENSIÓN
B19	FUSIBLE XS
B15	CT MT
B22	RECONECTADOR

Se recomienda revisar la aplicación de estos códigos, asegurando que todos los responsables de cumplimentar el SGI interpreten que en *Ins\_Origen* debe indicarse la instalación directamente afectada por el contacto o la proximidad del árbol.

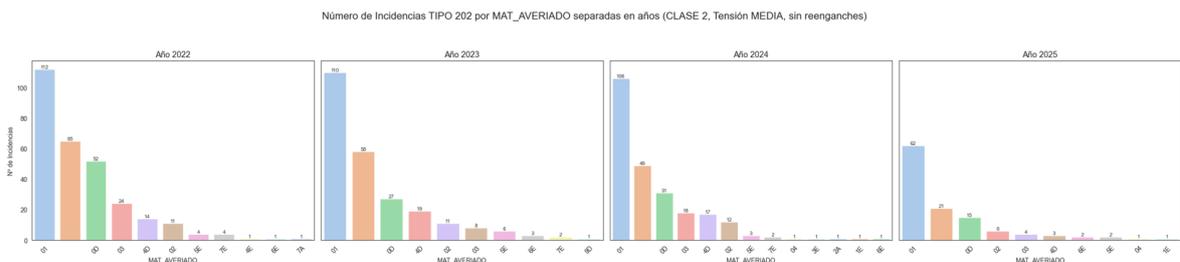


Gráfico 59: Distribución del campo *MAT\_AVERIADO* para incidencias Tipo 202 de MT, por año.

El Gráfico 59 representa la frecuencia de valores registrados en el campo *Mat\_Averiado*. Los códigos más empleado son, ordenados de mayor a menor frecuencia:

01	NINGUN MATERIAL
	VACIO
0D	CABLE AÉREO
03	OTROS ELEMENTOS
4D	FUSIBLE XS
02	PTE. REPARACION DEFI

Para las incidencias Tipo 202 –Árboles, el valor más frecuente es 01 Ningún Material, seguido de registros con el campo vacío, lo que se relaciona con casos en los que los árboles provocan una apertura del circuito sin que se llegue a producir daño físico en los elementos de la instalación.

El código *03-Otros Elementos* es ambiguo y no aporta información específica y el código *02-Pendiente de Reparación Definitiva* parece describir un estado operativo más que un material averiado, por lo que su uso en este campo debería revisarse.

### **Análisis de los campos de texto libre en las incidencias tipo 202-Árboles de MT .**

Para este caso, además del análisis de los campos estructurados, se ha analizado el contenido del campo *Descripción* con el fin de identificar términos recurrentes que puedan aportar información adicional sobre las causas de las incidencias.

El Gráfico 58 muestra la frecuencia de las palabras más repetidas en dicho campo. Se observa que las palabras predominantes están en línea con la información registrada en los campos *Ins\_Origen* y *Mat\_Averiado*, lo que sugiere una coherencia general entre los datos estructurados y el texto libre.

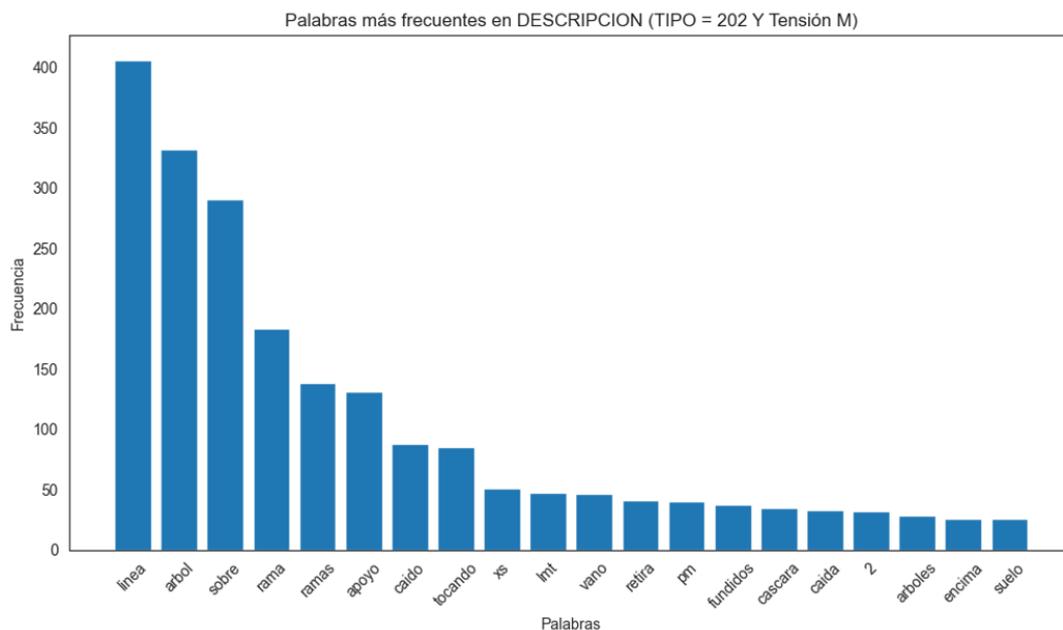
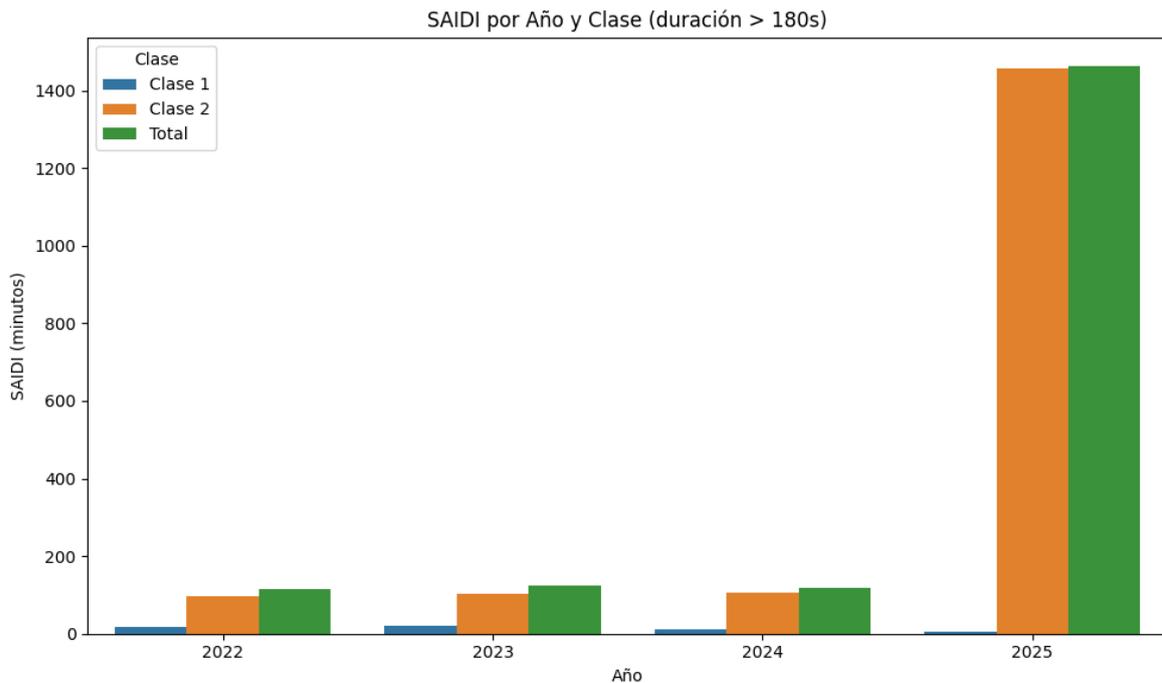


Gráfico 60: Frecuencia de palabras repetidas en Observaciones en las Incidencias Imprevistas de MT Tipo 202-Árboles

## **DISTRIBUCIÓN DEL SAIDI Y SAIFI POR NIVEL DE TENSIÓN (MT Y BT) (2022-2025)**

En este anexo se incluyen los gráficos correspondientes al análisis del reparto de los indicadores SAIDI y SAIFI entre los niveles de tensión Media Tensión (MT) y Baja Tensión (BT), a lo largo del periodo 2022-2025.



*Gráfico 61: Evolución del SAIDI en minutos, por clase 1: Incidencias Programadas, 2: Incidencias Imprevistas*

El objetivo principal de este análisis es evaluar cómo varía la aportación relativa de cada nivel de tensión al total de los indicadores, diferenciando entre incidencias programadas (Clase 1) e imprevistas (Clase 2). Para ello, se han utilizado los datos extraídos de la base Master\_SGI del Data Lake corporativo, aplicando los siguientes criterios de filtrado:

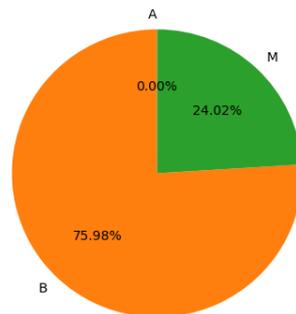
- Se consideran únicamente las interrupciones con una duración superior a 180 segundos (3 minutos), en coherencia con la metodología de cálculo de los indicadores regulados TIEPI y NIEPI.
- Se excluyen las incidencias clasificadas como Suministro Único (SU), centrándose el análisis en los niveles de MT y BT.

Los gráficos generados representan, para cada año del periodo de estudio y para cada clase de incidencia, el porcentaje de SAIDI y SAIFI que corresponde a BT respecto del total (BT + MT). Este enfoque permite visualizar de manera directa el peso relativo que tendría la

inclusión de las interrupciones de BT en el cálculo de estos indicadores, tradicionalmente centrados en Media Tensión.

## SAIDI

Distribución SAIDI - Año 2022, Clase 1



Distribución SAIDI - Año 2022, Clase 2

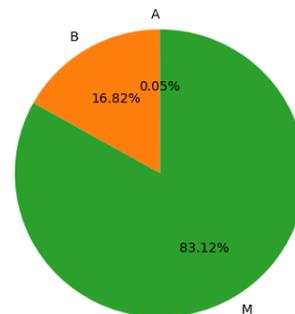
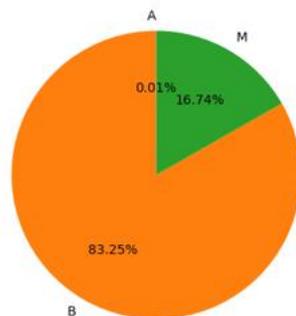


Gráfico 62: Distribución SAIDI por Clase y Nivel de Tensión. AÑO 2022

Distribución SAIDI - Año 2023, Clase 1



Distribución SAIDI - Año 2023, Clase 2

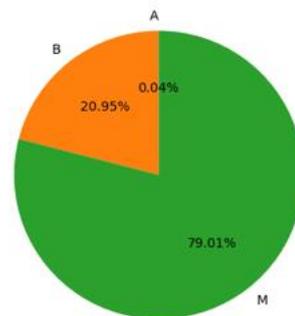


Gráfico 63: Distribución SAIDI por Clase y Nivel de Tensión. AÑO 2023

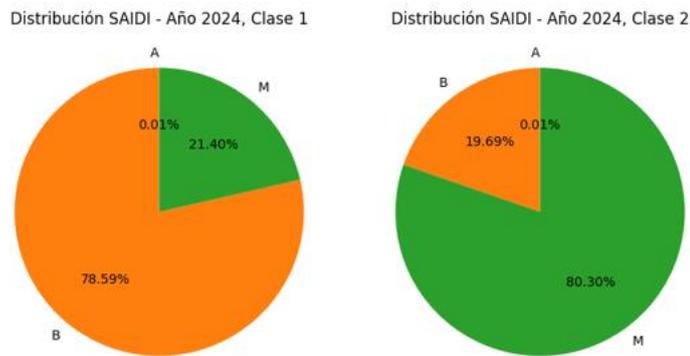


Gráfico 64: Distribución SAIDI por Clase y Nivel de Tensión. AÑO 2024

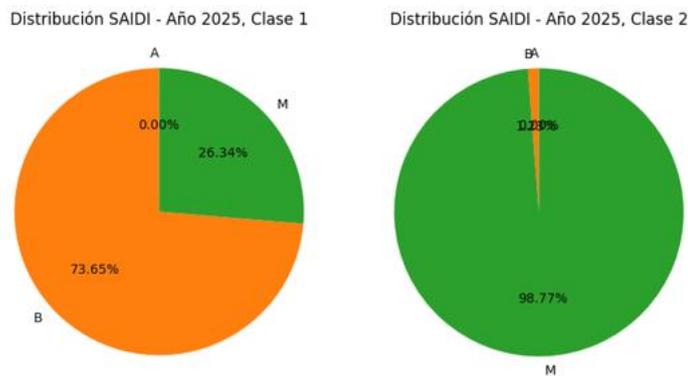


Gráfico 65: Distribución SAIDI por Clase y Nivel de Tensión. AÑO 2025

Los resultados reflejan que, en el caso de las incidencias programadas (Clase 1), el SAIDI asociado a la Baja Tensión (BT) supera el 73% en todos los años analizados, lo que evidencia que las interrupciones programadas en BT representan la mayor parte del tiempo total de interrupción percibido por los clientes.

Por el contrario, en el caso de las incidencias imprevistas (Clase 2), se observa que más del 79% del SAIDI corresponde a interrupciones en Media Tensión (MT), lo cual es consistente con la naturaleza de este tipo de incidencias, que habitualmente afectan a tramos de red de mayor extensión y a un mayor número de clientes.

Este resultado pone de manifiesto que, en el contexto de las incidencias programadas, la consideración de las interrupciones en BT adquiere un peso muy relevante, especialmente desde la perspectiva del número de clientes afectados. Esto se alinea con las tendencias observadas en los apartados anteriores, donde se constata un incremento sostenido de los

trabajos programados en Baja Tensión en los últimos años. En consecuencia, si se incorporan estos trabajos en el cálculo de los indicadores, su impacto sobre el SAIDI puede ser significativo, modificando sustancialmente la percepción de la calidad de suministro desde el punto de vista del cliente.

## **SAIFI**

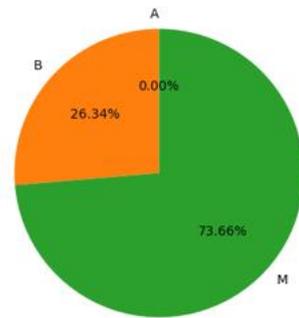
El comportamiento del SAIFI refleja que, tanto en las interrupciones programadas (Clase 1) como en las imprevistas (Clase 2), la mayor parte del indicador está asociada a la Media Tensión (MT) a lo largo de todo el periodo de estudio.

En el caso de las interrupciones programadas, la aportación de la Baja Tensión (BT) resulta relevante, situándose entre aproximadamente un 26% y un 39%, en función del año considerado. Este comportamiento pone de manifiesto que las actuaciones planificadas en BT generan un impacto significativo sobre el número de interrupciones por cliente, especialmente si se considera desde la perspectiva del conjunto de suministros afectados.

Por el contrario, en las interrupciones imprevistas, el predominio de la Media Tensión es prácticamente absoluto, con valores que superan en todos los casos el 98% del SAIFI. Esto refleja que los fallos en la red de MT suelen implicar la desconexión simultánea de un número elevado de clientes, mientras que las interrupciones imprevistas en BT, de carácter más localizado, tienen un peso muy reducido en el total del indicador.

Este análisis refuerza la importancia de incorporar de forma diferenciada la contribución de cada nivel de tensión en la evaluación de los indicadores.

Distribución SAIFI - Año 2022, Clase 1



Distribución SAIFI - Año 2022, Clase 2

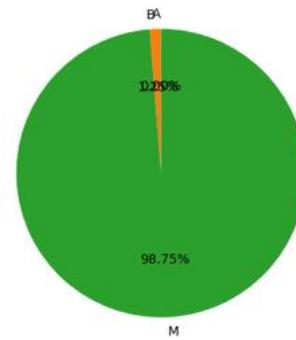
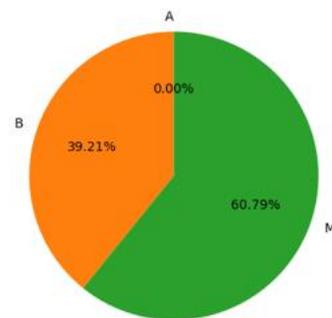


Gráfico 66: Distribución SAIFI por Clase y Nivel de Tensión. AÑO 2022

Distribución SAIFI - Año 2023, Clase 1



Distribución SAIFI - Año 2023, Clase 2

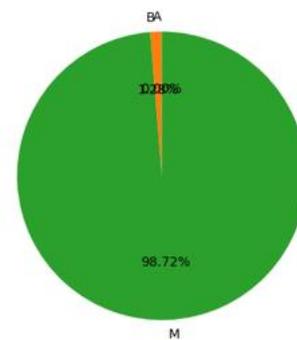
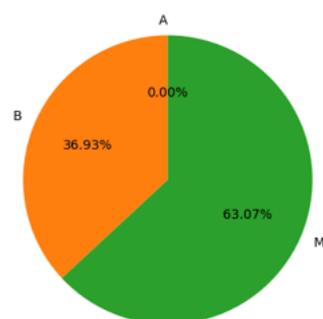


Gráfico 67: Distribución SAIFI por Clase y Nivel de Tensión. AÑO 2023

Distribución SAIFI - Año 2024, Clase 1



Distribución SAIFI - Año 2024, Clase 2

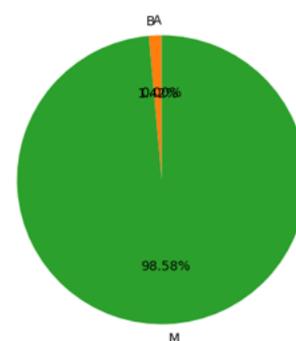


Gráfico 68: Distribución SAIFI por Clase y Nivel de Tensión. AÑO 2024

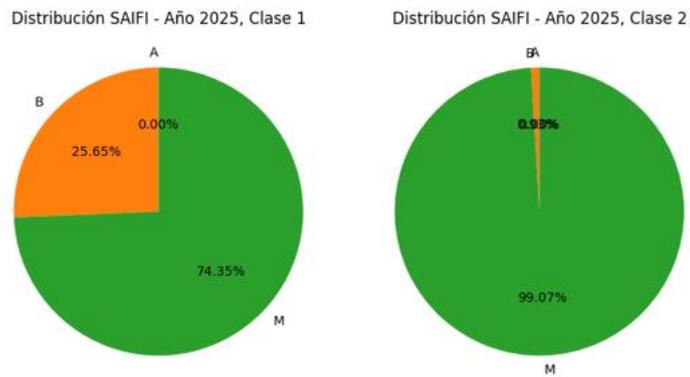


Gráfico 69: Distribución SAIFI por Clase y Nivel de Tensión. AÑO 2025

# ANEXO III: CÓDIGO EMPLEADO PARA EL DESARROLLO DEL TFM

## ANÁLISIS EXPLOTARIO DE LOS DATOS DE INCIDENCIAS

### #Importar librerías

```
! pip install pyathena
from pyathena import connect
import pandas as pd
from pandasql import sqldf
#from IPython.display import Markdown, display

from pyathena import connect
#from IPython.display import Markdown, display
```

### # Configuración de la conexión

**# Se establece la conexión con AWS Athena utilizando el perfil y región especificados.**

```
region_name = 'eu-west-1' # Ejemplo: 'us-west-2'
# Establecer la conexión
conn = connect(profile_name='000370836235_PS-DLAKEUFD-PRO-ANALYTICS',
               region_name='eu-west-1',
               s3_staging_dir='s3://naturgy-dlakeufd-pro/aws-query-athena-
results/')
```

### # Consulta SQL.

**# Consulta que extrae información de incidencias desde múltiples tablas y agrupa por código**

```
consulta = """ SELECT
    i.codigo AS CODIGO,
    i.tipo AS TIPO,
    i4."desc" AS DES_TIPO,
    i.ins_origen AS INS_ORIGEN,
    i2."desc" AS DESC_ORIGEN,
    i.niv_ten AS NIV_TENS,
    i.ins_afectada_sgd AS INS_AFECTADA_SGD,
    b.nombre AS NOMBRE_INSTALACION,
    i.mat_averiado AS MAT_AVERIADO,
    i3."desc" AS DESC_MAT_AVER,
    i.estado AS ESTADO,
    i.descargo AS CLASE,
    i.potencia AS POTENCIA,
    i.n_clien AS N_CLIEN,
    i.pxt AS PXT,
```

```

i.duracion AS DURACION,
i.fecha_det AS FECHA_DET,
i.hora_det AS HORA_DET,
i.cod_grupo AS COD_GRUPO,
i.centro AS CENTRO,
i.not_plazo AS NOT_PLAZO,
i.tip_trabajo AS TIPO_TRABAJO,
i.mercado AS MERCADO,
i.sin_tiepi AS SIN_TIEPI,
i.documentado AS DOCUMENTADO,
i.cod_descargo AS CODIGO_DESCARGO,
i.observaciones AS OBSERVACIONES,
i.localizacion AS LOCALIZACION,
i.descripcion as DESCRIPCION,
LISTAGG(ct.codigo, ', ') WITHIN GROUP (ORDER BY ct.codigo) AS
CTS_PARTICULARES
FROM master_sgi.sgi_incidencias i
LEFT JOIN master_sgi.sgi_tablas_i2 i2 ON i.ins_origen = i2.ins_origen
LEFT JOIN master_sgi.sgi_tablas_i4 i4 ON i4.tipo_incidencia = i.tipo
LEFT JOIN master_sgi.sgi_instalaciones b ON i.ins_afectada_sgd = b.cod_ope
LEFT JOIN master_sgi.sgi_cts_particulares ct ON i.codigo = ct.codigo
LEFT JOIN master_sgi.sgi_tablas_i3 i3 ON TRIM(UPPER(i.mat_averiado)) =
TRIM(UPPER(i3.tipo_material))
WHERE SUBSTRING(i.fecha_det, 1, 4) IN ('2022','2023','2024','2025') and i.estado
!= 'AN'
GROUP BY
    i.codigo, i.tipo, i4."desc", i.ins_origen, i2."desc", i.niv_ten,
    i.ins_afectada_sgd, i.estado, i.descargo, i.potencia, i.n_clien, i.pxt,
    i.duracion, i.fecha_det, i.hora_det, i.cod_grupo, i.centro, i.not_plazo,
    i.mercado, i.sin_tiepi, i.documentado, i.cod_descargo,
    i.localizacion,i.observaciones,i.descripcion,
    b.nombre,i.mat_averiado , i3."desc",i.tip_trabajo ""

```

**#Carga de datos**  
**#Se ejecuta la consulta y se carga el resultado en un DataFrame.**

```

df1 = pd.read_sql(consulta,conn)
df1.head()
df = df1.copy()

```

**#Exploración inicial: Se inspecciona la estructura del DataFrame y los tipos de datos.**

```

print(df.shape)
print(df.dtypes)

```

**#Conversión de fechas y creación de variables temporales**  
**#FECHA\_DET a formato fecha y Creación de una nueva columna llamada año**

```

df["FECHA_DET"] = pd.to_datetime(df["FECHA_DET"], errors="coerce", dayfirst=True)
df["AÑO"] = df["FECHA_DET"].dt.year

```

**#Conversión y estandarización de variables: Se ajustan los tipos de datos de cada columna según su naturaleza (numérica, categórica o texto), lo que facilita el análisis posterior y mejora el rendimiento**

```

df["CODIGO"] = df["CODIGO"] = pd.to_numeric(df["CODIGO"],
errors="coerce").astype("Int64")
df["TIPO"] = pd.to_numeric(df["TIPO"], errors="coerce").astype("Int64")
df["TIPO"] = df["TIPO"].astype("category")
df["DES_TIPO"] = df["DES_TIPO"].astype("category")

```

```

df["INS_ORIGEN"] = df["INS_ORIGEN"].astype("category")
df["DESC_ORIGEN"] = df["DESC_ORIGEN"].astype("category")
df["NIV_TENS"] = df["NIV_TENS"].astype("category")
df["INS_AFECTADA_SGD"] = df["INS_AFECTADA_SGD"].astype("string")
df["NOMBRE_INSTALACION"] = df["NOMBRE_INSTALACION"].astype("string")
df["ESTADO"] = df["ESTADO"].astype("category")
df["CLASE"] = df["CLASE"].astype("category")
df["POTENCIA"] = pd.to_numeric(df["POTENCIA"], errors="coerce")
df["N_CLIEN"] = pd.to_numeric(df["N_CLIEN"], errors="coerce").astype("Int64")
df["POTENCIA"] = pd.to_numeric(df["POTENCIA"], errors="coerce")
df["PXT"] = pd.to_numeric(df["PXT"], errors="coerce")
df["DURACION"] = pd.to_numeric(df["DURACION"], errors="coerce").astype("Int64")
df["HORA_DET"] = pd.to_datetime(df["HORA_DET"], format="%H:%M:%S",
errors="coerce").dt.time
df["MAT_AVERIADO"] = df["MAT_AVERIADO"].astype("category")
df["DESC_MAT_AVER"] = df["DESC_MAT_AVER"].astype("category")
print(df.dtypes)

# Gráficas filtrando por año y por Nivel de Tension
# Agrupar por año y nivel de tensión
df_agrupado = df.groupby(["AÑO", "NIV_TENS"]).size().reset_index(name="CODIGO")
# Filtrar niveles válidos
niveles_validos = ["A", "B", "M", "S"]
df_agrupado = df_agrupado[df_agrupado["NIV_TENS"].isin(niveles_validos)].copy()
# Mapear etiquetas legibles
mapeo_niveles = {"A": "AT", "B": "BT", "M": "MT", "S": "SU"}
df_agrupado["NIV_TENS_LABEL"] = df_agrupado["NIV_TENS"].map(mapeo_niveles)
# Gráfico
fig, axes = plt.subplots(1, 4, figsize=(24, 6), sharey=True)
años = [2022, 2023, 2024, 2025] # Se ha añadido el año 2025
for i, año in enumerate(años):
    datos_año = df_agrupado[df_agrupado["AÑO"] == año]
    sns.barplot(data=datos_año, x="NIV_TENS_LABEL", y="CODIGO", ax=axes[i],
palette="pastel")
    axes[i].set_title(f"Año {año}", fontsize=14)
    axes[i].set_xlabel("Nivel de Tensión", fontsize=12)
    if i == 0:
        axes[i].set_ylabel("Nº de Incidencias", fontsize=12)
    else:
        axes[i].set_ylabel("")
    axes[i].grid(False)

    for p in axes[i].patches:
        height = p.get_height()
        if height > 0:
            axes[i].annotate(f'{int(height):,}',
                (p.get_x() + p.get_width() / 2., height),
                ha='center', va='bottom', fontsize=9)
fig.suptitle("Número de Incidencias por Nivel de Tensión (2022 - 2025)",
fontsize=16, y=1.02)

plt.tight_layout()
plt.show()

```

**#Crear Tabla con el número de incidencias por año y por nivel de tensión.**

```
import pandas as pd

mapeo_niveles = {
    "A": "AT",
    "B": "BT",
    "M": "MT",
    "S": "SU"
}

df_agrupado["NIV_TENS_LABEL"] = df_agrupado["NIV_TENS"].map(mapeo_niveles)

tabla_pivote = df_agrupado.pivot_table(
    index="NIV_TENS_LABEL",
    columns="AÑO",
    values="CODIGO",
    aggfunc="sum"
).fillna(0).astype(int)

orden_niveles = ["AT", "BT", "MT", "SU"]
tabla_pivote = tabla_pivote.loc[orden_niveles]

tabla_formateada = tabla_pivote.style.format("{:,}").set_table_styles([
    {"selector": "th", "props": [("font-weight", "bold"), ("text-align",
"center")]},
    {"selector": "td", "props": [("text-align", "center")]})
]).set_caption("Número de Incidencias por Nivel de Tensión y Año (Niveles A, B,
M, S)")
```

tabla\_formateada

**# Ahora teniendo en cuenta SOLO CLASE=1 y CLASE=2: Grafico indicando año y nivel de tensión sin tener en cuenta la clase 3**

```
import seaborn as sns
import matplotlib.pyplot as plt

# Estilo sin grid
sns.set_style("white")

# Filtramos niveles de tensión válidos (sin 'T' ni vacíos)
clases_validos = ["1", "2"]
niveles_validos = ["A", "B", "M", "S"]

df_filtrado = df[
    df["NIV_TENS"].isin(niveles_validos) &
    (df["CLASE"] != "3")
]

import seaborn as sns
import matplotlib.pyplot as plt
import pandas as pd

# Asegúrate de definir niveles_validos si no está en el entorno
niveles_validos = ["A", "B", "M", "S"]
```

```
# Ordenamos niveles
df_filtrado["NIV_TENS"] = pd.Categorical(df_filtrado["NIV_TENS"],
categories=niveles_validos, ordered=True)

# Mapeamos etiquetas legibles
mapeo_niveles = {
    "A": "AT",
    "B": "BT",
    "M": "MT",
    "S": "SU"
}
df_filtrado["NIV_TENS_LABEL"] = df_filtrado["NIV_TENS"].map(mapeo_niveles)

# Agrupamos por año y nivel
df_agrupado = df_filtrado.groupby(["AÑO",
"NIV_TENS_LABEL"]).size().reset_index(name="CODIGO")

# Lista fija de años
años = [2022, 2023, 2024, 2025]

# Creamos 4 subgráficos
fig, axes = plt.subplots(1, 4, figsize=(24, 6), sharey=True)

for i, año in enumerate(años):
    datos_año = df_agrupado[df_agrupado["AÑO"] == año]

    sns.barplot(data=datos_año, x="NIV_TENS_LABEL", y="CODIGO", ax=axes[i],
palette="pastel")

    axes[i].set_title(f"Año {año}", fontsize=14)
    axes[i].set_xlabel("Nivel de Tensión", fontsize=12)
    if i == 0:
        axes[i].set_ylabel("Nº de Incidencias", fontsize=12)
    else:
        axes[i].set_ylabel("")

    axes[i].grid(False)

# Anotaciones de cantidad encima de las barras
for p in axes[i].patches:
    height = p.get_height()
    if height > 0:
        axes[i].annotate(f'{int(height):,}',
            (p.get_x() + p.get_width() / 2., height),
            ha='center', va='bottom', fontsize=9)

# Título principal
fig.suptitle("Número de Incidencias por Nivel de Tensión (2022 - 2025) y CLASE #
3", fontsize=16, y=1.02)

plt.tight_layout()
plt.show()
```

**#Crear Tabla con el número de incidencias por año y por nivel de tensión sin tener en cuenta la clase 3**

```

pip install Jinja2
import pandas as pd
# Crear la tabla pivote: filas = nivel tensión, columnas = año, valores = número
de incidencias
tabla_pivote = df_agrupado.pivot_table(
    index="NIV_TENS_LABEL",
    columns="AÑO",
    values="CODIGO",
    aggfunc="sum"
).fillna(0).astype(int)

# Reordenar
orden_niveles = ["AT", "BT", "MT", "SU"]
tabla_pivote = tabla_pivote.loc[orden_niveles]

tabla_formateada = tabla_pivote.style.format("{:,}").set_table_styles([
    {"selector": "th", "props": [("font-weight", "bold"), ("text-align",
"center")]},
    {"selector": "td", "props": [("text-align", "center")]})
]).set_caption("Número de Incidencias por Nivel de Tensión y Año")

tabla_formateada

```

**#TABLA DEL NUMERO DE INCIDENCIAS TOTAL POR TIPO DE INCIDENCIA**

**Número de incidencias según su tipo (1xx,2xx, 3xx y 4xx) y según ins\_origen (BXX, CXX, DXX, TXX, GXX)**

```

#Agrupar por TIPO para saber cuántas hay en cada tipo y ver coherencia
tipos_conteo = df.groupby("TIPO").agg({"CODIGO":
"count"}).rename(columns={"CODIGO": "N_INCIDENCIAS"}).reset_index()
# Agrupar y contar, manteniendo la descripción asociada a cada tipo
tipos_conteo = (
    df.groupby("TIPO")
        .agg(
            N_INCIDENCIAS=("CODIGO", "count"),
            DESC_TIPO=("DES_TIPO", "first") # O "max" si estás segura que es
siempre única
        )
        .reset_index()
)

# Ordenar por número de incidencias
tipos_conteo = tipos_conteo.sort_values(by="N_INCIDENCIAS", ascending=False)
tipos_conteo = tipos_conteo[["TIPO", "DESC_TIPO", "N_INCIDENCIAS"]]

# Mostrar tabla
tabla_formateada = tipos_conteo.style.format({
    "N_INCIDENCIAS": "{:,}"
}).set_table_styles([
    {"selector": "th", "props": [("font-weight", "bold"), ("text-align",
"center")]},
    {"selector": "td", "props": [("text-align", "center")]})
]).set_caption("Número de Incidencias por Tipo con Descripción")

tabla_formateada

```

```

#Tabla para tener código de tipo y su descripción
df_distintos = df.drop_duplicates(subset='TIPO')[['TIPO', 'DES_TIPO']]
print(df_distintos)

#TABLA PARA CONTAR NÚMERO DE INCIDENCIAS POR INS_ORIGEN
tipos_conteo = df.gr      .agg(
oupby("INS_ORIGEN").agg({"CODIGO": "count"}).rename(columns={"CODIGO":
"N_INCIDENCIAS"}).reset_index()

# Agrupar y contar, manteniendo la descripción asociada a cada tipo
tipos_conteo = (
    df.groupby("INS_ORIGEN")
        N_INCIDENCIAS=("CODIGO", "count"),
        DESC_TIPO=("DESC_ORIGEN", "first")
    )
    .reset_index()
)

# Ordenar por número de incidencias
tipos_conteo = tipos_conteo.sort_values(by="N_INCIDENCIAS", ascending=False)
tipos_conteo = tipos_conteo[["INS_ORIGEN", "DESC_TIPO", "N_INCIDENCIAS"]]

# Mostrar tabla
tabla_formateada = tipos_conteo.style.format({
    "N_INCIDENCIAS": "{:,}"
}).set_table_styles([
    {"selector": "th", "props": [("font-weight", "bold"), ("text-align",
"center")]},
    {"selector": "td", "props": [("text-align", "center")]})
]).set_caption("Número de Incidencias por Tipo con Descripción")

tabla_formateada
#Tabla para tener Código y la descripción de cada INS_ORIGEN
df_distintos = df.drop_duplicates(subset='INS_ORIGEN')[['INS_ORIGEN',
'DESC_ORIGEN']]
print(df_distintos)

#Análisis de las incidencias por tipo 1xx, 2xx, 3xx y 4xx
# TABLA CON EL NUMERO CE INCIDENCIAS PARA TIPO 1XX
tipos_conteo = (
    df.groupby("TIPO")
        .agg(
            N_INCIDENCIAS=("CODIGO", "count"),
            DESC_TIPO=("DES_TIPO", "first") # O "max" si estás segura que es
siempre única
        )
        .reset_index()
)
df["TIPO"] = pd.to_numeric(df["TIPO"],
errors="coerce").astype("Int64").astype(str)
df["TIPO"].head()
tipos_menor_200 = df[df["TIPO"].str.startswith("1")]
# 1. Asegurarse de que TIPO es string sin decimales
tipos_conteo["TIPO"] = tipos_conteo["TIPO"].astype(str)

```

```
# 2. Filtrar los tipos que empiezan por '1'
tipos_menor_200 = tipos_conteo[
    tipos_conteo["TIPO"].str.startswith("1")
]

# 3. Ordenar por número de incidencias de mayor a menor
tipos_menor_200 = tipos_menor_200.sort_values(by="N_INCIDENCIAS",
ascending=False)

# 4. Reordenar columnas
tipos_menor_200 = tipos_menor_200[["TIPO", "DESC_TIPO", "N_INCIDENCIAS"]]

# 5. Mostrar como tabla formateada
tabla_formateada = tipos_menor_200.style.format({
    "N_INCIDENCIAS": "{:,}"
}).set_table_styles([
    {"selector": "th", "props": [("font-weight", "bold"), ("text-align",
"center")]},
    {"selector": "td", "props": [("text-align", "center")]})
].set_caption("Tipos que comienzan por '1' ordenados por Número de Incidencias")
tabla_formateada
```

#### **# TABLA CON EL NUMERO DE INCIDENCIAS PARA TIPO 1XX por AÑOS**

```
# Asegurar conversión y limpieza
df["TIPO"] = df["TIPO"].astype(str).str.rstrip(".0")
df["TIPO_NUM"] = pd.to_numeric(df["TIPO"], errors='coerce')
df = df[df["TIPO_NUM"].notna()]
df["TIPO_NUM"] = df["TIPO_NUM"].astype(int)
df["AÑO"] = df["AÑO"].astype(int)

# Filtrar tipos entre 100 y 199
df_1xx = df[(df["TIPO_NUM"] >= 100) & (df["TIPO_NUM"] < 200)].copy()

# Crear columna combinada TIPO + DESCRIPCIÓN
df_1xx["TIPO_DESC"] = df_1xx["TIPO"] + " - " + df_1xx["DESC_TIPO"].astype(str)

# Crear tabla pivote
tabla_1xx = (
    df_1xx.pivot_table(
        index="TIPO_DESC",
        columns="AÑO",
        values="CODIGO",
        aggfunc="count",
        fill_value=0
    )
)

# Agregar columna de TOTAL por tipo
tabla_1xx["TOTAL"] = tabla_1xx.sum(axis=1)

# Ordenar de mayor a menor por TOTAL
tabla_1xx = tabla_1xx.sort_values(by="TOTAL", ascending=False)

# Mover columna TOTAL al principio
cols = ["TOTAL"] + [col for col in tabla_1xx.columns if col != "TOTAL"]
tabla_1xx = tabla_1xx[cols]
```

```
# Mostrar tabla
tabla_lxx.style.set_caption("Número de Incidencias por Tipos 1XX y Año
(Ordenado)").format("{:,}")

# TABLA CON EL NUMERO CE INCIDENCIAS PARA TIPO 1XX por NIVEL DE TENSION

# Asegurar que TIPO es texto sin .0
df["TIPO"] = df["TIPO"].astype(str).str.rstrip(".0")

# Convertir TIPO a número y filtrar los 1XX
df["TIPO_NUM"] = pd.to_numeric(df["TIPO"], errors='coerce')
df = df[df["TIPO_NUM"].notna()]
df["TIPO_NUM"] = df["TIPO_NUM"].astype(int)

# Definir los tipos deseados (100 a 199 como string)
tipos_deseados = df[(df["TIPO_NUM"] >= 100) & (df["TIPO_NUM"] <
200)]["TIPO"].unique().tolist()

# Filtrar el DataFrame original
df_filtrado = df[df["TIPO"].isin(tipos_deseados)].copy()

# Crear columna TIPO + descripción
df_filtrado["DES_TIPO"] = df_filtrado["TIPO"] + " - " +
df_filtrado["DES_TIPO"].astype(str)

# Crear tabla pivote: filas = tipo-desc, columnas = nivel de tensión
tabla_pivote = (
    df_filtrado.pivot_table(
        index="DES_TIPO",
        columns="NIV_TENS",
        values="CODIGO",
        aggfunc="count",
        fill_value=0
    )
)

# Añadir columna TOTAL y ordenar de mayor a menor
tabla_pivote["TOTAL"] = tabla_pivote.sum(axis=1)
tabla_pivote = tabla_pivote.sort_values(by="TOTAL", ascending=False)

# Reordenar columnas: TOTAL al inicio
cols = ["TOTAL"] + [col for col in tabla_pivote.columns if col != "TOTAL"]
tabla_pivote = tabla_pivote[cols]

# Mostrar tabla con formato
tabla_pivote.style.format("{:,}").set_caption("Número de Incidencias por Tipos
1XX y Nivel de Tensión")

# GRAFICO CON EL NUMERO de INCIDENCIAS PARA TIPO 1XX por NIVELES DE TENSION

import matplotlib.pyplot as plt
import seaborn as sns

# Paso 1: Asegurar formato correcto
```

```
df["TIPO"] = df["TIPO"].astype(str).str.rstrip(".0")
df["NIV_TENS"] = df["NIV_TENS"].astype(str)

# Paso 2: Filtrar DataFrame por tipos deseados
df_filtrado = df[df["TIPO"].isin(tipos_deseados)].copy()

# Paso 3: Crear columna combinada
df_filtrado["DES_TIPO"] = df_filtrado["TIPO"] + " - " +
df_filtrado["DES_TIPO"].astype(str)

# Paso 4: Agrupar por TIPO+desc y NIV_TENS
df_agrupado = df_filtrado.groupby(["DES_TIPO", "NIV_TENS"]).agg({
    "CODIGO": "count"
}).reset_index().rename(columns={"CODIGO": "N_INCIDENCIAS"})

# Paso 5: Calcular total por TIPO para ordenar
orden =
df_agrupado.groupby("DES_TIPO")["N_INCIDENCIAS"].sum().sort_values(ascending=False).index
df_agrupado["DES_TIPO"] = pd.Categorical(df_agrupado["DES_TIPO"],
categories=orden, ordered=True)

# Paso 6: Crear gráfico
plt.figure(figsize=(14, 7))
sns.barplot(data=df_agrupado, x="DES_TIPO", y="N_INCIDENCIAS", hue="NIV_TENS",
palette="pastel")
plt.title("Número de Incidencias por Tipos 1XX y Nivel de Tensión", fontsize=16)
plt.xlabel("TIPO - DESCRIPCIÓN")
plt.ylabel("Nº de Incidencias")
plt.xticks(rotation=90)
plt.legend(title="Nivel Tensión")
plt.tight_layout()
plt.show()
```

