



Universitat Oberta
de Catalunya

Aplicación de Inteligencia Empresarial para el análisis del comportamiento de incidencias de una Plataforma de Gestión y Tarificación de Usuarios de Operadoras de Telefonía Móvil.

Julio Pedro Guerrero Hernández
Grado de Ingeniería Informática
Área de Business Intelligence

Nombre Consultor/a:
Humberto Andrés Sanz

**Nombre Profesor/a responsable de la
asignatura:**
Atanasi Daradoumis Haralabus

Fecha Entrega: 15/01/2023



Esta obra está sujeta a una licencia de Reconocimiento-NoComercial-SinObraDerivada [3.0 España de Creative Commons](https://creativecommons.org/licenses/by-nc-nd/3.0/es/)

FICHA DEL TRABAJO FINAL

Título del trabajo:	<i>Aplicación de Inteligencia Empresarial para el análisis del comportamiento de incidencias de una Plataforma de Gestión y Tarificación de Usuarios de Operadoras de Telefonía Móvil.</i>
Nombre del autor:	<i>Julio Pedro Guerrero Hernández</i>
Nombre del consultor/a:	<i>Humberto Andrés Sanz</i>
Nombre del PRA:	<i>Atanasi Daradoumis Haralabus</i>
Fecha de entrega (mm/aaaa):	<i>01/2023</i>
Titulación o programa:	Grado de Ingeniería Informática
Área del Trabajo Final:	<i>Business Intelligence</i>
Idioma del trabajo:	<i>Castellano</i>
Palabras clave	<i>Business Intelligence, ETL, Almacén de datos</i>
Resumen del Trabajo	
<p>Este proyecto tiene como finalidad principal el análisis del comportamiento de las incidencias generadas a la plataforma que gestiona y tarifica todo lo relacionado con abonados de telefonía móvil para 16 operadoras en América y Europa, para cumplir con la protección comercial de los datos no se mencionará ningún nombre comercial ni se hará referencia a datos personales.</p> <p>Para el desarrollo de este proyecto se utilizará la metodología y las herramientas que se usarían en un entorno de trabajo real, por lo cual primero se procedió a extraer los datos de una base de datos relacional en remoto, posterior a esto se transformaron y cargaron haciendo uso de una herramienta ETL, el repositorio final será un almacén de datos configurado en local usando un gestor de base de datos.</p> <p>Con la data cargada se procederá a usar una herramienta de análisis de negocio para plasmar la información que nos ira desvelando la data, generando un cuadro de mando principal a modo de resumen y cinco reportes cada uno con una categoría definida, estas son criticidad, país, estado, reabiertas, parches y causa raíz asociadas a las incidencias.</p> <p>Los resultados han sido tal cual se esperaban, se ha conseguido todo lo relacionado con los aspectos técnicos, es decir generado un almacén con la data transformada y todo lo relacionado con el aspecto de análisis de negocios, ya que se ha podido visualizar los reportes, generar los gráficos necesarios para entender o poder analizar el comportamiento de las incidencias.</p>	
Abstract	
<p>The main purpose of this project is to analyze the behavior of the incidents generated by the platform that manages and rates everything related to mobile telephony subscribers for 16 operators in America and Europe. To comply with the commercial protection of the data, no commercial name will be mentioned, nor will any reference be made to personal data.</p> <p>For the development of this project, we will use the methodology and tools that would be used in a real working environment, so first, we proceeded to extract the data from a relational database remotely, after this they were transformed and loaded using an ETL tool, the final repository will be a data warehouse configured locally using a database manager.</p>	

With the data loaded, we proceeded to use a business analysis tool to capture the information that will reveal the data, generating a main scorecard as a summary and five reports each with a defined category. These are criticality, country, status, reopened, patches, and root causes associated with the incidents.

The results were as expected. We have achieved everything related to the technical aspects, a warehouse with the transformed data and we have been able to visualize the reports and generate the necessary graphs to understand or to be able to analyze the behavior of the incidents.