**Se comprueba que todos los que tienen el nivel de tensión vacío no tienen interrupciones (PTX=0) y el tipo de trabajo es un RE, CA o IR. Se genera tabla**

```
# Asegurar formato correcto
df["TIPO"] = df["TIPO"].astype(str).str.rstrip(".0")

# Filtrar tipo que empieza por '1'
df_1xx = df[df["TIPO"].str.startswith("1")].copy()

# Filtrar los que tienen NIV_TENS vacío o nulo
df_1xx_sin_tension = df_1xx[df_1xx["NIV_TENS"].isna() |
(df_1xx["NIV_TENS"].str.strip() == "")]

# Filtrar con PXT = 0
df_filtrado = df_1xx_sin_tension[df_1xx_sin_tension["PXT"] == 0]

# Agrupar por TIPO_TRABAJO y contar registros + mostrar columna PXT
resumen = (
    df_filtrado
```

```
.groupby("TIPO_TRABAJO", dropna=False)
.agg(
    N_REGISTROS=("CODIGO", "count"),
    PXT=("PXT", "first") # o puedes usar "mean" si quieres el promedio
)
.reset_index()
)
```

```
resumen_styled = resumen.style.format({
    "N_REGISTROS": "{:,}",
    "PXT": "{:d}"
}).set_caption("Trabajos tipo 1 (sin NIV_TENS y PXT=0)").set_table_styles([
    {"selector": "th", "props": [("font-weight", "bold"), ("text-align",
"center")]},
    {"selector": "td", "props": [("text-align", "center")]
}])
```

resumen\_styled

#### # TABLA CON EL NUMERO CE INCIDENCIAS PARA TIPO 2XX

```
# 1. Asegurar que TIPO es string sin decimales
tipos_conteo["TIPO"] = tipos_conteo["TIPO"].astype(str).str.rstrip(".0")

# 2. Filtrar los TIPO que empiecen por '2'
tipos_2xx = tipos_conteo[tipos_conteo["TIPO"].str.startswith("2")]

# 3. Ordenar por N_INCIDENCIAS de mayor a menor
tipos_2xx = tipos_2xx.sort_values(by="N_INCIDENCIAS", ascending=False)

# 4. Reordenar columnas si es necesario
tipos_2xx = tipos_2xx[["TIPO", "DESC_TIPO", "N_INCIDENCIAS"]]

# 5. Mostrar en formato bonito
tabla_formateada = tipos_2xx.style.format({
    "N_INCIDENCIAS": "{:,}"
}).set_table_styles([
    {"selector": "th", "props": [("font-weight", "bold"), ("text-align",
"center")]},
    {"selector": "td", "props": [("text-align", "center")]
}]).set_caption("Tipos que comienzan por '2' ordenados por Número de Incidencias")

tabla_formateada
```

#### # TABLA CON EL NUMERO CE INCIDENCIAS PARA TIPO 2XX por AÑO

```
# Asegurar formato correcto
df["TIPO"] = df["TIPO"].astype(str).str.rstrip(".0")
df["AÑO"] = df["AÑO"].astype(int)

# Filtrar solo los tipos que empiezan por '2'
df_2xx = df[df["TIPO"].str.startswith("2")].copy()

# Crear columna combinada para mostrar TIPO + descripción
```

```
df_2xx["DES_TIPO"] = df_2xx["TIPO"] + " - " + df_2xx["DES_TIPO"].astype(str)

# Crear tabla pivote: filas = tipo+desc, columnas = año, valores = conteo
tabla_2xx_anual = (
    df_2xx.pivot_table(
        index="DES_TIPO",
        columns="AÑO",
        values="CODIGO",
        aggfunc="count",
        fill_value=0
    )
)

# Mostrar tabla en formato bonito
tabla_2xx_anual.style.format("{:,}").set_caption("Número de Incidencias por Tipo
2XX y Año")

# TABLA CON EL NUMERO CE INCIDENCIAS PARA TIPO 2XX por NIVEL DE TENSION

# Asegurar formato correcto
df["TIPO"] = df["TIPO"].astype(str).str.rstrip(".0")
df["NIV_TENS"] = df["NIV_TENS"]

# Filtrar solo los tipos que empiezan por '2'
df_2xx = df[df["TIPO"].str.startswith("2")].copy()

# Crear columna combinada para mostrar TIPO + descripción
df_2xx["DES_TIPO"] = df_2xx["TIPO"] + " - " + df_2xx["DES_TIPO"].astype(str)

# Crear tabla pivote: filas = tipo+desc, columnas = año, valores = conteo
tabla_2xx_anual = (
    df_2xx.pivot_table(
        index="DES_TIPO",
        columns="NIV_TENS",
        values="CODIGO",
        aggfunc="count",
        fill_value=0
    )
)

# Mostrar tabla en formato bonito
tabla_2xx_anual.style.format("{:,}").set_caption("Número de Incidencias por Tipo
2XX y Nivel de tension")

# TABLA CON EL NUMERO CE INCIDENCIAS PARA TIPO 3XX

# 1. Asegurar que TIPO es string sin decimales
tipos_conteo["TIPO"] = tipos_conteo["TIPO"].astype(str).str.rstrip(".0")

# 2. Filtrar los TIPO que empiecen por '3'
tipos_3xx = tipos_conteo[tipos_conteo["TIPO"].str.startswith("3")]

# 3. Ordenar por N_INCIDENCIAS de mayor a menor
tipos_3xx = tipos_3xx.sort_values(by="N_INCIDENCIAS", ascending=False)
```

```
# 4. Reordenar columnas si es necesario
tipos_3xx = tipos_3xx[["TIPO", "DESC_TIPO", "N_INCIDENCIAS"]]

# 5. Mostrar en formato bonito
tabla_formateada = tipos_3xx.style.format({
    "N_INCIDENCIAS": "{:,}"
}).set_table_styles([
    {"selector": "th", "props": [("font-weight", "bold"), ("text-align",
"center")]},
    {"selector": "td", "props": [("text-align", "center")]})
].set_caption("Tipos que comienzan por '3'ordenados por Número de Incidencias")

tabla_formateada
```

#### **# TABLA CON EL NUMERO CE INCIDENCIAS PARA TIPO 3XX por AÑO**

```
# Asegurar que TIPO y AÑO están bien formateados
df["TIPO"] = df["TIPO"].astype(str).str.rstrip(".0")
df["AÑO"] = df["AÑO"].astype(int)

# Filtrar solo TIPO que empieza por '3'
df_3xx = df[df["TIPO"].str.startswith("3")].copy()

# Crear columna TIPO + DESC_TIPO
df_3xx["DES_TIPO"] = df_3xx["TIPO"] + " - " + df_3xx["DESC_TIPO"].astype(str)

# Crear tabla pivote por AÑO
tabla_3xx = df_3xx.pivot_table(
    index="DES_TIPO",
    columns="AÑO",
    values="CODIGO",
    aggfunc="count",
    fill_value=0
)

# Mostrar la tabla en formato bonito
tabla_3xx.style.format("{:,}").set_caption("Número de Incidencias por Tipo 3XX y
Año")
```

#### **# TABLA CON EL NUMERO CE INCIDENCIAS PARA TIPO 3XX por NIVEL DE TENSION**

```
# Asegurar que TIPO y AÑO están bien formateados
df["TIPO"] = df["TIPO"].astype(str).str.rstrip(".0")
df["NIV_TENS"] = df["NIV_TENS"]

# Filtrar solo TIPO que empieza por '3'
df_3xx = df[df["TIPO"].str.startswith("3")].copy()

# Crear columna TIPO + DESC_TIPO
df_3xx["DES_TIPO"] = df_3xx["TIPO"] + " - " + df_3xx["DESC_TIPO"].astype(str)

# Crear tabla pivote por AÑO
tabla_3xx = df_3xx.pivot_table(
    index="DES_TIPO",
```

```
columns="NIV_TENS",
values="CODIGO",
aggfunc="count",
fill_value=0
)

# Mostrar la tabla en formato bonito
tabla_3xx.style.format("{:,}").set_caption("Número de Incidencias por Tipo 3XX y
NIVEL DE TENSION ")

# TABLA CON EL NUMERO CE INCIDENCIAS PARA TIPO 4XX

# 1. Asegurar que TIPO es string sin decimales
tipos_conteo["TIPO"] = tipos_conteo["TIPO"].astype(str).str.rstrip(".0")

# 2. Filtrar los TIPO que empiecen por '4'
tipos_4xx = tipos_conteo[tipos_conteo["TIPO"].str.startswith("4")]

# 3. Ordenar por N_INCIDENCIAS de mayor a menor
tipos_4xx = tipos_4xx.sort_values(by="N_INCIDENCIAS", ascending=False)

# 4. Reordenar columnas si es necesario
tipos_4xx = tipos_4xx[["TIPO", "DESC_TIPO", "N_INCIDENCIAS"]]

# 5. Mostrar en formato bonito
tabla_formateada = tipos_4xx.style.format({
    "N_INCIDENCIAS": "{:,}"
}).set_table_styles([
    {"selector": "th", "props": [("font-weight", "bold"), ("text-align",
"center")]},
    {"selector": "td", "props": [("text-align", "center")]})
].set_caption("Tipos que comienzan por '4' (4XX) ordenados por Número de
Incidencias")

tabla_formateada

# TABLA CON EL NUMERO CE INCIDENCIAS PARA TIPO 4XX por AÑO
# Asegurar que TIPO y AÑO están bien formateados
df["TIPO"] = df["TIPO"].astype(str).str.rstrip(".0")
df["AÑO"] = df["AÑO"].astype(int)

# Filtrar solo TIPO que empieza por '4'
df_4xx = df[df["TIPO"].str.startswith("4")].copy()

# Crear columna TIPO + DESC_TIPO
df_4xx["DES_TIPO"] = df_4xx["TIPO"] + " - " + df_4xx["DESC_TIPO"].astype(str)

# Crear tabla pivote por AÑO
tabla_4xx = df_4xx.pivot_table(
    index="DES_TIPO",
    columns="AÑO",
    values="CODIGO",
    aggfunc="count",
    fill_value=0
)
```

```
# Mostrar la tabla en formato bonito
tabla_4xx.style.format("{:,}") .set_caption("Número de Incidencias por Tipo 4XX y Año")
```

**# TABLA CON EL NUMERO DE INCIDENCIAS PARA TIPO 4XX por NIVEL DE TENSION**

```
# Asegurar que TIPO y AÑO están bien formateados
df["TIPO"] = df["TIPO"].astype(str).str.rstrip(".0")
df["NIV_TENS"] = df["NIV_TENS"]

# Filtrar solo TIPO que empieza por '4'
df_4xx = df[df["TIPO"].str.startswith("4")].copy()

# Crear columna TIPO + DESC_TIPO
df_4xx["DES_TIPO"] = df_4xx["TIPO"] + " - " + df_4xx["DESC_TIPO"].astype(str)

# Crear tabla pivote por AÑO
tabla_4xx = df_4xx.pivot_table(
    index="DES_TIPO",
    columns="NIV_TENS",
    values="CODIGO",
    aggfunc="count",
    fill_value=0
)

# Mostrar la tabla en formato bonito
tabla_4xx.style.format("{:,}") .set_caption("Número de Incidencias por Tipo 4XX y Nivel de Tension")
```

**# TABLA CON EL NUMERO DE INCIDENCIAS DEL TIPO 1XX (PROGRAMADAS POR NIVEL DE TENSION Y AÑO**

```
tipos_menor_200 = df[df["TIPO"].str.startswith("1")].copy()
tipos_mayor_200 = df[df["TIPO"].str.startswith(("2", "3", "4"))].copy()
#Agrupar por NIV_TENS para los tipo 1xx
tabla_menor = (
    tipos_menor_200
    .groupby(["AÑO", "NIV_TENS"])
    .size()
    .reset_index(name="N_REGISTROS")
    .pivot_table(index="NIV_TENS", columns="AÑO", values="N_REGISTROS",
    fill_value=0)
)

print("Número de registros por Nivel de Tensión y Año (TIPOS 1XX):")
print(tabla_menor)
```

**#FILTRAR SOLO LOS NIVELES DE TENSION VALIDOS PARA LAS INCIDENCIAS tipo 1xx (QUITAR LAS TIPO T) DE LA TABLA**

```
# Eliminar registros con NIV_TENS vacío o igual a 'T'
tipos_menor_200 = tipos_menor_200[
    tipos_menor_200["NIV_TENS"].notna() &
    (tipos_menor_200["NIV_TENS"].str.strip() != "") &
```

```

    (tipos_menor_200["NIV_TENS"].str.strip() != "T")
]

# Crear tabla pivote
tabla_menor = (
    tipos_menor_200
    .groupby(["AÑO", "NIV_TENS"])
    .size()
    .reset_index(name="N_REGISTROS")
    .pivot_table(index="NIV_TENS", columns="AÑO", values="N_REGISTROS",
    fill_value=0)
)

# Mostrar resultado
print("Número de registros por Nivel de Tensión y Año (TIPOS 1XX):")
print(tabla_menor)

# GRÁFICO CON EL NÚMERO DE INCIDENCIAS POR NIVEL DE TENSION Y AÑO

import seaborn as sns
import matplotlib.pyplot as plt

# Resetear el índice para que 'NIV_TENS' sea una columna y hacer melt para
convertirlo a formato largo
datos_grafico = tabla_menor.reset_index().melt(id_vars="NIV_TENS",
var_name="AÑO", value_name="N_REGISTROS")

# Crear el gráfico de barras
plt.figure(figsize=(10, 6))
ax = sns.barplot(data=datos_grafico, x="NIV_TENS", y="N_REGISTROS", hue="AÑO",
palette="pastel")

plt.title("Número de Incidencias por Nivel de Tensión y Año (TIPOS 1XX)",
fontsize=14)
plt.xlabel("Nivel de Tensión", fontsize=12)
plt.ylabel("Nº de Incidencias", fontsize=12)

# Etiquetas encima de las barras
for p in ax.patches:
    height = p.get_height()
    if height > 0:
        ax.annotate(f'{int(height):,}',
                    (p.get_x() + p.get_width() / 2., height),
                    ha='center', va='bottom', fontsize=8, rotation=90)

plt.tight_layout()
plt.legend(title="Año")
plt.show()

# TABLA CON EL NÚMERO DE INCIDENCIAS DEL TIPO 2XX, 3XX Y 4XX POR AÑO
tabla_mayor = (
    tipos_mayor_200
    .groupby(["AÑO", "NIV_TENS"])
    .size()
    .reset_index(name="N_REGISTROS")

```

```

    .pivot_table(index="NIV_TENS", columns="AÑO", values="N_REGISTROS",
fill_value=0)
)

print("Número de registros por Nivel de Tensión y Año (TIPOS >= 2XX):")
print(tabla_mayor)

# Eliminar registros con NIV_TENS vacío o igual a 'T'
tipos_mayor_200 = tipos_mayor_200
[
    tipos_menor_200["NIV_TENS"].notna() &
    (tipos_menor_200["NIV_TENS"].str.strip() != "") &
    (tipos_menor_200["NIV_TENS"].str.strip() != "T")
]

# Crear tabla pivote
tabla_mayor =(tipos_mayor_200
    .groupby(["AÑO", "NIV_TENS"])
    .size()
    .reset_index(name="N_REGISTROS")
    .pivot_table(index="NIV_TENS", columns="AÑO", values="N_REGISTROS",
fill_value=0)
)

# Mostrar resultado
print("Número de registros por Nivel de Tensión y Año (TIPOS 1XX):")
print(tabla_mayor)
# Resetea el índice para que 'NIV_TENS' sea una columna y hacer melt para
convertirlo a formato largo
datos_grafico_mayor = tabla_mayor.reset_index().melt(id_vars="NIV_TENS",
var_name="AÑO", value_name="N_REGISTROS")

#GRAFICO DE BARRAS CON EL NUMERO DE INCIDENCIAS DEL TIPO 2XX O 3XX O 4XX, POR
NIVEL DE TENSION Y AÑO

# Crear el gráfico de barras
plt.figure(figsize=(10, 6))
ax = sns.barplot(data=datos_grafico_mayor, x="NIV_TENS", y="N_REGISTROS",
hue="AÑO", palette="pastel")

plt.title("Número de Incidencias por Nivel de Tensión y Año (TIPOS 2XX,3xx y
4xx)", fontsize=14)
plt.xlabel("Nivel de Tensión", fontsize=12)
plt.ylabel("N° de Incidencias", fontsize=12)

# Etiquetas encima de las barras
for p in ax.patches:
    height = p.get_height()
    if height > 0:
        ax.annotate(f'{int(height):,}',
            (p.get_x() + p.get_width() / 2., height),
            ha='center', va='bottom', fontsize=8, rotation=90)

plt.tight_layout()

```

```
plt.legend(title="Año")
plt.show()
```

**#MISMO GRAFICO AGRUPADO PARA VER CADA AÑO, EL NUMERO DE INCIDENCIAS IMPREVISTAS (TIPO 2XX, 3XX Y 4XX) POR NIVELES DE TENSION**

```
tipos_conteo = df.groupby("TIPO").agg({"CODIGO":
"count"}).rename(columns={"CODIGO": "N_INCIDENCIAS"}).reset_index()

tipos_mayor_200 = tipos_conteo[tipos_conteo["N_INCIDENCIAS"] > 200]["TIPO"]
tipos_menor_igual_200 = tipos_conteo[tipos_conteo["N_INCIDENCIAS"] <=
200]["TIPO"]

df_mayor_200 = df[df["TIPO"].isin(tipos_mayor_200)]
df_menor_200 = df[df["TIPO"].isin(tipos_menor_igual_200)]
# AGRUPAR POR NIVEL DE TENSION
grupo_mayor = df_mayor_200.groupby(["AÑO", "NIV_TENS"]).agg({
"CODIGO": "count"
}).rename(columns={"CODIGO": "N_INCIDENCIAS"}).reset_index()

grupo_menor = df_menor_200.groupby(["AÑO", "NIV_TENS"]).agg({
"CODIGO": "count"
}).rename(columns={"CODIGO": "N_INCIDENCIAS"}).reset_index()
# FILTRAR NIVELES VALIDOS
niveles_validos = ["A", "B", "M", "S"]
grupo_mayor = grupo_mayor[grupo_mayor["NIV_TENS"].isin(niveles_validos)]
grupo_menor = grupo_menor[grupo_menor["NIV_TENS"].isin(niveles_validos)]

grupo_menor["NIV_TENS"] = grupo_menor["NIV_TENS"].astype("category")
grupo_menor["NIV_TENS"] = grupo_menor["NIV_TENS"].cat.remove_unused_categories()

import seaborn as sns
import matplotlib.pyplot as plt

# Añadir el año 2025
años = [2022, 2023, 2024, 2025]

# --- GRÁFICOS PARA TIPOS > 200 ---
fig, axes = plt.subplots(1, 4, figsize=(24, 6), sharey=True)

for i, año in enumerate(años):
    datos = grupo_mayor[grupo_mayor["AÑO"] == año]

    sns.barplot(data=datos, x="NIV_TENS", y="N_INCIDENCIAS", ax=axes[i],
palette="pastel")

    axes[i].set_title(f"Año {año}", fontsize=14)
    axes[i].set_xlabel("Nivel de Tensión", fontsize=12)

    if i == 0:
        axes[i].set_ylabel("N° de Incidencias", fontsize=12)
    else:
        axes[i].set_ylabel("")
```

```
axes[i].grid(False)

# Añadir etiquetas de valor encima de las barras
for p in axes[i].patches:
    height = p.get_height()
    if height > 0:
        axes[i].annotate(f'{int(height):,}',
                        (p.get_x() + p.get_width() / 2, height),
                        ha='center', va='bottom', fontsize=9)

# Título general del gráfico
fig.suptitle("Tipos con > 200 Incidencias por Nivel de Tensión", fontsize=16,
y=1.02)

plt.tight_layout()
plt.show()
```

**# GRAFICO AGRUPADO PARA VER EL NUMERO DE INCIDENCIAS PROGRAMADAS (TIPO 1XX) POR NIVELES DE TENSION**

```
import seaborn as sns
import matplotlib.pyplot as plt

# Asegúrate de que 'años' está definido
años = [2022, 2023, 2024, 2025]

# --- GRÁFICOS PARA TIPOS ≤ 200 ---
fig, axes = plt.subplots(1, 4, figsize=(24, 6), sharey=True)

for i, año in enumerate(años):
    datos = grupo_menor[grupo_menor["AÑO"] == año]

    sns.barplot(data=datos, x="NIV_TENS", y="N_INCIDENCIAS", ax=axes[i],
palette="pastel")

    axes[i].set_title(f"Año {año}", fontsize=14)
    axes[i].set_xlabel("Nivel de Tensión", fontsize=12)

    if i == 0:
        axes[i].set_ylabel("N° de Incidencias", fontsize=12)
    else:
        axes[i].set_ylabel("")

    axes[i].grid(False)

# Añadir etiquetas de número encima de las barras
for p in axes[i].patches:
    height = p.get_height()
    if height > 0:
        axes[i].annotate(f'{int(height):,}',
                        (p.get_x() + p.get_width()/2, height),
                        ha='center', va='bottom', fontsize=9)
```

```
# Título principal
fig.suptitle("Tipos con ≤ 200 Incidencias por Nivel de Tensión", fontsize=16,
y=1.02)

plt.tight_layout()
plt.show()

#GRAFICO PARA VER EL NUMERO DE INCIDENCIAS TOTAL PARA TRABAJOS DE CLASE=1
(COMPROBAR SI COINCIDEN CON LOS 1XX) POR NIVELES DE TENSION
df["CLASE"] = df["CLASE"].astype("category")

grupo_clases = df[df["CLASE"].isin(["1", "2"]).groupby(["CLASE",
"NIV_TENS"]).agg({
    "CODIGO": "count"
}).rename(columns={"CODIGO": "N_INCIDENCIAS"}).reset_index()
grupo_1 = grupo_clases[grupo_clases["CLASE"] == "1"]
grupo_2 = grupo_clases[grupo_clases["CLASE"] == "2"]

import seaborn as sns
import matplotlib.pyplot as plt

#GRAFICO PARA VER EL NUMERO DE INCIDENCIAS TOTAL PARA TRABAJOS DE
CLASE=2 (COMPROBAR SI COINCIDEN CON LOS 2XX) POR NIVELES DE TENSION
plt.figure(figsize=(8,5))
ax1 = sns.barplot(data=grupo_1, x="NIV_TENS", y="N_INCIDENCIAS",
palette="pastel")
plt.title("Número de Incidencias por Nivel de Tensión (CLASE 1)")
plt.xlabel("Nivel de Tensión")
plt.ylabel("Nº de Incidencias")
plt.grid(False)
for p in ax1.patches:
    height = p.get_height()
    if height > 0:
        ax1.annotate(f'{int(height):,}',
                    (p.get_x() + p.get_width()/2., height),
                    ha='center', va='bottom', fontsize=9)

plt.tight_layout()
plt.show()

# ----- Gráfico para CLASE 2 -----
plt.figure(figsize=(8,5))
ax2 = sns.barplot(data=grupo_2, x="NIV_TENS", y="N_INCIDENCIAS",
palette="pastel")
plt.title("Número de Incidencias por Nivel de Tensión (CLASE 2)")
plt.xlabel("Nivel de Tensión")
plt.ylabel("Nº de Incidencias")
plt.grid(False)
for p in ax2.patches:
    height = p.get_height()
    if height > 0:
        ax2.annotate(f'{int(height):,}',
                    (p.get_x() + p.get_width()/2., height),
                    ha='center', va='bottom', fontsize=9)

plt.tight_layout()
plt.show()

#GRAFICO NUMERO DE INCIDENCIAS CLASE 1 Y CLASE 2, POR AÑO Y POR NIVEL DE TENSION.
```

```
import seaborn as sns
import matplotlib.pyplot as plt

# Aseguramos que CLASE es string
df["CLASE"] = df["CLASE"].astype(str)

# Filtramos solo CLASE 1 y CLASE 2
df_clase_12 = df[df["CLASE"].isin(["1", "2"])]

# Agrupamos por año, clase y nivel de tensión (sin filtrar niveles)
grupo = df_clase_12.groupby(["AÑO", "CLASE", "NIV_TENS"]).agg({
    "CODIGO": "count"
}).rename(columns={"CODIGO": "N_INCIDENCIAS"}).reset_index()

import seaborn as sns
import matplotlib.pyplot as plt

# Lista de años
años = [2022, 2023, 2024, 2025]

# Creamos 4 subgráficos (uno por año)
fig, axes = plt.subplots(1, 4, figsize=(24, 6), sharey=True)

for i, año in enumerate(años):
    datos = grupo[grupo["AÑO"] == año]
    sns.barplot(data=datos, x="NIV_TENS", y="N_INCIDENCIAS", hue="CLASE",
ax=axes[i], palette="pastel")

    axes[i].set_title(f"Año {año}", fontsize=14)
    axes[i].set_xlabel("Nivel de Tensión", fontsize=12)

    if i == 0:
        axes[i].set_ylabel("Nº de Incidencias", fontsize=12)
    else:
        axes[i].set_ylabel("")

    axes[i].grid(False)

    # Etiquetas encima de cada barra
    for p in axes[i].patches:
        height = p.get_height()
        if height > 0:
            axes[i].annotate(f'{int(height):,}',
                (p.get_x() + p.get_width() / 2., height),
                ha='center', va='bottom', fontsize=9)

# Título general del gráfico
fig.suptitle("Incidencias por Nivel de Tensión y CLASE (sin filtrar niveles)",
    fontsize=16, y=1.02)

plt.tight_layout()
plt.show()
#GRAFICO NUMERO DE INCIDENCIAS AT POR AÑO Y CLASE 1 Y CLASE 2
import seaborn as sns
```

```
import matplotlib.pyplot as plt

nivel = "A"
datos = grupo[grupo["NIV_TENS"] == nivel]

plt.figure(figsize=(8, 6))
ax = sns.barplot(data=datos, x="AÑO", y="N_INCIDENCIAS", hue="CLASE",
palette="pastel")
ax.set_title(f"Nivel de Tensión: {nivel}", fontsize=14)
ax.set_xlabel("Año", fontsize=12)
ax.set_ylabel("Nº de Incidencias", fontsize=12)
ax.grid(False)

for p in ax.patches:
    height = p.get_height()
    if height > 0:
        ax.annotate(f'{int(height):,}',
                    (p.get_x() + p.get_width() / 2., height),
                    ha='center', va='bottom', fontsize=9)

plt.tight_layout()
plt.show()
plt.show()
#GRAFICO NUMERO DE INCIDENCIAS BT POR AÑO Y CLASE 1 Y CLASE 2
nivel = "B"
datos = grupo[grupo["NIV_TENS"] == nivel]

plt.figure(figsize=(8, 6))
ax = sns.barplot(data=datos, x="AÑO", y="N_INCIDENCIAS", hue="CLASE",
palette="pastel")
ax.set_title(f"Nivel de Tensión: {nivel}", fontsize=14)
ax.set_xlabel("Año", fontsize=12)
ax.set_ylabel("Nº de Incidencias", fontsize=12)
ax.grid(False)

for p in ax.patches:
    height = p.get_height()
    if height > 0:
        ax.annotate(f'{int(height):,}',
                    (p.get_x() + p.get_width() / 2., height),
                    ha='center', va='bottom', fontsize=9)

plt.tight_layout()
plt.show()
plt.show()
#GRAFICO NUMERO DE INCIDENCIAS MT POR AÑO Y CLASE 1 Y CLASE 2
nivel = "M"
datos = grupo[grupo["NIV_TENS"] == nivel]

plt.figure(figsize=(8, 6))
ax = sns.barplot(data=datos, x="AÑO", y="N_INCIDENCIAS", hue="CLASE",
palette="pastel")
ax.set_title(f"Nivel de Tensión: {nivel}", fontsize=14)
ax.set_xlabel("Año", fontsize=12)
```

```

ax.set_ylabel("N° de Incidencias", fontsize=12)
ax.grid(False)

for p in ax.patches:
    height = p.get_height()
    if height > 0:
        ax.annotate(f'{int(height):,}',
                    (p.get_x() + p.get_width() / 2., height),
                    ha='center', va='bottom', fontsize=9)

plt.tight_layout()
plt.show()
plt.show()

#GRAFICO NUMERO DE INCIDENCIAS SU POR AÑO Y CLASE 1 Y CLASE 2
nivel = "S"
datos = grupo[grupo["NIV_TENS"] == nivel]
plt.figure(figsize=(8, 6))
ax = sns.barplot(data=datos, x="AÑO", y="N_INCIDENCIAS", hue="CLASE",
                palette="pastel")
ax.set_title(f"Nivel de Tensión: {nivel}", fontsize=14)
ax.set_xlabel("Año", fontsize=12)
ax.set_ylabel("N° de Incidencias", fontsize=12)
ax.grid(False)

for p in ax.patches:
    height = p.get_height()
    if height > 0:
        ax.annotate(f'{int(height):,}',
                    (p.get_x() + p.get_width() / 2., height),
                    ha='center', va='bottom', fontsize=9)

plt.tight_layout()
plt.show()
plt.show()

#GRAFICO NUMERO DE INCIDENCIAS BT, CLASE=1 Y TIPOLOGIA. ¿HAY TIPOS ¡=1XX?
df_clase1_b = df[(df["CLASE"] == "1") & (df["NIV_TENS"] == "B")]

df_clase1_b = df[(df["CLASE"] == "1") & (df["NIV_TENS"] == "B")]

grupo_tipo = df_clase1_b.groupby("TIPO").agg({
    "CODIGO": "count"
}).rename(columns={"CODIGO": "N_INCIDENCIAS"}).reset_index()
import seaborn as sns
import matplotlib.pyplot as plt

# Ordenar el DataFrame por N_INCIDENCIAS de forma descendente
grupo_tipo_sorted = grupo_tipo.sort_values(by="N_INCIDENCIAS", ascending=False)

plt.figure(figsize=(12,6))
ax = sns.barplot(data=grupo_tipo_sorted, x="TIPO", y="N_INCIDENCIAS",
                palette="pastel")

plt.title("Número de Incidencias por TIPO (CLASE 1, Nivel de Tensión B)",
        fontsize=14)

```

```
plt.xlabel("TIPO")
plt.ylabel("N° de Incidencias")
plt.grid(False)

# Etiquetas encima de las barras
for p in ax.patches:
    height = p.get_height()
    if height > 0:
        ax.annotate(f'{int(height):,}',
                    (p.get_x() + p.get_width()/2., height),
                    ha='center', va='bottom', fontsize=9)

plt.xticks(rotation=45)
plt.tight_layout()
plt.show()

plt.show()
#TABLA NUMERO DE INCIDENCIAS BT CALSE=1 POR TIPO (INFO GRAFICA AHORA EN TABLA)

tipos_descripciones = df[["TIPO", "DES_TIPO"]].drop_duplicates()

# Unir la descripción al DataFrame de grupo
grupo_tipo = grupo_tipo.merge(tipos_descripciones, on="TIPO", how="left")

# Ordenar por N_INCIDENCIAS y resetear índice
grupo_tipo = grupo_tipo.sort_values(by="N_INCIDENCIAS",
                                     ascending=False).reset_index(drop=True)

# Mostrar tabla resultante
print(grupo_tipo)

#GRAFICO NUMERO DE INCIDENCIAS BT CALSE=1 POR TIPO Y AÑO
df_clase1_b = df[(df["CLASE"] == "1") & (df["NIV_TENS"] == "B")]
grupo_tipo_año = df_clase1_b.groupby(["AÑO", "TIPO"]).agg({
    "CODIGO": "count"
}).rename(columns={"CODIGO": "N_INCIDENCIAS"}).reset_index()
import seaborn as sns
import matplotlib.pyplot as plt

# Añadir el año 2025 a la lista
años = [2022, 2023, 2024, 2025]

# Crear 4 subgráficos (uno por año)
fig, axes = plt.subplots(1, 4, figsize=(24, 6), sharey=True)

for i, año in enumerate(años):
    datos = grupo_tipo_año[grupo_tipo_año["AÑO"] == año].sort_values(
        by="N_INCIDENCIAS", ascending=False
    )

    sns.barplot(data=datos, x="TIPO", y="N_INCIDENCIAS", ax=axes[i],
               palette="pastel")

    axes[i].set_title(f"Año {año}", fontsize=14)
```

```

axes[i].set_xlabel("TIPO", fontsize=12)

if i == 0:
    axes[i].set_ylabel("N° de Incidencias", fontsize=12)
else:
    axes[i].set_ylabel("")

axes[i].grid(False)
axes[i].tick_params(axis='x', rotation=45)

# Etiquetas encima de las barras
for p in axes[i].patches:
    height = p.get_height()
    if height > 0:
        axes[i].annotate(f'{int(height):,}',
                        (p.get_x() + p.get_width()/2., height),
                        ha='center', va='bottom', fontsize=9)

# Título general
fig.suptitle("Incidencias por TIPO (CLASE 1, Nivel de Tensión B) por Año",
            fontsize=16, y=1.02)

plt.tight_layout()
plt.show()

#GRAFICO NUMERO DE INCIDENCIAS MT, CLASE=1 Y TIPOLOGIA. ¿HAY TIPOS ¡=1XX?
df_clasel_m = df[(df["CLASE"] == "1") & (df["NIV_TENS"] == "M")]
grupo_tipo = df_clasel_m.groupby("TIPO").agg({
    "CODIGO": "count"
}).rename(columns={"CODIGO": "N_INCIDENCIAS"}).reset_index()

import seaborn as sns
import matplotlib.pyplot as plt

# Ordenar por número de incidencias descendente
grupo_tipo_ordenado = grupo_tipo.sort_values(by="N_INCIDENCIAS", ascending=False)

plt.figure(figsize=(12, 6))
ax = sns.barplot(data=grupo_tipo_ordenado, x="TIPO", y="N_INCIDENCIAS",
                palette="pastel")

plt.title("Número de Incidencias por TIPO (CLASE 1, Nivel de Tensión MEDIA)",
        fontsize=14)
plt.xlabel("TIPO")
plt.ylabel("N° de Incidencias")
plt.grid(False)

# Etiquetas encima de las barras
for p in ax.patches:
    height = p.get_height()
    if height > 0:
        ax.annotate(f'{int(height):,}',
                    (p.get_x() + p.get_width() / 2., height),
                    ha='center', va='bottom', fontsize=9)

```

```
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()

#TABLA NUMERO DE INCIDENCIAS MT, CLASE=1 Y TIPOLOGIA. ¿HAY TIPOS ¡=1XX?
# Obtener descripciones únicas
tipos_descripciones = df[["TIPO", "DES_TIPO"]].drop_duplicates()

# Unir descripciones al DataFrame de incidencias
grupo_tipo = grupo_tipo.merge(tipos_descripciones, on="TIPO", how="left")

# Ordenar por número de incidencias y resetear índice
grupo_tipo = grupo_tipo.sort_values(by="N_INCIDENCIAS",
ascending=False).reset_index(drop=True)

# Mostrar en formato tabla con separador visual
print("Número de Incidencias por TIPO (con descripción):")
print(grupo_tipo.to_string(index=False))

#GRAFICO NUMERO DE INCIDENCIAS MT, CLASE=1 Y TIPOLOGIA. ¿HAY TIPOS ¡=1XX?
grupo_tipo_año = df_clase1_m.groupby(["AÑO", "TIPO"]).agg({
    "CODIGO": "count"
}).rename(columns={"CODIGO": "N_INCIDENCIAS"}).reset_index()

import seaborn as sns
import matplotlib.pyplot as plt

# Lista de años ampliada
años = [2022, 2023, 2024, 2025]

# Crear 4 subplots (uno por año)
fig, axes = plt.subplots(1, 4, figsize=(24, 6), sharey=True)

for i, año in enumerate(años):
    # Filtrar datos del año correspondiente
    datos = grupo_tipo_año[grupo_tipo_año["AÑO"] == año]

    # Ordenar por número de incidencias
    datos = datos.sort_values(by="N_INCIDENCIAS", ascending=False)

    # Gráfico de barras por TIPO
    sns.barplot(data=datos, x="TIPO", y="N_INCIDENCIAS", ax=axes[i],
palette="pastel")

    axes[i].set_title(f"Año {año}", fontsize=14)
    axes[i].set_xlabel("TIPO", fontsize=12)
    if i == 0:
        axes[i].set_ylabel("N° de Incidencias", fontsize=12)
    else:
        axes[i].set_ylabel("")

    axes[i].grid(False)
    axes[i].tick_params(axis='x', rotation=45)
```

```
# Anotar valores encima de las barras
for p in axes[i].patches:
    height = p.get_height()
    if height > 0:
        axes[i].annotate(f'{int(height):,}',
                        (p.get_x() + p.get_width()/2., height),
                        ha='center', va='bottom', fontsize=9)

# Título general del gráfico
fig.suptitle("Incidencias por TIPO (CLASE 1, Nivel de Tensión Media) por Año",
            fontsize=16, y=1.02)

plt.tight_layout()
plt.show()

#GRAFICO NUMERO DE INCIDENCIAS BT, CLASE=1 E INS_ORIGEN
# Filtrar por CLASE 1 y NIV_TENS Baja
df_clase1_bm = df[(df["CLASE"] == "1") & (df["NIV_TENS"].isin(["B"]))]

# Agrupar por INS_ORIGEN y contar número de incidencias (según columna CODIGO)
grupo_origen = df_clase1_bm.groupby("INS_ORIGEN").agg({
    "CODIGO": "count"
}).rename(columns={"CODIGO": "N_INCIDENCIAS"}).reset_index()

# Visualización
import seaborn as sns
import matplotlib.pyplot as plt

plt.figure(figsize=(12, 6))
ax = sns.barplot(data=grupo_origen, x="INS_ORIGEN", y="N_INCIDENCIAS",
                palette="pastel")

plt.title("Número de Incidencias por INS_ORIGEN (CLASE 1, Tensión BAJA)",
        fontsize=14)
plt.xlabel("INS_ORIGEN")
plt.ylabel("Nº de Incidencias")
plt.grid(False)

# Etiquetas encima de las barras
for p in ax.patches:
    height = p.get_height()
    if height > 0:
        ax.annotate(f'{int(height):,}',
                    (p.get_x() + p.get_width() / 2., height),
                    ha='center', va='bottom', fontsize=9)

plt.xticks(rotation=45)
plt.tight_layout()
plt.show()

#TABLA NUMERO DE INCIDENCIAS BT, CLASE=1 E INS_ORIGEN
# Filtrar por CLASE 1 y NIV_TENS Baja
df_clase1_bm = df[(df["CLASE"] == "1") & (df["NIV_TENS"] == "B")]
```

```
# Agrupar por INS_ORIGEN y DESC_ORIGEN y contar incidencias
grupo_origen = (
    df_clasel_bm.groupby(["INS_ORIGEN", "DESC_ORIGEN"])
    .agg(N_INCIDENCIAS=("CODIGO", "count"))
    .reset_index()
    .sort_values(by="N_INCIDENCIAS", ascending=False)
)

# Mostrar como tabla
print("Tabla de incidencias por INS_ORIGEN y su descripción (CLASE 1, Tensión BAJA):")
print(grupo_origen.to_string(index=False))

#GRAFICO NUMERO DE INCIDENCIAS MT, CLASE=1 E INS_ORIGEN POR AÑO
import seaborn as sns
import matplotlib.pyplot as plt

# Filtrar por CLASE 1 y NIV_TENS M (Media)
df_clasel_b = df[(df["CLASE"] == "1") & (df["NIV_TENS"] == "M")]

# Agrupar por INS_ORIGEN y AÑO
grupo_origen_año = df_clasel_b.groupby(["AÑO", "INS_ORIGEN"]).agg({
    "CODIGO": "count"
}).rename(columns={"CODIGO": "N_INCIDENCIAS"}).reset_index()

# Años a graficar, incluyendo 2025
años = [2022, 2023, 2024, 2025]

# Crear 4 subplots (uno por año)
fig, axes = plt.subplots(1, 4, figsize=(32, 10), sharey=True) # tamaño ampliado

for i, año in enumerate(años):
    datos = grupo_origen_año[grupo_origen_año["AÑO"] == año]

    sns.barplot(data=datos, x="INS_ORIGEN", y="N_INCIDENCIAS", ax=axes[i],
                palette="pastel")

    axes[i].set_title(f"Año {año}", fontsize=18)
    axes[i].set_xlabel("INS_ORIGEN", fontsize=14)

    if i == 0:
        axes[i].set_ylabel("N° de Incidencias", fontsize=14)
    else:
        axes[i].set_ylabel("")

    axes[i].grid(False)
    axes[i].tick_params(axis='x', rotation=60, labelsz=11)

# Etiquetas sobre las barras
for p in axes[i].patches:
    height = p.get_height()
    if height > 0:
        axes[i].annotate(f'{int(height):,}',
```

```
(p.get_x() + p.get_width() / 2., height),
ha='center', va='bottom', fontsize=11)

# Título principal
fig.suptitle("Incidencias por INS_ORIGEN (CLASE 1, Nivel de Tensión MEDIA) por
Año", fontsize=20, y=1.05)

plt.tight_layout()
plt.show()

#TABLA NUMERO DE INCIDENCIAS MT, CLASE=1 E INS_ORIGEN TOTAL TODOS LOS AÑOS

# Filtrar por CLASE 1 y NIV_TENS Baja o Media
df_clase1_bm = df[(df["CLASE"] == "1") & (df["NIV_TENS"].isin(["M"]))]

# Agrupar por INS_ORIGEN y contar número de incidencias (según columna CODIGO)
grupo_origen = df_clase1_bm.groupby("INS_ORIGEN").agg({
    "CODIGO": "count"
}).rename(columns={"CODIGO": "N_INCIDENCIAS"}).reset_index()

# Visualización
import seaborn as sns
import matplotlib.pyplot as plt

plt.figure(figsize=(12, 6))
ax = sns.barplot(data=grupo_origen, x="INS_ORIGEN", y="N_INCIDENCIAS",
palette="pastel")

plt.title("Número de Incidencias por INS_ORIGEN (CLASE 1, Tensión MEDIA)",
fontsize=14)
plt.xlabel("INS_ORIGEN")
plt.ylabel("Nº de Incidencias")
plt.grid(False)

# Etiquetas encima de las barras
for p in ax.patches:
    height = p.get_height()
    if height > 0:
        ax.annotate(f'{int(height):,}',
                    (p.get_x() + p.get_width() / 2., height),
                    ha='center', va='bottom', fontsize=9)

plt.xticks(rotation=45)
plt.tight_layout()
plt.show()

#TABLA NUMERO DE INCIDENCIAS BT, CLASE=1 POR INS_ORIGEN

# Filtrar por CLASE 1 y NIV_TENS Baja
df_clase1_bm = df[(df["CLASE"] == "1") & (df["NIV_TENS"] == "M")]

# Agrupar por INS_ORIGEN y DESC_ORIGEN y contar incidencias
grupo_origen = (
    df_clase1_bm.groupby(["INS_ORIGEN", "DESC_ORIGEN"])
    .agg(N_INCIDENCIAS=("CODIGO", "count"))
```

```
.reset_index()
.sort_values(by="N_INCIDENCIAS", ascending=False)
)

# Mostrar como tabla
print("Tabla de incidencias por INS_ORIGEN y su descripción (CLASE 1, Tensión BAJA):")
print(grupo_origen.to_string(index=False))
#MAPA DE CALOR DE NUMERO DE INCIDENCIAS POR TIPO E INSTALACION ORIGEN PARA CLASE 1 Y BT

import seaborn as sns
import matplotlib.pyplot as plt
import pandas as pd
# 1. Filtrar por CLASE 1 y NIV_TENS = B
df_filtrado = df[(df["CLASE"] == "1") & (df["NIV_TENS"] == "B")].copy()

# 2. Obtener los TIPO e INS_ORIGEN que aparecen al menos una vez
tipos_usados = df_filtrado["TIPO"].unique()
origenes_usados = df_filtrado["INS_ORIGEN"].unique()
# 3. Agrupar por INS_ORIGEN y TIPO
grupo_heatmap = (
    df_filtrado.groupby(["INS_ORIGEN", "TIPO"])
    .size()
    .reset_index(name="N_INCIDENCIAS")
)

# 4. Crear tabla pivot con todos los TIPO e INS_ORIGEN usados
tabla_heatmap = grupo_heatmap.pivot(index="INS_ORIGEN", columns="TIPO",
values="N_INCIDENCIAS")
# 5. Rellenar con 0 las combinaciones que no existen
tabla_heatmap = tabla_heatmap.reindex(index=origenes_usados,
columns=tipos_usados, fill_value=0)
# 6. Generar el heatmap
plt.figure(figsize=(14, 8))
sns.heatmap(tabla_heatmap, cmap="YlGnBu", annot=True, fmt=".0f", linewidths=.5,
linecolor='gray')
plt.title("Mapa de Calor de Incidencias por TIPO e INS_ORIGEN (CLASE 1, Tensión BAJA)",
fontsize=16)
plt.xlabel("TIPO", fontsize=12)
plt.ylabel("INS_ORIGEN", fontsize=12)
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
#MAPA DE CALOR DE NUMERO DE INCIDENCIAS POR TIPO E INSTALACION ORIGEN PARA CLASE 1 Y BT EN % CON BASE LAS QUE SI SE HAN CUMPLIMENTADO (VALORES DISTINTOS DE CERO)
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
# 1. Filtrar CLASE 1 y Baja Tensión
df_bt_clase1 = df[(df["CLASE"] == "1") & (df["NIV_TENS"] == "B")].copy()
# 2. Crear tabla de contingencia INS_ORIGEN vs TIPO (con todos los valores posibles)
tabla = pd.crosstab(df_bt_clase1["INS_ORIGEN"], df_bt_clase1["TIPO"])
# 3. Asegurar que todos los valores estén como enteros
```

```
tabla = tabla.astype(int)

# 4. Calcular el total global
total_global = tabla.values.sum()

# 5. Calcular porcentaje sobre total global
porcentaje = (tabla / total_global * 100).round(1).astype(str) + "%"

# 6. Crear anotaciones: valor absoluto + porcentaje
anotaciones = tabla.astype(str) + "\n" + porcentaje

# 7. Crear el heatmap
plt.figure(figsize=(18, 10))
sns.heatmap(tabla, annot=anotaciones, fmt='', cmap="YlGnBu", linewidths=0.5,
            cbar_kws={"label": "N° Incidencias"}, annot_kws={"fontsize": 9})

plt.title("Mapa de Calor: N° de Incidencias por TIPO e INS_ORIGEN (CLASE 1, Baja
Tensión)", fontsize=16)
plt.xlabel("TIPO")
plt.ylabel("INS_ORIGEN")
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()

#MAPA DE CALOR DE NUMERO DE INCIDENCIAS POR TIPO E INSTALACION ORIGEN PARA CLASE
1 Y MT
# 1. Filtrar CLASE 1 y Media Tensión
df_mt_clase1 = df[(df["CLASE"] == "1") & (df["NIV_TENS"] == "M")].copy()

# 2. Crear tabla de contingencia INS_ORIGEN vs TIPO
tabla_mt = pd.crosstab(df_mt_clase1["INS_ORIGEN"],
df_mt_clase1["TIPO"]).astype(int)

# 3. Calcular total global
total_mt = tabla_mt.values.sum()

# 4. Calcular porcentaje respecto al total
porcentaje_mt = (tabla_mt / total_mt * 100).round(1).astype(str) + "%"

# 5. Crear anotaciones combinando valor absoluto y porcentaje
anotaciones_mt = tabla_mt.astype(str) + "\n" + porcentaje_mt

# 6. Crear mapa de calor
plt.figure(figsize=(18, 10))
sns.heatmap(tabla_mt, annot=anotaciones_mt, fmt='', cmap="YlGnBu",
linewidths=0.5,
            cbar_kws={"label": "N° Incidencias"}, annot_kws={"fontsize": 9})

plt.title("Mapa de Calor: N° de Incidencias por TIPO e INS_ORIGEN (CLASE 1, Media
Tensión)", fontsize=16)
plt.xlabel("TIPO")
plt.ylabel("INS_ORIGEN")
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```

**# CALCULO DEL PORCENTAJE DE INCIDENCIAS CLASE=1 QUE CONTIENEN "SABT" EN EL CAMPO OBSERVACIONES CADA AÑO**

```
# Aseguramos que sea texto
df["OBSERVACIONES"] = df["OBSERVACIONES"].astype(str)
# Filtramos clase 1
df_clase1 = df[df["CLASE"] == "1"]
# Buscamos SABT (sin A) - ignorando mayúsculas
contiene_sabt = df_clase1["OBSERVACIONES"].str.upper().str.contains("SABT",
na=False)
# Contamos
n_con_sabt = contiene_sabt.sum()
total_clase1 = len(df_clase1)
porcentaje = (n_con_sabt / total_clase1) * 100
# Resultado
print(f"Porcentaje de CLASE 1 con 'SABT' en OBSERVACIONES: {porcentaje:.2f}%")
# Aseguramos que OBSERVACIONES sea texto
df["OBSERVACIONES"] = df["OBSERVACIONES"].astype(str)
# Filtramos solo CLASE 1
df_clase1 = df[df["CLASE"] == "1"]
# Creamos una lista de años que nos interesan
años = [2022, 2023, 2024]
# Recorremos cada año y calculamos el porcentaje
for año in años:
    df_año = df_clase1[df_clase1["AÑO"] == año]
    contiene_sabt = df_año["OBSERVACIONES"].str.upper().str.contains("SABT",
na=False)
    n_con_sabt = contiene_sabt.sum()
    total = len(df_año)
    porcentaje = (n_con_sabt / total) * 100 if total > 0 else 0

    print(f"Año {año}: {porcentaje:.2f}% de CLASE 1 con 'SABT' en OBSERVACIONES")
```

**# CALCULO DEL PORCENTAJE DE INCIDENCIAS CLASE=1 QUE CONTIENEN "SABT" EN EL CAMPO DESCRIPCION.**

```
# Aseguramos que sea texto
df["DESCRIPCION"] = df["DESCRIPCION"].astype(str)

# Filtramos clase 1
df_clase1 = df[df["CLASE"] == "1"]

# Buscamos SABT - ignorando mayúsculas
contiene_sabt = df_clase1["DESCRIPCION"].str.upper().str.contains("SABT",
na=False)
# Contamos
n_con_sabt = contiene_sabt.sum()
total_clase1 = len(df_clase1)
porcentaje = (n_con_sabt / total_clase1) * 100
# Resultado
print(f"Porcentaje de CLASE 1 con 'SABT' en DESCRIPCION: {porcentaje:.2f}%")
print(n_con_sabt)
print(total_clase1)
```

**# CALCULO DEL PORCENTAJE DE INCIDENCIAS CLASE=1 QUE CONTIENEN "SABT" EN EL CAMPO DESCRIPCION.**

```
# Aseguramos que DESCRIPCION sea texto
df["DESCRIPCION"] = df["DESCRIPCION"].astype(str)

# Filtramos solo CLASE 1
df_clase1 = df[df["CLASE"] == "1"]

# Creamos una lista de años que nos interesan
años = [2022, 2023, 2024]

# Recorremos cada año y calculamos el porcentaje
for año in años:
    df_año = df_clase1[df_clase1["AÑO"] == año]
    contiene_sabt = df_año["DESCRIPCION"].str.upper().str.contains("SABT",
na=False)

    n_con_sabt = contiene_sabt.sum()
    total = len(df_año)
    porcentaje = (n_con_sabt / total) * 100 if total > 0 else 0

    print(f"Año {año}: {porcentaje:.2f}% de CLASE 1 con 'SABT' en DESCRIPCION")
```

**# CONTAR NUMERO DE INCIDENCIAS CLASE=1 QUE CONTIENEN "SABT" EN EL CAMPO DESCRIPCION O EN EL CAMPO OBSERVACIONES POR AÑO.**

```
df['OBSERVACIONES'] = df['OBSERVACIONES'].astype(str)
df["DESCRIPCION"] = df["DESCRIPCION"].astype(str)

# filtrar la clase 1
df_clase1 = df[df["CLASE"] == "1"]
# lista de años
años = [2022, 2023, 2024]

#recorremos los años
for año in años:
    df_año = df_clase1[df_clase1["AÑO"] == año]
    contiene_sabt_obs = df_año["OBSERVACIONES"].str.upper().str.contains("SABT",
na=False)
    contiene_sabt_desc = df_año["DESCRIPCION"].str.upper().str.contains("SABT",
na=False)
    contiene_sabt_total = contiene_sabt_obs | contiene_sabt_desc
    n_con_sabt = contiene_sabt_total.sum()

    print(f"Año {año}: {n_con_sabt} incidencias CLASE 1 con 'SABT' en
OBSERVACIONES o DESCRIPCION")
```

**# PARTIENDO DE LOS TRABAJOS QUE TIENEN SABT EN DESCRIPCION O EN OBSERVACIONES, VEAMOS QUE TIPO Y QUE INS\_ORIGEN TIENEN. CREAMOS TABLA**

```
# Filtrar registros CLASE = '1' que contienen 'sabt' en OBSERVACIONES o
DESCRIPCION
filtro_sabt = (
    (df["CLASE"] == "1") &
    (
```

```
df["OBSERVACIONES"].str.contains("sabt", case=False, na=False) |
df["DESCRIPCION"].str.contains("sabt", case=False, na=False)
)
)

df_sabt_clase1 = df[filtro_sabt].copy()

# Agrupar por TIPO e INS_ORIGEN y contar incidencias
tabla_dupla = (
    df_sabt_clase1
    .groupby(["TIPO", "INS_ORIGEN"])
    .agg(N_INCIDENCIAS=("CODIGO", "count"))
    .reset_index()
    .sort_values(by="N_INCIDENCIAS", ascending=False)
)

# Mostrar tabla
print(tabla_dupla)
# CREAR TABLA CON LAS PALABRAS MAS REPETIDAS EN EL CAMPO DESCRIPCION EN TODO EL PERIODO DE ESTUDIO. (INCIDENCIAS CLASE=1)
!pip install nltk
import pandas as pd
from collections import Counter
import re
import nltk
# Descargar stopwords (solo si no lo has hecho antes)
nltk.download('stopwords')
from nltk.corpus import stopwords
# 1. Aseguramos que es texto
df["DESCRIPCION"] = df["DESCRIPCION"].astype(str)
# Filtramos clase 1
df_clase1 = df[df["CLASE"] == "1"]
# 2. Unimos todas las descripciones en un solo texto
texto = " ".join(df_clase1["DESCRIPCION"].dropna()).upper()
# 3. Eliminamos caracteres no alfabéticos
texto = re.sub(r"[^A-ZÁÉÍÓÚÑ ]", "", texto)
# 4. Separamos en palabras
palabras = texto.split()
# 5. Lista de palabras vacías en español
stopwords_es = set(stopwords.words('spanish'))
# 6. Quitamos palabras vacías
palabras_filtradas = [palabra for palabra in palabras if palabra not in
stopwords_es and len(palabra) > 2]
# 7. Contamos frecuencia
conteo = Counter(palabras_filtradas)
# 8. Crear DataFrame con resultados
df_conteo = pd.DataFrame(conteo.items(), columns=["PALABRA", "FRECUENCIA"])
df_conteo = df_conteo.sort_values(by="FRECUENCIA",
ascending=False).reset_index(drop=True)
# Filtrar palabras con frecuencia > 1000
df_top100 = df_conteo[df_conteo["FRECUENCIA"] > 1000]
# Mostrar tabla resultante
print(df_top100)
```

```
# CREAR GRAFICO CON LAS PALABRAS MAS REPETIDAS EN EL CAMPO DESCRIPCION EN AÑO
2024 PARA MT PARA INCIDENCIAS CLASE=1
import pandas as pd
from collections import Counter
import re
from nltk.corpus import stopwords
import nltk
# Asegúrate de tener las stopwords descargadas
nltk.download('stopwords')
stop_words = set(stopwords.words('spanish'))
# Paso 1 y 2: Filtrar por NIV_TENS y AÑO
df_filtrado = df[
    (df['NIV_TENS'].isin(['M'])) &
    (df['AÑO'] == 2024) &
    (df['CLASE'] == "1")
]
# Paso 3: Trabajar solo con columna DESCRIPCION (y eliminar nulos)
descripciones = df_filtrado['DESCRIPCION'].dropna().astype(str)
# Paso 4: Limpiar y tokenizar
def limpiar_y_tokenizar(texto):
    texto = texto.lower()
    texto = re.sub(r'^a-záéíóúñ0-9\s', '', texto) # quitar puntuación
    tokens = texto.split()
    tokens = [t for t in tokens if t not in stop_words and len(t) > 2]
    return tokens

# Tokenizar todas las descripciones
tokens = descripciones.apply(limpiar_y_tokenizar)

# Paso 5: Contar frecuencia
contador = Counter()
for lista in tokens:
    contador.update(lista)

# Paso 6: Filtrar palabras con más de 1000 repeticiones
palabras_mas_1000 = {palabra: cuenta for palabra, cuenta in contador.items() if
cuenta > 500}

# Ordenar
palabras_ordenadas = dict(sorted(palabras_mas_1000.items(), key=lambda item:
item[1], reverse=True))

# Mostrar resultados
print(palabras_ordenadas)

# Ordenar y graficar
palabras_ordenadas = dict(sorted(palabras_mas_1000.items(), key=lambda item:
item[1], reverse=True))

# Crear gráfica
plt.figure(figsize=(10, 6))
plt.bar(palabras_ordenadas.keys(), palabras_ordenadas.values())
plt.xlabel('Palabras')
```

```
plt.ylabel('Frecuencia')
plt.title('Palabras más frecuentes en DESCRIPCION (Año 2024, Tensión M)')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()

# CREAR GRAFICO CON LAS PALABRAS MAS REPETIDAS EN EL CAMPO DESCRIPCION EN AÑO
# 2024 PARA BT PARA INCIDENCIAS CLASE=1
import pandas as pd
from collections import Counter
import re
from nltk.corpus import stopwords
import nltk

# Asegúrate de tener las stopwords descargadas
nltk.download('stopwords')
stop_words = set(stopwords.words('spanish'))

# Paso 1 y 2: Filtrar por NIV_TENS y AÑO
df_filtrado = df[
    (df['NIV_TENS'].isin(['B'])) &
    (df['AÑO'] == 2024) &
    (df['CLASE'] == "1")
]

# Paso 3: Trabajar solo con columna DESCRIPCION (y eliminar nulos)
descripciones = df_filtrado['DESCRIPCION'].dropna().astype(str)

# Paso 4: Limpiar y tokenizar
def limpiar_y_tokenizar(texto):
    texto = texto.lower()
    texto = re.sub(r'^a-záéíóúüñ0-9\s|', '', texto) # quitar puntuación
    tokens = texto.split()
    tokens = [t for t in tokens if t not in stop_words and len(t) > 2]
    return tokens

# Tokenizar todas las descripciones
tokens = descripciones.apply(limpiar_y_tokenizar)

# Paso 5: Contar frecuencia
contador = Counter()
for lista in tokens:
    contador.update(lista)

# Paso 6: Filtrar palabras con más de 1000 repeticiones
palabras_mas_1000 = {palabra: cuenta for palabra, cuenta in contador.items() if
cuenta > 500}

# Ordenar
palabras_ordenadas = dict(sorted(palabras_mas_1000.items(), key=lambda item:
item[1], reverse=True))

# Mostrar resultados
print(palabras_ordenadas)
# Paso 6: Filtrar palabras con más de 1000 repeticiones
```

```
palabras_mas_1000 = {palabra: cuenta for palabra, cuenta in contador.items() if
cuenta > 500}

# Ordenar
palabras_ordenadas = dict(sorted(palabras_mas_1000.items(), key=lambda item:
item[1], reverse=True))

# Ordenar y graficar
palabras_ordenadas = dict(sorted(palabras_mas_1000.items(), key=lambda item:
item[1], reverse=True))

# Crear gráfica
plt.figure(figsize=(10, 6))
plt.bar(palabras_ordenadas.keys(), palabras_ordenadas.values())
plt.xlabel('Palabras')
plt.ylabel('Frecuencia')
plt.title('Palabras más frecuentes en DESCRIPCION (Año 2024, Tensión B)')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
# CREAR GRAFICO CON LAS PALABRAS MAS REPETIDAS EN EL CAMPO DESCRIPCION EN AÑO
2025 PARA BT PARA INCIDENCIAS CLASE=1
import pandas as pd
from collections import Counter
import re
from nltk.corpus import stopwords
import nltk
import matplotlib.pyplot as plt

# Asegúrate de tener las stopwords descargadas
nltk.download('stopwords')
stop_words = set(stopwords.words('spanish'))

# Paso 1 y 2: Filtrar por NIV_TENS, AÑO y CLASE
df_filtrado_2025 = df[
    (df['NIV_TENS'] == 'B') &
    (df['AÑO'] == 2025) &
    (df['CLASE'] == "1")
]

# Paso 3: Trabajar solo con DESCRIPCION
descripciones = df_filtrado_2025['DESCRIPCION'].dropna().astype(str)

# Paso 4: Limpiar y tokenizar
def limpiar_y_tokenizar(texto):
    texto = texto.lower()
    texto = re.sub(r'^a-záéíóúüñ0-9\s|', '', texto) # quitar puntuación
    tokens = texto.split()
    tokens = [t for t in tokens if t not in stop_words and len(t) > 2]
    return tokens

tokens = descripciones.apply(limpiar_y_tokenizar)

# Paso 5: Contar frecuencia
```

```
contador = Counter()
for lista in tokens:
    contador.update(lista)

# Mostrar palabras con más de 100 apariciones (en lugar de 500)
palabras_mas_100 = {palabra: cuenta for palabra, cuenta in contador.items() if
cuenta > 100}

# Ordenar por frecuencia descendente
palabras_ordenadas = dict(sorted(palabras_mas_100.items(), key=lambda item:
item[1], reverse=True))

# Paso 8: Mostrar gráfico
plt.figure(figsize=(10, 6))
plt.bar(palabras_ordenadas.keys(), palabras_ordenadas.values())
plt.xlabel('Palabras')
plt.ylabel('Frecuencia')
plt.title('Palabras más frecuentes en DESCRIPCION (Año 2025, Tensión B)')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```

**# CREAR GRAFICO CON LAS PALABRAS MAS REPETIDAS EN EL CAMPO DESCRIPCION EN AÑO 2025 PARA MT PARA INCIDENCIAS CLASE=1**

```
import pandas as pd
from collections import Counter
import re
from nltk.corpus import stopwords
import nltk
import matplotlib.pyplot as plt

# Asegúrate de tener las stopwords descargadas
nltk.download('stopwords')
stop_words = set(stopwords.words('spanish'))

# Paso 1 y 2: Filtrar por NIV_TENS, AÑO y CLASE
df_filtrado_2025 = df[
    (df['NIV_TENS'] == 'M') &
    (df['AÑO'] == 2025) &
    (df['CLASE'] == "1")
]

# Paso 3: Trabajar solo con DESCRIPCION
descripciones = df_filtrado_2025['DESCRIPCION'].dropna().astype(str)

# Paso 4: Limpiar y tokenizar
def limpiar_y_tokenizar(texto):
    texto = texto.lower()
    texto = re.sub(r'^a-záéíóúüñ0-9\s|', '', texto) # quitar puntuación
    tokens = texto.split()
    tokens = [t for t in tokens if t not in stop_words and len(t) > 2]
    return tokens
```

```
tokens = descripciones.apply(limpiar_y_tokenizar)

# Paso 5: Contar frecuencia
contador = Counter()
for lista in tokens:
    contador.update(lista)

# Mostrar palabras con más de 100 apariciones (en lugar de 500)
palabras_mas_100 = {palabra: cuenta for palabra, cuenta in contador.items() if
cuenta > 100}

# Ordenar por frecuencia descendente
palabras_ordenadas = dict(sorted(palabras_mas_100.items(), key=lambda item:
item[1], reverse=True))

# Paso 8: Mostrar gráfico
plt.figure(figsize=(10, 6))
plt.bar(palabras_ordenadas.keys(), palabras_ordenadas.values())
plt.xlabel('Palabras')
plt.ylabel('Frecuencia')
plt.title('Palabras más frecuentes en DESCRIPCION (Año 2025, Tensión M)')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```

**# ESTUDIAR INCIDENCIAS CLASE=2 QUE SON IMPREVISTAS**  
**GRAFICO DEL TOTAL DE INCIDENCIAS CLASE=2 EN EL PERIODO POR NIVEL DE TENSIO**  
**(AT,MT,BT, SU)**

```
import seaborn as sns
import matplotlib.pyplot as plt

grupo_2 = grupo_clases[grupo_clases["CLASE"] == "2"]

# ----- Gráfico para CLASE 2 -----
plt.figure(figsize=(8,5))
ax1 = sns.barplot(data=grupo_2, x="NIV_TENS", y="N_INCIDENCIAS",
palette="pastel")
plt.title("Número de Incidencias por Nivel de Tensión (CLASE 2)")
plt.xlabel("Nivel de Tensión")
plt.ylabel("Nº de Incidencias")
plt.grid(False)
for p in ax1.patches:
    height = p.get_height()
    if height > 0:
        ax1.annotate(f'{int(height):,}',
                    (p.get_x() + p.get_width()/2., height),
                    ha='center', va='bottom', fontsize=9)
plt.tight_layout()
plt.show()
```

**GRAFICO DE EVOLUCION DE INCIDENCIAS CLASE=2 POR AÑO Y NIVEL DE TENSION**

```
print(grupo_clases.columns)
import seaborn as sns
import matplotlib.pyplot as plt
```

```
# Agrupación por año y nivel de tensión
df_clase2 = df[df["CLASE"] == "2"]

grupo_tens_año = df_clase2.groupby(["AÑO", "NIV_TENS"]).agg({
    "CODIGO": "count"
}).rename(columns={"CODIGO": "N_INCIDENCIAS"}).reset_index()

años = sorted(grupo_tens_año["AÑO"].unique())
fig, axes = plt.subplots(1, len(años), figsize=(6 * len(años), 6), sharey=True)

for i, año in enumerate(años):
    datos = grupo_tens_año[grupo_tens_año["AÑO"] == año]
    sns.barplot(data=datos, x="NIV_TENS", y="N_INCIDENCIAS", ax=axes[i],
        palette="pastel")

    axes[i].set_title(f"Año {año}")
    axes[i].set_xlabel("Nivel de Tensión")
    if i == 0:
        axes[i].set_ylabel("N° de Incidencias")
    else:
        axes[i].set_ylabel("")

    axes[i].grid(False)
    axes[i].tick_params(axis='x', rotation=45)

    for p in axes[i].patches:
        height = p.get_height()
        if height > 0:
            axes[i].annotate(f'{int(height):,}',
                (p.get_x() + p.get_width() / 2., height),
                ha='center', va='bottom', fontsize=9)

fig.suptitle("Incidencias por Nivel de Tensión (CLASE 2) por Año", fontsize=16,
y=1.02)
plt.tight_layout()
plt.show()
```

**GRAFICO DEL TOTAL DE INCIDENCIAS CLASE=2 Y NIVEL = BT POR TIPO**

```
import seaborn as sns
import matplotlib.pyplot as plt

# Filtrar clase 2 y nivel de tensión B
df_clase2_b = df[(df["CLASE"] == "2") & (df["NIV_TENS"] == "B")]

# Agrupar por TIPO y contar incidencias
grupo_tipo = df_clase2_b.groupby("TIPO").agg({
    "CODIGO": "count"
}).rename(columns={"CODIGO": "N_INCIDENCIAS"}).reset_index()

# Ordenar por número de incidencias de mayor a menor
grupo_tipo = grupo_tipo.sort_values(by="N_INCIDENCIAS", ascending=False)

# Gráfico
plt.figure(figsize=(12,6))
```

```
ax = sns.barplot(data=grupo_tipo, x="TIPO", y="N_INCIDENCIAS", palette="pastel")

plt.title("Número de Incidencias por TIPO (CLASE 2, Nivel de Tensión B)",
fontsize=14)
plt.xlabel("TIPO")
plt.ylabel("N° de Incidencias")
plt.grid(False)

# Etiquetas encima de las barras
for p in ax.patches:
    height = p.get_height()
    if height > 0:
        ax.annotate(f'{int(height):,}',
                    (p.get_x() + p.get_width()/2., height),
                    ha='center', va='bottom', fontsize=9)

plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```

**TABLA DEL TOTAL DE INCIDENCIAS CLASE=2 Y BT EN EL PERIODO POR TIPO (2XX, 3XX Y 4XX)**

```
# Filtrar CLASE 2 y NIV_TENS = 'B'
df_clase2_b = df[(df["CLASE"] == "2") & (df["NIV_TENS"] == "B")]

# Agrupar por TIPO y contar incidencias
grupo_tipo = (
    df_clase2_b
    .groupby("TIPO")
    .agg(N_INCIDENCIAS=("CODIGO", "count"))
    .reset_index()
)

# Obtener descripciones únicas
tipos_descripciones = df[["TIPO", "DES_TIPO"]].drop_duplicates()
# Unir descripción
grupo_tipo = grupo_tipo.merge(tipos_descripciones, on="TIPO", how="left")
# Ordenar de mayor a menor
grupo_tipo = grupo_tipo.sort_values(by="N_INCIDENCIAS",
ascending=False).reset_index(drop=True)
# Mostrar como tabla
import pandas as pd
from IPython.display import display
display(grupo_tipo)
```

**GRAFICO DE INCIDENCIAS CLASE=2 Y BT POR AÑOS (2XX, 3XX Y 4XX)**

```
import seaborn as sns
import matplotlib.pyplot as plt

# Limpiar columnas
for col in ["CLASE", "NIV_TENS", "TIPO"]:
    df[col] = df[col].astype(str).str.strip()

# Crear columna AÑO si no existe
if "AÑO" not in df.columns:
    if "FECHA" in df.columns:
        df["FECHA"] = pd.to_datetime(df["FECHA"])
        df["AÑO"] = df["FECHA"].dt.year
    else:
```

```

    raise ValueError("No hay columna AÑO ni FECHA para segmentar por años.")

# Filtrar CLASE 2 y Tensión B
df_clase2_b = df[(df["CLASE"] == "2") & (df["NIV_TENS"] == "B")]

# Lista de años únicos ordenados
años = sorted(df_clase2_b["AÑO"].unique())

# Crear subplots por año
fig, axs = plt.subplots(1, len(años), figsize=(6 * len(años), 6), sharey=True)

# Asegurar que axs es una lista si hay un solo año
if len(años) == 1:
    axs = [axs]

for i, año in enumerate(años):
    df_anual = df_clase2_b[df_clase2_b["AÑO"] == año]

    grupo_tipo = df_anual.groupby("TIPO").agg(
        N_INCIDENCIAS=("CODIGO", "count")
    ).reset_index().sort_values(by="N_INCIDENCIAS", ascending=False)

    ax = axs[i]
    sns.barplot(data=grupo_tipo, x="TIPO", y="N_INCIDENCIAS", palette="pastel",
ax=ax)
    ax.set_title(f"Año {año}", fontsize=13)
    ax.set_xlabel("TIPO")
    ax.set_ylabel("Nº de Incidencias" if i == 0 else "")
    ax.tick_params(axis='x', rotation=45)
    ax.grid(False)

    # Etiquetas encima de las barras
    for p in ax.patches:
        height = p.get_height()
        if height > 0:
            ax.annotate(f'{int(height):,}',
                        (p.get_x() + p.get_width()/2., height),
                        ha='center', va='bottom', fontsize=8)

plt.suptitle("Número de Incidencias por TIPO (CLASE 2, Tensión B) separadas por
año", fontsize=16)
plt.tight_layout(rect=[0, 0.03, 1, 0.92])
plt.show()

```

**GRAFICO DE INCIDENCIAS CLASE=2 Y BT Y TIPO=303 EN EL PERIODO POR INSTALACION  
ORIGEN PARA TODO EL PERIODO**

```

import seaborn as sns
import matplotlib.pyplot as plt
# Limpiar columnas
for col in ["CLASE", "NIV_TENS", "INS_ORIGEN", "TIPO"]:
    df[col] = df[col].astype(str).str.strip()
# Filtrar
df_clase2_B_303 = df[(df["CLASE"] == "2") &
                    (df["NIV_TENS"] == "B") &
                    (df["TIPO"] == "303") ]

```

```
# Agrupar por INS_ORIGEN y contar incidencias totales
grupo = df_clase2_B_303.groupby("INS_ORIGEN").agg(
    N_INCIDENCIAS=("CODIGO", "count")
).reset_index().sort_values(by="N_INCIDENCIAS", ascending=False)

# Gráfico
plt.figure(figsize=(14, 6))
ax = sns.barplot(data=grupo, x="INS_ORIGEN", y="N_INCIDENCIAS", palette="pastel")

plt.title("Número de Incidencias TIPO = 303 por INS_ORIGEN (CLASE 2, Tensión B)", fontsize=16)
plt.xlabel("INS_ORIGEN")
plt.ylabel("Nº de Incidencias")
plt.grid(False)

# Etiquetas encima de las barras
for p in ax.patches:
    height = p.get_height()
    if height > 0:
        ax.annotate(f'{int(height):,}',
                    (p.get_x() + p.get_width()/2., height),
                    ha='center', va='bottom', fontsize=9)

plt.xticks(rotation=45, ha='right')
plt.tight_layout()
plt.show()
```

**TABLA DE INCIDENCIAS CLASE=2 Y BT Y TIPO=303 EN EL PERIODO POR INS\_ORIGEN PARA TODO EL PERIODO**

```
import pandas as pd
from IPython.display import display

# Agrupar por INS_ORIGEN y calcular número de incidencias
grupo_tipo = df_clase2_B_303.groupby(["INS_ORIGEN", "DESC_ORIGEN"]).agg({
    "CODIGO": "count"
}).rename(columns={"CODIGO": "N_INCIDENCIAS"}).reset_index()

# Ordenar de forma descendente
grupo_tipo = grupo_tipo.sort_values(by="N_INCIDENCIAS", ascending=False)

# Mostrar la tabla
display(grupo_tipo)
```

**GRAFICO DE INCIDENCIAS CLASE=2 Y BT Y TIPO=303 POR INS\_ORIGEN Y POR AÑO**

```
import seaborn as sns
import matplotlib.pyplot as plt

# Limpiar columnas necesarias
for col in ["CLASE", "NIV_TENS", "INS_ORIGEN"]:
    df[col] = df[col].astype(str).str.strip()

# Asegurar columna AÑO
if "AÑO" not in df.columns:
    if "FECHA" in df.columns:
        df["FECHA"] = pd.to_datetime(df["FECHA"])
```

```

df["AÑO"] = df["FECHA"].dt.year
else:
    raise ValueError("No hay columna AÑO ni FECHA")

# Lista de años únicos ordenados
años = sorted(df_clase2_B_303["AÑO"].unique())

# Crear subplots (uno por año)
fig, axs = plt.subplots(1, len(años), figsize=(6 * len(años), 6), sharey=True)

# Asegurar que axs es lista
if len(años) == 1:
    axs = [axs]

for i, año in enumerate(años):
    df_anual = df_clase2_B_303[df_clase2_B_303["AÑO"] == año]

    grupo = df_anual.groupby("INS_ORIGEN").agg(
        N_INCIDENCIAS=("CODIGO", "count")
    ).reset_index().sort_values(by="N_INCIDENCIAS", ascending=False)

    ax = axs[i]
    sns.barplot(data=grupo, x="INS_ORIGEN", y="N_INCIDENCIAS", palette="pastel",
ax=ax)
    ax.set_title(f"Año {año}", fontsize=13)
    ax.set_xlabel("INS_ORIGEN")
    ax.set_ylabel("Nº de Incidencias" if i == 0 else "")
    ax.tick_params(axis='x', rotation=45)
    ax.grid(False)

    # Etiquetas encima de las barras
    for p in ax.patches:
        height = p.get_height()
        if height > 0:
            ax.annotate(f'{int(height):,}',
                (p.get_x() + p.get_width()/2., height),
                ha='center', va='bottom', fontsize=8)

plt.suptitle("Número de Incidencias TIPO = 303 por INS_ORIGEN (CLASE 2, Tensión
B) separadas por año", fontsize=16)
plt.tight_layout(rect=[0, 0.03, 1, 0.92])
plt.show()