Índice

1. Introducción.....	5
1.1 Contexto y justificación del Trabajo.....	5
1.2 Objetivos del Trabajo.....	5
1.3 Enfoque y método seguido.....	6
1.3.1 Programas informáticos a utilizar.....	7
1.4 Planificación del Trabajo.....	7
1.4.1 Tiempo.....	7
1.4.2 Recursos humanos:.....	8
1.4.3 Actividades y tareas del proyecto.....	8
1.4.4 Hitos.....	9
1.4.5 Recursos técnicos.....	12
1.5 Breve resumen de productos obtenidos.....	12
1.6 Breve descripción de los otros capítulos de la memoria.....	12
2. Análisis de la Solución.....	14
2.1 Análisis de Requisitos.....	15
2.1.1 Valoraciones.....	15
2.1.2 Requisitos funcionales.....	15
2.1.3 Requisitos no funcionales.....	17
2.2 Análisis de Indicadores e Informes.....	18
2.2.1 Datos y contexto de los indicadores.....	18
2.2.2 Indicadores relacionados con criticidad/estado de las incidencias.....	19
2.2.3 Indicadores relacionados con origen de las incidencias.....	21
2.2.4 Indicadores relacionados el tiempo de solución.....	21
2.4 Análisis Inicial del Conjunto de Datos de Origen.....	22
2.4.1 Tablas necesarias para obtener el estatus de incidencias.....	23
2.4.2 Tablas complementarias para los KPI e Informes.....	24
2.4.3 Tablas complementarias para asociar parches.....	26
2.4.4 Tablas complementarias para la causa raíz.....	26
2.4.5 Tablas complementarias para medir la afectación.....	27
2.4.6 Tablas complementarias para asociar MOP.....	27
2.5 Análisis de Modelo de Datos.....	31
2.5.1 Modelo de Estrella.....	32
2.5.2 Funcionamiento del modelo.....	33
2.5.3 Ventajas de usar este modelo.....	33
3. Diseño de la Solución.....	35
3.1 Diseño de la estructura de la base de datos.....	35
3.1.1 Estructura OLTP.....	35
3.1.1 Estructura OLAP.....	38
3.2 Diseño del Almacén de Datos (DWH).....	39
3.2.1 Esquema del almacén de datos.....	40
3.3 Diseño presentación grafica.....	40
3.3.1 Diseño Reportes.....	41
4. Elección de las herramientas de la Solución.....	44
4.1 Criterios de elección de herramientas.....	44
4.1.1 Elección de herramienta de análisis e inteligencia empresarial.....	44
4.1.2 Elección de herramienta de ETL.....	48
4.1.3 Elección de herramienta de soporte del almacén de datos.....	50
5. Implementación de la Solución.....	52

5.1 ETL.....	52
5.1.1 ETL Primera etapa	52
5.1.1.1 Creación base de datos local en PostgreSQL.....	52
5.1.1.2 Creación trabajo para poblar tablas en Pentaho	53
5.1.2 ETL Segunda etapa.....	59
5.1.2.1 Configuración previa dimensión	60
5.1.2.2 Creación de las dimensiones	66
5.1.2.3 Validación de la inserción de las tablas.....	69
5.1.2.4 Creación de la tabla de hechos	70
5.2 Validaciones Datos Post ETL	72
5.3 Implementación Reportes.....	73
5.3.1 Implementación Dashboard Resumen	73
5.3.2 Implementación Reporte Criticidad.....	74
5.3.3 Implementación Reporte Estado	74
5.3.4 Implementación Reporte País	75
5.3.5 Implementación Reporte Reabiertas y Parches	75
5.3.6 Implementación Reporte Causa Raíz.....	76
5.3.7 Implementación KPIs.....	76
6. Conclusiones.....	77
6.1 Asociadas al desarrollo del proyecto.....	77
6.2 Asociadas a lo alcanzado en el proyecto	77
6.3 Asociadas a una posible mejora.....	77
6.4 Asociadas a lo personal	78
7. Glosario	79
8. Bibliografía	80
9. Anexos	81
9.1 Introducción al ecosistema de las herramientas.....	81
9.1.1 Introducción MS Power BI.....	81
9.1.1.1 Descarga e instalación de la herramienta	81
9.1.1.2 Términos a usar debido a esta herramienta	82
9.1.2 Introducción Pentaho Data Integration	86
9.1.2.1 Descarga e instalación de la herramienta	86
9.1.2.2 Términos a usar debido a esta herramienta	88
9.1.3 Introducción PostgreSQL	89
9.1.3.1 Descarga e instalación de la herramienta	89
9.1.3.2 Términos a usar debido a esta herramienta	91
9.2 Transformaciones Posteriores al ETL	93

Índice de Tablas

Tabla 1. Dedicación en horas por cada PEC. Fuente: Elaboración Propia	8
Tabla 2. Fechas de Entrega PEC. Fuente: Elaboración Propia.....	8
Tabla 3. Tabla de actividades y tareas del proyecto (Versión 3). Fuente: Elaboración Propia	9
Tabla 4. Hitos del proyecto. Fuente: Elaboración Propia.....	10
Tabla 5. Tabla comparativa herramientas BI. Fuente: Elaboración Propia	47
Tabla 6. Tabla comparativa herramientas ETL. Fuente: Elaboración Propia	49
Tabla 7. Tabla comparativa soporte almacén de datos. Fuente: Elaboración Propia	51
Tabla 8. ETL1 Creación Pentaho Conteo registros tablas. Fuente: Elaboración propia	59