```

**GRAFICO DE INCIDENCIAS CLASE=2 Y BT Y TIPO=303 EN EL PERIODO POR MATERIAL AVERIADO**

```

import seaborn as sns
import matplotlib.pyplot as plt

# Limpiar columnas necesarias
for col in ["CLASE", "NIV_TENS", "MAT_AVERIADO"]:
    df[col] = df[col].astype(str).str.strip()

# Agrupar por MAT_AVERIADO y contar
grupo = df_clase2_B_303.groupby("MAT_AVERIADO").agg(
    N_INCIDENCIAS=("CODIGO", "count")

```

```

).reset_index().sort_values(by="N_INCIDENCIAS", ascending=False)

# Gráfico
plt.figure(figsize=(14, 6))
ax = sns.barplot(data=grupo, x="MAT_AVERIADO", y="N_INCIDENCIAS",
palette="pastel")

plt.title("Número de Incidencias TIPO = 303 por MAT_AVERIADO (CLASE 2, Tensión
B)", fontsize=16)
plt.xlabel("MAT_AVERIADO")
plt.ylabel("Nº de Incidencias")
plt.grid(False)

# Etiquetas encima de las barras
for p in ax.patches:
    height = p.get_height()
    if height > 0:
        ax.annotate(f'{int(height):,}',
                    (p.get_x() + p.get_width()/2., height),
                    ha='center', va='bottom', fontsize=9)

plt.xticks(rotation=45, ha='right')
plt.tight_layout()
plt.show()

```

**TABLA DE INCIDENCIAS CLASE=2 Y BT Y TIPO=303 EN EL PERIODO POR MATERIAL AVERIADO CON SU CODIGO Y DESCRIPCION**

```

import pandas as pd
from IPython.display import display
# Agrupar por INS_ORIGEN y calcular número de incidencias
grupo_tipo = df_clase2_B_303.groupby(["MAT_AVERIADO", "DESC_MAT_AVER"]).agg({
    "CODIGO": "count"
}).rename(columns={"CODIGO": "N_INCIDENCIAS").reset_index()
# Ordenar de forma descendente
grupo_tipo = grupo_tipo.sort_values(by="N_INCIDENCIAS", ascending=False)
# Mostrar la tabla
display(grupo_tipo)

```

**GRAFICO DE EVOLUCION DE NUMERO DE INCIDENCIAS CLASE=2 Y BT Y TIPO=303 POR MATERIAL AVERIADO Y POR AÑO PARA PODER COMPARAR**

```

import seaborn as sns
import matplotlib.pyplot as plt
# Limpiar columnas necesarias
for col in ["CLASE", "NIV_TENS", "MAT_AVERIADO"]:
    df[col] = df[col].astype(str).str.strip()
# Asegurar columna AÑO
if "AÑO" not in df.columns:
    if "FECHA" in df.columns:
        df["FECHA"] = pd.to_datetime(df["FECHA"])
        df["AÑO"] = df["FECHA"].dt.year
    else:
        raise ValueError("No hay columna AÑO ni FECHA")
# Lista de años únicos ordenados
años = sorted(df_clase2_B_303["AÑO"].unique())

```

```
# Crear subplots (uno por año)
fig, axs = plt.subplots(1, len(años), figsize=(6 * len(años), 6), sharey=True)

# Asegurar que axs es lista
if len(años) == 1:
    axs = [axs]

for i, año in enumerate(años):
    df_anual = df_clase2_B_303[df_clase2_B_303["AÑO"] == año]

    grupo = df_anual.groupby("MAT_AVERIADO").agg(
        N_INCIDENCIAS=("CODIGO", "count")
    ).reset_index().sort_values(by="N_INCIDENCIAS", ascending=False)

    ax = axs[i]
    sns.barplot(data=grupo, x="MAT_AVERIADO", y="N_INCIDENCIAS",
palette="pastel", ax=ax)
    ax.set_title(f"Año {año}", fontsize=13)
    ax.set_xlabel("MAT_AVERIADO")
    ax.set_ylabel("Nº de Incidencias" if i == 0 else "")
    ax.tick_params(axis='x', rotation=45)
    ax.grid(False)

    # Etiquetas encima de las barras
    for p in ax.patches:
        height = p.get_height()
        if height > 0:
            ax.annotate(f'{int(height):,}',
                (p.get_x() + p.get_width()/2., height),
                ha='center', va='bottom', fontsize=8)

plt.suptitle("Número de Incidencias TIPO = 303 por MAT_AVERIADO (CLASE 2, Tensión
B) separadas por año", fontsize=16)
plt.tight_layout(rect=[0, 0.03, 1, 0.92])
plt.show()
```

**GRAFICO DE NUMERO DE INCIDENCIAS CLASE=2 Y BT Y TIPO=301 POR MATERIAL AVERIADO**

### CLASE 2, TIPO = 301 Y NIVEL DE TENSION = BAJA POR INS ORIGEN :

```
import seaborn as sns
import matplotlib.pyplot as plt

# Limpiar columnas
for col in ["CLASE", "NIV_TENS", "INS_ORIGEN", "TIPO"]:
    df[col] = df[col].astype(str).str.strip()

# Filtrar
df_clase2_B_301 = df[(df["CLASE"] == "2") &
    (df["NIV_TENS"] == "B") &
    (df["TIPO"] == "301") ]

# Agrupar por INS_ORIGEN y contar incidencias totales
grupo = df_clase2_B_301.groupby("INS_ORIGEN").agg(
    N_INCIDENCIAS=("CODIGO", "count")
).reset_index().sort_values(by="N_INCIDENCIAS", ascending=False)
```

```
# Gráfico
plt.figure(figsize=(14, 6))
ax = sns.barplot(data=grupo, x="INS_ORIGEN", y="N_INCIDENCIAS", palette="pastel")

plt.title("Número de Incidencias TIPO = 301 por INS_ORIGEN (CLASE 2, Tensión B)", fontsize=16)
plt.xlabel("INS_ORIGEN")
plt.ylabel("Nº de Incidencias")
plt.grid(False)

# Etiquetas encima de las barras
for p in ax.patches:
    height = p.get_height()
    if height > 0:
        ax.annotate(f'{int(height):,}',
                    (p.get_x() + p.get_width()/2., height),
                    ha='center', va='bottom', fontsize=9)

plt.xticks(rotation=45, ha='right')
plt.tight_layout()
plt.show()
```

**TABLA DE NUMERO DE INCIDENCIAS CLASE=2 Y BT Y TIPO=301 POR INS\_ORIGEN**

```
import pandas as pd
from IPython.display import display

# Agrupar por INS_ORIGEN y calcular número de incidencias
grupo_tipo = df_clase2_B_301.groupby(["INS_ORIGEN", "DESC_ORIGEN"]).agg({
    "CODIGO": "count"
}).rename(columns={"CODIGO": "N_INCIDENCIAS"}).reset_index()

# Ordenar de forma descendente
grupo_tipo = grupo_tipo.sort_values(by="N_INCIDENCIAS", ascending=False)

# Mostrar la tabla
display(grupo_tipo)
```

**GRAFICO DE EVOLUCION DE NUMERO DE INCIDENCIAS CLASE=2 Y BT Y TIPO=301 POR INS\_ORIGEN Y POR AÑO PARA PODER COMPARAR**

```
import seaborn as sns
import matplotlib.pyplot as plt

# Limpiar columnas necesarias
for col in ["CLASE", "NIV_TENS", "INS_ORIGEN"]:
    df[col] = df[col].astype(str).str.strip()

# Asegurar columna AÑO
if "AÑO" not in df.columns:
    if "FECHA" in df.columns:
        df["FECHA"] = pd.to_datetime(df["FECHA"])
        df["AÑO"] = df["FECHA"].dt.year
    else:
        raise ValueError("No hay columna AÑO ni FECHA")
```

```
# Lista de años únicos ordenados
años = sorted(df_clase2_B_301["AÑO"].unique())

# Crear subplots (uno por año)
fig, axs = plt.subplots(1, len(años), figsize=(6 * len(años), 6), sharey=True)

# Asegurar que axs es lista
if len(años) == 1:
    axs = [axs]

for i, año in enumerate(años):
    df_anual = df_clase2_B_301[df_clase2_B_301["AÑO"] == año]

    grupo = df_anual.groupby("INS_ORIGEN").agg(
        N_INCIDENCIAS=("CODIGO", "count")
    ).reset_index().sort_values(by="N_INCIDENCIAS", ascending=False)

    ax = axs[i]
    sns.barplot(data=grupo, x="INS_ORIGEN", y="N_INCIDENCIAS", palette="pastel",
ax=ax)
    ax.set_title(f"Año {año}", fontsize=13)
    ax.set_xlabel("INS_ORIGEN")
    ax.set_ylabel("N° de Incidencias" if i == 0 else "")
    ax.tick_params(axis='x', rotation=45)
    ax.grid(False)

    # Etiquetas encima de las barras
    for p in ax.patches:
        height = p.get_height()
        if height > 0:
            ax.annotate(f'{int(height):,}',
                        (p.get_x() + p.get_width()/2., height),
                        ha='center', va='bottom', fontsize=8)

plt.suptitle("Número de Incidencias TIPO = 301 por INS_ORIGEN (CLASE 2, Tensión
B) separadas por año", fontsize=16)
plt.tight_layout(rect=[0, 0.03, 1, 0.92])
plt.show()

GRAFICO NUMERO DE INCIDENCIAS CLASE=2 Y BT Y TIPO=301 POR MAT_AVERIADO
### CLASE 2, TIPO = 301 Y MAT AVERIADO :
import seaborn as sns
import matplotlib.pyplot as plt

# Limpiar columnas necesarias
for col in ["CLASE", "NIV_TENS", "MAT_AVERIADO"]:
    df[col] = df[col].astype(str).str.strip()

# Agrupar por MAT_AVERIADO y contar
grupo = df_clase2_B_301.groupby("MAT_AVERIADO").agg(
    N_INCIDENCIAS=("CODIGO", "count")
).reset_index().sort_values(by="N_INCIDENCIAS", ascending=False)

# Gráfico
plt.figure(figsize=(14, 6))
```

```
ax = sns.barplot(data=grupo, x="MAT_AVERIADO", y="N_INCIDENCIAS",
palette="pastel")
plt.title("Número de Incidencias TIPO = 301 por MAT_AVERIADO (CLASE 2, Tensión
B)", fontsize=16)
plt.xlabel("MAT_AVERIADO")
plt.ylabel("N° de Incidencias")
plt.grid(False)

# Etiquetas encima de las barras
for p in ax.patches:
    height = p.get_height()
    if height > 0:
        ax.annotate(f'{int(height):,}',
                    (p.get_x() + p.get_width()/2., height),
                    ha='center', va='bottom', fontsize=9)
plt.xticks(rotation=45, ha='right')
plt.tight_layout()
plt.show()
```

**TABLA NUMERO DE INCIDENCIAS CLASE=2 Y BT Y TIPO=301 MAT\_AVERIADO CON SU DESCRIPCION**

```
import pandas as pd
from IPython.display import display
# Agrupar por INS_ORIGEN y calcular número de incidencias
grupo_tipo = df_clase2_B_301.groupby(["MAT_AVERIADO", "DESC_MAT_AVER"]).agg({
    "CODIGO": "count"
}).rename(columns={"CODIGO": "N_INCIDENCIAS"}).reset_index()
# Ordenar de forma descendente
grupo_tipo = grupo_tipo.sort_values(by="N_INCIDENCIAS", ascending=False)
# Mostrar la tabla
display(grupo_tipo)
```

**GRAFICO DE EVOLUCION DE NUMERO DE INCIDENCIAS CLASE=2 Y BT Y TIPO=301 POR MAT\_AVERIADO Y POR AÑO PARA PODER COMPARAR**

```
import seaborn as sns
import matplotlib.pyplot as plt
# Limpiar columnas necesarias
for col in ["CLASE", "NIV_TENS", "MAT_AVERIADO"]:
    df[col] = df[col].astype(str).str.strip()
# Asegurar columna AÑO
if "AÑO" not in df.columns:
    if "FECHA" in df.columns:
        df["FECHA"] = pd.to_datetime(df["FECHA"])
        df["AÑO"] = df["FECHA"].dt.year
    else:
        raise ValueError("No hay columna AÑO ni FECHA")
# Lista de años únicos ordenados
años = sorted(df_clase2_B_301["AÑO"].unique())
# Crear subplots (uno por año)
fig, axs = plt.subplots(1, len(años), figsize=(6 * len(años), 6), sharey=True)
# Asegurar que axs es lista
if len(años) == 1:
    axs = [axs]
for i, año in enumerate(años):
    df_anual = df_clase2_B_301[df_clase2_B_301["AÑO"] == año]
    grupo = df_anual.groupby("MAT_AVERIADO").agg(
        N_INCIDENCIAS=("CODIGO", "count")
```

```

).reset_index().sort_values(by="N_INCIDENCIAS", ascending=False)
ax = axs[i]
sns.barplot(data=grupo, x="MAT_AVERIADO", y="N_INCIDENCIAS",
palette="pastel", ax=ax)
ax.set_title(f"Año {año}", fontsize=13)
ax.set_xlabel("MAT_AVERIADO")
ax.set_ylabel("N° de Incidencias" if i == 0 else "")
ax.tick_params(axis='x', rotation=45)
ax.grid(False)
# Etiquetas encima de las barras
for p in ax.patches:
    height = p.get_height()
    if height > 0:
        ax.annotate(f'{int(height):,}',
                    (p.get_x() + p.get_width()/2., height),
                    ha='center', va='bottom', fontsize=8)
plt.suptitle("Número de Incidencias TIPO = 301 por MAT_AVERIADO (CLASE 2, Tensión
B) separadas por año", fontsize=16)
plt.tight_layout(rect=[0, 0.03, 1, 0.92])
plt.show()

```

```

GRAFICO DE NUMERO DE INCIDENCIAS CLASE=2 Y BT Y TIPO=416 POR INS_ORIGEN
### CLASE 2, TIPO = 416 Y NIVEL DE TENSION = BAJA POR INS ORIGEN :
import seaborn as sns
import matplotlib.pyplot as plt
# Limpiar columnas
for col in ["CLASE", "NIV_TENS", "INS_ORIGEN", "TIPO"]:
    df[col] = df[col].astype(str).str.strip()
# Filtrar
df_clase2_B_416 = df[(df["CLASE"] == "2") &
                    (df["NIV_TENS"] == "B") &
                    (df["TIPO"] == "416")]
# Agrupar por INS_ORIGEN y contar incidencias totales
grupo = df_clase2_B_416.groupby("INS_ORIGEN").agg(
    N_INCIDENCIAS=("CODIGO", "count")
).reset_index().sort_values(by="N_INCIDENCIAS", ascending=False)
# Gráfico
plt.figure(figsize=(14, 6))
ax = sns.barplot(data=grupo, x="INS_ORIGEN", y="N_INCIDENCIAS", palette="pastel")
Pl. Title("Número de Incidencias TIPO = 416 por INS_ORIGEN (CLASE 2, Tensión
B)", fontsize=16)
plt.xlabel("INS_ORIGEN")
plt.ylabel("N° de Incidencias")
plt.grid(False)
# Etiquetas encima de las barras
for p in ax.patches:
    height = p.get_height()
    if height > 0:
        ax.annotate(f'{int(height):,}',
                    (p.get_x() + p.get_width()/2., height),
                    ha='center', va='bottom', fontsize=9)
plt.xticks(rotation=45, ha='right')
plt.tight_layout()
plt.show()

```

**TABLA NUMERO DE INCIDENCIAS CLASE=2 Y BT Y TIPO=416 POR INS\_ORIGEN**

```
import pandas as pd
from IPython.display import display
# Agrupar por INS_ORIGEN y calcular número de incidencias
grupo_tipo = df_clase2_B_416.groupby(["INS_ORIGEN", "DESC_ORIGEN"]).agg({
    "CODIGO": "count"
}).rename(columns={"CODIGO": "N_INCIDENCIAS"}).reset_index()
# Ordenar de forma descendente
grupo_tipo = grupo_tipo.sort_values(by="N_INCIDENCIAS", ascending=False)
# Mostrar la tabla
display(grupo_tipo)
```

**GRAFICO DE EVOLUCION DE NUMERO DE INCIDENCIAS CLASE=2 Y BT Y TIPO=416 POR INS\_ORIGEN Y POR AÑO PARA PODER COMPARAR**

```
import seaborn as sns
import matplotlib.pyplot as plt
# Limpiar columnas necesarias
for col in ["CLASE", "NIV_TENS", "INS_ORIGEN"]:
    df[col] = df[col].astype(str).str.strip()
# Asegurar columna AÑO
if "AÑO" not in df.columns:
    if "FECHA" in df.columns:
        df["FECHA"] = pd.to_datetime(df["FECHA"])
        df["AÑO"] = df["FECHA"].dt.year
    else:
        raise ValueError("No hay columna AÑO ni FECHA")
# Lista de años únicos ordenados
años = sorted(df_clase2_B_416["AÑO"].unique())
# Crear subplots (uno por año)
fig, axs = plt.subplots(1, len(años), figsize=(6 * len(años), 6), sharey=True)
# Asegurar que axs es lista
if len(años) == 1:
    axs = [axs]
for i, año in enumerate(años):
    df_anual = df_clase2_B_416[df_clase2_B_416["AÑO"] == año]
    grupo = df_anual.groupby("INS_ORIGEN").agg(
        N_INCIDENCIAS=("CODIGO", "count")
    ).reset_index().sort_values(by="N_INCIDENCIAS", ascending=False)
    ax = axs[i]
    sns.barplot(data=grupo, x="INS_ORIGEN", y="N_INCIDENCIAS", palette="pastel",
ax=ax)
    ax.set_title(f"Año {año}", fontsize=13)
    ax.set_xlabel("INS_ORIGEN")
    ax.set_ylabel("N° de Incidencias" if i == 0 else "")
    ax.tick_params(axis='x', rotation=45)
    ax.grid(False)
    # Etiquetas encima de las barras
    for p in ax.patches:
        height = p.get_height()
        if height > 0:
            ax.annotate(f'{int(height):,}',
                (p.get_x() + p.get_width()/2., height),
                ha='center', va='bottom', fontsize=8)
plt.suptitle("Número de Incidencias TIPO = 416 por INS_ORIGEN (CLASE 2, Tensión B) separadas por año", fontsize=16)
```

```
plt.tight_layout(rect=[0, 0.03, 1, 0.92])  
plt.show()
```

**GRAFICO DE NUMERO DE INCIDENCIAS CLASE=2 Y BT Y TIPO=416 POR MAT\_AVERIADO**

```
import seaborn as sns  
import matplotlib.pyplot as plt  
# Limpiar columnas necesarias  
for col in ["CLASE", "NIV_TENS", "MAT_AVERIADO"]:  
    df[col] = df[col].astype(str).str.strip()  
# Agrupar por MAT_AVERIADO y contar  
grupo = df_clase2_B_416.groupby("MAT_AVERIADO").agg(  
    N_INCIDENCIAS=("CODIGO", "count")  
) .reset_index().sort_values(by="N_INCIDENCIAS", ascending=False)  
# Gráfico  
plt.figure(figsize=(14, 6))  
ax = sns.barplot(data=grupo, x="MAT_AVERIADO", y="N_INCIDENCIAS",  
    palette="pastel")  
plt.title("Número de Incidencias TIPO = 416 por MAT_AVERIADO (CLASE 2, Tensión  
B)", fontsize=16)  
plt.xlabel("MAT_AVERIADO")  
plt.ylabel("N° de Incidencias")  
plt.grid(False)  
# Etiquetas encima de las barras  
for p in ax.patches:  
    height = p.get_height()  
    if height > 0:  
        ax.annotate(f'{int(height):,}',  
            (p.get_x() + p.get_width()/2., height),  
            ha='center', va='bottom', fontsize=9)  
plt.xticks(rotation=45, ha='right')  
plt.tight_layout()  
plt.show()
```

**TABLA DE NUMERO DE INCIDENCIAS CLASE=2 Y BT Y TIPO=301 POR MAT\_AVERIADO**

```
import pandas as pd  
from IPython.display import display  
  
# Agrupar por INS_ORIGEN y calcular número de incidencias  
grupo_tipo = df_clase2_B_416.groupby(["MAT_AVERIADO", "DESC_MAT_AVER"]).agg(  
    "CODIGO": "count"  
) .rename(columns={"CODIGO": "N_INCIDENCIAS"}).reset_index()  
  
# Ordenar de forma descendente  
grupo_tipo = grupo_tipo.sort_values(by="N_INCIDENCIAS", ascending=False)  
  
# Mostrar la tabla  
display(grupo_tipo)
```

**GRAFICO DE EVOLUCION DE NUMERO DE INCIDENCIAS CLASE=2 Y BT Y TIPO=416 POR  
MAT\_AVRIADO Y POR AÑO PARA PODER COMPARAR**

```
import seaborn as sns  
import matplotlib.pyplot as plt
```

```
# Limpiar columnas necesarias
for col in ["CLASE", "NIV_TENS", "MAT_AVERIADO"]:
    df[col] = df[col].astype(str).str.strip()
# Asegurar columna AÑO
if "AÑO" not in df.columns:
    if "FECHA" in df.columns:
        df["FECHA"] = pd.to_datetime(df["FECHA"])
        df["AÑO"] = df["FECHA"].dt.year
    else:
        raise ValueError("No hay columna AÑO ni FECHA")
# Lista de años únicos ordenados
años = sorted(df_clase2_B_416["AÑO"].unique())
# Crear subplots (uno por año)
fig, axs = plt.subplots(1, len(años), figsize=(6 * len(años), 6), sharey=True)
# Asegurar que axs es lista
if len(años) == 1:
    axs = [axs]
for i, año in enumerate(años):
    df_anual = df_clase2_B_416[df_clase2_B_416["AÑO"] == año]
    grupo = df_anual.groupby("MAT_AVERIADO").agg(
        N_INCIDENCIAS=("CODIGO", "count")
    ).reset_index().sort_values(by="N_INCIDENCIAS", ascending=False)
    ax = axs[i]
    sns.barplot(data=grupo, x="MAT_AVERIADO", y="N_INCIDENCIAS",
palette="pastel", ax=ax)
    ax.set_title(f"Año {año}", fontsize=13)
    ax.set_xlabel("MAT_AVERIADO")
    ax.set_ylabel("Nº de Incidencias" if i == 0 else "")
    ax.tick_params(axis='x', rotation=45)
    ax.grid(False)

    # Etiquetas encima de las barras
    for p in ax.patches:
        height = p.get_height()
        if height > 0:
            ax.annotate(f'{int(height):,}',
                (p.get_x() + p.get_width()/2., height),
                ha='center', va='bottom', fontsize=8)

plt.suptitle("Número de Incidencias TIPO = 416 por MAT_AVERIADO (CLASE 2, Tensión
B) separadas por año", fontsize=16)
plt.tight_layout(rect=[0, 0.03, 1, 0.92])
plt.show()
```

#### **GRAFICO DE NUMERO DE INCIDENCIAS CLASE=2 Y BT Y TIPO=202 POR INS\_ORIGEN**

```
import seaborn as sns
import matplotlib.pyplot as plt
# Limpiar columnas
for col in ["CLASE", "NIV_TENS", "INS_ORIGEN", "TIPO"]:
    df[col] = df[col].astype(str).str.strip()
# Filtrar
df_clase2_B_202 = df[(df["CLASE"] == "2") &
    (df["NIV_TENS"] == "B") &
    (df["TIPO"] == "202")] ]
```

```
# Agrupar por INS_ORIGEN y contar incidencias totales
grupo = df_clase2_B_202.groupby("INS_ORIGEN").agg(
    N_INCIDENCIAS=("CODIGO", "count")
).reset_index().sort_values(by="N_INCIDENCIAS", ascending=False)
# Gráfico
plt.figure(figsize=(14, 6))
ax = sns.barplot(data=grupo, x="INS_ORIGEN", y="N_INCIDENCIAS", palette="pastel")
plt.title("Número de Incidencias TIPO = 202 por INS_ORIGEN (CLASE 2, Tensión B)", fontsize=16)
plt.xlabel("INS_ORIGEN")
plt.ylabel("Nº de Incidencias")
plt.grid(False)
# Etiquetas encima de las barras
for p in ax.patches:
    height = p.get_height()
    if height > 0:
        ax.annotate(f'{int(height):,}',
                    (p.get_x() + p.get_width()/2., height),
                    ha='center', va='bottom', fontsize=9)
plt.xticks(rotation=45, ha='right')
plt.tight_layout()
plt.show()
```

**TABLA DEL NUMERO DE INCIDENCIAS CLASE=2 Y BT Y TIPO=202 POR INS\_ORIGEN**

```
import pandas as pd
from IPython.display import display

# Agrupar por INS_ORIGEN y calcular número de incidencias
grupo_tipo = df_clase2_B_202.groupby(["INS_ORIGEN", "DESC_ORIGEN"]).agg({
    "CODIGO": "count"
}).rename(columns={"CODIGO": "N_INCIDENCIAS"}).reset_index()

# Ordenar de forma descendente
grupo_tipo = grupo_tipo.sort_values(by="N_INCIDENCIAS", ascending=False)

# Mostrar la tabla
display(grupo_tipo)
```

**GRAFICO DE EVOLUCION DE NUMERO DE INCIDENCIAS CLASE=2 Y BT Y TIPO=202 POR INS\_ORIGEN Y POR AÑO PARA PODER COMPARAR**

```
import seaborn as sns
import matplotlib.pyplot as plt

# Limpiar columnas necesarias
for col in ["CLASE", "NIV_TENS", "INS_ORIGEN"]:
    df[col] = df[col].astype(str).str.strip()

# Asegurar columna AÑO
if "AÑO" not in df.columns:
    if "FECHA" in df.columns:
        df["FECHA"] = pd.to_datetime(df["FECHA"])
        df["AÑO"] = df["FECHA"].dt.year
    else:
        raise ValueError("No hay columna AÑO ni FECHA")
```

```
# Lista de años únicos ordenados
años = sorted(df_clase2_B_202["AÑO"].unique())

# Crear subplots (uno por año)
fig, axs = plt.subplots(1, len(años), figsize=(6 * len(años), 6), sharey=True)

# Asegurar que axs es lista
if len(años) == 1:
    axs = [axs]

for i, año in enumerate(años):
    df_anual = df_clase2_B_202[df_clase2_B_202["AÑO"] == año]

    grupo = df_anual.groupby("INS_ORIGEN").agg(
        N_INCIDENCIAS=("CODIGO", "count")
    ).reset_index().sort_values(by="N_INCIDENCIAS", ascending=False)

    ax = axs[i]
    sns.barplot(data=grupo, x="INS_ORIGEN", y="N_INCIDENCIAS", palette="pastel",
ax=ax)
    ax.set_title(f"Año {año}", fontsize=13)
    ax.set_xlabel("INS_ORIGEN")
    ax.set_ylabel("N° de Incidencias" if i == 0 else "")
    ax.tick_params(axis='x', rotation=45)
    ax.grid(False)

    # Etiquetas encima de las barras
    for p in ax.patches:
        height = p.get_height()
        if height > 0:
            ax.annotate(f'{int(height):,}',
                (p.get_x() + p.get_width()/2., height),
                ha='center', va='bottom', fontsize=8)

plt.suptitle("Número de Incidencias TIPO = 202 por INS_ORIGEN (CLASE 2, Tensión
B) separadas por año", fontsize=16)
plt.tight_layout(rect=[0, 0.03, 1, 0.92])
plt.show()
```

#### **GRAFICO NUMERO DE INCIDENCIAS CLASE=2 Y BT Y TIPO=202 POR MAT\_AVERIADO**

```
import seaborn as sns
import matplotlib.pyplot as plt

# Limpiar columnas necesarias
for col in ["CLASE", "NIV_TENS", "MAT_AVERIADO"]:
    df[col] = df[col].astype(str).str.strip()

# Agrupar por MAT_AVERIADO y contar
grupo = df_clase2_B_202.groupby("MAT_AVERIADO").agg(
    N_INCIDENCIAS=("CODIGO", "count")
).reset_index().sort_values(by="N_INCIDENCIAS", ascending=False)
```

```
# Gráfico
plt.figure(figsize=(14, 6))
ax = sns.barplot(data=grupo, x="MAT_AVERIADO", y="N_INCIDENCIAS",
palette="pastel")

plt.title("Número de Incidencias TIPO = 202 por MAT_AVERIADO (CLASE 2, Tensión
B)", fontsize=16)
plt.xlabel("MAT_AVERIADO")
plt.ylabel("N° de Incidencias")
plt.grid(False)

# Etiquetas encima de las barras
for p in ax.patches:
    height = p.get_height()
    if height > 0:
        ax.annotate(f'{int(height):,}',
                    (p.get_x() + p.get_width()/2., height),
                    ha='center', va='bottom', fontsize=9)

plt.xticks(rotation=45, ha='right')
plt.tight_layout()
plt.show()
```

**TABLA DE NUMERO DE INCIDENCIAS CLASE=2 Y BT Y TIPO=202 POR MAT\_AVERIADO**

```
import pandas as pd
from IPython.display import display
# Agrupar por INS_ORIGEN y calcular número de incidencias
grupo_tipo = df_clase2_B_202.groupby(["MAT_AVERIADO", "DESC_MAT_AVER"]).agg({
    "CODIGO": "count"
}).rename(columns={"CODIGO": "N_INCIDENCIAS"}).reset_index()

# Ordenar de forma descendente
grupo_tipo = grupo_tipo.sort_values(by="N_INCIDENCIAS", ascending=False)
# Mostrar la tabla
display(grupo_tipo)
```

**GRAFICO DE EVOLUCION DE NUMERO DE INCIDENCIAS CLASE=2 Y BT Y TIPO=202 POR MAT\_AVERIADO Y POR AÑO PARA PODER COMPARAR**

```
import seaborn as sns
import matplotlib.pyplot as plt
# Limpiar columnas necesarias
for col in ["CLASE", "NIV_TENS", "MAT_AVERIADO"]:
    df[col] = df[col].astype(str).str.strip()
# Asegurar columna AÑO
if "AÑO" not in df.columns:
    if "FECHA" in df.columns:
        df["FECHA"] = pd.to_datetime(df["FECHA"])
        df["AÑO"] = df["FECHA"].dt.year
    else:
        raise ValueError("No hay columna AÑO ni FECHA")
# Lista de años únicos ordenados
años = sorted(df_clase2_B_202["AÑO"].unique())
# Crear subplots (uno por año)
fig, axs = plt.subplots(1, len(años), figsize=(6 * len(años), 6), sharey=True)
```

```
# Asegurar que axs es lista
if len(años) == 1:
    axs = [axs]

for i, año in enumerate(años):
    df_anual = df_clase2_B_202[df_clase2_B_202["AÑO"] == año]

    grupo = df_anual.groupby("MAT_AVERIADO").agg(
        N_INCIDENCIAS=("CODIGO", "count")
    ).reset_index().sort_values(by="N_INCIDENCIAS", ascending=False)

    ax = axs[i]
    sns.barplot(data=grupo, x="MAT_AVERIADO", y="N_INCIDENCIAS",
palette="pastel", ax=ax)
    ax.set_title(f"Año {año}", fontsize=13)
    ax.set_xlabel("MAT_AVERIADO")
    ax.set_ylabel("Nº de Incidencias" if i == 0 else "")
    ax.tick_params(axis='x', rotation=45)
    ax.grid(False)

    # Etiquetas encima de las barras
    for p in ax.patches:
        height = p.get_height()
        if height > 0:
            ax.annotate(f'{int(height):,}',
                (p.get_x() + p.get_width()/2., height),
                ha='center', va='bottom', fontsize=8)

plt.suptitle("Número de Incidencias TIPO = 202 por MAT_AVERIADO (CLASE 2, Tensión
B) separadas por año", fontsize=16)
plt.tight_layout(rect=[0, 0.03, 1, 0.92])
plt.show()
```

## MEDIA TENSION

### GRAFICO NUMERO DE INCIDENCIAS CLASE=2 Y MT POR TIPO DE INCIDENCIA

```
import seaborn as sns
import matplotlib.pyplot as plt

# Filtrar por CLASE 2 y Nivel de Tensión MEDIA (M)
df_clase2_m = df[(df["CLASE"] == "2") & (df["NIV_TENS"] == "M")]

# Agrupar y contar
grupo_tipo = df_clase2_m.groupby("TIPO").agg({
    "CODIGO": "count"
}).rename(columns={"CODIGO": "N_INCIDENCIAS"}).reset_index()

# Ordenar por número de incidencias (descendente)
grupo_tipo = grupo_tipo.sort_values(by="N_INCIDENCIAS", ascending=False)

# Gráfico
plt.figure(figsize=(12,6))
ax = sns.barplot(data=grupo_tipo, x="TIPO", y="N_INCIDENCIAS", palette="pastel")
```

```
plt.title("Número de Incidencias por TIPO (CLASE 2, Nivel de Tensión MEDIA)",
fontsize=14)
plt.xlabel("TIPO")
plt.ylabel("N° de Incidencias")
plt.grid(False)

# Etiquetas encima de las barras
for p in ax.patches:
    height = p.get_height()
    if height > 0:
        ax.annotate(f'{int(height):,}',
                    (p.get_x() + p.get_width()/2., height),
                    ha='center', va='bottom', fontsize=9)

plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```

#### **TABLA DE NUMERO DE INCIDENCIAS CLASE=2 Y MT POR TIPO DE INCIDENCIA**

```
# Filtrar CLASE 2 y NIV_TENS = 'M'
df_clase2_m = df[(df["CLASE"] == "2") & (df["NIV_TENS"] == "M")]

# Agrupar por TIPO y contar incidencias
grupo_tipo_m = (
    df_clase2_m
    .groupby("TIPO")
    .agg(N_INCIDENCIAS=("CODIGO", "count"))
    .reset_index()
)

# Obtener descripciones únicas
tipos_descripciones = df[["TIPO", "DES_TIPO"]].drop_duplicates()
# Unir descripción
grupo_tipo_m = grupo_tipo_m.merge(tipos_descripciones, on="TIPO", how="left")

# Ordenar de mayor a menor
grupo_tipo_m = grupo_tipo_m.sort_values(by="N_INCIDENCIAS",
ascending=False).reset_index(drop=True)
# Mostrar tabla
print("Número de Incidencias por TIPO (CLASE 2, Nivel de Tensión MEDIA):")
import pandas as pd
from IPython.display import display
display(grupo_tipo_m)
```

#### **GRAFICO DE EVOLUCION DE NUMERO DE INCIDENCIAS CLASE=2 Y MT POR TIPO Y POR AÑO PARA PODER COMPARAR**

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
# Lista de años a graficar, ahora con 2025
años = [2022, 2023, 2024, 2025]
# Crear una figura con 4 subplots (uno por año)
fig, axs = plt.subplots(1, 4, figsize=(24, 6), sharey=True)
```

```
for i, año in enumerate(años):

    # Filtrar por año
    datos_anio = df_clase2_m[df_clase2_m["AÑO"] == año]

    # Agrupar por TIPO
    grupo_tipo = datos_anio.groupby("TIPO").agg({
        "CODIGO": "count"
    }).rename(columns={"CODIGO": "N_INCIDENCIAS"}).reset_index().sort_values(
        by="N_INCIDENCIAS", ascending=False
    )

    # Gráfico
    ax = axs[i]
    sns.barplot(data=grupo_tipo, x="TIPO", y="N_INCIDENCIAS", palette="pastel",
ax=ax)
    ax.set_title(f"Año {año}", fontsize=13)
    ax.set_xlabel("TIPO")
    ax.set_ylabel("Nº de Incidencias" if i == 0 else "")
    ax.tick_params(axis='x', rotation=45)

    ax.grid(False)

    # Etiquetas sobre las barras
    for p in ax.patches:
        height = p.get_height()
        if height > 0:
            ax.annotate(f'{int(height):,}',
                        (p.get_x() + p.get_width()/2., height),
                        ha='center', va='bottom', fontsize=8)
plt.suptitle("Número de Incidencias por TIPO (CLASE 2, Tensión MEDIA)",
fontsize=16)
plt.tight_layout(rect=[0, 0.03, 1, 0.95])
plt.show()
```

**#GRAFICO DE NUMERO DE INCIDENCIAS CLASE=2 Y MT POR TIPO ELIMINANDO LAS SIN TIEPI (NIVEL DE TENSION!=T)**

```
# Filtrar clase 2, nivel de tensión media y eliminar reenganches (SIN_TIEPI == 'S')
df_clase2_sin_tiepi = df[(df["CLASE"] == "2") &
                        (df["NIV_TENS"] == "M") &
                        (df["SIN_TIEPI"] != "S")]

# Agrupar y contar
grupo_tipo = df_clase2_sin_tiepi.groupby("TIPO").agg({
    "CODIGO": "count"
}).rename(columns={"CODIGO": "N_INCIDENCIAS"}).reset_index()

# Ordenar por número de incidencias (descendente)
grupo_tipo = grupo_tipo.sort_values(by="N_INCIDENCIAS", ascending=False)

# Gráfico
plt.figure(figsize=(12,6))
ax = sns.barplot(data=grupo_tipo, x="TIPO", y="N_INCIDENCIAS", palette="pastel")
```

```
plt.title("Número de Incidencias por TIPO (CLASE 2, Nivel de Tensión MEDIA ,
eliminando los reenganches)", fontsize=14)
plt.xlabel("TIPO")
plt.ylabel("N° de Incidencias")
plt.grid(False)

# Etiquetas encima de las barras
for p in ax.patches:
    height = p.get_height()
    if height > 0:
        ax.annotate(f'{int(height):,}',
                    (p.get_x() + p.get_width()/2., height),
                    ha='center', va='bottom', fontsize=9)

plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```

**TABLA DE NUMERO DE INCIDENCIAS CLASE=2 Y MT POR TIPO Y SIN CONTAR LAS SIN TIEPI (REENGANCHES)**

```
import pandas as pd
from tabulate import tabulate

# Agrupar por TIPO y contar número de incidencias
grupo_tipo = df_clase2_sin_tiepi.groupby("TIPO").agg({
    "CODIGO": "count"
}).rename(columns={"CODIGO": "N_INCIDENCIAS"}).reset_index()

# Ordenar de forma descendente
grupo_tipo = grupo_tipo.sort_values(by="N_INCIDENCIAS", ascending=False)

# Mostrar tabla
import pandas as pd
from IPython.display import display
display(grupo_tipo)
```

**GRAFICO DE NUMERO DE INCIDENCIAS CLASE=2 Y MT Y TIPO=301 POR INS\_ORIGEN TRAS ELIMINAR LAS SIN TIEPI (SIN REENGANCHES DE LINEAS AEREAS)**

```
import seaborn as sns
import matplotlib.pyplot as plt

df["CLASE"] = df["CLASE"].astype(str).str.strip()
df["NIV_TENS"] = df["NIV_TENS"].astype(str).str.strip()
df["TIPO"] = df["TIPO"].astype(str).str.strip()
df["SIN_TIEPI"] = df["SIN_TIEPI"].astype(str).str.strip()

# Filtro estricto con todas las condiciones obligatorias
df_clase2_m_301_st = df[
    (df["CLASE"] == "2") &
    (df["NIV_TENS"] == "M") &
    (df["TIPO"] == "301") &
    (df["SIN_TIEPI"] != "S")]
```

```
# Agrupar por INS_ORIGEN y contar
grupo_origen = df_clase2_m_301_st.groupby("INS_ORIGEN").agg({
    "CODIGO": "count"
}).rename(columns={"CODIGO": "N_INCIDENCIAS"}).reset_index()

# Ordenar por número de incidencias
grupo_origen = grupo_origen.sort_values(by="N_INCIDENCIAS", ascending=False)

# Gráfico
plt.figure(figsize=(12,6))
ax = sns.barplot(data=grupo_origen, x="INS_ORIGEN", y="N_INCIDENCIAS",
    palette="pastel")
plt.title("Número de Incidencias Tipo 301 por INS_ORIGEN (CLASE 2, Tensión MEDIA
, sin reenganches)", fontsize=14)
plt.xlabel("INS_ORIGEN")
plt.ylabel("Nº de Incidencias")
plt.grid(False)

# Etiquetas encima de las barras
for p in ax.patches:
    height = p.get_height()
    if height > 0:
        ax.annotate(f'{int(height):,}',
            (p.get_x() + p.get_width()/2., height),
            ha='center', va='bottom', fontsize=9)

plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```

**TABLA DE NUMERO DE INCIDENCIAS CLASE=2 Y MT Y TIPO=301 POR INS\_ORIGEN TRAS ELIMINAR LAS SIN TIEPI (SIN REENGANCHES DE LINEAS AEREAS)**

```
import pandas as pd
from tabulate import tabulate

# Seleccionar solo columnas relevantes para agrupar
df_subset = df_clase2_m_301_st[["INS_ORIGEN", "DESC_ORIGEN", "CODIGO"]]

# Agrupar por INS_ORIGEN y DESC_ORIGEN y contar incidencias
grupo_origen = df_subset.groupby(["INS_ORIGEN", "DESC_ORIGEN"]).agg({
    "CODIGO": "count"
}).rename(columns={"CODIGO": "N_INCIDENCIAS"}).reset_index()

# Ordenar por número de incidencias descendente
grupo_origen = grupo_origen.sort_values(by="N_INCIDENCIAS", ascending=False)

# Mostrar tabla en consola (formato bonito)
# Mostrar tabla
import pandas as pd
from IPython.display import display

display(grupo_origen)
```

**GRAFICO DE EVOLUCION DEL NUMERO DE INCIDENCIAS CLASE=2 Y MT Y TIPO=301 POR INS\_ORIGEN TRAS ELIMINAR LAS SIN TIEPI (SIN REENGANCHES) POR AÑO**

```
import matplotlib.pyplot as plt
import seaborn as sns

# Lista de años que quieres graficar
años = [2022, 2023, 2024, 2025]

# Crear subplots por año
fig, axs = plt.subplots(1, 4, figsize=(26, 6), sharey=True)

for i, año in enumerate(años):
    # Filtrado por condiciones
    df_filtrado = df[
        (df["CLASE"] == "2") &
        (df["NIV_TENS"] == "M") &
        (df["TIPO"] == "301") &
        (df["SIN_TIEPI"] != "S") &
        (df["AÑO"] == año)
    ]

    # Agrupación por INS_ORIGEN
    grupo_origen = df_filtrado.groupby("INS_ORIGEN").agg({
        "CODIGO": "count"
    }).rename(columns={"CODIGO": "N_INCIDENCIAS"}).reset_index().sort_values(
        by="N_INCIDENCIAS", ascending=False
    )

    # Crear gráfico de barras
    ax = axs[i]
    sns.barplot(data=grupo_origen, x="INS_ORIGEN", y="N_INCIDENCIAS",
        palette="pastel", ax=ax)
    ax.set_title(f"Año {año}", fontsize=13)
    ax.set_xlabel("INS_ORIGEN")
    if i == 0:
        ax.set_ylabel("N° de Incidencias")
    else:
        ax.set_ylabel("")

    ax.tick_params(axis='x', rotation=45)
    ax.grid(False)

    # Etiquetas encima de las barras
    for p in ax.patches:
        height = p.get_height()
        if height > 0:
            ax.annotate(f'{int(height):,}',
                (p.get_x() + p.get_width()/2., height),
                ha='center', va='bottom', fontsize=8)

# Título general
plt.suptitle("Número de Incidencias TIPO 301 por INS_ORIGEN (CLASE 2, Tensión
MEDIA, sin reenganches)", fontsize=16)
plt.tight_layout(rect=[0, 0.03, 1, 0.92])
plt.show()
```

**GRAFICO DE NUMERO DE INCIDENCIAS CLASE=2 Y MT Y TIPO=301 POR AÑO Y POR MAT\_AVERIADO TRAS ELIMINAR LAS SIN TIEPI (SIN REENGANCHES)**

```
import seaborn as sns
import matplotlib.pyplot as plt

# Agrupar por INS_ORIGEN y contar
grupo_origen = df_clase2_m_301_st.groupby("MAT_AVERIADO").agg({
    "CODIGO": "count"
}).rename(columns={"CODIGO": "N_INCIDENCIAS"}).reset_index()

# Ordenar por número de incidencias
grupo_origen = grupo_origen.sort_values(by="N_INCIDENCIAS", ascending=False)

# Gráfico
plt.figure(figsize=(12,6))
ax = sns.barplot(data=grupo_origen, x="MAT_AVERIADO", y="N_INCIDENCIAS",
palette="pastel")

plt.title("Número de Incidencias Tipo 301 por MAT_AVERIADO (CLASE 2, Tensión
MEDIA , sin reenganchos)", fontsize=14)
plt.xlabel("INS_ORIGEN")
plt.ylabel("N° de Incidencias")
plt.grid(False)

# Etiquetas encima de las barras
for p in ax.patches:
    height = p.get_height()
    if height > 0:
        ax.annotate(f'{int(height):,}',
                    (p.get_x() + p.get_width()/2., height),
                    ha='center', va='bottom', fontsize=9)

plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```

**TABLA NUMERO DE INCIDENCIAS CLASE=2 Y MT Y TIPO=301 POR MAT\_AVERIADO TRAS ELIMINAR LAS SIN TIEPI (REENGANCHES)**

```
import pandas as pd
from tabulate import tabulate

# Seleccionar solo columnas relevantes para agrupar
df_subset = df_clase2_m_301_st[["MAT_AVERIADO", "DESC_MAT_AVER", "CODIGO"]]

grupo_origen = df_subset.groupby(["MAT_AVERIADO", "DESC_MAT_AVER"]).agg({
    "CODIGO": "count"
}).rename(columns={"CODIGO": "N_INCIDENCIAS"}).reset_index()

# Ordenar por número de incidencias descendente
grupo_origen = grupo_origen.sort_values(by="N_INCIDENCIAS", ascending=False)
# Mostrar tabla en consola (formato bonito)
# Mostrar table
import pandas as pd
from IPython.display import display
display(grupo_origen)
```

**GRAFICO DE EVOLUCION DEL NUMERO DE INCIDENCIAS CLASE=2 Y MT Y TIPO=301 POR AÑOS Y POR MAT\_AVERIADO TRAS ELIMINAR LAS SIN TIEPI (REENGANCHES)**

```
import matplotlib.pyplot as plt
import seaborn as sns
# Lista de años que quieres graficar
años = [2022, 2023, 2024, 2025]
# Crear subplots por año (4 columnas)
fig, axs = plt.subplots(1, 4, figsize=(26, 6), sharey=True)

for i, año in enumerate(años):
    # Filtrado por condiciones
    df_filtrado = df[
        (df["CLASE"] == "2") &
        (df["NIV_TENS"] == "M") &
        (df["TIPO"] == "301") &
        (df["SIN_TIEPI"] != "S") &
        (df["AÑO"] == año)
    ]
    # Agrupación por MAT_AVERIADO
    grupo_origen = df_filtrado.groupby("MAT_AVERIADO").agg({
        "CODIGO": "count"
    }).rename(columns={"CODIGO": "N_INCIDENCIAS"}).reset_index().sort_values(
        by="N_INCIDENCIAS", ascending=False
    )
    # Crear gráfico de barras
    ax = axs[i]
    sns.barplot(data=grupo_origen, x="MAT_AVERIADO", y="N_INCIDENCIAS",
        palette="pastel", ax=ax)
    ax.set_title(f"Año {año}", fontsize=13)
    ax.set_xlabel("MAT_AVERIADO")
    if i == 0:
        ax.set_ylabel("N° de Incidencias")
    else:
        ax.set_ylabel("")
    ax.tick_params(axis='x', rotation=45)
    ax.grid(False)

    # Etiquetas encima de las barras
    for p in ax.patches:
        height = p.get_height()
        if height > 0:
            ax.annotate(f'{int(height):,}',
                (p.get_x() + p.get_width()/2., height),
                ha='center', va='bottom', fontsize=8)

# Título general
plt.suptitle("Número de Incidencias TIPO 301 por MAT_AVERIADO separadas en años
(CLASE 2, Tensión MEDIA, sin reenganches)", fontsize=16)
plt.tight_layout(rect=[0, 0.03, 1, 0.92])
plt.show()

plt.show()
```

**GRAFICO DEL NUMERO DE INCIDENCIAS CLASE=2 Y MT Y TIPO=303 POR INS\_ORIGEN TRAS ELIMINAR LAS SIN TIEPI (REENGANCHES)**

```
import seaborn as sns
import matplotlib.pyplot as plt

# Normalizar columnas al tipo string y quitar espacios
for col in ["CLASE", "NIV_TENS", "TIPO", "SIN_TIEPI"]:
    df[col] = df[col].astype(str).str.strip()

# Filtrar por CLASE 2, Nivel de Tensión MEDIA y TIPO 303, SIN_TIEPI distinto de "S"
df_clase2_m_303_st = df[
    (df["CLASE"] == "2") &
    (df["NIV_TENS"] == "M") &
    (df["TIPO"] == "303") &
    (df["SIN_TIEPI"] != "S")
]

# Agrupar por INS_ORIGEN y contar incidencias
grupo_origen = df_clase2_m_303_st.groupby("INS_ORIGEN").agg(
    N_INCIDENCIAS=("CODIGO", "count")
).reset_index().sort_values(by="N_INCIDENCIAS", ascending=False)

# Gráfico de barras
plt.figure(figsize=(12, 6))
ax = sns.barplot(data=grupo_origen, x="INS_ORIGEN", y="N_INCIDENCIAS",
palette="pastel")

plt.title("Número de Incidencias Tipo 303 por INS_ORIGEN (CLASE 2, Tensión MEDIA,
sin reenganches)", fontsize=14)
plt.xlabel("INS_ORIGEN")
plt.ylabel("Nº de Incidencias")
plt.grid(False)

# Etiquetas encima de cada barra
for p in ax.patches:
    height = p.get_height()
    if height > 0:
        ax.annotate(f'{int(height):,}',
                    (p.get_x() + p.get_width()/2., height),
                    ha='center', va='bottom', fontsize=9)

plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```

**TABLA DEL NUMERO DE INCIDENCIAS CLASE=2 Y MT Y TIPO=303 POR INS\_ORIGEN TRAS ELIMINAR LAS SIN TIEPI (REENGANCHES)**

```
import pandas as pd
from tabulate import tabulate

# Seleccionar solo columnas relevantes para agrupar
df_subset = df_clase2_m_303_st[["INS_ORIGEN", "DESC_ORIGEN", "CODIGO"]]
```

```
# Agrupar por INS_ORIGEN y DESC_ORIGEN y contar incidencias
grupo_origen = df_subset.groupby(["INS_ORIGEN", "DESC_ORIGEN"]).agg({
    "CODIGO": "count"
}).rename(columns={"CODIGO": "N_INCIDENCIAS"}).reset_index()

# Ordenar por número de incidencias descendente
grupo_origen = grupo_origen.sort_values(by="N_INCIDENCIAS", ascending=False)

# Mostrar tabla en consola (formato bonito)
# Mostrar table
```

**GRAFICO DE EVOLUCION DEL NUMERO DE INCIDENCIAS CLASE=2 Y MT Y TIPO=303 POR INS\_ORIGEN TRAS ELIMINAR LAS SIN TIEPI (REENGANCHES)**

```
import pandas as pd
from IPython.display import display

display(grupo_origen)
import seaborn as sns
import matplotlib.pyplot as plt

# Lista de años que quieres graficar
años = [2022, 2023, 2024, 2025]

# Crear subplots por año
fig, axs = plt.subplots(1, 4, figsize=(26, 6), sharey=True)

for i, año in enumerate(años):
    # Filtrar por año y condiciones
    df_filtrado = df[
        (df["CLASE"] == "2") &
        (df["NIV_TENS"] == "M") &
        (df["TIPO"] == "303") &
        (df["SIN_TIEPI"] != "S") &
        (df["AÑO"] == año)
    ]

    # Agrupar por INS_ORIGEN
    grupo_origen = df_filtrado.groupby("INS_ORIGEN").agg(
        N_INCIDENCIAS=("CODIGO", "count")
    ).reset_index().sort_values(by="N_INCIDENCIAS", ascending=False)

    # Crear gráfico de barras
    ax = axs[i]
    sns.barplot(data=grupo_origen, x="INS_ORIGEN", y="N_INCIDENCIAS",
        palette="pastel", ax=ax)
    ax.set_title(f"Año {año}", fontsize=13)
    ax.set_xlabel("INS_ORIGEN")
    if i == 0:
        ax.set_ylabel("Nº de Incidencias")
    else:
        ax.set_ylabel("")
    ax.tick_params(axis='x', rotation=45)
    ax.grid(False)
```

```
# Etiquetas encima de las barras
for p in ax.patches:
    height = p.get_height()
    if height > 0:
        ax.annotate(f'{int(height):,}',
                    (p.get_x() + p.get_width()/2., height),
                    ha='center', va='bottom', fontsize=8)

# Título general
plt.suptitle("Número de Incidencias TIPO 303 por INS_ORIGEN (CLASE 2, Tensión
MEDIA, sin reenganches)", fontsize=16)
plt.tight_layout(rect=[0, 0.03, 1, 0.92])
plt.show()
```

**GRAFICO DEL NUMERO DE INCIDENCIAS CLASE=2 Y MT Y TIPO=303 POR MAT\_AVERIADO TRAS ELIMINAR LAS SIN TIEPI (REENGANCHES)**

```
import seaborn as sns
import matplotlib.pyplot as plt

# Agrupar por INS_ORIGEN y contar
grupo_origen = df_clase2_m_303_st.groupby("MAT_AVERIADO").agg({
    "CODIGO": "count"
}).rename(columns={"CODIGO": "N_INCIDENCIAS"}).reset_index()

# Ordenar por número de incidencias
grupo_origen = grupo_origen.sort_values(by="N_INCIDENCIAS", ascending=False)

# Gráfico
plt.figure(figsize=(12,6))
ax = sns.barplot(data=grupo_origen, x="MAT_AVERIADO", y="N_INCIDENCIAS",
palette="pastel")

plt.title("Número de Incidencias Tipo 303 por MAT_AVERIADO (CLASE 2, Tensión
MEDIA , sin reenganchos)", fontsize=14)
plt.xlabel("MAT_AVERIADO")
plt.ylabel("N° de Incidencias")
plt.grid(False)

# Etiquetas encima de las barras
for p in ax.patches:
    height = p.get_height()
    if height > 0:
        ax.annotate(f'{int(height):,}',
                    (p.get_x() + p.get_width()/2., height),
                    ha='center', va='bottom', fontsize=9)

plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```

**TABLA DEL NUMERO DE INCIDENCIAS CLASE=2 Y MT Y TIPO=303 POR MAT\_AVERIADO TRAS ELIMINAR LAS SIN TIEPI (REENGANCHES)**

```
import pandas as pd
```

```

from tabulate import tabulate

# Seleccionar solo columnas relevantes para agrupar
df_subset = df_clase2_m_303_st[["MAT_AVERIADO", "DESC_MAT_AVER", "CODIGO"]]

grupo_origen = df_subset.groupby(["MAT_AVERIADO", "DESC_MAT_AVER"]).agg({
    "CODIGO": "count"
}).rename(columns={"CODIGO": "N_INCIDENCIAS"}).reset_index()

# Ordenar por número de incidencias descendente
grupo_origen = grupo_origen.sort_values(by="N_INCIDENCIAS", ascending=False)

# Mostrar tabla en consola (formato bonito)
# Mostrar table
import pandas as pd
from IPython.display import display
display(grupo_origen)

```

**GRAFICO DE EVOLUCION DEL NUMERO DE INCIDENCIAS CLASE=2 Y MT Y TIPO=303 POR AÑOS Y POR MAT\_AVERIADO TRAS ELIMINAR LAS SIN TIEPI (REENGANCHES)**

```

import seaborn as sns
import matplotlib.pyplot as plt

# Lista de años que quieres graficar
años = [2022, 2023, 2024, 2025]

# Crear subplots por año
fig, axs = plt.subplots(1, 4, figsize=(26, 6), sharey=True)

for i, año in enumerate(años):
    df_filtrado = df[
        (df["CLASE"] == "2") &
        (df["NIV_TENS"] == "M") &
        (df["TIPO"] == "303") &
        (df["SIN_TIEPI"] != "S") &
        (df["AÑO"] == año)
    ]

    # Agrupación por MAT_AVERIADO
    grupo_origen = df_filtrado.groupby("MAT_AVERIADO").agg({
        "CODIGO": "count"
    }).rename(columns={"CODIGO": "N_INCIDENCIAS"}).reset_index().sort_values(
        by="N_INCIDENCIAS", ascending=False
    )

    # Crear gráfico de barras
    ax = axs[i]
    sns.barplot(data=grupo_origen, x="MAT_AVERIADO", y="N_INCIDENCIAS",
        palette="pastel", ax=ax)
    ax.set_title(f"Año {año}", fontsize=13)
    ax.set_xlabel("MAT_AVERIADO")
    if i == 0:
        ax.set_ylabel("N° de Incidencias")

```

```
else:
    ax.set_ylabel("")
ax.tick_params(axis='x', rotation=45)
ax.grid(False)

# Etiquetas encima de las barras
for p in ax.patches:
    height = p.get_height()
    if height > 0:
        ax.annotate(f'{int(height):,}',
                    (p.get_x() + p.get_width()/2., height),
                    ha='center', va='bottom', fontsize=8)

# Título general
plt.suptitle("Número de Incidencias TIPO 303 por MAT_AVERIADO separadas en años
(CLASE 2, Tensión MEDIA, sin reenganches)", fontsize=16)
plt.tight_layout(rect=[0, 0.03, 1, 0.92])
plt.show()
```

**GRAFICO DEL NUMERO DE INCIDENCIAS CLASE=2 Y MT Y TIPO=304 Y POR INS\_ORIGEN TRAS ELIMINAR LAS SIN TIEPI (REENGANCHES)**

```
import seaborn as sns
import matplotlib.pyplot as plt

# Normalizar columnas al tipo string y quitar espacios
for col in ["CLASE", "NIV_TENS", "TIPO", "SIN_TIEPI"]:
    df[col] = df[col].astype(str).str.strip()

# Filtrar por CLASE 2, Tensión MEDIA, TIPO 304 y SIN_TIEPI distinto de "S"
df_clase2_m_304_st = df[
    (df["CLASE"] == "2") &
    (df["NIV_TENS"] == "M") &
    (df["TIPO"] == "304") &
    (df["SIN_TIEPI"] != "S")
]

# Agrupar por INS_ORIGEN y contar incidencias
grupo_origen = df_clase2_m_304_st.groupby("INS_ORIGEN").agg(
    N_INCIDENCIAS=("CODIGO", "count")
).reset_index().sort_values(by="N_INCIDENCIAS", ascending=False)

# Gráfico de barras
plt.figure(figsize=(12, 6))
ax = sns.barplot(data=grupo_origen, x="INS_ORIGEN", y="N_INCIDENCIAS",
palette="pastel")

plt.title("Número de Incidencias Tipo 304 por INS_ORIGEN (CLASE 2, Tensión MEDIA,
sin reenganches)", fontsize=14)
plt.xlabel("INS_ORIGEN")
plt.ylabel("N° de Incidencias")
plt.grid(False)
```

```
# Etiquetas encima de las barras
for p in ax.patches:
    height = p.get_height()
    if height > 0:
        ax.annotate(f'{int(height):,}',
                    (p.get_x() + p.get_width()/2., height),
                    ha='center', va='bottom', fontsize=9)

plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```

**TABLA DEL NUMERO DE INCIDENCIAS CLASE=2 Y MT Y TIPO=304 y POR INS\_ORIGEN TRAS ELIMINAR LAS SIN TIEPI (REENGANCHES)**

```
import pandas as pd
from tabulate import tabulate

# Seleccionar solo columnas relevantes para agrupar
df_subset = df_clase2_m_304_st[["INS_ORIGEN", "DESC_ORIGEN", "CODIGO"]]

# Agrupar por INS_ORIGEN y DESC_ORIGEN y contar incidencias
grupo_origen = df_subset.groupby(["INS_ORIGEN", "DESC_ORIGEN"]).agg({
    "CODIGO": "count"
}).rename(columns={"CODIGO": "N_INCIDENCIAS"}).reset_index()

# Ordenar por número de incidencias descendente
grupo_origen = grupo_origen.sort_values(by="N_INCIDENCIAS", ascending=False)

# Mostrar tabla en consola (formato bonito)
# Mostrar tabla
import pandas as pd
from IPython.display import display

display(grupo_origen)
```

**GRAFICO DE EVOLUCION DEL NUMERO DE INCIDENCIAS CLASE=2 Y MT Y TIPO=304 y POR INS\_ORIGEN TRAS ELIMINAR LAS SIN TIEPI (REENGANCHES)**

```
import seaborn as sns
import matplotlib.pyplot as plt
import pandas as pd

# Asegurar limpieza de columnas clave
for col in ["CLASE", "NIV_TENS", "TIPO", "SIN_TIEPI"]:
    df[col] = df[col].astype(str).str.strip()

# Asegurarse de que exista la columna AÑO
if "AÑO" not in df.columns:
    if "FECHA" in df.columns:
        df["FECHA"] = pd.to_datetime(df["FECHA"])
        df["AÑO"] = df["FECHA"].dt.year
```

```
else:
    raise ValueError("No existe la columna AÑO ni FECHA para segmentar por
años.")

# Lista de años a graficar (incluyendo 2025)
años = [2022, 2023, 2024, 2025]

# Crear subplots por año
fig, axs = plt.subplots(1, 4, figsize=(26, 6), sharey=True)

for i, año in enumerate(años):
    # Filtrado por condiciones y año
    df_filtrado = df[
        (df["CLASE"] == "2") &
        (df["NIV_TENS"] == "M") &
        (df["TIPO"] == "304") &
        (df["SIN_TIEPI"] != "S") &
        (df["AÑO"] == año)
    ]

    # Agrupación por INS_ORIGEN
    grupo_origen = df_filtrado.groupby("INS_ORIGEN").agg(
        N_INCIDENCIAS=("CODIGO", "count")
    ).reset_index().sort_values(by="N_INCIDENCIAS", ascending=False)

    # Crear gráfico de barras
    ax = axs[i]
    sns.barplot(data=grupo_origen, x="INS_ORIGEN", y="N_INCIDENCIAS",
palette="pastel", ax=ax)
    ax.set_title(f"Año {año}", fontsize=13)
    ax.set_xlabel("INS_ORIGEN")
    if i == 0:
        ax.set_ylabel("N° de Incidencias")
    else:
        ax.set_ylabel("")
    ax.tick_params(axis='x', rotation=45)
    ax.grid(False)

    # Etiquetas encima de las barras
    for p in ax.patches:
        height = p.get_height()
        if height > 0:
            ax.annotate(f'{int(height):,}',
                (p.get_x() + p.get_width()/2., height),
                ha='center', va='bottom', fontsize=8)

# Título general
plt.suptitle("Número de Incidencias TIPO 304 por INS_ORIGEN (CLASE 2, Tensión
MEDIA, sin reenganches)", fontsize=16)
plt.tight_layout(rect=[0, 0.03, 1, 0.92])
plt.show()
```

**GRAFICO DEL NUMERO DE INCIDENCIAS CLASE=2 Y MT Y TIPO=304 y POR MAT\_AVERIADO TRAS ELIMINAR LAS SIN TIEPI (REENGANCHES)**

```
import seaborn as sns
import matplotlib.pyplot as plt

# Agrupar por INS_ORIGEN y contar
grupo_origen = df_clase2_m_304_st.groupby("MAT_AVERIADO").agg({
    "CODIGO": "count"
}).rename(columns={"CODIGO": "N_INCIDENCIAS"}).reset_index()

# Ordenar por número de incidencias
grupo_origen = grupo_origen.sort_values(by="N_INCIDENCIAS", ascending=False)

# Gráfico
plt.figure(figsize=(12,6))
ax = sns.barplot(data=grupo_origen, x="MAT_AVERIADO", y="N_INCIDENCIAS",
palette="pastel")

plt.title("Número de Incidencias Tipo 304 por MAT_AVERIADO (CLASE 2, Tensión
MEDIA , sin reenganchos)", fontsize=14)
plt.xlabel("MAT_AVERIADO")
plt.ylabel("N° de Incidencias")
plt.grid(False)

# Etiquetas encima de las barras
for p in ax.patches:
    height = p.get_height()
    if height > 0:
        ax.annotate(f'{int(height):,}',
                    (p.get_x() + p.get_width()/2., height),
                    ha='center', va='bottom', fontsize=9)

plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```

**TABLA DEL NUMERO DE INCIDENCIAS CLASE=2 Y MT Y TIPO=304 y POR MAT\_AVERIADO TRAS ELIMINAR LAS SIN TIEPI (REENGANCHES)**

```
import pandas as pd
from tabulate import tabulate

# Seleccionar solo columnas relevantes para agrupar
df_subset = df_clase2_m_304_st[["MAT_AVERIADO", "DESC_MAT_AVER", "CODIGO"]]

grupo_origen = df_subset.groupby(["MAT_AVERIADO", "DESC_MAT_AVER"]).agg({
    "CODIGO": "count"
}).rename(columns={"CODIGO": "N_INCIDENCIAS"}).reset_index()

# Ordenar por número de incidencias descendente
grupo_origen = grupo_origen.sort_values(by="N_INCIDENCIAS", ascending=False)

# Mostrar tabla en consola (formato bonito)
# Mostrar tabla
import pandas as pd
```

```
from IPython.display import display

display(grupo_origen)
```

**GRAFICO DE EVOLUCION DEL NUMERO DE INCIDENCIAS CLASE=2 Y MT Y TIPO=304 y POR MAT\_AVERIADO Y POR AÑOS TRAS ELIMINAR LAS SIN TIEPI (REENGANCHES)**

```
import seaborn as sns
import matplotlib.pyplot as plt

# Limpieza y normalización previa si no se ha hecho ya
for col in ["CLASE", "NIV_TENS", "TIPO", "SIN_TIEPI"]:
    df[col] = df[col].astype(str).str.strip()

# Asegurar columna AÑO a partir de una columna de fecha si no existe
if "AÑO" not in df.columns:
    if "FECHA" in df.columns:
        df["FECHA"] = pd.to_datetime(df["FECHA"])
        df["AÑO"] = df["FECHA"].dt.year
    else:
        raise ValueError("No se encuentra la columna AÑO ni FECHA para segmentar los datos por año.")

# Lista de años a graficar (incluyendo 2025)
años = [2022, 2023, 2024, 2025]

# Crear subplots
fig, axs = plt.subplots(1, 4, figsize=(26, 6), sharey=True)

for i, año in enumerate(años):
    # Filtrar por condiciones más año
    df_filtrado = df[
        (df["CLASE"] == "2") &
        (df["NIV_TENS"] == "M") &
        (df["TIPO"] == "304") &
        (df["SIN_TIEPI"] != "S") &
        (df["AÑO"] == año)
    ]

    # Agrupar por MAT_AVERIADO
    grupo_origen = df_filtrado.groupby("MAT_AVERIADO").agg(
        N_INCIDENCIAS=("CODIGO", "count")
    ).reset_index().sort_values(by="N_INCIDENCIAS", ascending=False)

    # Crear gráfico de barras
    ax = axs[i]
    sns.barplot(data=grupo_origen, x="MAT_AVERIADO", y="N_INCIDENCIAS",
        palette="pastel", ax=ax)
    ax.set_title(f"Año {año}", fontsize=13)
    ax.set_xlabel("MAT_AVERIADO")
    if i == 0:
        ax.set_ylabel("N° de Incidencias")
    else:
        ax.set_ylabel("")
```

```
ax.tick_params(axis='x', rotation=45)
ax.grid(False)

# Etiquetas encima de las barras
for p in ax.patches:
    height = p.get_height()
    if height > 0:
        ax.annotate(f'{int(height):,}',
                    (p.get_x() + p.get_width()/2., height),
                    ha='center', va='bottom', fontsize=8)

# Título general
plt.suptitle("Número de Incidencias TIPO 304 por MAT_AVERIADO separadas en años
(CLASE 2, Tensión MEDIA, sin reenganches)", fontsize=16)
plt.tight_layout(rect=[0, 0.03, 1, 0.92])
plt.show()
```

**GRAFICO DEL NUMERO DE INCIDENCIAS CLASE=2 Y MT Y TIPO=202 POR INS\_ORIGEN TRAS ELIMINAR LAS SIN TIEPI (REENGANCHES)**

```
### TIPO = 202 ESTUDIO POR INS_ORIGEN :

import seaborn as sns
import matplotlib.pyplot as plt

# Normalización previa si no se ha hecho
for col in ["CLASE", "NIV_TENS", "TIPO", "SIN_TIEPI"]:
    df[col] = df[col].astype(str).str.strip()

# Filtrar por CLASE 2, Tensión MEDIA, TIPO 399 y SIN_TIEPI distinto de "S"
df_clase2_m_202_st = df[
    (df["CLASE"] == "2") &
    (df["NIV_TENS"] == "M") &
    (df["TIPO"] == "202") &
    (df["SIN_TIEPI"] != "S")
]

# Agrupar por INS_ORIGEN y contar incidencias
grupo_origen = df_clase2_m_202_st.groupby("INS_ORIGEN").agg(
    N_INCIDENCIAS=("CODIGO", "count")
).reset_index().sort_values(by="N_INCIDENCIAS", ascending=False)

# Gráfico de barras
plt.figure(figsize=(12, 6))
ax = sns.barplot(data=grupo_origen, x="INS_ORIGEN", y="N_INCIDENCIAS",
                 palette="pastel")

plt.title("Número de Incidencias Tipo 202 por INS_ORIGEN (CLASE 2, Tensión MEDIA,
sin reenganches)", fontsize=14)
plt.xlabel("INS_ORIGEN")
plt.ylabel("Nº de Incidencias")
plt.grid(False)
```

```
# Etiquetas encima de las barras
for p in ax.patches:
    height = p.get_height()
    if height > 0:
        ax.annotate(f'{int(height):,}',
                    (p.get_x() + p.get_width()/2., height),
                    ha='center', va='bottom', fontsize=9)

plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```

**TABLA DEL NUMERO DE INCIDENCIAS CLASE=2 Y MT Y TIPO=202 POR INS\_ORIGEN TRAS ELIMINAR LAS SIN TIEPI (REENGANCHES)**

```
import pandas as pd
from tabulate import tabulate

# Seleccionar solo columnas relevantes para agrupar
df_subset = df_clase2_m_202_st[["INS_ORIGEN", "DESC_ORIGEN", "CODIGO"]]

# Agrupar por INS_ORIGEN y DESC_ORIGEN y contar incidencias
grupo_origen = df_subset.groupby(["INS_ORIGEN", "DESC_ORIGEN"]).agg({
    "CODIGO": "count"
}).rename(columns={"CODIGO": "N_INCIDENCIAS"}).reset_index()

# Ordenar por número de incidencias descendente
grupo_origen = grupo_origen.sort_values(by="N_INCIDENCIAS", ascending=False)

# Mostrar tabla en consola (formato bonito)
# Mostrar tabla
import pandas as pd
from IPython.display import display

display(grupo_origen)
```

**GRAFICO DE EVOLUCION DEL NUMERO DE INCIDENCIAS CLASE=2 Y MT Y TIPO=202 POR INS\_ORIGEN Y POR AÑO TRAS ELIMINAR LAS SIN TIEPI (REENGANCHES)**

```
import seaborn as sns
import matplotlib.pyplot as plt

# Normalizar columnas clave
for col in ["CLASE", "NIV_TENS", "TIPO", "SIN_TIEPI"]:
    df[col] = df[col].astype(str).str.strip()

# Crear columna 'AÑO' si no existe
if "AÑO" not in df.columns:
    if "FECHA" in df.columns:
        df["FECHA"] = pd.to_datetime(df["FECHA"])
        df["AÑO"] = df["FECHA"].dt.year
    else:
        raise ValueError("No se encuentra la columna AÑO ni FECHA para segmentar los datos por año.")
```

```
# Lista de años incluyendo 2025
años = [2022, 2023, 2024, 2025]

# Crear subplots
fig, axs = plt.subplots(1, 4, figsize=(26, 6), sharey=True)

for i, año in enumerate(años):
    # Filtrar por condiciones + año actual
    df_filtrado = df[
        (df["CLASE"] == "2") &
        (df["NIV_TENS"] == "M") &
        (df["TIPO"] == "202") &
        (df["SIN_TIEPI"] != "S") &
        (df["AÑO"] == año)
    ]

    # Agrupar por INS_ORIGEN
    grupo_origen = df_filtrado.groupby("INS_ORIGEN").agg(
        N_INCIDENCIAS=("CODIGO", "count")
    ).reset_index().sort_values(by="N_INCIDENCIAS", ascending=False)

    # Crear gráfico de barras
    ax = axs[i]
    sns.barplot(data=grupo_origen, x="INS_ORIGEN", y="N_INCIDENCIAS",
        palette="pastel", ax=ax)
    ax.set_title(f"Año {año}", fontsize=13)
    ax.set_xlabel("INS_ORIGEN")
    if i == 0:
        ax.set_ylabel("N° de Incidencias")
    else:
        ax.set_ylabel("")
    ax.tick_params(axis='x', rotation=45)
    ax.grid(False)

    # Etiquetas encima de las barras
    for p in ax.patches:
        height = p.get_height()
        if height > 0:
            ax.annotate(f'{int(height):,}',
                (p.get_x() + p.get_width()/2., height),
                ha='center', va='bottom', fontsize=8)

# Título general
plt.suptitle("Número de Incidencias TIPO 202 por INS_ORIGEN (CLASE 2, Tensión
MEDIA, sin reenganches)", fontsize=16)
plt.tight_layout(rect=[0, 0.03, 1, 0.92])
plt.show()
```

**GRAFICO DEL NUMERO DE INCIDENCIAS CLASE=2 Y MT Y TIPO=202 POR MAT\_AVERIADO Y POR AÑO TRAS ELIMINAR LAS SIN TIEPI (REENGANCHES)**

```
import seaborn as sns
import matplotlib.pyplot as plt

# Agrupar por INS_ORIGEN y contar
```

```

grupo_origen = df_clase2_m_202_st.groupby("MAT_AVERIADO").agg({
    "CODIGO": "count"
}).rename(columns={"CODIGO": "N_INCIDENCIAS"}).reset_index()
# Ordenar por número de incidencias
grupo_origen = grupo_origen.sort_values(by="N_INCIDENCIAS", ascending=False)
# Gráfico
plt.figure(figsize=(12,6))
ax = sns.barplot(data=grupo_origen, x="MAT_AVERIADO", y="N_INCIDENCIAS",
palette="pastel")

plt.title("Número de Incidencias Tipo 202 por MAT_AVERIADO (CLASE 2, Tensión
MEDIA , sin reenganchos)", fontsize=14)
plt.xlabel("mat_averiado")
plt.ylabel("N° de Incidencias")
plt.grid(False)

# Etiquetas encima de las barras
for p in ax.patches:
    height = p.get_height()
    if height > 0:
        ax.annotate(f'{int(height):,}',
                    (p.get_x() + p.get_width()/2., height),
                    ha='center', va='bottom', fontsize=9)
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()

```

**TABLA DEL NUMERO DE INCIDENCIAS CLASE=2 Y MT Y TIPO=202 POR MAT\_AVERIADO Y POR AÑO TRAS ELIMINAR LAS SIN TIEPI (REENGANCHES)**

```

import pandas as pd
from tabulate import tabulate
# Seleccionar solo columnas relevantes para agrupar
df_subset = df_clase2_m_202_st[["MAT_AVERIADO", "DESC_MAT_AVER", "CODIGO"]]

grupo_origen = df_subset.groupby(["MAT_AVERIADO", "DESC_MAT_AVER"]).agg({
    "CODIGO": "count"
}).rename(columns={"CODIGO": "N_INCIDENCIAS"}).reset_index()

# Ordenar por número de incidencias descendente
grupo_origen = grupo_origen.sort_values(by="N_INCIDENCIAS", ascending=False)

# Mostrar tabla en consola (formato bonito)
# Mostrar tabla
import pandas as pd
from IPython.display import display
display(grupo_origen)

```

**GRAFICO DE LA EVOLUCION DEL NUMERO DE INCIDENCIAS CLASE=2 Y MT Y TIPO=202 POR MAT\_AVERIADO Y POR AÑO TRAS ELIMINAR LAS SIN TIEPI (REENGANCHES)**

```

import seaborn as sns
import matplotlib.pyplot as plt
# Asegurar que las columnas están limpias
for col in ["CLASE", "NIV_TENS", "TIPO", "SIN_TIEPI"]:
    df[col] = df[col].astype(str).str.strip()

```

```
# Crear columna AÑO si no existe
if "AÑO" not in df.columns:
    if "FECHA" in df.columns:
        df["FECHA"] = pd.to_datetime(df["FECHA"])
        df["AÑO"] = df["FECHA"].dt.year
    else:
        raise ValueError("No se encuentra columna AÑO ni FECHA para segmentar por
año.")

# Lista de años que quieres graficar (ahora incluyendo 2025)
años = [2022, 2023, 2024, 2025]

# Crear subplots
fig, axs = plt.subplots(1, 4, figsize=(26, 6), sharey=True)

for i, año in enumerate(años):
    # Filtrar por condiciones y año
    df_filtrado = df[
        (df["CLASE"] == "2") &
        (df["NIV_TENS"] == "M") &
        (df["TIPO"] == "202") &
        (df["SIN_TIEPI"] != "S") &
        (df["AÑO"] == año)
    ]
    # Agrupar por MAT_AVERIADO
    grupo_origen = df_filtrado.groupby("MAT_AVERIADO").agg(
        N_INCIDENCIAS=("CODIGO", "count")
    ).reset_index().sort_values(by="N_INCIDENCIAS", ascending=False)
    # Crear gráfico
    ax = axs[i]
    sns.barplot(data=grupo_origen, x="MAT_AVERIADO", y="N_INCIDENCIAS",
palette="pastel", ax=ax)
    ax.set_title(f"Año {año}", fontsize=13)
    ax.set_xlabel("MAT_AVERIADO")
    if i == 0:
        ax.set_ylabel("N° de Incidencias")
    else:
        ax.set_ylabel("")
    ax.tick_params(axis='x', rotation=45)
    ax.grid(False)

    # Etiquetas encima de las barras
    for p in ax.patches:
        height = p.get_height()
        if height > 0:
            ax.annotate(f'{int(height):,}',
                (p.get_x() + p.get_width() / 2., height),
                ha='center', va='bottom', fontsize=8)

# Título general
plt.suptitle("Número de Incidencias TIPO 202 por MAT_AVERIADO separadas en años
(CLASE 2, Tensión MEDIA, sin reenganches)", fontsize=16)
plt.tight_layout(rect=[0, 0.03, 1, 0.92])
plt.show()
```