Índice de Figuras

Figura 1. Planificación del Proyecto – Gantt (Versión 4). Fuente: Elaboración Propia	11
Figura 2. Flujo y transformación a Conocimiento entre las Capas del Negocio. Fuente: Elaboración Propia	14
Figura 3. Componentes Análisis Funcional del Proyecto Fuente: Elaboración Propia.....	14
Figura 4. Arquitectura Plataforma. Fuente: Elaboración Propia	23
Figura 5. Diagrama Entidad Relación (Flujo Incidencias). Fuente: Elaboración Propia	29
Figura 6 Diagrama Entidad Relación (Flujo Incidencia vs MOP). Fuente: Elaboración Propia ..	30
Figura 7. Esquema del Modelo en Estrella. Fuente: https://learn.microsoft.com/en-us/power-bi/guidance/star-schema	33
Figura 8. Esquema Inicial Modelo Estrella: Elaboración propia.....	34
Figura 9. OLTP CicloVida. Elaboración propia	35
Figura 10. OLTP Tiempo. Elaboración propia.....	35
Figura 11. OLTP DGrupo. Elaboración propia	36
Figura 12. OLTP DOperadora. Elaboración propia	36
Figura 13. OLTP ROpen. Elaboración propia	36
Figura 14. OLTP Patch. Elaboración propia	37
Figura 15. OLTP RootCause. Elaboración propia.....	37
Figura 16. OLTP Extra. Elaboración propia	37
Figura 17. Estructura OLAP. Elaboración propia	38
Figura 18. Esquema Final Proyecto Modelo Estrella: Elaboración propia.....	39
Figura 19. Arquitectura Almacén de Datos. Elaboración propia	40
Figura 20. Escala de grises DEMO. Elaboración propia.....	41
Figura 21. Prototipo Resumen Reporte. Elaboración Propia	41
Figura 22. Prototipo Reporte Criticidad. Elaboración Propia	42
Figura 23. Prototipo Reporte Estado. Elaboración Propia	42
Figura 24. Prototipo Reporte País. Elaboración Propia	42
Figura 25. Prototipo Reporte Reabiertas-Parches. Elaboración Propia	43
Figura 26. Prototipo Reporte Causa Raíz. Elaboración Propia	43
Figura 27. Cuadrante Mágico de Gartner para plataformas de análisis e inteligencia de negocio. Fuente: https://www.gartner.com/document/4012759?ref=solrAll&refval=347522631	44
Figura 28. Interface Tableau Desktop. Fuente: Internet	45
Figura 29. Interface Qlik. Fuente: Internet	45
Figura 30. Interface Power BI. Fuente: Internet	46
Figura 31. Interface Pentaho Data Integration. Fuente: Internet	48
Figura 32. Interface Talend Data Integration. Fuente: Internet.....	49
Figura 33. ETL1 Creación BD PG Interfaz pgAdmin. Fuente: Elaboración propia	52
Figura 34. ETL1 Creación BD PG Menú. Fuente: Elaboración propia	52
Figura 35. ETL1 Creación BD PG Creación Paso 1. Fuente: Elaboración propia.....	53
Figura 36. ETL1 Creación BD PG Creación Paso 2. Fuente: Elaboración propia.....	53
Figura 37. ETL1 Creación BD Carga Spoon. Fuente: Elaboración propia	53
Figura 38. ETL1 Creación Pentaho Menú Trabajo. Fuente: Elaboración propia	54
Figura 39. ETL1 Creación Pentaho Nuevo Trabajo. Fuente: Elaboración propia	54
Figura 40. ETL1 Creación Pentaho Conexión Oracle. Fuente: Elaboración propia	54
Figura 41. ETL1 Creación Pentaho Conexión Oracle Datos 1. Fuente: Elaboración propia.....	54
Figura 42. ETL1 Creación Pentaho Conexión Oracle Datos 2. Fuente: Elaboración propia.....	55
Figura 43. ETL1 Creación Pentaho Conexión Oracle Datos 3. Fuente: Elaboración propia.....	55
Figura 44. ETL1 Creación Pentaho Conexión Oracle Test. Fuente: Elaboración propia	55
Figura 45. ETL1 Creación Pentaho Conexión Postgres Test. Fuente: Elaboración propia.....	55
Figura 46. ETL1 Creación Pentaho Copia Tablas 1. Fuente: Elaboración propia	56
Figura 47. ETL1 Creación Pentaho Agregación Conexiones. Fuente: Elaboración propia.....	56
Figura 48. ETL1 Creación Pentaho Tablas Importar. Fuente: Elaboración propia	56
Figura 49. ETL1 Creación Pentaho Listado Tablas. Fuente: Elaboración propia	56
Figura 50. ETL1 Creación trabajo Configuración directorio. Fuente: Elaboración propia	57
Figura 51. ETL1 Creación Pentaho Panel Tareas. Fuente: Elaboración propia	57
Figura 52. ETL1 Creación Pentaho Ejecución Tarea. Fuente: Elaboración propia	57
Figura 53. ETL1 Creación Pentaho Monitorización Postgres. Fuente: Elaboración propia.....	58
Figura 54. ETL1 Creación Pentaho Jobs Metrics. Fuente: Elaboración propia	58
Figura 55. ETL1 Creación Pentaho Grafico Tablas Creadas. Fuente: Elaboración propia	59

Figura 56. ETL2 Creación Usuario dwh. Fuente: Elaboración propia.....	59
Figura 57. ETL2 Creación BD dwh_EXP. Fuente: Elaboración propia.....	60
Figura 58. ETL2 Creación tarea Pentaho. Fuente: Elaboración propia	60
Figura 59. ETL2 Conf DIM_Ciclovida. Fuente: Elaboración propia	60
Figura 60. ETL2 Conf DIM_Ciclovida 2. Fuente: Elaboración propia	61
Figura 61. ETL2 Creación Tabla DIM_Ciclovida. Fuente: Elaboración propia	62
Figura 62. ETL2 Panel Pentaho Pasos 1. Fuente: Elaboración propia	66
Figura 63. ETL2 Agregar conexión BD dwh. Fuente: Elaboración propia	66
Figura 64. ETL2 Test conexión BD dwh. Fuente: Elaboración propia	66
Figura 65. ETL2 Agregar Paso Insertar en DWH. Fuente: Elaboración propia	67
Figura 66. ETL2 Configurar Paso Insertar en DWH. Fuente: Elaboración propia	67
Figura 67. ETL2 Agregar Creación Dimension DWH. Fuente: Elaboración propia	68
Figura 68. ETL2 Validación Dimension Postgres. Fuente: Elaboración propia	68
Figura 69. ETL2 Foto Final Dimensiones DWH. Fuente: Elaboración propia	68
Figura 70. ETL2 Pentaho Ejecutar tarea poblar DWH. Fuente: Elaboración propia	69
Figura 71. ETL2 Pentaho Fin Ejecución. Fuente: Elaboración propia	69
Figura 72. ETL2 Pentaho Resumen Ejecución. Fuente: Elaboración propia	69
Figura 73. ETL2 Registros Dimensiones y Grafico PostgreSQL. Fuente: Elaboración propia...	70
Figura 74. Creación tabla de hechos FACT_CRS	72
Figura 75. Limpieza PowerBI tabla FACT_CRS. Elaboración propia	72

1. Introducción

La gestión de incidencias es una parte importante del día a día de aquellas empresas que brindan el servicio integral de una plataforma, si a eso le sumamos que no se puede negar la importancia de los análisis y la gestión del rendimiento ahora que las empresas están más orientadas a los datos que nunca. A partir de estos datos es que se puede mejorar la eficiencia y el rendimiento del servicio de soporte siendo este una práctica esencial de la gestión de servicios de TI, ya que es un proceso crítico por la influencia que tiene en la sensación de satisfacción del cliente final.

En este ámbito es donde he decidido desarrollar el proyecto asociado al TFG, ya que apoyados por los métodos de inteligencia empresarial y explotando su enorme potencial podemos usar los datos que puede generar la gestión de incidencias de manera que, a través de la recopilación, análisis y presentación de información de estos, se pueda hacer explícito el conocimiento implícito, acelerando los procesos de corrección necesarios que la plataforma necesite.

1.1 Contexto y justificación del Trabajo

Dado el contexto en el que nos movemos actualmente, donde todos los datos son cuantificables, clasificables y de los cuales se puede inferir información y en las manos adecuadas puede llegar a ser conocimiento, desde lo más macro como las tendencias bursátiles de un mercado creciente hasta lo más micro como la cantidad de veces que le damos a un botón al usar un buscador en el teléfono móvil, es que se decide enfocar este TFG.

Si esta tendencia, le sumamos mi creciente intereses por el análisis de datos, vi la conjunción perfecta para poder profundizar en conocimientos, tecnologías y tendencias que he ido adquiriendo durante mi formación universitaria, a esto hay que sumarle que, debido a mi trabajo actual, tengo un conjunto de datos del día a día de una plataforma que da soporte a una plataforma, que será la base para desarrollar mi proyecto.

1.2 Objetivos del Trabajo

Para este proyecto asociado al TFG se han planteado los siguientes objetivos:

1.2.1 Objetivos Generales

- Desarrollar una solución de inteligencia empresarial con la finalidad de analizar datos de una plataforma de gestión y tarificación de usuarios de telefonía móvil de manera que pueda facilitar la toma de decisiones y en caso de ser necesario aplicar correcciones.

1.2.2 Objetivos Específicos

- Desarrollar capacidades en lo referente con el uso de tecnologías que se usan actualmente en el área de análisis de datos, como es Pentaho y Power Bi.
- Desarrollar capacidades de transformar un conjunto de datos en información, a través de la aplicación de técnicas con una herramienta de ETL, para luego generar un almacén de datos hasta llegar a plasmar esta visualmente en un cuadro de mando, indicadores y reportes.
- Diseñar e implementar un almacén de datos y los procesos ETL necesarios para poder estandarizar la data obtenida del conjunto de datos inicial.
- Diseñar e implementar los cuadros de mando, reportes, indicadores e informes para simplificar el análisis aplicando las mejores prácticas.