**# CREAR TABLA CON LAS 20 PALABRAS MAS REPETIDAS EN EL CAMPO DESCRIPCION PARA INCIDENCIAS CLASE 2, MT Y TIPO 202, ELIMINANDO LAS SIN TIEPI**

```
import pandas as pd
from collections import Counter
import re
from IPython.display import display

# stop words , palabras comunes que no quiero q salgan
stopwords = {
    "de", "la", "y", "en", "el", "los", "por", "con", "una", "del", "se", "las",
    "para", "un", "al", "sin", "que", "como", "a", "es", "su", "no", "lo", "o",
    "más", "ha", "ya", "le", "sus", "entre", "también", "pero", "fue", "tras"
}

# 1. Filtrar el DataFrame según tus condiciones
df_clase2_m_202_st = df[
    (df["CLASE"] == "2") &
    (df["NIV_TENS"] == "M") &
    (df["TIPO"] == "202") &
    (df["SIN_TIEPI"] != "S")
]

# 2. Concatenar todas las descripciones
texto_completo = " ".join(df_clase2_m_202_st["DESCRIPCION"].dropna().astype(str))

# 3. Tokenizar, pasar a minúsculas y filtrar stopwords
palabras = [
    palabra for palabra in re.findall(r'\b\w+\b', texto_completo.lower())
    if palabra not in stopwords
]

# 4. Contar frecuencia
contador = Counter(palabras)

# 5. Convertir a DataFrame y ordenar
df_palabras = pd.DataFrame(contador.items(), columns=["PALABRA", "FRECUENCIA"])
df_palabras = df_palabras.sort_values(by="FRECUENCIA",
ascending=False).reset_index(drop=True)

# 6. Mostrar las 20 palabras más frecuentes
display(df_palabras.head(60))
```

**# GRAFICO CON LAS 20 PALABRAS MAS REPETIDAS EN EL CAMPO DESCRIPCION PARA INCIDENCIAS CLASE 2, MT Y TIPO 202, ELIMINANDO LAS SIN TIEPI**

```
top_palabras = df_palabras.head(20)
# Crear gráfica
plt.figure(figsize=(10, 6))
plt.bar(top_palabras["PALABRA"], top_palabras["FRECUENCIA"])
plt.xlabel('Palabras')
plt.ylabel('Frecuencia')
plt.title('Palabras más frecuentes en DESCRIPCION (TIPO = 202 Y Tensión M)')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```

**GRAFICO DEL NUMERO DE INCIDENCIAS CLASE=2 Y MT Y TIPO=406 POR INS\_ORIGEN TRAS ELIMINAR LAS SIN TIEPI (REENGANCHES)**

```
import seaborn as sns
import matplotlib.pyplot as plt

# Normalización previa si no se ha hecho
for col in ["CLASE", "NIV_TENS", "TIPO", "SIN_TIEPI"]:
    df[col] = df[col].astype(str).str.strip()

df_clase2_m_406_st = df[
    (df["CLASE"] == "2") &
    (df["NIV_TENS"] == "M") &
    (df["TIPO"] == "406") &
    (df["SIN_TIEPI"] != "S")
]

# Agrupar por INS_ORIGEN y contar incidencias
grupo_origen = df_clase2_m_406_st.groupby("INS_ORIGEN").agg(
    N_INCIDENCIAS=("CODIGO", "count")
).reset_index().sort_values(by="N_INCIDENCIAS", ascending=False)

# Gráfico de barras
plt.figure(figsize=(12, 6))
ax = sns.barplot(data=grupo_origen, x="INS_ORIGEN", y="N_INCIDENCIAS",
palette="pastel")

plt.title("Número de Incidencias Tipo 406 por INS_ORIGEN (CLASE 2, Tensión MEDIA,
sin reenganches)", fontsize=14)
plt.xlabel("INS_ORIGEN")
plt.ylabel("N° de Incidencias")
plt.grid(False)

# Etiquetas encima de las barras
for p in ax.patches:
    height = p.get_height()
    if height > 0:
        ax.annotate(f'{int(height):,}',
                    (p.get_x() + p.get_width()/2., height),
                    ha='center', va='bottom', fontsize=9)

plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```

**TABLA DEL NUMERO DE INCIDENCIAS CLASE=2 Y MT Y TIPO=406 POR INS\_ORIGEN TRAS ELIMINAR LAS SIN TIEPI (REENGANCHES)**

```
import pandas as pd
from tabulate import tabulate

# Seleccionar solo columnas relevantes para agrupar
df_subset = df_clase2_m_406_st[["INS_ORIGEN", "DESC_ORIGEN", "CODIGO"]]

# Agrupar por INS_ORIGEN y DESC_ORIGEN y contar incidencias
```

```
grupo_origen = df_subset.groupby(["INS_ORIGEN", "DESC_ORIGEN"]).agg({
    "CODIGO": "count"
}).rename(columns={"CODIGO": "N_INCIDENCIAS"}).reset_index()

# Ordenar por número de incidencias descendente
grupo_origen = grupo_origen.sort_values(by="N_INCIDENCIAS", ascending=False)

# Mostrar tabla en consola (formato bonito)
# Mostrar tabla
import pandas as pd
from IPython.display import display

display(grupo_origen)
```

**GRAFICO DE EVOLUCION DEL NUMERO DE INCIDENCIAS CLASE=2 Y MT Y TIPO=406 POR INS\_ORIGEN Y POR AÑO TRAS ELIMINAR LAS SIN TIEPI (REENGANCHES)**

```
import seaborn as sns
import matplotlib.pyplot as plt

# Normalizar columnas clave
for col in ["CLASE", "NIV_TENS", "TIPO", "SIN_TIEPI"]:
    df[col] = df[col].astype(str).str.strip()

# Crear columna 'AÑO' si no existe
if "AÑO" not in df.columns:
    if "FECHA" in df.columns:
        df["FECHA"] = pd.to_datetime(df["FECHA"])
        df["AÑO"] = df["FECHA"].dt.year
    else:
        raise ValueError("No se encuentra la columna AÑO ni FECHA para segmentar los datos por año.")

# Lista de años que quieres graficar (incluyendo 2025)
años = [2022, 2023, 2024, 2025]

# Crear subplots por año
fig, axs = plt.subplots(1, 4, figsize=(26, 6), sharey=True)

for i, año in enumerate(años):
    # Filtrar por condiciones + año actual
    df_filtrado = df[
        (df["CLASE"] == "2") &
        (df["NIV_TENS"] == "M") &
        (df["TIPO"] == "406") &
        (df["SIN_TIEPI"] != "S") &
        (df["AÑO"] == año)
    ]

    # Agrupar por INS_ORIGEN
    grupo_origen = df_filtrado.groupby("INS_ORIGEN").agg(
        N_INCIDENCIAS=("CODIGO", "count")
    ).reset_index().sort_values(by="N_INCIDENCIAS", ascending=False)
```

```
# Crear gráfico de barras
ax = axs[i]
sns.barplot(data=grupo_origen, x="INS_ORIGEN", y="N_INCIDENCIAS",
palette="pastel", ax=ax)
ax.set_title(f"Año {año}", fontsize=13)
ax.set_xlabel("INS_ORIGEN")
if i == 0:
    ax.set_ylabel("N° de Incidencias")
else:
    ax.set_ylabel("")
ax.tick_params(axis='x', rotation=45)
ax.grid(False)

# Etiquetas encima de las barras
for p in ax.patches:
    height = p.get_height()
    if height > 0:
        ax.annotate(f'{int(height):,}',
                    (p.get_x() + p.get_width()/2., height),
                    ha='center', va='bottom', fontsize=8)

# Título general
plt.suptitle("Número de Incidencias TIPO 406 por INS_ORIGEN (CLASE 2, Tensión
MEDIA, sin reenganches)", fontsize=16)
plt.tight_layout(rect=[0, 0.03, 1, 0.92])
plt.show()
```

**GRAFICO DEL NUMERO DE INCIDENCIAS CLASE=2 Y MT Y TIPO=406 POR MAT\_AVERIADO TRAS ELIMINAR LAS SIN TIEPI (REENGANCHES)**

```
import seaborn as sns
import matplotlib.pyplot as plt

# Agrupar por INS_ORIGEN y contar
grupo_origen = df_clase2_m_406_st.groupby("MAT_AVERIADO").agg({
    "CODIGO": "count"
}).rename(columns={"CODIGO": "N_INCIDENCIAS"}).reset_index()

# Ordenar por número de incidencias
grupo_origen = grupo_origen.sort_values(by="N_INCIDENCIAS", ascending=False)

# Gráfico
plt.figure(figsize=(12,6))
ax = sns.barplot(data=grupo_origen, x="MAT_AVERIADO", y="N_INCIDENCIAS",
palette="pastel")

plt.title("Número de Incidencias Tipo 406 por MAT_AVERIADO (CLASE 2, Tensión
MEDIA , sin reenganchos)", fontsize=14)
plt.xlabel("MAT_AVERIADO")
plt.ylabel("N° de Incidencias")
plt.grid(False)
```

```
# Etiquetas encima de las barras
for p in ax.patches:
    height = p.get_height()
    if height > 0:
        ax.annotate(f'{int(height):,}',
                    (p.get_x() + p.get_width()/2., height),
                    ha='center', va='bottom', fontsize=9)

plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```

**TABLA DEL NUMERO DE INCIDENCIAS CLASE=2 Y MT Y TIPO=406 POR MAT\_AVERIADO TRAS ELIMINAR LAS SIN TIEPI (REENGANCHES)**

```
import pandas as pd
from tabulate import tabulate

# Seleccionar solo columnas relevantes para agrupar
df_subset = df_clase2_m_406_st[["MAT_AVERIADO", "DESC_MAT_AVER", "CODIGO"]]

grupo_origen = df_subset.groupby(["MAT_AVERIADO", "DESC_MAT_AVER"]).agg({
    "CODIGO": "count"
}).rename(columns={"CODIGO": "N_INCIDENCIAS"}).reset_index()

# Ordenar por número de incidencias descendente
grupo_origen = grupo_origen.sort_values(by="N_INCIDENCIAS", ascending=False)

# Mostrar tabla en consola (formato bonito)
# Mostrar tabla
import pandas as pd
from IPython.display import display

display(grupo_origen)
```

**GRAFICO DE EVOLUCION DEL NUMERO DE INCIDENCIAS CLASE=2 Y MT Y TIPO=406 POR MAT\_AVERIADO TRAS ELIMINAR LAS SIN TIEPI (REENGANCHES)**

```
import seaborn as sns
import matplotlib.pyplot as plt

# Asegurar que las columnas están limpias
for col in ["CLASE", "NIV_TENS", "TIPO", "SIN_TIEPI"]:
    df[col] = df[col].astype(str).str.strip()

# Crear columna AÑO si no existe
if "AÑO" not in df.columns:
    if "FECHA" in df.columns:
        df["FECHA"] = pd.to_datetime(df["FECHA"])
        df["AÑO"] = df["FECHA"].dt.year
    else:
        raise ValueError("No se encuentra columna AÑO ni FECHA para segmentar por año.")

# Lista de años a graficar (incluyendo 2025)
```

```

años = [2022, 2023, 2024, 2025]

# Crear subplots
fig, axs = plt.subplots(1, 4, figsize=(26, 6), sharey=True)

for i, año in enumerate(años):
    # Filtrar por condiciones y año
    df_filtrado = df[
        (df["CLASE"] == "2") &
        (df["NIV_TENS"] == "M") &
        (df["TIPO"] == "406") &
        (df["SIN_TIEPI"] != "S") &
        (df["AÑO"] == año)
    ]
    # Agrupar por MAT_AVERIADO
    grupo_origen = df_filtrado.groupby("MAT_AVERIADO").agg(
        N_INCIDENCIAS=("CODIGO", "count")
    ).reset_index().sort_values(by="N_INCIDENCIAS", ascending=False)
    # Crear gráfico
    ax = axs[i]
    sns.barplot(data=grupo_origen, x="MAT_AVERIADO", y="N_INCIDENCIAS",
palette="pastel", ax=ax)
    ax.set_title(f"Año {año}", fontsize=13)
    ax.set_xlabel("MAT_AVERIADO")
    if i == 0:
        ax.set_ylabel("N° de Incidencias")
    else:
        ax.set_ylabel("")
    ax.tick_params(axis='x', rotation=45)
    ax.grid(False)

    # Etiquetas encima de las barras
    for p in ax.patches:
        height = p.get_height()
        if height > 0:
            ax.annotate(f'{int(height):,}',
                (p.get_x() + p.get_width() / 2., height),
                ha='center', va='bottom', fontsize=8)

# Título general
plt.suptitle("Número de Incidencias TIPO 406 por MAT_AVERIADO separadas en años
(CLASE 2, Tensión MEDIA, sin reenganches)", fontsize=16)
plt.tight_layout(rect=[0, 0.03, 1, 0.92])
plt.show()

```

#### NIVEL DE TESIION = ALTA TENSION

##### GRAFICA DEL NUMERO DE INCIDENCIAS CLASE=2 Y AT POR TIPO

```

import seaborn as sns
import matplotlib.pyplot as plt
# Filtrar por CLASE 2 y NIV_TENS = A
df_clase2_a = df[(df["CLASE"] == "2") & (df["NIV_TENS"] == "A")]
# Agrupar por TIPO
grupo_tipo = df_clase2_a.groupby("TIPO").agg({
    "CODIGO": "count"
}).rename(columns={"CODIGO": "N_INCIDENCIAS"}).reset_index()

```

```
# Ordenar de mayor a menor
grupo_tipo = grupo_tipo.sort_values(by="N_INCIDENCIAS", ascending=False)

# Crear gráfico
plt.figure(figsize=(12,6))
ax = sns.barplot(data=grupo_tipo, x="TIPO", y="N_INCIDENCIAS", palette="pastel")

plt.title("Número de Incidencias por TIPO (CLASE 2, Nivel de Tensión ALTA)",
fontsize=14)
plt.xlabel("TIPO")
plt.ylabel("N° de Incidencias")
plt.grid(False)
# Etiquetas encima de las barras
for p in ax.patches:
    height = p.get_height()
    if height > 0:
        ax.annotate(f'{int(height):,}',
                    (p.get_x() + p.get_width()/2., height),
                    ha='center', va='bottom', fontsize=9)
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```

#### **TABLA DEL NUMERO DE INCIDENCIAS CLASE=2 Y AT POR TIPO**

```
# Agrupar por TIPO y contar incidencias
grupo_tipo = (
    df_clase2_a
    .groupby("TIPO")
    .agg(N_INCIDENCIAS=("CODIGO", "count"))
    .reset_index()
)

# Obtener descripciones únicas
tipos_descripciones = df[["TIPO", "DES_TIPO"]].drop_duplicates()

# Unir con descripciones
grupo_tipo = grupo_tipo.merge(tipos_descripciones, on="TIPO", how="left")

# Ordenar de mayor a menor
grupo_tipo = grupo_tipo.sort_values(by="N_INCIDENCIAS",
ascending=False).reset_index(drop=True)

# Mostrar como tabla

# Formatear y mostrar
grupo_tipo["N_INCIDENCIAS"] = grupo_tipo["N_INCIDENCIAS"].map("{:,}".format)
display(grupo_tipo[["TIPO", "DES_TIPO", "N_INCIDENCIAS"]])
```

#### **GRAFICA DE LA EVOLUCION DEL NUMERO DE INCIDENCIAS CLASE=2 Y AT POR TIPO Y POR AÑO.**

```
import seaborn as sns
```

```
import matplotlib.pyplot as plt

# Agrupación
grupo_tipo_año = df_clase2_a.groupby(["AÑO", "TIPO"]).agg({
    "CODIGO": "count"
}).rename(columns={"CODIGO": "N_INCIDENCIAS"}).reset_index()

# Gráficos por año (ahora incluyendo 2025)
años = [2022, 2023, 2024, 2025]
fig, axes = plt.subplots(1, 4, figsize=(24, 6), sharey=True)

for i, año in enumerate(años):
    datos = grupo_tipo_año[grupo_tipo_año["AÑO"] ==
año].sort_values(by="N_INCIDENCIAS", ascending=False)

    sns.barplot(data=datos, x="TIPO", y="N_INCIDENCIAS", ax=axes[i],
palette="pastel")

    axes[i].set_title(f"Año {año}")
    axes[i].set_xlabel("TIPO")
    if i == 0:
        axes[i].set_ylabel("Nº de Incidencias")
    else:
        axes[i].set_ylabel("")

    axes[i].grid(False)
    axes[i].tick_params(axis='x', rotation=45)

    for p in axes[i].patches:
        height = p.get_height()
        if height > 0:
            axes[i].annotate(f'{int(height):,}',
                (p.get_x() + p.get_width() / 2., height),
                ha='center', va='bottom', fontsize=9)

fig.suptitle("Incidencias por TIPO (CLASE 2, Nivel de Tensión A) por Año",
fontsize=16, y=1.02)
plt.tight_layout()
plt.show()
```

## ESTUDIO DE LAS SU

### GRAFICA DEL NUMERO DE INCIDENCIAS CLASE=2 Y NIVEL DE TENSION= SU (SUMINISTRO UNICO) POR TIPO

```
import seaborn as sns
import matplotlib.pyplot as plt

# Filtrar CLASE 2 y Nivel de Tensión S
df_clase2_s = df[(df["CLASE"] == "2") & (df["NIV_TENS"] == "S")]
# Agrupar por TIPO y contar incidencias
grupo_tipo = (
    df_clase2_s
```

```

.groupby("TIPO")
.agg(N_INCIDENCIAS=("CODIGO", "count"))
.reset_index()
.sort_values(by="N_INCIDENCIAS", ascending=False) # ordenar de mayor a menor
)

# Gráfico
plt.figure(figsize=(12, 6))
ax = sns.barplot(data=grupo_tipo, x="TIPO", y="N_INCIDENCIAS", palette="pastel")

plt.title("Número de Incidencias por TIPO (CLASE 2, Nivel de Tensión S)",
fontsize=14)
plt.xlabel("TIPO")
plt.ylabel("Nº de Incidencias")
plt.grid(False)

# Etiquetas encima de las barras
for p in ax.patches:
    height = p.get_height()
    if height > 0:
        ax.annotate(f'{int(height):,}',
                    (p.get_x() + p.get_width()/2., height),
                    ha='center', va='bottom', fontsize=9)

plt.xticks(rotation=45)
plt.tight_layout()
plt.show()

```

**TABLA DEL NUMERO DE INCIDENCIAS CLASE=2 Y NIVEL DE TENSION=SU (SUMINISTRO UNICO) POR TIPO**

```

# Filtrar CLASE 2 y Nivel de Tensión S
df_clase2_s = df[(df["CLASE"] == "2") & (df["NIV_TENS"] == "S")]

# Agrupar por TIPO y contar incidencias
grupo_tipo = (
    df_clase2_s
    .groupby("TIPO")
    .agg(N_INCIDENCIAS=("CODIGO", "count"))
    .reset_index()
)

# Obtener descripciones únicas de tipo
tipos_descripciones = df[["TIPO", "DES_TIPO"]].drop_duplicates()

# Unir descripción al grupo
grupo_tipo = grupo_tipo.merge(tipos_descripciones, on="TIPO", how="left")

# Ordenar por N_INCIDENCIAS
grupo_tipo = grupo_tipo.sort_values(by="N_INCIDENCIAS",
ascending=False).reset_index(drop=True)

# Mostrar como tabla
import pandas as pd
from IPython.display import display

```

```
# Formatear y mostrar
grupo_tipo["N_INCIDENCIAS"] = grupo_tipo["N_INCIDENCIAS"].map("{:,.}").format)
display(grupo_tipo[["TIPO", "DES_TIPO", "N_INCIDENCIAS"]])
```

**GRAFICA DE LA EVOLUCION DEL NUMERO DE INCIDENCIAS CLASE=2 Y NIVEL DE TENSION= SU (SUMINISTRO UNICO) POR TIPO Y POR AÑOS**

```
import seaborn as sns
import matplotlib.pyplot as plt

grupo_tipo_año = df_clase2_s.groupby(["AÑO", "TIPO"]).agg({
    "CODIGO": "count"
}).rename(columns={"CODIGO": "N_INCIDENCIAS"}).reset_index()

años = [2022, 2023, 2024, 2025]
fig, axes = plt.subplots(1, 4, figsize=(24, 6), sharey=True)

for i, año in enumerate(años):
    datos = grupo_tipo_año[grupo_tipo_año["AÑO"] ==
año].sort_values(by="N_INCIDENCIAS", ascending=False)
    sns.barplot(data=datos, x="TIPO", y="N_INCIDENCIAS", ax=axes[i],
palette="pastel")

    axes[i].set_title(f"Año {año}")
    axes[i].set_xlabel("TIPO")
    if i == 0:
        axes[i].set_ylabel("N° de Incidencias")
    else:
        axes[i].set_ylabel("")

    axes[i].grid(False)
    axes[i].tick_params(axis='x', rotation=45)

    for p in axes[i].patches:
        height = p.get_height()
        if height > 0:
            axes[i].annotate(f'{int(height):,}',
                (p.get_x() + p.get_width()/2., height),
                ha='center', va='bottom', fontsize=9)

fig.suptitle("Incidencias por TIPO (CLASE 2, Nivel de Tensión S) por Año",
fontsize=16, y=1.02)
plt.tight_layout()
plt.show()
```

**GRAFICO DE LA EVOLUCION DEL NUMERO DE INCIDENCIAS TIPO 2XX Y 3XX (CAUSAS PROPIAS DE DISTRIBUCION), POR TIPO. SE INCLUYEN TODOS LOS NIVELES DE TENSION (NO SE ELIMINAN LAS NIVEL DE TENSION=T, REENGANCHES)**

```
import pandas as pd
```

```
import seaborn as sns
import matplotlib.pyplot as plt

# Filtrar TIPO que empiece por '2' o '3'
df_tipo_23 = df[df["TIPO"].str.startswith(("2", "3"))]

# Filtrar TIPO que empiece por '4'
df_tipo_4 = df[df["TIPO"].str.startswith("4")]

# Años a mostrar
años = [2022, 2023, 2024]

# ===== GRÁFICO TIPO 2XX y 3XX =====
grupo_tipo_año_23 = df_tipo_23.groupby(["AÑO", "TIPO"]).agg(
    N_INCIDENCIAS=("CODIGO", "count")
).reset_index()

fig, axes = plt.subplots(1, 3, figsize=(18, 6), sharey=True)

for i, año in enumerate(años):
    datos = grupo_tipo_año_23[grupo_tipo_año_23["AÑO"] == año].sort_values(
        by="N_INCIDENCIAS", ascending=False
    )

    sns.barplot(data=datos, x="TIPO", y="N_INCIDENCIAS", ax=axes[i],
palette="pastel")
    axes[i].set_title(f"Año {año}")
    axes[i].set_xlabel("TIPO")
    if i == 0:
        axes[i].set_ylabel("N° de Incidencias")
    else:
        axes[i].set_ylabel("")
    axes[i].grid(False)
    axes[i].tick_params(axis='x', rotation=45)

    for p in axes[i].patches:
        height = p.get_height()
        if height > 0:
            axes[i].annotate(f'{int(height):,}',
                (p.get_x() + p.get_width()/2., height),
                ha='center', va='bottom', fontsize=9)

fig.suptitle("Incidencias por TIPO (2XX y 3XX) por Año", fontsize=16, y=1.02)
plt.tight_layout()
plt.show()
```

**GRAFICO DE LA EVOLUCION DEL NUMERO DE INCIDENCIAS TIPO 4XX (TERCEROS) , POR TIPO. SE INCLUYEN TODOS LOS NIVELES DE TENSION (NO SE ELIMINAN LAS NIVEL DE TENSION=T, REENGANCHES)**

```
# ===== GRÁFICO TIPO 4XX =====
grupo_tipo_año_4 = df_tipo_4.groupby(["AÑO", "TIPO"]).agg(
    N_INCIDENCIAS=("CODIGO", "count")
).reset_index()
```

```
fig, axes = plt.subplots(1, 3, figsize=(18, 6), sharey=True)

for i, año in enumerate(años):
    datos = grupo_tipo_año_4[grupo_tipo_año_4["AÑO"] == año].sort_values(
        by="N_INCIDENCIAS", ascending=False
    )

    sns.barplot(data=datos, x="TIPO", y="N_INCIDENCIAS", ax=axes[i],
        palette="pastel")
    axes[i].set_title(f"Año {año}")
    axes[i].set_xlabel("TIPO")
    if i == 0:
        axes[i].set_ylabel("N° de Incidencias")
    else:
        axes[i].set_ylabel("")
    axes[i].grid(False)
    axes[i].tick_params(axis='x', rotation=45)

    for p in axes[i].patches:
        height = p.get_height()
        if height > 0:
            axes[i].annotate(f'{int(height):,}',
                (p.get_x() + p.get_width()/2., height),
                ha='center', va='bottom', fontsize=9)

fig.suptitle("Incidencias por TIPO 4XX por Año", fontsize=16, y=1.02)
plt.tight_layout()
plt.show()
```

### **GRAFICO DE LA EVOLUCION DEL NUMERO DE INCIDENCIAS TIPO 2XX Y 3XX (CAUSAS PROPIAS DE DISTRIBUCION) Y NIVEL DE TENSION=MT POR TIPO Y POR AÑO.**

Tipo 2xx y 3xx (causas propias de distribución) separado por MT

```
# Filtrar nivel de tensión M
df_tipo_23_m = df_tipo_23[df_tipo_23["NIV_TENS"] == "M"].copy()

# Agrupar por año y tipo
grupo_tipo_año_23_m = df_tipo_23_m.groupby(["AÑO", "TIPO"]).agg(
    N_INCIDENCIAS=("CODIGO", "count")
).reset_index()

# Graficar
años = [2022, 2023, 2024]
fig, axes = plt.subplots(1, 3, figsize=(18, 6), sharey=True)

for i, año in enumerate(años):
    datos = grupo_tipo_año_23_m[grupo_tipo_año_23_m["AÑO"] ==
año].sort_values(by="N_INCIDENCIAS", ascending=False)

    sns.barplot(data=datos, x="TIPO", y="N_INCIDENCIAS", ax=axes[i],
        palette="pastel")
    axes[i].set_title(f"Año {año}")
```

```

axes[i].set_xlabel("TIPO")
if i == 0:
    axes[i].set_ylabel("N° de Incidencias")
else:
    axes[i].set_ylabel("")
axes[i].grid(False)
axes[i].tick_params(axis='x', rotation=45)

for p in axes[i].patches:
    height = p.get_height()
    if height > 0:
        axes[i].annotate(f'{int(height):,}',
                        (p.get_x() + p.get_width()/2., height),
                        ha='center', va='bottom', fontsize=9)

fig.suptitle("Incidencias por TIPO (2XX y 3XX) con NIV_TENS = M por Año",
            fontsize=16, y=1.02)
plt.tight_layout()
plt.show()

```

### **GRAFICO DE LA EVOLUCION DEL NUMERO DE INCIDENCIAS TIPO 2XX Y 3XX (CAUSAS PROPIAS DE DISTRIBUCION) Y NIVEL DE TENSION=BT POR TIPO.**

Tipo 2xx y 3xx (causas propias de distribución) separado por BT

```

# Filtrar nivel de tensión B
df_tipo_23_b = df_tipo_23[df_tipo_23["NIV_TENS"] == "B"].copy()

# Agrupar por año y tipo
grupo_tipo_año_23_b = df_tipo_23_b.groupby(["AÑO", "TIPO"]).agg(
    N_INCIDENCIAS=("CODIGO", "count")
).reset_index()

# Graficar
fig, axes = plt.subplots(1, 3, figsize=(18, 6), sharey=True)

for i, año in enumerate(años):
    datos = grupo_tipo_año_23_b[grupo_tipo_año_23_b["AÑO"] ==
año].sort_values(by="N_INCIDENCIAS", ascending=False)

    sns.barplot(data=datos, x="TIPO", y="N_INCIDENCIAS", ax=axes[i],
palette="pastel")
    axes[i].set_title(f"Año {año}")
    axes[i].set_xlabel("TIPO")
    if i == 0:
        axes[i].set_ylabel("N° de Incidencias")
    else:
        axes[i].set_ylabel("")
    axes[i].grid(False)
    axes[i].tick_params(axis='x', rotation=45)

    for p in axes[i].patches:
        height = p.get_height()
        if height > 0:
            axes[i].annotate(f'{int(height):,}',

```

```

        (p.get_x() + p.get_width()/2., height),
        ha='center', va='bottom', fontsize=9)

fig.suptitle("Incidencias por TIPO (2XX y 3XX) con NIV_TENS = B por Año",
            fontsize=16, y=1.02)
plt.tight_layout()
plt.show()

```

### **GRAFICO DE LA EVOLUCION DEL NUMERO DE INCIDENCIAS TIPO 4XX (CAUSADA POR TERCEROS) Y NIVEL DE TENSION=MT POR TIPO Y POR AÑO.**

```

# Filtrar nivel de tensión M
df_tipo_4_m = df_tipo_4[df_tipo_4["NIV_TENS"] == "M"].copy()

# Agrupar por año y tipo
grupo_tipo_año_4_m = df_tipo_4_m.groupby(["AÑO", "TIPO"]).agg(
    N_INCIDENCIAS=("CODIGO", "count")
).reset_index()

# Graficar
años = [2022, 2023, 2024]
fig, axes = plt.subplots(1, 3, figsize=(18, 6), sharey=True)

for i, año in enumerate(años):
    datos = grupo_tipo_año_4_m[grupo_tipo_año_4_m["AÑO"] ==
año].sort_values(by="N_INCIDENCIAS", ascending=False)

    sns.barplot(data=datos, x="TIPO", y="N_INCIDENCIAS", ax=axes[i],
palette="pastel")
    axes[i].set_title(f"Año {año}")
    axes[i].set_xlabel("TIPO")
    if i == 0:
        axes[i].set_ylabel("N° de Incidencias")
    else:
        axes[i].set_ylabel("")
    axes[i].grid(False)
    axes[i].tick_params(axis='x', rotation=45)

    for p in axes[i].patches:
        height = p.get_height()
        if height > 0:
            axes[i].annotate(f'{int(height):,}',
                (p.get_x() + p.get_width()/2., height),
                ha='center', va='bottom', fontsize=9)

fig.suptitle("Incidencias por TIPO 4XX con NIV_TENS = M por Año", fontsize=16,
y=1.02)
plt.tight_layout()
plt.show()

```

### **GRAFICO DE LA EVOLUCION DEL NUMERO DE INCIDENCIAS TIPO 4XX (CAUSADA POR TERCEROS) Y NIVEL DE TENSION=BT POR TIPO Y POR AÑO.**

```

# Filtrar nivel de tensión B
df_tipo_4_b = df_tipo_4[df_tipo_4["NIV_TENS"] == "B"].copy()

```

```
# Agrupar por año y tipo
grupo_tipo_año_4_b = df_tipo_4_b.groupby(["AÑO", "TIPO"]).agg(
    N_INCIDENCIAS=("CODIGO", "count")
).reset_index()

# Graficar
fig, axes = plt.subplots(1, 3, figsize=(18, 6), sharey=True)

for i, año in enumerate(años):
    datos = grupo_tipo_año_4_b[grupo_tipo_año_4_b["AÑO"] ==
año].sort_values(by="N_INCIDENCIAS", ascending=False)

    sns.barplot(data=datos, x="TIPO", y="N_INCIDENCIAS", ax=axes[i],
palette="pastel")
    axes[i].set_title(f"Año {año}")
    axes[i].set_xlabel("TIPO")
    if i == 0:
        axes[i].set_ylabel("N° de Incidencias")
    else:
        axes[i].set_ylabel("")
    axes[i].grid(False)
    axes[i].tick_params(axis='x', rotation=45)

    for p in axes[i].patches:
        height = p.get_height()
        if height > 0:
            axes[i].annotate(f'{int(height):,}',
                (p.get_x() + p.get_width()/2., height),
                ha='center', va='bottom', fontsize=9)

fig.suptitle("Incidencias por TIPO 4XX con NIV_TENS = B por Año", fontsize=16,
y=1.02)
plt.tight_layout()
plt.show()
```

## GRAFICO DEL NUMERO DE INCIDENCIAS TIPO 303 Y NIVEL DE TENSION=S POR INS\_ORIGEN

1. Estudiamso los 303 de nivel tensión S por INS\_ORIGEN y por año
  - Distribución de ins origen para tipo 303 y niv tensión S por año eliminando las D, las G y las T ya q no nos ineresan

```
import seaborn as sns
import matplotlib.pyplot as plt

# Asegurarse de tener los datos ordenados antes de graficar
tabla_origen_sorted = tabla_origen.sort_values(by="N_INCIDENCIAS",
ascending=False)

plt.figure(figsize=(16, 6))
ax = sns.barplot(
    data=tabla_origen_sorted,
    x="INS_ORIGEN",
```

```
y="N_INCIDENCIAS",
palette="pastel"
)

plt.title("Incidencias TIPO 303 y NIV_TENS = 'S' por INS_ORIGEN", fontsize=16)
plt.xlabel("INS_ORIGEN", fontsize=12)
plt.ylabel("N° de Incidencias", fontsize=12)
plt.xticks(rotation=45, ha='right')

# Etiquetas encima de las barras - ¡esta es la parte corregida!
for bar in ax.patches:
    height = bar.get_height()
    ax.annotate(f'{int(height)}',
                xy=(bar.get_x() + bar.get_width() / 2, height),
                xytext=(0, 3), # separación respecto a la barra
                textcoords="offset points",
                ha='center', va='bottom', fontsize=9)

plt.tight_layout()
plt.show()
```

### **TABLA DEL NUMERO DE INCIDENCIAS TIPO 303 Y NIVEL DE TENSION=S POR INS\_ORIGEN**

```
import pandas as pd

# Asegurar limpieza
df["INS_ORIGEN"] = df["INS_ORIGEN"].astype(str).str.strip()
df["DESC_ORIGEN"] = df["DESC_ORIGEN"].astype(str).str.strip()
df["TIPO"] = pd.to_numeric(df["TIPO"], errors="coerce").astype("Int64")
df["NIV_TENS"] = df["NIV_TENS"].astype(str).str.strip().str.upper()

# Filtrar TIPO 303 y NIV_TENS = 'S'
df_303_s = df[(df["TIPO"] == 303) & (df["NIV_TENS"] == "S")].copy()

# Agrupación con descripción
tabla_origen = (
    df_303_s.groupby(["INS_ORIGEN", "DESC_ORIGEN"])
    .agg(N_INCIDENCIAS=("CODIGO", "count"))
    .reset_index()
    .sort_values(by="N_INCIDENCIAS", ascending=False)
)

# Mostrar tabla ordenada
print("Resumen de incidencias por INS_ORIGEN y descripción:")
display(tabla_origen)
```

### **GRAFICO DE LA EVOLUCION DEL NUMERO DE INCIDENCIAS TIPO 303 Y NIVEL DE TENSION=S POR MAT\_AVERIADO**

```
import pandas as pd
import seaborn as sns
```

```
import matplotlib.pyplot as plt

# Limpieza de columnas relevantes
df["MAT_AVERIADO"] = df["MAT_AVERIADO"].astype(str).str.strip()
df["DESC_MAT_AVER"] = df["DESC_MAT_AVER"].astype(str).str.strip()

# Filtro igual que antes
df_303_s = df[(df["TIPO"] == 303) & (df["NIV_TENS"] == "S")].copy()

# Agrupación por material averiado
tabla_material = (
    df_303_s.groupby(["MAT_AVERIADO", "DESC_MAT_AVER"])
    .agg(N_INCIDENCIAS=("CODIGO", "count"))
    .reset_index()
    .sort_values(by="N_INCIDENCIAS", ascending=False)
)

# Visualización con histograma
plt.figure(figsize=(16, 6))
ax = sns.barplot(
    data=tabla_material,
    x="MAT_AVERIADO",
    y="N_INCIDENCIAS",
    palette="pastel"
)

plt.title("Incidencias TIPO 303 y NIV_TENS = 'S' por MAT_AVERIADO", fontsize=16)
plt.xlabel("MAT_AVERIADO", fontsize=12)
plt.ylabel("N° de Incidencias", fontsize=12)
plt.xticks(rotation=45, ha='right')

# Etiquetas encima de las barras (corregidas)
for bar in ax.patches:
    height = bar.get_height()
    ax.annotate(f'{int(height)}',
                xy=(bar.get_x() + bar.get_width() / 2, height),
                xytext=(0, 3),
                textcoords="offset points",
                ha='center', va='bottom', fontsize=9)

plt.tight_layout()
plt.show()
```

### **TABLA DEL NUMERO DE INCIDENCIAS TIPO 303 Y NIVEL DE TENSION=S POR MAT\_AVERIADO**

```
# Filtrar TIPO 303 y NIV_TENS = 'S'
df_303_s = df[(df["TIPO"] == 303) & (df["NIV_TENS"] == "S")].copy()

# Agrupación con descripción
tabla_origen = (
    df_303_s.groupby(["MAT_AVERIADO", "DESC_MAT_AVER"])
    .agg(N_INCIDENCIAS=("CODIGO", "count"))
    .reset_index()
```

```
.sort_values(by="N_INCIDENCIAS", ascending=False)
)

display(tabla_origen)
```

### **TABLA DE LA EVOLUCION DEL NUMERO DE INCIDENCIAS, TIPO 303 Y NIVEL DE TENSION=S POR MAT\_AVERIADO Y POR AÑO**

```
import seaborn as sns
import matplotlib.pyplot as plt

# Agrupar por MAT_AVERIADO y AÑO
grupo_origen_año = df_303_s.groupby(["AÑO", "MAT_AVERIADO"]).agg({
    "CODIGO": "count"
}).rename(columns={"CODIGO": "N_INCIDENCIAS"}).reset_index()

# Añadir 2025
años = [2022, 2023, 2024, 2025]

# Crear 4 subplots
fig, axes = plt.subplots(1, 4, figsize=(26, 7), sharey=True)

for i, año in enumerate(años):
    datos = grupo_origen_año[grupo_origen_año["AÑO"] == año]
    sns.barplot(data=datos, x="MAT_AVERIADO", y="N_INCIDENCIAS", ax=axes[i],
        palette="pastel")

    axes[i].set_title(f"Año {año} - TIPO 303", fontsize=14)
    axes[i].set_xlabel("MAT_AVERIADO")
    if i == 0:
        axes[i].set_ylabel("N° de Incidencias")
    else:
        axes[i].set_ylabel("")

    axes[i].grid(False)
    axes[i].tick_params(axis='x', rotation=45)

    for p in axes[i].patches:
        height = p.get_height()
        if height > 0:
            axes[i].annotate(f'{int(height):,}',
                (p.get_x() + p.get_width()/2., height),
                ha='center', va='bottom', fontsize=8)

fig.suptitle("Distribución de MAT_AVERIADO para TIPO 303 y NIV_TENS 'S' por Año
(INS_ORIGEN filtrados)", fontsize=18, y=1.05)
plt.tight_layout()
plt.show()
```