1.3 Enfoque y método seguido

En lo referente a la gestión del TFG como proyecto usaremos las recomendaciones del PMBOK, en este caso, tal y como iremos desglosando el proyecto en hitos y etapas, la gestión estará asociada a la metodología EDT (**WBS**¹), nos hará de herramienta para gestionar este proyecto en base a sus entregables. Los cuales ya los hemos dividido en etapas para que nos permita planificar de manera fácil el proyecto. Lo que nos aportara simplicidad en la programación, supervisión, control y gestión.

En lo referente al entregable asociado a este TFG se usará la metodología **Waterfall**², siendo esta una serie de etapas lineales de gestión de proyectos de desarrollo. El modelo que usaremos incluirá cinco fases independientes, y cada fase se concatenará con los resultados de la fase anterior. Por lo tanto, será necesario completar cada una de ellas antes de poder pasar a la siguiente.

Entre las fases que iremos recorriendo están:

- **Análisis y requisitos**, se desarrollará en casi toda la Etapa 2, se generarán los puntos relacionados con los requisitos del producto, de manera que se genere y enumere lo que se espera que haga el producto final (en este caso nuestro entregable final).

¹ Work Breakdown Structure (WBS)—Basic Principles | PMI. (s. f.), de <https://www.pmi.org/learning/library/work-breakdown-structure-basic-principles-4883>

² Waterfall model. (2022). En Wikipedia. https://en.wikipedia.org/w/index.php?title=Waterfall_model&oldid=1114454036

- **Diseño**, se diseñará la arquitectura de base de datos, del almacén de datos y de los reportes e informes, se harán en base a los requisitos especificados en el punto anterior.
- **Implementación**, se desarrollará la tarea de ETL, integración de los sistemas o programas informáticos que usaremos y también implementaremos los reportes e informes, esta estará asociada a la Etapa 3.
- **Verificación**, se probará todo lo implementado para descubrir y corregir errores y defectos e identificar riesgos, aquí también añadiremos de ser necesario todas las mejoras o correcciones asociadas a la retroalimentación recibida en las Etapas anteriores.
- **Mantenimiento**, en este caso este punto no se llegará a cumplir debido a que este será un proyecto dentro del ámbito académico u no se dejará en funcionamiento en ningún entorno productivo.

1.3.1 Programas informáticos a utilizar

En este apartado se definirán los productos a nivel de software que serán utilizados posteriormente, como las herramientas para la creación del almacén de datos, creación del proceso del ETL, análisis y visualización.

Para este caso, se han elegido los siguientes productos:

- Windows 10 PRO, que es el que viene por defecto en el portátil donde se desarrollara el proyecto de TFG.
- MS Office 365 para empresas.

1.4 Planificación del Trabajo

En harás de desarrollar este TFG se han estimado los siguientes recursos:

1.4.1 Tiempo

Las horas de dedicación se han calculado en base a los créditos universitarios que pesa esta asignatura (12 ECTS) y estos se han convertido en horas 300 horas (1 ECTS = **25 horas**³) de dedicación para poder llevar a cabo este proyecto.

El proyecto se inició el día 16/09/2022 con el inicio del semestre y tendrá una duración de aproximadamente 16 semanas, finalizando el tiempo

³ FAQ - Estructura de las enseñanzas—Calidad—(UOC). (s. f.). , de <https://www.uoc.edu/portal/es/qualitat/qualitat-titulacions/estructura-ensenyaments/faqs/faq-generals.html>

para presentar todos los avances el día 15/01/2023, ya que se cuenta con 4 PECs.

Inicialmente se ha planteado una dedicación de 3 horas por día, con esta y el número de días entre inicio y fin de proyecto nos da un total de 303 horas, esto nos deja la siguiente relación de horas (aproximadas) por PEC que se tendrá:

Etapa	Cantidad Horas
PEC1	54
PEC2	90
PEC3	93
Entrega Final	66

Tabla 1. Dedicación en horas por cada PEC. Fuente: Elaboración Propia

Siendo realistas se espera algo de flexibilidad, moviendo horas hacia los fines de semana debido a la carga personal, navidades y a la dedicación de otras asignaturas que se están cursando en paralelo.

Por lo tanto, definiremos 4 etapas claramente delimitadas con fecha de fin asociada al día de entrega de cada PEC, en este caso las fechas límites para los entregables serán las siguientes:

Etapa	Fecha Limite
PEC1	16/10/2022
PEC2	16/11/2022
PEC3	18/12/2022
Entrega Final	15/01/2023

Tabla 2. Fechas de Entrega PEC. Fuente: Elaboración Propia

1.4.2 Recursos humanos:

Para desarrollar el proyecto, siendo este realizado por una sola persona, se contará con 1 recurso que desempeñará los siguientes roles, entre paréntesis tendremos la sigla que usaremos en el Gantt de la planificación:

- Gestor del proyecto (PM)
- Analista de Datos (AD)
- Administrador de Base de Datos (ABD)
- Analista BI (ABI)

1.4.3 Actividades y tareas del proyecto

El tener claros los delimitadores de tiempo, fechas de entregas de PEC, nuestros hitos serán 4, que se corresponderá con las 3 entregas parciales más la entrega final:

TAREAS	ROL	INICIO	FIN	DÍAS
Propuesta TFG	PM	28-Sep-22	3-Oct-22	5
Etapa1 PEC1 Gestión	PM	3-Oct-22	16-Oct-22	13
Definición de objetivos	PM	3-Oct-22	6-Oct-22	3
Planificación	PM	6-Oct-22	12-Oct-22	6
Documentación Entregable PEC1	PM	12-Oct-22	16-Oct-22	4
Entrega PEC1	PM	16-Oct-22	16-Oct-22	1
Etapa2 PEC2 Gestión	PM	17-Oct-22	16-Nov-22	30
Análisis de Requisitos	AD	17-Oct-22	24-Oct-22	7
Análisis de Indicadores e Informes	AD	24-Oct-22	28-Oct-22	4
Análisis Inicial Dataset en Origen	AD	28-Oct-22	4-Nov-22	7
Análisis Modelo de Datos	ABD	4-Nov-22	13-Nov-22	9
Documentación Entregable PEC2	PM	13-Nov-22	16-Nov-22	3
Entrega PEC2	PM	16-Nov-22	16-Nov-22	1
Etapa3 PEC3 Gestión	PM	17-Nov-22	18-Dec-22	31
Diseño Estructura Datos y DWH	ABD	17-Nov-22	27-Nov-22	10
Revisión Documentación Técnica Software	AD	27-Nov-22	29-Nov-22	2
ETL Fase1	ABD	29-Nov-22	5-Dec-22	6
Delay por problema personal	-	5-Dec-22	14-Dec-22	9
Validación Datos Post ETL	AD	14-Dec-22	15-Dec-22	1
Diseño Reportes e Informes	ABI	15-Dec-22	16-Dec-22	1
Implementación Inicial Reporte e Informes	ABI	16-Dec-22	17-Dec-22	1
Documentación Entregable PEC3	PM	16-Dec-22	18-Dec-22	2
Entrega PEC3	PM	18-Dec-22	18-Dec-22	1
Etapa4 Entrega Final Gestión	PM	19-Dec-22	15-Jan-23	22
Integración Feedback, Mejoras, NFs	PM	19-Dec-22	23-Dec-22	4
Integración Documento Final (PEC1-2-3)	PM	26-Dec-22	30-Dec-22	4
Revisión Documento Integrado	PM	2-Jan-23	6-Jan-23	4
Finalizar Documentos a Entregar	PM	6-Jan-23	15-Jan-23	9
Entrega Final	PM	15-Jan-23	15-Jan-23	1
Duración TFG	PM/AD/ABD/ABI	28-sep-22	15-Jan-23	110
Tribunal de Evaluación	PM/AD/ABD/ABI	27-Jan-23	28-Jan-23	1
Duración Semestre		28-sep-22	06-feb-23	131

Tabla 3. Tabla de actividades y tareas del proyecto (Versión 3). Fuente: Elaboración Propia

1.4.4 Hitos

En este punto expondremos los hitos no solo a nivel de entregables sino también de aquellas tareas que serán de obligatoria consecución para poder controlar el correcto avance de este proyecto:

TAREAS	INICIO	FINALIZACIÓN
Entrega PEC1	16-Oct-22	16-Oct-22
Análisis de Indicadores e Informes	24-Oct-22	28-Oct-22
Entrega PEC2	16-Nov-22	16-Nov-22
Diseño Estructura Datos y DWH	17-Nov-22	29-Nov-22
ETL Fase1	29-Nov-22	2-Dec-22
Implementación Inicial Reporte e Informes	12-Dec-22	15-Dec-22
Entrega PEC3	18-Dec-22	18-Dec-22
Integración <i>Feedbacks</i> , Mejoras, NFs	19-Dec-22	23-Dec-22
Entrega Final	15-Jan-23	15-Jan-23

Tabla 4. Hitos del proyecto. Fuente: Elaboración Propia

Si bien el plan inicial contempla alrededor de 327 horas según calendario y dedicación diaria, esta se verá reducida a 303 horas, quedando el siguiente Plan del Proyecto:

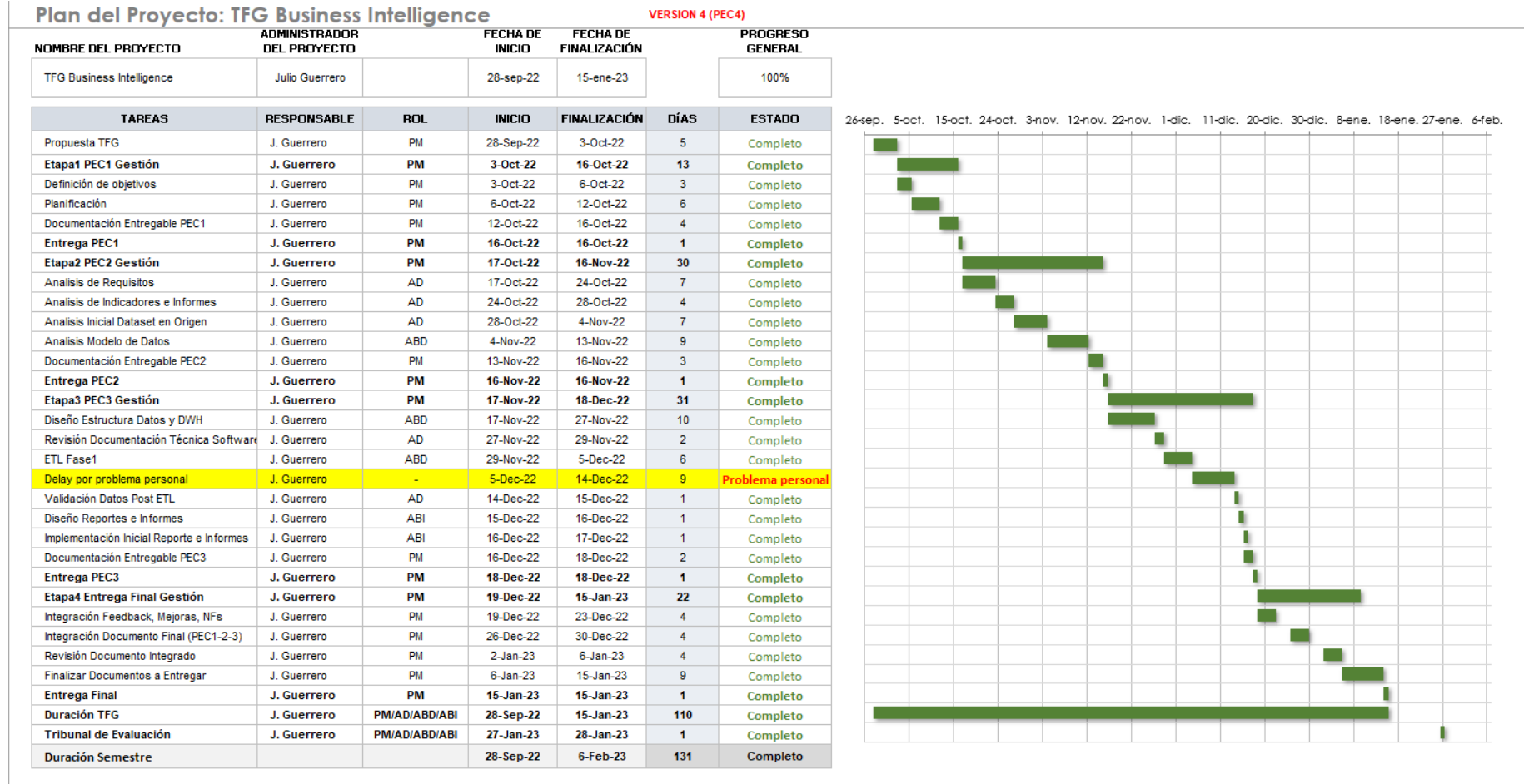


Figura 1. Planificación del Proyecto – Gantt (Versión 4). Fuente: Elaboración Propia

1.4.5 Recursos técnicos

Para desarrollar el proyecto usaremos íntegramente un ordenador portátil personal con las siguientes características:

- Procesador - Intel Skylake Core i7-6700HQ CPU, quad-core 2.6 GHz (3.5 GHz TBoost)
- Memoria - 32 GB DDR3 RAM
- Tarjeta gráfica - Integrated Intel HD 530 + Nvidia GTX 960M 2GB
- Disco Duro - SSD 2 TB

1.5 Breve resumen de productos obtenidos

Al finalizar este Proyecto se espera obtener los siguientes productos finales:

- Memoria final del TFG
- Ficheros multimedia y de presentación asociados con lo requerido para dar por finalizado el TFG.
- Conjunto de datos finales, en texto plano, empaquetado en un fichero de tipo zip.
- Ficheros de tipo pibx con los cuadros de mando final, indicadores y reportes finales generados.

1.6 Breve descripción de los otros capítulos de la memoria

Asociado al desarrollo de este Proyecto se espera obtener los siguientes capítulos para parte de la memoria final:

Etapa 2 – PEC2

- Capítulo 2 – Análisis de requisitos, indicadores e informes, conjunto de datos en origen y del modelo de datos a adoptar para llevarlo al almacén de datos.

Etapa 3 – PEC3

- Capítulo 3 – Diseño de la Estructura de datos final y del repositorio que usaremos como almacén de datos, desde el cual se lanzarán luego las consultas que provisionarán de información al análisis visual.

- Capítulo 4 – Elección de las herramientas de la Solución.
- Capítulo 5 – Descripción del proceso ETL y validación de los datos posterior a este proceso e Implementación del contenido de Reportes, indicadores e informes, se incluirá el detalle visual, paleta de colores a usar.
- Capítulo 6 – Bibliografía
- Capítulo 7 – Anexo compuesto por un Resumen de los aspectos técnicos de las tecnologías usadas en este proyecto.