### **GRAFICO DEL NUMERO DE INCIDENCIAS TIPO 317 Y NIVEL DE TENSION=S POR INS\_ORIGEN**

```
import matplotlib.pyplot as plt
import seaborn as sns
```

```
# Filtrar incidencias TIPO 317
df_317 = df[df["TIPO"] == 317].copy()

# Agrupar por INS_ORIGEN y contar incidencias
tabla_317_origen = (
    df_317.groupby("INS_ORIGEN")
    .agg(N_INCIDENCIAS=("CODIGO", "count"))
    .reset_index()
    .sort_values(by="N_INCIDENCIAS", ascending=False)
)

# Ordenar INS_ORIGEN como categoría para que seaborn respete el orden
tabla_317_origen["INS_ORIGEN"] = pd.Categorical(
    tabla_317_origen["INS_ORIGEN"],
    categories=tabla_317_origen["INS_ORIGEN"],
    ordered=True
)

# Graficar
plt.figure(figsize=(20, 6))
ax = sns.barplot(
    data=tabla_317_origen,
    x="INS_ORIGEN",
    y="N_INCIDENCIAS",
    palette="pastel"
)

# Añadir etiquetas numéricas encima de cada barra
for container in ax.containers:
    ax.bar_label(container, fmt='%d', label_type='edge', fontsize=9)

plt.title("Incidencias TIPO 317 por INS_ORIGEN")
plt.xlabel("INS_ORIGEN")
plt.ylabel("N° de Incidencias")
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```

### **TABLA DEL NUMERO DE INCIDENCIAS TIPO 317 Y NIVEL DE TENSION=S POR INS\_ORIGEN**

```
import pandas as pd

# Asegura limpieza de columnas relevantes
df["TIPO"] = pd.to_numeric(df["TIPO"], errors="coerce").astype("Int64")
df["INS_ORIGEN"] = df["INS_ORIGEN"].astype(str).str.strip()
df["DESC_ORIGEN"] = df["DESC_ORIGEN"].astype(str).str.strip()

# Filtra las incidencias tipo 317
df_317 = df[df["TIPO"] == 317].copy()

# Agrupa por material averiado y su descripción
tabla_mat_averiado = (
```

```
df_317.groupby(["INS_ORIGEN", "DESC_ORIGEN"])
    .agg(N_INCIDENCIAS=("CODIGO", "count"))
    .reset_index()
    .sort_values(by="N_INCIDENCIAS", ascending=False)
)

# Mostrar la tabla
print(tabla_mat_averiado)
```

2.

## **GRAFICO DEL NUMERO DE INCIDENCIAS TIPO 317 Y NIVEL DE TENSION=S POR INS\_ORIGEN Y POR AÑO ELIMINANDO LAS D,G Y T**

Estudiamos los 317 de nivel de tensión S por ins origen por año eliminando las D, G y T

```
import seaborn as sns
import matplotlib.pyplot as plt

# Agrupar por MAT_AVERIADO y AÑO
grupo_origen_año_df_317 = df_317.groupby(["AÑO", "INS_ORIGEN"]).agg({
    "CODIGO": "count"
}).rename(columns={"CODIGO": "N_INCIDENCIAS"}).reset_index()

# Añadir 2025
años = [2022, 2023, 2024, 2025]

# Crear 4 subplots
fig, axes = plt.subplots(1, 4, figsize=(26, 7), sharey=True)

for i, año in enumerate(años):
    datos = grupo_origen_año_df_317[grupo_origen_año_df_317["AÑO"] == año]
    sns.barplot(data=datos, x="INS_ORIGEN", y="N_INCIDENCIAS", ax=axes[i],
        palette="pastel")

    axes[i].set_title(f"Año {año} - TIPO 303", fontsize=14)
    axes[i].set_xlabel("INS_ORIGEN")
    if i == 0:
        axes[i].set_ylabel("Nº de Incidencias")
    else:
        axes[i].set_ylabel("")

    axes[i].grid(False)
    axes[i].tick_params(axis='x', rotation=45)

    for p in axes[i].patches:
        height = p.get_height()
        if height > 0:
            axes[i].annotate(f'{int(height):,}',
                (p.get_x() + p.get_width()/2., height),
                ha='center', va='bottom', fontsize=8)

fig.suptitle("Distribución de MAT_AVERIADO para TIPO 303 y NIV_TENS 'S' por Año
(INS_ORIGEN filtrados)", fontsize=18, y=1.05)
plt.tight_layout()
plt.show()
```

### GRAFICO DEL NUMERO DE INCIDENCIAS TIPO 317 Y NIVEL DE TENSION=S POR MAT\_AVERIADO

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

# Limpieza previa
df["TIPO"] = pd.to_numeric(df["TIPO"], errors="coerce").astype("Int64")
df["NIV_TENS"] = df["NIV_TENS"].astype(str).str.strip().str.upper()
df["MAT_AVERIADO"] = df["MAT_AVERIADO"].astype(str).str.strip()

# Filtro TIPO 317 y NIV_TENS = 'S'
df_317_s = df[(df["TIPO"] == 317) & (df["NIV_TENS"] == "S")]

# Verificación de datos
if df_317_s.empty:
    print("✗ No hay datos con TIPO = 317 y NIV_TENS = 'S'.")
else:
    # Agrupación total por MAT_AVERIADO
    resumen_mat = (
        df_317_s.groupby("MAT_AVERIADO")
        .size()
        .reset_index(name="N_INCIDENCIAS")
        .sort_values(by="N_INCIDENCIAS", ascending=False)
    )

    # Gráfico
    plt.figure(figsize=(14, 6))
    ax = sns.barplot(data=resumen_mat, x="MAT_AVERIADO", y="N_INCIDENCIAS",
palette="pastel")
    plt.title("Distribución de MAT_AVERIADO para TIPO = 317 y NIV_TENS = 'S'",
fontsize=16)
    plt.xlabel("MAT_AVERIADO", fontsize=12)
    plt.ylabel("N° de Incidencias", fontsize=12)
    plt.xticks(rotation=45, ha='right')

    # Etiquetas
    for p in ax.patches:
        height = p.get_height()
        if height > 0:
            ax.annotate(f'{int(height):,}',
                (p.get_x() + p.get_width() / 2., height),
                ha='center', va='bottom', fontsize=9)

    plt.tight_layout()
    plt.show()
```

### TABLA DEL NUMERO DE INCIDENCIAS TIPO 317 Y NIVEL DE TENSION=S POR MAT\_AVERIADO

```
# Agrupa por material averiado y su descripción
tabla_mat_averiado = (
    df_317.groupby(["MAT_AVERIADO", "DESC_MAT_AVER"])
    .agg(N_INCIDENCIAS=("CODIGO", "count"))
    .reset_index()
```

```
.sort_values(by="N_INCIDENCIAS", ascending=False)
)

# Mostrar la tabla
print("Resumen de incidencias TIPO 317 por MAT_AVERIADO y descripción:")
print(tabla_mat_averiado)

GRAFICO DE LA EVOLUCION DEL NUMERO DE INCIDENCIAS TIPO 317 Y NIVEL DE TENSION=S POR MAT_AVERIADO Y POR AÑO

import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

# Limpieza previa
df["TIPO"] = pd.to_numeric(df["TIPO"], errors="coerce").astype("Int64")
df["NIV_TENS"] = df["NIV_TENS"].astype(str).str.strip().str.upper()
df["MAT_AVERIADO"] = df["MAT_AVERIADO"].astype(str).str.strip()

# Asegurar columna AÑO si no existe
if "AÑO" not in df.columns:
    if "FECHA" in df.columns:
        df["FECHA"] = pd.to_datetime(df["FECHA"], errors="coerce")
        df["AÑO"] = df["FECHA"].dt.year
    else:
        raise ValueError("No se encuentra la columna 'AÑO' ni 'FECHA' en el DataFrame.")

# Filtro TIPO 317 y NIV_TENS = 'S'
df_317_s = df[(df["TIPO"] == 317) & (df["NIV_TENS"] == "S")]

# Verificación de datos
if df_317_s.empty:
    print("No hay datos con TIPO = 317 y NIV_TENS = 'S'.")
else:
    # Agrupar por AÑO y MAT_AVERIADO
    resumen_mat_año = (
        df_317_s.groupby(["AÑO", "MAT_AVERIADO"])
        .size()
        .reset_index(name="N_INCIDENCIAS")
        .sort_values(by=["AÑO", "N_INCIDENCIAS"], ascending=[True, False])
    )

    # Lista de años únicos
    años = sorted(resumen_mat_año["AÑO"].dropna().unique())

    # Crear subplots
    fig, axes = plt.subplots(1, len(años), figsize=(6 * len(años), 7),
sharey=True)

    # Asegurar que axes sea iterable
    if len(años) == 1:
        axes = [axes]

    for i, año in enumerate(años):
        datos = resumen_mat_año[resumen_mat_año["AÑO"] == año]
```

```

    if not datos.empty:
        sns.barplot(data=datos, x="MAT_AVERIADO", y="N_INCIDENCIAS",
ax=axes[i], palette="pastel")
        axes[i].tick_params(axis='x', rotation=45)
        for p in axes[i].patches:
            height = p.get_height()
            if height > 0:
                axes[i].annotate(f'{int(height):,}',
                                (p.get_x() + p.get_width() / 2., height),
                                ha='center', va='bottom', fontsize=9)
    else:
        axes[i].text(0.5, 0.5, "Sin datos", ha='center', va='center',
fontsize=12)
        axes[i].set_xticks([])
        axes[i].set_yticks([])

    axes[i].set_title(f"Año {año} - TIPO 317", fontsize=14)
    axes[i].set_xlabel("MAT_AVERIADO")
    axes[i].set_ylabel("N° de Incidencias" if i == 0 else "")
    axes[i].grid(False)

fig.suptitle("Distribución de MAT_AVERIADO para TIPO 317 y NIV_TENS = 'S' por
Año", fontsize=18, y=1.05)
plt.tight_layout()
plt.show()

```

### **GRAFICO DEL NUMERO DE INCIDENCIAS TIPO 416 Y NIVEL DE TENSION=S POR INS\_ORIGEN**

```

df = df1.copy()
print(df[(df["TIPO"] == 416) & (df["NIV_TENS"] == "S")].shape)
print(df["TIPO"].unique())
print(df["NIV_TENS"].unique())

df["TIPO"] = pd.to_numeric(df["TIPO"], errors="coerce").astype("Int64")
df["NIV_TENS"] = df["NIV_TENS"].astype(str).str.strip().str.upper()
df["INS_ORIGEN"] = df["INS_ORIGEN"].astype(str).str.strip()
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

# Limpieza previa
df["TIPO"] = pd.to_numeric(df["TIPO"], errors="coerce").astype("Int64")
df["NIV_TENS"] = df["NIV_TENS"].astype(str).str.strip().str.upper()
df["INS_ORIGEN"] = df["INS_ORIGEN"].astype(str).str.strip()

# Filtro TIPO 416 y NIV_TENS = 'S'
df_416_s = df[(df["TIPO"] == 416) & (df["NIV_TENS"] == "S")]

# Verificación por si el DataFrame está vacío
if df_416_s.empty:
    print("No hay datos con TIPO = 416 y NIV_TENS = 'S'. Revisa los valores.")
else:
    # Agrupamos por INS_ORIGEN

```

```

resumen_origen =
df_416_s.groupby("INS_ORIGEN").size().reset_index(name="N_INCIDENCIAS")
resumen_origen = resumen_origen.sort_values(by="N_INCIDENCIAS",
ascending=False)

# Gráfico
plt.figure(figsize=(14, 6))
ax = sns.barplot(data=resumen_origen, x="INS_ORIGEN", y="N_INCIDENCIAS",
palette="pastel")
plt.title("Distribución de INS_ORIGEN para TIPO = 416 y NIV_TENS = 'S'",
fontsize=16)
plt.xlabel("INS_ORIGEN", fontsize=12)
plt.ylabel("N° de Incidencias", fontsize=12)
plt.xticks(rotation=45, ha='right')

# Etiquetas
for p in ax.patches:
    height = p.get_height()
    if height > 0:
        ax.annotate(f'{int(height):,}',
                    (p.get_x() + p.get_width() / 2., height),
                    ha='center', va='bottom', fontsize=9)

plt.tight_layout()
plt.show()

```

### **TABLA DEL NUMERO DE INCIDENCIAS TIPO 416 Y NIVEL DE TENSION=S POR INS\_ORIGEN**

```

import pandas as pd

# Limpieza previa
df["TIPO"] = pd.to_numeric(df["TIPO"], errors="coerce").astype("Int64")
df["NIV_TENS"] = df["NIV_TENS"].astype(str).str.strip().str.upper()
df["INS_ORIGEN"] = df["INS_ORIGEN"].astype(str).str.strip()
df["DESC_ORIGEN"] = df["DESC_ORIGEN"].astype(str).str.strip()

# Filtro TIPO 416 y NIV_TENS = 'S'
df_416_s = df[(df["TIPO"] == 416) & (df["NIV_TENS"] == "S")]

# Verificación por si el DataFrame está vacío
if df_416_s.empty:
    print("No hay datos con TIPO = 416 y NIV_TENS = 'S'. Revisa los valores.")
else:
    # Agrupación con descripción
    tabla_origen_desc = (
        df_416_s.groupby(["INS_ORIGEN", "DESC_ORIGEN"])
        .agg(N_INCIDENCIAS=("CODIGO", "count"))
        .reset_index()
        .sort_values(by="N_INCIDENCIAS", ascending=False)
    )

    print("\nTabla ordenada de incidencias por INS_ORIGEN y descripción:")
    print(tabla_origen_desc.to_string(index=False))

```

### **GRAFICO DE LA EVOLUCION DEL NUMERO DE INCIDENCIAS TIPO 416 Y NIVEL DE TENSION=S POR INS\_ORIGEN Y POR AÑO ELIMINANDO LAS D, G, T**

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

# Limpieza previa
df["TIPO"] = pd.to_numeric(df["TIPO"], errors="coerce").astype("Int64")
df["NIV_TENS"] = df["NIV_TENS"].astype(str).str.strip().str.upper()
df["INS_ORIGEN"] = df["INS_ORIGEN"].astype(str).str.strip()

# Asegurar columna AÑO a partir de FECHA si no existe
if "AÑO" not in df.columns:
    if "FECHA" in df.columns:
        df["FECHA"] = pd.to_datetime(df["FECHA"], errors="coerce")
        df["AÑO"] = df["FECHA"].dt.year
    else:
        raise ValueError("No hay columna AÑO ni FECHA disponible en el DataFrame.")

# Filtro TIPO 416 y NIV_TENS = 'S'
df_416_s = df[(df["TIPO"] == 416) & (df["NIV_TENS"] == "S")]

# Eliminar INS_ORIGEN que empiezan por D, G o T
df_416_s = df_416_s[~df_416_s["INS_ORIGEN"].str.startswith(("D", "G", "T"))]

# Agrupar por AÑO e INS_ORIGEN
grupo_origen_año = df_416_s.groupby(["AÑO",
"INS_ORIGEN"]).size().reset_index(name="N_INCIDENCIAS")

# Lista de años a graficar (puedes modificarla si quieres más años)
años = sorted(grupo_origen_año["AÑO"].dropna().unique())

# Crear subplots dinámicamente
fig, axes = plt.subplots(1, len(años), figsize=(6 * len(años), 7), sharey=True)

# Si solo hay un año, convertir axes a lista para compatibilidad
if len(años) == 1:
    axes = [axes]

for i, año in enumerate(años):
    datos = grupo_origen_año[grupo_origen_año["AÑO"] == año]

    if not datos.empty:
        datos = datos.sort_values(by="N_INCIDENCIAS", ascending=False)
        sns.barplot(data=datos, x="INS_ORIGEN", y="N_INCIDENCIAS", ax=axes[i],
palette="pastel")
        axes[i].tick_params(axis='x', rotation=45)

    for p in axes[i].patches:
        height = p.get_height()
```

```

        if height > 0:
            axes[i].annotate(f'{int(height):,}',
                            (p.get_x() + p.get_width()/2., height),
                            ha='center', va='bottom', fontsize=9)
        else:
            axes[i].text(0.5, 0.5, "Sin datos", ha='center', va='center',
            fontsize=12)
            axes[i].set_xticks([])
            axes[i].set_yticks([])

            axes[i].set_title(f"Año {año} - TIPO 416 (sin D/G/T)", fontsize=14)
            axes[i].set_xlabel("INS_ORIGEN")
            axes[i].set_ylabel("N° de Incidencias" if i == 0 else "")
            axes[i].grid(False)

fig.suptitle("Distribución de INS_ORIGEN para TIPO 416 y NIV_TENS = 'S' por Año
(Excluyendo D/G/T)", fontsize=18, y=1.05)
plt.tight_layout()
plt.show()

```

### **GRAFICO DEL NUMERO DE INCIDENCIAS TIPO 416 Y NIVEL DE TENSION=S POR MAT\_AVERIADO**

```

import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

# Limpieza previa
df["TIPO"] = pd.to_numeric(df["TIPO"], errors="coerce").astype("Int64")
df["NIV_TENS"] = df["NIV_TENS"].astype(str).str.strip().str.upper()
df["MAT_AVERIADO"] = df["MAT_AVERIADO"].astype(str).str.strip()

# Filtro TIPO 416 y NIV_TENS = 'S'
df_416_s = df[(df["TIPO"] == 416) & (df["NIV_TENS"] == "S")]

# Verificación por si el DataFrame está vacío
if df_416_s.empty:
    print("No hay datos con TIPO = 416 y NIV_TENS = 'S'. Revisa los valores.")
else:
    # Agrupamos por MAT_AVERIADO
    resumen_mat =
df_416_s.groupby("MAT_AVERIADO").size().reset_index(name="N_INCIDENCIAS")
    resumen_mat = resumen_mat.sort_values(by="N_INCIDENCIAS", ascending=False)

    # Gráfico
    plt.figure(figsize=(14, 6))
    ax = sns.barplot(data=resumen_mat, x="MAT_AVERIADO", y="N_INCIDENCIAS",
    palette="pastel")
    plt.title("Distribución de MAT_AVERIADO para TIPO = 416 y NIV_TENS = 'S'",
    fontsize=16)
    plt.xlabel("MAT_AVERIADO", fontsize=12)
    plt.ylabel("N° de Incidencias", fontsize=12)
    plt.xticks(rotation=45, ha='right')

```

```
# Etiquetas
for p in ax.patches:
    height = p.get_height()
    if height > 0:
        ax.annotate(f'{int(height):,}',
                    (p.get_x() + p.get_width() / 2., height),
                    ha='center', va='bottom', fontsize=9)

plt.tight_layout()
plt.show()
```

### **TABLA DEL NUMERO DE INCIDENCIAS TIPO 416 Y NIVEL DE TENSION=S POR MAT\_AVERIADO**

```
import pandas as pd

# Limpieza previa
df["TIPO"] = pd.to_numeric(df["TIPO"], errors="coerce").astype("Int64")
df["NIV_TENS"] = df["NIV_TENS"].astype(str).str.strip().str.upper()
df["MAT_AVERIADO"] = df["MAT_AVERIADO"].astype(str).str.strip()
df["DESC_MAT_AVER"] = df["DESC_MAT_AVER"].astype(str).str.strip()

# Filtrar TIPO 416 y NIV_TENS = 'S'
df_416_s = df[(df["TIPO"] == 416) & (df["NIV_TENS"] == "S")]

# Verificación por si el DataFrame está vacío
if df_416_s.empty:
    print("No hay datos con TIPO = 416 y NIV_TENS = 'S'. Revisa los valores.")
else:
    # Agrupación con descripción
    tabla_mat_desc = (
        df_416_s.groupby(["MAT_AVERIADO", "DESC_MAT_AVER"])
        .agg(N_INCIDENCIAS=("CODIGO", "count"))
        .reset_index()
        .sort_values(by="N_INCIDENCIAS", ascending=False)
    )

    # Mostrar la tabla ordenada
    print("\nTabla ordenada de incidencias por MAT_AVERIADO y descripción:")
    print(tabla_mat_desc.to_string(index=False))
```

### **GRAFICO DE LA EVOLUCION DEL NUMERO DE INCIDENCIAS TIPO 416 Y NIVEL DE TENSION=S POR MAT\_AVERIADO Y POR AÑO**

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

# Limpieza previa
df["TIPO"] = pd.to_numeric(df["TIPO"], errors="coerce").astype("Int64")
df["NIV_TENS"] = df["NIV_TENS"].astype(str).str.strip().str.upper()
df["MAT_AVERIADO"] = df["MAT_AVERIADO"].astype(str).str.strip()

# Asegurar columna AÑO si no existe
if "AÑO" not in df.columns:
```

```

if "FECHA" in df.columns:
    df["FECHA"] = pd.to_datetime(df["FECHA"], errors="coerce")
    df["AÑO"] = df["FECHA"].dt.year
else:
    raise ValueError("No existe columna AÑO ni FECHA en el DataFrame")

# Filtro TIPO 416 y NIV_TENS = 'S'
df_416_s = df[(df["TIPO"] == 416) & (df["NIV_TENS"] == "S")]

# Verificación de datos
if df_416_s.empty:
    print("✗ No hay datos con TIPO = 416 y NIV_TENS = 'S'.")
else:
    # Agrupamos por AÑO y MAT_AVERIADO
    resumen_mat_año = (
        df_416_s.groupby(["AÑO", "MAT_AVERIADO"])
        .size()
        .reset_index(name="N_INCIDENCIAS")
        .sort_values(by=["AÑO", "N_INCIDENCIAS"], ascending=[True, False])
    )

    # Años únicos ordenados
    años = sorted(resumen_mat_año["AÑO"].dropna().unique())

    # Crear subplots
    fig, axes = plt.subplots(1, len(años), figsize=(6 * len(años), 7),
sharey=True)

    # Si solo hay un año, convertir axes a lista
    if len(años) == 1:
        axes = [axes]

    for i, año in enumerate(años):
        datos = resumen_mat_año[resumen_mat_año["AÑO"] == año]

        if not datos.empty:
            sns.barplot(data=datos, x="MAT_AVERIADO", y="N_INCIDENCIAS",
ax=axes[i], palette="pastel")
            axes[i].tick_params(axis='x', rotation=45)
            for p in axes[i].patches:
                height = p.get_height()
                if height > 0:
                    axes[i].annotate(f'{int(height):,}',
(p.get_x() + p.get_width() / 2., height),
ha='center', va='bottom', fontsize=9)
        else:
            axes[i].text(0.5, 0.5, "Sin datos", ha='center', va='center',
fontsize=12)
            axes[i].set_xticks([])
            axes[i].set_yticks([])

            axes[i].set_title(f"Año {año} - TIPO 416", fontsize=14)
            axes[i].set_xlabel("MAT_AVERIADO")
            axes[i].set_ylabel("N° de Incidencias" if i == 0 else "")

```

```
axes[i].grid(False)

fig.suptitle("Distribución de MAT_AVERIADO para TIPO 416 y NIV_TENS = 'S' por
Año", fontsize=18, y=1.05)
plt.tight_layout()
plt.show()
```

### **GRAFICO DEL NUMERO DE INCIDENCIAS TIPO 417 Y NIVEL DE TENSION=S POR INS\_ORIGEN**

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

# Limpieza básica
df["TIPO"] = pd.to_numeric(df["TIPO"], errors="coerce").astype("Int64")
df["NIV_TENS"] = df["NIV_TENS"].astype(str).str.strip().str.upper()
df["INS_ORIGEN"] = df["INS_ORIGEN"].astype(str).str.strip()

# Filtro TIPO 417 y NIV_TENS = 'S'
df_417_s = df[(df["TIPO"] == 417) & (df["NIV_TENS"] == "S")]

# Agrupar por INS_ORIGEN
resumen_origen = (
    df_417_s.groupby("INS_ORIGEN")
    .size()
    .reset_index(name="N_INCIDENCIAS")
    .sort_values(by="N_INCIDENCIAS", ascending=False)
)

# Verificación previa
if resumen_origen.empty:
    print("✘ No hay datos con TIPO = 417 y NIV_TENS = 'S'.")
else:
    # Gráfico
    plt.figure(figsize=(14, 6))
    ax = sns.barplot(data=resumen_origen, x="INS_ORIGEN", y="N_INCIDENCIAS",
palette="pastel")
    plt.title("Distribución de INS_ORIGEN para TIPO = 417 y NIV_TENS = 'S'",
    fontsize=16)
    plt.xlabel("INS_ORIGEN", fontsize=12)
    plt.ylabel("N° de Incidencias", fontsize=12)
    plt.xticks(rotation=45, ha='right')

    # Etiquetas en las barras
    for p in ax.patches:
        height = p.get_height()
        if height > 0:
            ax.annotate(f'{int(height)}',
                (p.get_x() + p.get_width() / 2., height),
                ha='center', va='bottom', fontsize=9)

    plt.tight_layout()
    plt.show()
```

### GRAFICO DEL NUMERO DE INCIDENCIAS TIPO 417 Y NIVEL DE TENSION=S POR INS\_ORIGEN Y POR AÑO ELIMINANDO LAS D, G Y T

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

# Limpieza de columnas necesarias
df_clase2_b["TIPO"] = pd.to_numeric(df_clase2_b["TIPO"],
errors="coerce").astype("Int64")
df_clase2_b["NIV_TENS"] =
df_clase2_b["NIV_TENS"].astype(str).str.strip().str.upper()
df_clase2_b["INS_ORIGEN"] = df_clase2_b["INS_ORIGEN"].astype(str).str.strip()

# Asegurar columna AÑO
if "AÑO" not in df_clase2_b.columns:
    if "FECHA" in df_clase2_b.columns:
        df_clase2_b["FECHA"] = pd.to_datetime(df_clase2_b["FECHA"],
errors="coerce")
        df_clase2_b["AÑO"] = df_clase2_b["FECHA"].dt.year
    else:
        raise ValueError("No se encuentra la columna 'AÑO' ni 'FECHA'.")

# Filtrar TIPO 417 y NIV_TENS = 'S'
df_417_s = df_clase2_b[(df_clase2_b["TIPO"] == 417) & (df_clase2_b["NIV_TENS"] ==
"S")]

# Filtrar INS_ORIGEN no deseados
ins_origen_excluir = ["D20", "D30", "D31", "D32", "D40", "D50", "G00", "T50",
"T60"]
df_417_s = df_417_s[~df_417_s["INS_ORIGEN"].isin(ins_origen_excluir)]

# Agrupar por AÑO e INS_ORIGEN
grupo_origen_año = (
    df_417_s.groupby(["AÑO", "INS_ORIGEN"])
        .size()
        .reset_index(name="N_INCIDENCIAS")
        .sort_values(by=["AÑO", "N_INCIDENCIAS"], ascending=[True, False])
)

# Años a graficar
años = [2022, 2023, 2024]

# Crear subplots
fig, axes = plt.subplots(1, len(años), figsize=(6 * len(años), 7), sharey=True)

for i, año in enumerate(años):
    datos = grupo_origen_año[grupo_origen_año["AÑO"] == año]

    if not datos.empty:
        sns.barplot(data=datos, x="INS_ORIGEN", y="N_INCIDENCIAS", ax=axes[i],
palette="pastel")
        axes[i].tick_params(axis='x', rotation=45)

    for p in axes[i].patches:
```

```

        height = p.get_height()
        if height > 0:
            axes[i].annotate(f'{int(height):,}',
                            (p.get_x() + p.get_width()/2., height),
                            ha='center', va='bottom', fontsize=9)
        else:
            axes[i].text(0.5, 0.5, "Sin datos", ha='center', va='center',
            fontsize=12)
            axes[i].set_xticks([])
            axes[i].set_yticks([])

            axes[i].set_title(f"Año {año} - TIPO 417 (Filtrado)", fontsize=14)
            axes[i].set_xlabel("INS_ORIGEN")
            axes[i].set_ylabel("N° de Incidencias" if i == 0 else "")
            axes[i].grid(False)

fig.suptitle("Distribución de INS_ORIGEN para TIPO 417 y NIV_TENS = 'S' por Año
(sin D/G/T)", fontsize=18, y=1.05)
plt.tight_layout()
plt.show()

```

### **GRAFICO DEL NUMERO DE INCIDENCIAS TIPO 417 Y NIVEL DE TENSION=S POR MAT\_AVERIADO**

```

# filtro las incidencias TIPO 417 y niv_tension = S

df_417_s = df_clase2_b[(df_clase2_b["TIPO"] == 417) & (df_clase2_b["NIV_TENS"] ==
"S")]

## agrupamos por Ins_Origen

resumen_origen = df_417_s.groupby("MAT_AVERIADO").size().reset_index(name =
"N_INCIDENCIAS")
resumen_origen = resumen_origen.sort_values(by="N_INCIDENCIAS", ascending=False)

## visualizamos
plt.figure(figsize=(14, 6))
sns.barplot(data=resumen_origen, x="MAT_AVERIADO", y="N_INCIDENCIAS",
palette="pastel")
plt.title("Distribución de mat_averiado para TIPO=417 y NIV_TENS = 'S'",
fontsize=16)
plt.xlabel("MAT_AVERIADO", fontsize=12)
plt.ylabel("N° de Incidencias", fontsize=12)
plt.xticks(rotation=45, ha='right')

for index, row in resumen_origen.iterrows():
    plt.text(index, row["N_INCIDENCIAS"] + 1, int(row["N_INCIDENCIAS"]),
ha='center', fontsize=9)

plt.tight_layout()
plt.show()

```

### **GRAFICO DE EVOLUCION DEL NUMERO DE INCIDENCIAS CALSE=2, NIVEL DE TENSION =BT POR TIPO Y POR AÑO**

```
import seaborn as sns
import matplotlib.pyplot as plt

grupo_tipo_año = df_clase2_b.groupby(["AÑO", "TIPO"]).agg({
    "CODIGO": "count"
}).rename(columns={"CODIGO": "N_INCIDENCIAS"}).reset_index()

años = [2022, 2023, 2024]
fig, axes = plt.subplots(1, 3, figsize=(18, 6), sharey=True)

for i, año in enumerate(años):
    datos = grupo_tipo_año[grupo_tipo_año["AÑO"] == año]
    sns.barplot(data=datos, x="TIPO", y="N_INCIDENCIAS", ax=axes[i],
        palette="pastel")

    axes[i].set_title(f"Año {año}")
    axes[i].set_xlabel("TIPO")
    if i == 0:
        axes[i].set_ylabel("N° de Incidencias")
    else:
        axes[i].set_ylabel("")

    axes[i].grid(False)
    axes[i].tick_params(axis='x', rotation=45)

    for p in axes[i].patches:
        height = p.get_height()
        if height > 0:
            axes[i].annotate(f'{int(height):,}',
                (p.get_x() + p.get_width()/2., height),
                ha='center', va='bottom', fontsize=9)

fig.suptitle("Incidencias por TIPO (CLASE 2, Nivel de Tensión Baja) por Año",
    fontsize=16, y=1.02)
plt.tight_layout()
plt.show()
```

### **GRAFICO DE EVOLUCION DEL NUMERO DE INCIDENCIAS PARA TIPO=417 Y NIVEL TENSION =s POR AÑO**

```
import seaborn as sns
import matplotlib.pyplot as plt

df_417_s = df_clase2_b[(df_clase2_b["TIPO"] == 417) & (df_clase2_b["NIV_TENS"] == "S")]

# Agrupar por INS_ORIGEN y AÑO
grupo_origen_año = df_417_s.groupby(["AÑO", "MAT_AVERIADO"]).agg({
```

```

"CODIGO": "count"
}).rename(columns={"CODIGO": "N_INCIDENCIAS"}).reset_index()

años = [2022, 2023, 2024]

fig, axes = plt.subplots(1, 3, figsize=(20, 7), sharey=True)

for i, año in enumerate(años):
    datos = grupo_origen_año[grupo_origen_año["AÑO"] == año]
    sns.barplot(data=datos, x="MAT_AVERIADO", y="N_INCIDENCIAS", ax=axes[i],
palette="pastel")

    axes[i].set_title(f"Año {año} - TIPO 303", fontsize=14)
    axes[i].set_xlabel("MAT_AVERIADO")
    if i == 0:
        axes[i].set_ylabel("N° de Incidencias")
    else:
        axes[i].set_ylabel("")

    axes[i].grid(False)
    axes[i].tick_params(axis='x', rotation=45)

    for p in axes[i].patches:
        height = p.get_height()
        if height > 0:
            axes[i].annotate(f'{int(height):,}',
                            (p.get_x() + p.get_width()/2., height),
                            ha='center', va='bottom', fontsize=8)

fig.suptitle("Distribución de MAT_AVERIADO para TIPO 317 y NIV_TENS 'S' por Año
(INS_ORIGEN filtrados)", fontsize=18, y=1.05)
plt.tight_layout()
plt.show()

```

## GRAFICO DE EVOLUCION DEL NUMERO DE INCIDENCIAS CLASE 2 POR AÑO

```

import seaborn as sns
import matplotlib.pyplot as plt

# Filtrar por CLASE 2
df_clase2 = df[(df["CLASE"] == "2")]

# Agrupar por AÑO
grupo_clase2_año = df_clase2.groupby("AÑO").agg({
    "CODIGO": "count"
}).rename(columns={"CODIGO": "N_INCIDENCIAS"}).reset_index()

# Crear el gráfico
plt.figure(figsize=(8, 5))
ax1 = sns.barplot(data=grupo_clase2_año, x="AÑO", y="N_INCIDENCIAS",
palette="pastel")

# Títulos y ejes

```

```
plt.title("Número de Incidencias por Año (CLASE 2)", fontsize=14)
plt.xlabel("Año", fontsize=12)
plt.ylabel("Nº de Incidencias", fontsize=12)
plt.grid(False)

# Etiquetas encima de las barras
for p in ax1.patches:
    height = p.get_height()
    if height > 0:
        ax1.annotate(f'{int(height):,}',
                    (p.get_x() + p.get_width() / 2., height),
                    ha='center', va='bottom', fontsize=10, fontweight='bold')

plt.tight_layout()
plt.show()
```

### **GRAFICO DE EVOLUCION DEL NUMERO DE INCIDENCIAS CLASE 2 Y NIV\_TENSION=MT POR AÑO**

```
import seaborn as sns
import matplotlib.pyplot as plt

#### CLASE 2 Y NIVEL DE TENSION = MEDIA , VEMOS LA INST ORIGEN QUE HAY POR AÑOS

# Filtrar por CLASE 2 y NIV_TENS B
df_clase2_b = df[(df["CLASE"] == "2") & (df["NIV_TENS"] == "M")]

# Agrupar por INS_ORIGEN y AÑO
grupo_origen_año = df_clase2_b.groupby(["AÑO", "INS_ORIGEN"]).agg({
    "CODIGO": "count"
}).rename(columns={"CODIGO": "N_INCIDENCIAS"}).reset_index()

# Años a graficar
años = [2022, 2023, 2024]
fig, axes = plt.subplots(1, 3, figsize=(24, 10), sharey=True) # tamaño más grande

import seaborn as sns
import matplotlib.pyplot as plt

for i, año in enumerate(años):
    datos = grupo_origen_año[grupo_origen_año["AÑO"] == año]
    sns.barplot(data=datos, x="INS_ORIGEN", y="N_INCIDENCIAS", ax=axes[i],
                palette="pastel")

    axes[i].set_title(f"Año {año}", fontsize=18)
    axes[i].set_xlabel("INS_ORIGEN", fontsize=14)
    if i == 0:
        axes[i].set_ylabel("Nº de Incidencias", fontsize=14)
    else:
        axes[i].set_ylabel("")

    axes[i].grid(False)
    axes[i].tick_params(axis='x', rotation=60, labelsize=11)
```

```

for p in axes[i].patches:
    height = p.get_height()
    if height > 0:
        axes[i].annotate(f'{int(height):,}',
                        (p.get_x() + p.get_width() / 2., height),
                        ha='center', va='bottom', fontsize=11)

fig.suptitle("Incidencias por INS_ORIGEN (CLASE 2, Nivel de Tensión MEDIA ) por
Año", fontsize=20, y=1.05)
plt.tight_layout()
plt.show()

```

### **GRAFICO DE EVOLUCION DEL NUMERO DE INCIDENCIAS CLASE 2 Y NIV\_TENSION=AT POR ORIGEN Y POR AÑOS**

```

import seaborn as sns
import matplotlib.pyplot as plt

##### CLASE 2 Y NIVEL DE TENSION = ALTA VEMOS LA INST ORIGEN QUE HAY POR AÑOS

# Filtrar por CLASE 2 y NIV_TENS B
df_clase2_b = df[(df["CLASE"] == "2") & (df["NIV_TENS"] == "A")]

# Agrupar por INS_ORIGEN y AÑO
grupo_origen_año = df_clase2_b.groupby(["AÑO", "INS_ORIGEN"]).agg({
    "CODIGO": "count"
}).rename(columns={"CODIGO": "N_INCIDENCIAS"}).reset_index()

# Años a graficar
años = [2022, 2023, 2024]
fig, axes = plt.subplots(1, 3, figsize=(24, 10), sharey=True) # tamaño más grande

import seaborn as sns
import matplotlib.pyplot as plt

for i, año in enumerate(años):
    datos = grupo_origen_año[grupo_origen_año["AÑO"] == año]
    sns.barplot(data=datos, x="INS_ORIGEN", y="N_INCIDENCIAS", ax=axes[i],
palette="pastel")

    axes[i].set_title(f"Año {año}", fontsize=18)
    axes[i].set_xlabel("INS_ORIGEN", fontsize=14)
    if i == 0:
        axes[i].set_ylabel("Nº de Incidencias", fontsize=14)
    else:
        axes[i].set_ylabel("")

    axes[i].grid(False)
    axes[i].tick_params(axis='x', rotation=60, labelsz=11)

```

```
for p in axes[i].patches:
    height = p.get_height()
    if height > 0:
        axes[i].annotate(f'{int(height):,}',
                        (p.get_x() + p.get_width() / 2., height),
                        ha='center', va='bottom', fontsize=11)

fig.suptitle("Incidencias por INS_ORIGEN (CLASE 2, Nivel de Tensión ALTA ) por
Año", fontsize=20, y=1.05)
plt.tight_layout()
plt.show()
```

### **GRAFICO DE EVOLUCION DEL NUMERO DE INCIDENCIAS CLASE 2 Y NIV\_TENSION=BT POR ORIGEN Y POR AÑOS**

```
import seaborn as sns
import matplotlib.pyplot as plt

# Filtrar por CLASE 2 y NIV_TENS = B (baja tensión)
df_clase2_b = df[(df["CLASE"] == "2") & (df["NIV_TENS"] == "B")]

# Agrupar por TIPO y AÑO
grupo_tipo_año = (
    df_clase2_b
    .groupby(["AÑO", "TIPO"])
    .agg({"CODIGO": "count"})
    .rename(columns={"CODIGO": "N_INCIDENCIAS"})
    .reset_index()
)

# Años a graficar
años = [2022, 2023, 2024]
fig, axes = plt.subplots(1, 3, figsize=(24, 10), sharey=True)

# Gráficos por año
for i, año in enumerate(años):
    datos = grupo_tipo_año[grupo_tipo_año["AÑO"] == año]

    sns.barplot(data=datos, x="TIPO", y="N_INCIDENCIAS", ax=axes[i],
                palette="pastel")

    axes[i].set_title(f"Año {año}", fontsize=18)
    axes[i].set_xlabel("TIPO", fontsize=14)

    if i == 0:
        axes[i].set_ylabel("N° de Incidencias", fontsize=14)
    else:
        axes[i].set_ylabel("")

    axes[i].tick_params(axis='x', rotation=60, labelsize=11)
    axes[i].grid(False)
```

```
# Etiquetas encima de las barras
for p in axes[i].patches:
    height = p.get_height()
    if height > 0:
        axes[i].annotate(f'{int(height):,}',
                        (p.get_x() + p.get_width() / 2., height),
                        ha='center', va='bottom', fontsize=11)

# Título principal
fig.suptitle("Incidencias por TIPO (CLASE 2, Nivel de Tensión BAJA) por Año",
            fontsize=20, y=1.05)
plt.tight_layout()
plt.show()
```

### **GRAFICO DE NUMERO DE INCIDENCIAS TIPO=302 Y NIV\_TENSION=BT Y POR AÑOS**

```
import seaborn as sns
import matplotlib.pyplot as plt

#### CLASE 2 Y NIVEL DE TENSION = BAJA , VEMOS LA INST ORIGEN QUE HAY POR AÑOS

# Filtrar por CLASE 2 y NIV_TENS B
df_clase2_b = df[(df["CLASE"] == "2") & (df["NIV_TENS"] == "B")]

# Agrupar por INS_ORIGEN y AÑO
grupo_origen_año = df_clase2_b.groupby(["AÑO", "INS_ORIGEN"]).agg({
    "CODIGO": "count"
}).rename(columns={"CODIGO": "N_INCIDENCIAS"}).reset_index()

# Años a graficar
años = [2022, 2023, 2024]
fig, axes = plt.subplots(1, 3, figsize=(24, 10), sharey=True) # tamaño más grande

import seaborn as sns
import matplotlib.pyplot as plt

for i, año in enumerate(años):
    datos = grupo_origen_año[grupo_origen_año["AÑO"] == año]
    sns.barplot(data=datos, x="INS_ORIGEN", y="N_INCIDENCIAS", ax=axes[i],
                palette="pastel")

    axes[i].set_title(f"Año {año}", fontsize=18)
    axes[i].set_xlabel("INS_ORIGEN", fontsize=14)
    if i == 0:
        axes[i].set_ylabel("N° de Incidencias", fontsize=14)
    else:
        axes[i].set_ylabel("")

    axes[i].grid(False)
    axes[i].tick_params(axis='x', rotation=60, labelsize=11)
```

```
for p in axes[i].patches:
    height = p.get_height()
    if height > 0:
        axes[i].annotate(f'{int(height):,}',
                        (p.get_x() + p.get_width() / 2., height),
                        ha='center', va='bottom', fontsize=11)

fig.suptitle("Incidencias por INS_ORIGEN (CLASE 2, Nivel de Tensión BAJA ) por
Año", fontsize=20, y=1.05)
plt.tight_layout()
plt.show()
```

### **GRAFICO DE NUMERO DE INCIDENCIAS TIPO=303 Y NIV\_TENSION=BT , CLASE =2 POR INS\_ORIGEN**

```
import matplotlib.pyplot as plt
import seaborn as sns

# Filtrar por TIPO 303 y NIV_TENS = B (baja tensión)
df_303_b = df_clase2_b[(df_clase2_b["TIPO"] == 303) & (df_clase2_b["NIV_TENS"] ==
"B")]

# Agrupar por INS_ORIGEN
resumen_origen = (
    df_303_b.groupby("INS_ORIGEN")
    .size()
    .reset_index(name="N_INCIDENCIAS")
    .sort_values(by="N_INCIDENCIAS", ascending=False)
)

# Visualización
plt.figure(figsize=(14, 6))
sns.barplot(data=resumen_origen, x="INS_ORIGEN", y="N_INCIDENCIAS",
palette="pastel")
plt.title("Distribución de INS_ORIGEN para TIPO = 303 y NIV_TENS = 'B'",
fontsize=16)
plt.xlabel("INS_ORIGEN", fontsize=12)
plt.ylabel("N° de Incidencias", fontsize=12)
plt.xticks(rotation=45, ha='right')

# Etiquetas encima de las barras
for index, row in resumen_origen.iterrows():
    plt.text(index, row["N_INCIDENCIAS"] + 1, int(row["N_INCIDENCIAS"]),
ha='center', fontsize=9)

plt.tight_layout()
plt.show()
```

### **GRAFICO DE NUMERO DE INCIDENCIAS TIPO=303 Y NIV\_TENSION=BT , CLASE =2 POR INS\_ORIGEN Y POR AÑO**

```
import seaborn as sns
```

```

import matplotlib.pyplot as plt

# Filtrar por CLASE 2, NIV_TENS = B y TIPO = 303
df_303_b = df[(df["CLASE"] == "2") & (df["NIV_TENS"] == "B") & (df["TIPO"] ==
303)]

# Agrupar por INS_ORIGEN y AÑO
grupo_303_origen_año = (
    df_303_b.groupby(["AÑO", "INS_ORIGEN"])
    .agg({"CODIGO": "count"})
    .rename(columns={"CODIGO": "N_INCIDENCIAS"})
    .reset_index()
)

# Años a graficar
años = [2022, 2023, 2024]
fig, axes = plt.subplots(1, 3, figsize=(24, 10), sharey=True)

# Gráfico de barras por año
for i, año in enumerate(años):
    datos = grupo_303_origen_año[grupo_303_origen_año["AÑO"] == año]

    sns.barplot(data=datos, x="INS_ORIGEN", y="N_INCIDENCIAS", ax=axes[i],
palette="pastel")

    axes[i].set_title(f"Año {año}", fontsize=18)
    axes[i].set_xlabel("INS_ORIGEN", fontsize=14)
    if i == 0:
        axes[i].set_ylabel("N° de Incidencias", fontsize=14)
    else:
        axes[i].set_ylabel("")

    axes[i].tick_params(axis='x', rotation=60, labelsz=11)
    axes[i].grid(False)

    for p in axes[i].patches:
        height = p.get_height()
        if height > 0:
            axes[i].annotate(f'{int(height):,}',
                (p.get_x() + p.get_width() / 2., height),
                ha='center', va='bottom', fontsize=11)

# Título global
fig.suptitle("Incidencias TIPO 303 por INS_ORIGEN (CLASE 2, Tensión BAJA) por
Año", fontsize=20, y=1.05)
plt.tight_layout()
plt.show()

```

### **GRAFICO DE NUMERO DE INCIDENCIAS TIPO=303 Y NIV\_TENSION=BT , CLASE =2 POR MAT\_AVERIADO**

```

import matplotlib.pyplot as plt
import seaborn as sns

# Filtrar por TIPO 303 y NIV_TENS = B (baja tensión)

```

```
df_303_b = df_clase2_b[(df_clase2_b["TIPO"] == 303) & (df_clase2_b["NIV_TENS"] == "B")]

# Agrupar por MAT_AVERIADO
resumen_mat = (
    df_303_b.groupby("MAT_AVERIADO")
    .size()
    .reset_index(name="N_INCIDENCIAS")
    .sort_values(by="N_INCIDENCIAS", ascending=False)
)

# Visualización
plt.figure(figsize=(14, 6))
sns.barplot(data=resumen_mat, x="MAT_AVERIADO", y="N_INCIDENCIAS",
            palette="pastel")
plt.title("Distribución de MAT_AVERIADO para TIPO = 303 y NIV_TENS = 'B'",
          fontsize=16)
plt.xlabel("MAT_AVERIADO", fontsize=12)
plt.ylabel("N° de Incidencias", fontsize=12)
plt.xticks(rotation=45, ha='right')

# Etiquetas encima de las barras
for index, row in resumen_mat.iterrows():
    plt.text(index, row["N_INCIDENCIAS"] + 1, int(row["N_INCIDENCIAS"]),
            ha='center', fontsize=9)

plt.tight_layout()
plt.show()
```

### **GRAFICO DE NUMERO DE INCIDENCIAS TIPO=303 Y NIV\_TENSION=BT , CLASE =2 POR MAT\_AVERIADO Y POR AÑO**

```
import seaborn as sns
import matplotlib.pyplot as plt

# Filtrar por CLASE 2, NIV_TENS = B y TIPO = 303
df_303_b = df[(df["CLASE"] == "2") & (df["NIV_TENS"] == "B") & (df["TIPO"] == 303)]

# Agrupar por MAT_AVERIADO y AÑO
grupo_303_mat_año = (
    df_303_b.groupby(["AÑO", "MAT_AVERIADO"])
    .agg({"CODIGO": "count"})
    .rename(columns={"CODIGO": "N_INCIDENCIAS"})
    .reset_index()
)

# Años a graficar
años = [2022, 2023, 2024]
fig, axes = plt.subplots(1, 3, figsize=(24, 10), sharey=True)

# Gráfico de barras por año
for i, año in enumerate(años):
    datos = grupo_303_mat_año[grupo_303_mat_año["AÑO"] == año]
```

```
sns.barplot(data=datos, x="MAT_AVERIADO", y="N_INCIDENCIAS", ax=axes[i],
palette="pastel")

axes[i].set_title(f"Año {año}", fontsize=18)
axes[i].set_xlabel("MAT_AVERIADO", fontsize=14)
if i == 0:
    axes[i].set_ylabel("N° de Incidencias", fontsize=14)
else:
    axes[i].set_ylabel("")

axes[i].tick_params(axis='x', rotation=60, labelsz=11)
axes[i].grid(False)

for p in axes[i].patches:
    height = p.get_height()
    if height > 0:
        axes[i].annotate(f'{int(height):,}',
                        (p.get_x() + p.get_width() / 2., height),
                        ha='center', va='bottom', fontsize=11)

# Título global
fig.suptitle("Incidencias TIPO 303 por MAT_AVERIADO (CLASE 2, Tensión BAJA) por
Año", fontsize=20, y=1.05)
plt.tight_layout()
plt.show()
```

### **GRAFICO DE NUMERO DE INCIDENCIAS TIPO=303 Y NIV\_TENSION=BT , CLASE =2 POR INST\_ORIGEN**

```
import matplotlib.pyplot as plt
import seaborn as sns

# Filtrar directamente desde df
df_202_b = df[(df["TIPO"] == 202) & (df["NIV_TENS"] == "B") & (df["CLASE"] ==
"2")]

# Agrupar por INS_ORIGEN
resumen_origen = (
    df_202_b.groupby("INS_ORIGEN")
    .size()
    .reset_index(name="N_INCIDENCIAS")
    .sort_values(by="N_INCIDENCIAS", ascending=False)
)

# Visualización
plt.figure(figsize=(14, 6))
sns.barplot(data=resumen_origen, x="INS_ORIGEN", y="N_INCIDENCIAS",
palette="pastel")
plt.title("Distribución de INS_ORIGEN para TIPO = 202 y NIV_TENS = 'B'",
fontsize=16)
plt.xlabel("INS_ORIGEN", fontsize=12)
plt.ylabel("N° de Incidencias", fontsize=12)
plt.xticks(rotation=45, ha='right')

# Etiquetas encima de las barras
```

```

for index, row in resumen_origen.iterrows():
    plt.text(index, row["N_INCIDENCIAS"] + 1, int(row["N_INCIDENCIAS"]),
             ha='center', fontsize=9)

plt.tight_layout()
plt.show()

GRAFICO DE NUMERO DE INCIDENCIAS TIPO=303 Y NIV_TENSION=BT , CLASE =2 POR INST_ORIGEN Y POR AÑO
import seaborn as sns
import matplotlib.pyplot as plt

# Filtrar por CLASE 2, NIV_TENS = B y TIPO = 303
df_202_b = df[(df["CLASE"] == "2") & (df["NIV_TENS"] == "B") & (df["TIPO"] ==
202)]

# Agrupar por INS_ORIGEN y AÑO
grupo_202_origen_año = (
    df_202_b.groupby(["AÑO", "INS_ORIGEN"])
    .agg({"CODIGO": "count"})
    .rename(columns={"CODIGO": "N_INCIDENCIAS"})
    .reset_index()
)

# Años a graficar
años = [2022, 2023, 2024]
fig, axes = plt.subplots(1, 3, figsize=(24, 10), sharey=True)

# Gráfico de barras por año
for i, año in enumerate(años):
    datos = grupo_202_origen_año[grupo_202_origen_año["AÑO"] == año]

    sns.barplot(data=datos, x="INS_ORIGEN", y="N_INCIDENCIAS", ax=axes[i],
palette="pastel")

    axes[i].set_title(f"Año {año}", fontsize=18)
    axes[i].set_xlabel("INS_ORIGEN", fontsize=14)
    if i == 0:
        axes[i].set_ylabel("N° de Incidencias", fontsize=14)
    else:
        axes[i].set_ylabel("")

    axes[i].tick_params(axis='x', rotation=60, labelsz=11)
    axes[i].grid(False)

    for p in axes[i].patches:
        height = p.get_height()
        if height > 0:
            axes[i].annotate(f'{int(height):,}',
                             (p.get_x() + p.get_width() / 2., height),
                             ha='center', va='bottom', fontsize=11)

# Título global
fig.suptitle("Incidencias TIPO 202 por INS_ORIGEN (CLASE 2, Tensión BAJA) por
Año", fontsize=20, y=1.05)

```

```
plt.tight_layout()  
plt.show()
```

### **GRAFICO DE NUMERO DE INCIDENCIAS TIPO=202 Y NIV\_TENSION=BT , CLASE =2 POR MAT\_AVERIADO**

```
import matplotlib.pyplot as plt  
import seaborn as sns  
  
df_202_b = df[(df["TIPO"] == 202) & (df["NIV_TENS"] == "B") & (df["CLASE"] == "2")]  
  
# Agrupar por MAT_AVERIADO  
resumen_mat = (  
    df_202_b.groupby("MAT_AVERIADO")  
        .size()  
        .reset_index(name="N_INCIDENCIAS")  
        .sort_values(by="N_INCIDENCIAS", ascending=False)  
)  
  
# Visualización  
plt.figure(figsize=(14, 6))  
ax = sns.barplot(data=resumen_mat, x="MAT_AVERIADO", y="N_INCIDENCIAS",  
                palette="pastel")  
plt.title("Distribución de MAT_AVERIADO para TIPO = 202 y NIV_TENS = 'B'",  
         fontsize=16)  
plt.xlabel("MAT_AVERIADO", fontsize=12)  
plt.ylabel("N° de Incidencias", fontsize=12)  
plt.xticks(rotation=45, ha='right')  
  
# Etiquetas encima de las barras  
for p in ax.patches:  
    height = p.get_height()  
    if height > 0:  
        ax.annotate(f'{int(height)}',  
                    (p.get_x() + p.get_width() / 2., height),  
                    ha='center', va='bottom', fontsize=9)  
  
plt.tight_layout()  
plt.show()
```

### **GRAFICO DE NUMERO DE INCIDENCIAS, CLASE =1 POR MAT\_AVERIADO Y BT**

```
import seaborn as sns  
import matplotlib.pyplot as plt  
  
df_clase2_b = df[(df["CLASE"] == "1") & (df["NIV_TENS"] == "B")]  
  
grupo_tipo = df_clase2_b.groupby("MAT_AVERIADO").agg(  
    "CODIGO": "count"  
).rename(columns={"CODIGO": "N_INCIDENCIAS"}).reset_index()  
  
plt.figure(figsize=(12,6))  
ax = sns.barplot(data=grupo_tipo, x="MAT_AVERIADO", y="N_INCIDENCIAS",  
                palette="pastel")
```

```
plt.title("Número de Incidencias por MAT_AVERIADO (CLASE 1, Nivel de Tensión BAJA)",
         fontsize=14)
plt.xlabel("MAT_AVERIADO")
plt.ylabel("Nº de Incidencias")
plt.grid(False)

# Etiquetas encima de las barras
for p in ax.patches:
    height = p.get_height()
    if height > 0:
        ax.annotate(f'{int(height):,}',
                    (p.get_x() + p.get_width()/2., height),
                    ha='center', va='bottom', fontsize=9)

plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```

### **GRAFICO DE NUMERO DE INCIDENCIAS, CLASE =1 POR MAT\_AVERIADO Y BT POR AÑO**

```
import seaborn as sns
import matplotlib.pyplot as plt

for i, año in enumerate(años):
    datos = grupo_origen_año[grupo_origen_año["AÑO"] == año]
    sns.barplot(data=datos, x="MAT_AVERIADO", y="N_INCIDENCIAS", ax=axes[i],
               palette="pastel")

    axes[i].set_title(f"Año {año}", fontsize=18)
    axes[i].set_xlabel("MAT_AVERIADO", fontsize=14)
    if i == 0:
        axes[i].set_ylabel("Nº de Incidencias", fontsize=14)
    else:
        axes[i].set_ylabel("")

    axes[i].grid(False)
    axes[i].tick_params(axis='x', rotation=60, labelsz=11)

    for p in axes[i].patches:
        height = p.get_height()
        if height > 0:
            axes[i].annotate(f'{int(height):,}',
                            (p.get_x() + p.get_width() / 2., height),
                            ha='center', va='bottom', fontsize=11)

fig.suptitle("Incidencias por MAT_AVERIADO (CLASE 1, Nivel de Tensión BAJA) por Año",
            fontsize=20, y=1.05)
plt.tight_layout()
plt.show()
```

### **GRAFICO DE NUMERO DE INCIDENCIAS, CLASE =1 POR MAT\_AVERIADO Y MT**

```
import seaborn as sns
import matplotlib.pyplot as plt

df_clase2_b = df[(df["CLASE"] == "1") & (df["NIV_TENS"] == "M")]
```

```

grupo_tipo = df_clase2_b.groupby("MAT_AVERIADO").agg({
    "CODIGO": "count"
}).rename(columns={"CODIGO": "N_INCIDENCIAS"}).reset_index()

plt.figure(figsize=(12,6))
ax = sns.barplot(data=grupo_tipo, x="MAT_AVERIADO", y="N_INCIDENCIAS",
palette="pastel")

plt.title("Número de Incidencias por MAT_AVERIADO (CLASE 1, Nivel de Tensión
MEDIA )", fontsize=14)
plt.xlabel("MAT_AVERIADO")
plt.ylabel("Nº de Incidencias")
plt.grid(False)

# Etiquetas encima de las barras
for p in ax.patches:
    height = p.get_height()
    if height > 0:
        ax.annotate(f'{int(height):,}',
            (p.get_x() + p.get_width()/2., height),
            ha='center', va='bottom', fontsize=9)

plt.xticks(rotation=45)
plt.tight_layout()
plt.show()

```

### **GRAFICO DE NUMERO DE INCIDENCIAS, CLASE =1 POR MAT\_AVERIADO Y MT Y POR AÑO**

```

import seaborn as sns
import matplotlib.pyplot as plt

df_clase2_b = df[(df["CLASE"] == "1") & (df["NIV_TENS"] == "M")]

# Agrupar por MAT_AVERIADO y AÑO
grupo_origen_año = df_clase2_b.groupby(["AÑO", "MAT_AVERIADO"]).agg({
    "CODIGO": "count"
}).rename(columns={"CODIGO": "N_INCIDENCIAS"}).reset_index()
# Años a graficar
años = [2022, 2023, 2024]
fig, axes = plt.subplots(1, 3, figsize=(24, 10), sharey=True)

import seaborn as sns
import matplotlib.pyplot as plt

for i, año in enumerate(años):
    datos = grupo_origen_año[grupo_origen_año["AÑO"] == año]
    sns.barplot(data=datos, x="MAT_AVERIADO", y="N_INCIDENCIAS", ax=axes[i],
palette="pastel")

    axes[i].set_title(f"Año {año}", fontsize=18)
    axes[i].set_xlabel("MAT_AVERIADO", fontsize=14)
    if i == 0:
        axes[i].set_ylabel("Nº de Incidencias", fontsize=14)
    else:
        axes[i].set_ylabel("")

```

```

axes[i].grid(False)
axes[i].tick_params(axis='x', rotation=60, labels=11)

for p in axes[i].patches:
    height = p.get_height()
    if height > 0:
        axes[i].annotate(f'{int(height):,}',
                        (p.get_x() + p.get_width() / 2., height),
                        ha='center', va='bottom', fontsize=11)

fig.suptitle("Incidencias por MAT_AVERIADO (CLASE 1, Nivel de Tensión MEDIA) por
Año", fontsize=20, y=1.05)
plt.tight_layout()
plt.show()

```

### **GRAFICO DE NUMERO DE INCIDENCIAS, CLASE =1 POR MAT\_AVERIADO Y AT**

```

import seaborn as sns
import matplotlib.pyplot as plt

df_clase2_b = df[(df["CLASE"] == "1") & (df["NIV_TENS"] == "A")]

grupo_tipo = df_clase2_b.groupby("MAT_AVERIADO").agg({
    "CODIGO": "count"
}).rename(columns={"CODIGO": "N_INCIDENCIAS"}).reset_index()

plt.figure(figsize=(12,6))
ax = sns.barplot(data=grupo_tipo, x="MAT_AVERIADO", y="N_INCIDENCIAS",
palette="pastel")

plt.title("Número de Incidencias por MAT_AVERIADO (CLASE 1, Nivel de Tensión ALTA
)", fontsize=14)
plt.xlabel("MAT_AVERIADO")
plt.ylabel("N° de Incidencias")
plt.grid(False)

# Etiquetas encima de las barras
for p in ax.patches:
    height = p.get_height()
    if height > 0:
        ax.annotate(f'{int(height):,}',
                    (p.get_x() + p.get_width()/2., height),
                    ha='center', va='bottom', fontsize=9)

plt.xticks(rotation=45)
plt.tight_layout()
plt.show()

```

### **GRAFICO DE NUMERO DE INCIDENCIAS, CLASE =1 POR MAT\_AVERIADO Y AT Y POR AÑO**

```

import seaborn as sns
import matplotlib.pyplot as plt

df_clase2_b = df[(df["CLASE"] == "1") & (df["NIV_TENS"] == "A")]

```

```
# Agrupar por MAT_AVERIADO y AÑO
grupo_origen_año = df_clase2_b.groupby(["AÑO", "MAT_AVERIADO"]).agg({
    "CODIGO": "count"
}).rename(columns={"CODIGO": "N_INCIDENCIAS"}).reset_index()
# Años a graficar
años = [2022, 2023, 2024]
fig, axes = plt.subplots(1, 3, figsize=(24, 10), sharey=True) # tamaño más grande

import seaborn as sns
import matplotlib.pyplot as plt

for i, año in enumerate(años):
    datos = grupo_origen_año[grupo_origen_año["AÑO"] == año]
    sns.barplot(data=datos, x="MAT_AVERIADO", y="N_INCIDENCIAS", ax=axes[i],
palette="pastel")

    axes[i].set_title(f"Año {año}", fontsize=18)
    axes[i].set_xlabel("MAT_AVERIADO", fontsize=14)
    if i == 0:
        axes[i].set_ylabel("Nº de Incidencias", fontsize=14)
    else:
        axes[i].set_ylabel("")

    axes[i].grid(False)
    axes[i].tick_params(axis='x', rotation=60, labels=11)

    for p in axes[i].patches:
        height = p.get_height()
        if height > 0:
            axes[i].annotate(f'{int(height):,}',
                (p.get_x() + p.get_width() / 2., height),
                ha='center', va='bottom', fontsize=11)

fig.suptitle("Incidencias por MAT_AVERIADO (CLASE 1, Nivel de Tensión ALTA) por Año",
    fontsize=20, y=1.05)
plt.tight_layout()
plt.show()
```

### **GRAFICO DE NUMERO DE INCIDENCIAS, CLASE =1 POR MAT\_AVERIADO Y NIVEL DE TENSION=S**

```
import seaborn as sns
import matplotlib.pyplot as plt

df_clase2_b = df[(df["CLASE"] == "1") & (df["NIV_TENS"] == "S")]

grupo_tipo = df_clase2_b.groupby("MAT_AVERIADO").agg({
    "CODIGO": "count"
}).rename(columns={"CODIGO": "N_INCIDENCIAS"}).reset_index()

plt.figure(figsize=(12,6))
ax = sns.barplot(data=grupo_tipo, x="MAT_AVERIADO", y="N_INCIDENCIAS",
palette="pastel")
```

```
plt.title("Número de Incidencias por MAT_AVERIADO (CLASE 1, Nivel de Tensión S)", fontsize=14)
plt.xlabel("MAT_AVERIADO")
plt.ylabel("Nº de Incidencias")
plt.grid(False)

# Etiquetas encima de las barras
for p in ax.patches:
    height = p.get_height()
    if height > 0:
        ax.annotate(f'{int(height):,}',
                    (p.get_x() + p.get_width()/2., height),
                    ha='center', va='bottom', fontsize=9)

plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```

### **GRAFICO DE NUMERO DE INCIDENCIAS, CLASE =1 POR MAT\_AVERIADO Y NIVEL DE TENSION=S Y POR AÑO**

```
import seaborn as sns
import matplotlib.pyplot as plt

df_clase2_b = df[(df["CLASE"] == "1") & (df["NIV_TENS"] == "S")]

# Agrupar por MAT_AVERIADO y AÑO
grupo_origen_año = df_clase2_b.groupby(["AÑO", "MAT_AVERIADO"]).agg({
    "CODIGO": "count"
}).rename(columns={"CODIGO": "N_INCIDENCIAS"}).reset_index()
# Años a graficar
años = [2022, 2023, 2024]
fig, axes = plt.subplots(1, 3, figsize=(24, 10), sharey=True) # tamaño más grande

import seaborn as sns
import matplotlib.pyplot as plt

for i, año in enumerate(años):
    datos = grupo_origen_año[grupo_origen_año["AÑO"] == año]
    sns.barplot(data=datos, x="MAT_AVERIADO", y="N_INCIDENCIAS", ax=axes[i],
                palette="pastel")

    axes[i].set_title(f"Año {año}", fontsize=18)
    axes[i].set_xlabel("MAT_AVERIADO", fontsize=14)
    if i == 0:
        axes[i].set_ylabel("Nº de Incidencias", fontsize=14)
    else:
        axes[i].set_ylabel("")

    axes[i].grid(False)
    axes[i].tick_params(axis='x', rotation=60, labelsize=11)

    for p in axes[i].patches:
        height = p.get_height()
        if height > 0:
```

```
axes[i].annotate(f'{int(height):,}',  
                (p.get_x() + p.get_width() / 2., height),  
                ha='center', va='bottom', fontsize=11)  
  
fig.suptitle("Incidencias por MAT_AVERIADO (CLASE 1, Nivel de Tensión S) por  
Año", fontsize=20, y=1.05)  
plt.tight_layout()  
plt.show()
```

## **CÁLCULO DE SAIDI Y SAIFI 2024-2025 POR CLASE (PROGRAMADA E IMPREVISTA)**

### **Conexión y Extracción desde Athena**

```
from pyathena import connect  
import pandas as pd  
  
# Conexión a Athena  
conn = connect(  
    profile_name='000370836235_PS-DLAKEUFD-PRO-ANALYTICS',  
    region_name='eu-west-1',  
    s3_staging_dir='s3://naturgy-dlakeufd-pro/aws-query-athena-results/'  
)  
  
# Consulta SQL  
consulta = """  
WITH datos_limpios AS (  
    SELECT  
        i.cod_incid AS INCIDENCIA,  
        i.cod_inst_sgd AS INSTALACION,  
        COALESCE(b.nombre, b2.nombre) AS NOMBRE,  
        i.propiedad AS PROPIEDAD,  
        p.nomb AS PROVINCIA,  
        i.fecha_det AS FECHA_DET,  
        i.hora_det AS HORA_DET,  
        i.fecha_rep AS FECHA_REP,  
        i.hora_rep AS HORA_REP,  
        i.potencia AS POTENCIA,  
        i.duracion AS DURACION,  
        TRIM(LEADING '0' FROM i.n_clien) AS CLIENTES_RAW,  
        i.tipo_seg AS TIPO_SEG,  
        i.municipio AS MUNICIPIO,  
        c.nomb AS NOMBRE_MUNICIPIO,  
        i.sin_niepi AS SIN_NIEPI,  
        inc.niv_ten AS NIV_TENS,  
        inc.tipo AS TIPO,  
        inc.descargo AS CLASE  
    FROM master_sgi.sgi_interrupciones i  
    LEFT JOIN master_sgi.sgi_instalaciones b ON i.cod_inst_sgd = b.cod_ope  
    LEFT JOIN master_sgi.sgi_bajainst b2 ON i.cod_inst_sgd = b2.ct_sgd  
    LEFT JOIN master_sgi.sgi_callejero c  
        ON SUBSTRING(c.cgh, 1, 5) = TRIM(LEADING '0' FROM i.municipio)  
        AND c.tip_reg = '20'  
    LEFT JOIN master_sgi.sgi_callejero p  
        ON SUBSTRING(p.cgh, 1, 2) = SUBSTRING(i.municipio, 1, 2)  
        AND p.tip_reg = '10'
```

```

LEFT JOIN master_sgi.sgi_incidencias inc ON i.cod_incid = inc.codigo
WHERE inc.codigo IS NOT NULL
      AND inc.estado != 'AN'
      AND SUBSTRING(i.fecha_det, 1, 4) IN ('2022','2023','2024','2025')
),
datos_con_ct AS (
  SELECT *,
        CASE
          WHEN REGEXP_LIKE(CLIENTES_RAW, '^[0-9]+$') THEN CAST(CLIENTES_RAW AS
INT)
          ELSE 0
        END AS CLIENTES_NUM
  FROM datos_limpios
)
SELECT
  INCIDENCIA,
  MAX(INSTALACION) AS INSTALACION,
  MAX(NOMBRE) AS NOMBRE,
  MAX(PROPIEDAD) AS PROPIEDAD,
  MAX(PROVINCIA) AS PROVINCIA,
  MAX(FECHA_DET) AS FECHA_DET,
  MAX(HORA_DET) AS HORA_DET,
  MAX(FECHA_REP) AS FECHA_REP,
  MAX(HORA_REP) AS HORA_REP,
  MAX(POTENCIA) AS POTENCIA,
  MAX(DURACION) AS DURACION,
  MAX(CLIENTES_RAW) AS CLIENTES,
  MAX(TIPO_SEG) AS TIPO_SEG,
  MAX(MUNICIPIO) AS MUNICIPIO,
  MAX(NOMBRE_MUNICIPIO) AS NOMBRE_MUNICIPIO,
  MAX(SIN_NIEPI) AS SIN_NIEPI,
  MAX(CLASE) AS CLASE,
  MAX(NIV_TENS) AS NIV_TENS,
  MAX(TIPO) AS TIPO,
  SUM(CLIENTES_NUM * DURACION) AS CxT,
  SUM(CLIENTES_NUM * DURACION) / 3857128.0 AS SAIDI,
  SUM(CLIENTES_NUM) AS TotalClientesAfectados,
  SUM(CLIENTES_NUM) / 3857128.0 AS SAIFI
FROM datos_con_ct
GROUP BY INCIDENCIA;
"""

# Carga de datos
df = pd.read_sql(consulta, conn)

```

## Limpieza y Preparación de Datos

```

# Procesamiento inicial
df['FECHA_DET'] = pd.to_datetime(df['FECHA_DET'], errors='coerce')
df['AÑO'] = df['FECHA_DET'].dt.year
df['NIV_TENS'] = df['NIV_TENS'].astype(str).str.strip().str.upper()
df['CLASE'] = pd.to_numeric(df['CLASE'], errors='coerce')

# Ver datos

```

```
print(df.head())
```

### **Cálculo de SAIDI por Clase y Nivel de Tensión**

```
# Filtrado por duración > 180 segundos y NIV_TENS distinto de 'S'
df_filtrado = df[(df['DURACION'] > 180) & (df['NIV_TENS'] != 'S') & (df['AÑO'] ==
2024)]

# Agrupar por NIV_TENS y CLASE
saidi_por_tension = df_filtrado.groupby(['CLASE',
'NIV_TENS'])['SAIDI'].sum().reset_index()

print(saidi_por_tension)
```

### **Gráfico de Circular por niveles de tensión para ver cuanto SAIDI aporta cada nivel**

```
import matplotlib.pyplot as plt

for clase in [1, 2]:
    datos = saidi_por_tension[saidi_por_tension['CLASE'] == clase]
    plt.figure(figsize=(6, 6))
    plt.pie(
        datos['SAIDI'],
        labels=datos['NIV_TENS'],
        autopct='%1.2f%%',
        startangle=90
    )
    plt.title(f'Distribución SAIDI por Nivel de Tensión - Clase {clase} (2024)')
    plt.show()
```

### **SAIDI Total por Año y Clase (Gráfico de Barras)**

```
# Crear tabla resumen
tabla_saidi = df_filtrado.groupby(['AÑO',
'CLASE'])['SAIDI'].sum().unstack(fill_value=0)
tabla_saidi['TOTAL'] = tabla_saidi.sum(axis=1)
print(tabla_saidi)

# Gráfico de barras
import seaborn as sns

df_resultados = tabla_saidi.reset_index().melt(id_vars='AÑO', var_name='CLASE',
value_name='SAIDI')

plt.figure(figsize=(10,6))
sns.barplot(data=df_resultados, x='AÑO', y='SAIDI', hue='CLASE')
plt.title("SAIDI por Año y Clase (duración > 180s)")
plt.ylabel("SAIDI (minutos)")
plt.xlabel("Año")
plt.tight_layout()
plt.show()
```

### **Código para generar:**

**Una tabla resumen con SAIDI y SAIFI por año y clase.**

**Dos gráficos de barras: uno para SAIDI y otro para SAIFI.**

```
from pyathena import connect
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# Conexión con Athena
conn = connect(
    profile_name='000370836235_PS-DLAKEUFD-PRO-ANALYTICS',
    region_name='eu-west-1',
    s3_staging_dir='s3://naturgy-dlakeufd-pro/aws-query-athena-results/'
)

# Consulta SQL para extraer datos agregados por incidencia
consulta = """
WITH datos_limpios AS (
    SELECT
        i.cod_incid AS INCIDENCIA,
        i.cod_inst_sgd AS INSTALACION,
        COALESCE(b.nombre, b2.nombre) AS NOMBRE,
        p.nomb AS PROVINCIA,
        i.fecha_det AS FECHA_DET,
        i.hora_det AS HORA_DET,
        i.fecha_rep AS FECHA_REP,
        i.hora_rep AS HORA_REP,
        i.potencia AS POTENCIA,
        i.duracion AS DURACION,
        TRIM(LEADING '0' FROM i.n_clien) AS CLIENTES_RAW,
        i.municipio AS MUNICIPIO,
        c.nomb AS NOMBRE_MUNICIPIO,
        inc.niv_ten AS NIV_TENS,
        inc.tipo AS TIPO,
        inc.descargo AS CLASE
    FROM master_sgi.sgi_interrupciones i
    LEFT JOIN master_sgi.sgi_instalaciones b ON i.cod_inst_sgd = b.cod_ope
    LEFT JOIN master_sgi.sgi_bajainst b2 ON i.cod_inst_sgd = b2.ct_sgd
    LEFT JOIN master_sgi.sgi_callejero c ON SUBSTRING(c.cgh, 1, 5) = TRIM(LEADING
'0' FROM i.municipio) AND c.tip_reg = '20'
    LEFT JOIN master_sgi.sgi_callejero p ON SUBSTRING(p.cgh, 1, 2) =
SUBSTRING(i.municipio, 1, 2) AND p.tip_reg = '10'
    LEFT JOIN master_sgi.sgi_incidencias inc ON i.cod_incid = inc.codigo
    WHERE inc.codigo IS NOT NULL
        AND inc.estado != 'AN'
        AND SUBSTRING(i.fecha_det, 1, 4) IN ('2022','2023','2024','2025')
),
datos_con_ct AS (
    SELECT *,
        CASE
            WHEN REGEXP_LIKE(CLIENTES_RAW, '^[0-9]+$') THEN CAST(CLIENTES_RAW AS
INT)
            ELSE 0
        END AS CLIENTES_NUM

```

```
FROM datos_limpios
)
SELECT
    INCIDENCIA,
    MAX(FECHA_DET) AS FECHA_DET,
    MAX(DURACION) AS DURACION,
    MAX(NIV_TENS) AS NIV_TENS,
    MAX(CLASE) AS CLASE,
    SUM(CLIENTES_NUM) AS TOTAL_CLIENTES,
    SUM(CLIENTES_NUM * DURACION) AS CxT,
    MAX(TIPO) AS TIPO
FROM datos_con_ct
GROUP BY INCIDENCIA;
"""

# Ejecutar la consulta
df = pd.read_sql(consulta, conn)

# Procesamiento de datos
df['FECHA_DET'] = pd.to_datetime(df['FECHA_DET'], errors='coerce')
df['AÑO'] = df['FECHA_DET'].dt.year
df['NIV_TENS'] = df['NIV_TENS'].astype(str).str.strip().str.upper()
df['CLASE'] = pd.to_numeric(df['CLASE'], errors='coerce')

# Supongamos que el total de clientes de referencia es este (ajustar si es otro):
TOTAL_CLIENTES = 3857128

# Filtrar incidencias válidas (sin nivel de tensión 'S' y duración > 180
segundos)
df_filtrado = df[(df['DURACION'] > 180) & (df['NIV_TENS'] != 'S')]

# Agrupar por año y clase para calcular SAIDI y SAIFI
resultados = df_filtrado.groupby(['AÑO', 'CLASE']).agg({
    'CxT': 'sum',
    'TOTAL_CLIENTES': 'sum'
}).reset_index()

# Calcular SAIDI y SAIFI
resultados['SAIDI'] = resultados['CxT'] / TOTAL_CLIENTES / 60 # Convertido a
minutos
resultados['SAIFI'] = resultados['TOTAL_CLIENTES'] / TOTAL_CLIENTES

# Tabla resumen
tabla = resultados.pivot(index='AÑO', columns='CLASE', values=['SAIDI', 'SAIFI'])
print(tabla.round(3))

# ♦ Gráfico de barras comparativo SAIDI y SAIFI
plt.figure(figsize=(10, 6))
sns.barplot(data=resultados, x='AÑO', y='SAIDI', hue='CLASE')
plt.title("SAIDI por Año y Clase (Duración > 180s)")
plt.ylabel("SAIDI (minutos)")
plt.tight_layout()
plt.show()
```

```
plt.figure(figsize=(10, 6))
sns.barplot(data=resultados, x='AÑO', y='SAIFI', hue='CLASE')
plt.title("SAIFI por Año y Clase (Duración > 180s)")
plt.ylabel("SAIFI (n° interrupciones por cliente)")
plt.tight_layout()
plt.show()
```