Etapas 4 – Entrega final

- Capítulo 8 – Conclusiones del proyecto, lecciones aprendidas y experiencias/reflexiones personales, en caso de aplicar, también se agregaría un capítulo con mejoras o posibles cambios que provengan de la retroalimentación del profesor.

2. Análisis de la Solución

Este proyecto tal y como se ha definido en los capítulos iniciales, tendrá como punto de partida el volcado de datos, por lo que el procedimiento se empezará a desarrollar enfocándose en estos datos existentes en una base de datos relacional, al que procederá un volcado de estos a un almacén de datos que se implementará en local, en este caso en el ordenador portátil definido en el punto 1.4.5 de este documento.

Posterior a este procedimiento, el cual se llevará a cabo aplicando un proceso de ETL (Extracción, Transformación y Carga), siendo este el que se encargará de la transformación de los datos a información, haciendo una transición entre lo que nos proporciona la capa de operaciones de negocios y la información que vamos a empezar a utilizar en la inteligencia de negocios.

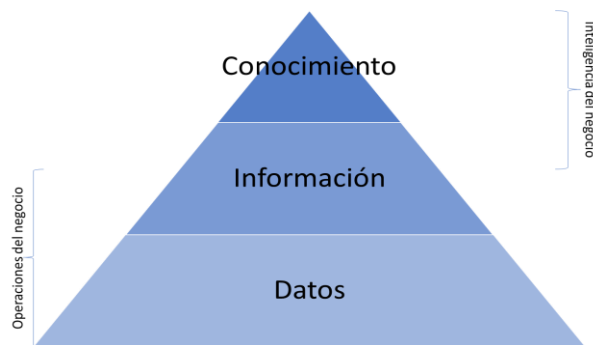


Figura 2. Flujo y transformación a Conocimiento entre las Capas del Negocio. Fuente: Elaboración Propia

En este punto del análisis interviene el conocimiento del negocio y también influirá el conocimiento técnico, para esto se intentará plasmar gráficamente la posición funcional que cumplirá nuestro proyecto y los componentes de los que dependerá:

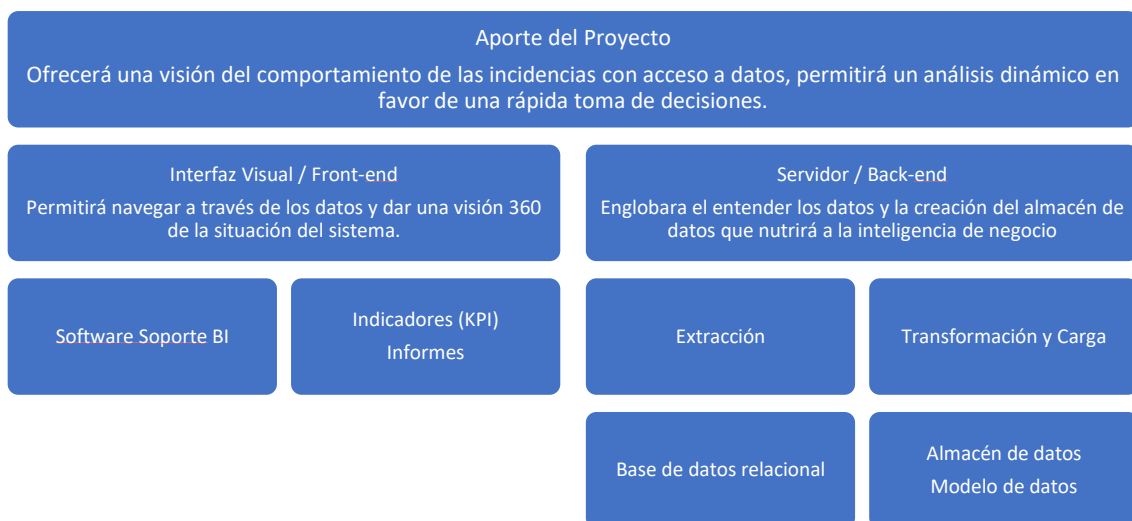


Figura 3. Componentes Análisis Funcional⁴ del Proyecto Fuente: Elaboración Propia

4 Diagrama de bloques funcional. (2022). En Wikipedia, la enciclopedia libre. https://es.wikipedia.org/w/index.php?title=Diagrama_de_bloques_funcional&oldid=141197037

Por lo que, para que el resultado final de estos procedimientos esté en concordancia con lo esperado, tenemos que realizar una fase de análisis en los que se deberá tener en cuenta los requisitos funcionales, no funcionales, los indicadores e informes que deseamos obtener, así como un análisis del modelo de datos del que provienen los datos, para entender el origen y del modelo de datos final del que se nutrirá el cuadro de mando.

2.1 Análisis de Requisitos

Este apartado tiene como objetivo determinar, puntualizar y detallar claramente los requerimientos funcionales y no funcionales con los que el proyecto se deberá implementar, por lo que, este mismo servirá de base para el desarrollo de este.

2.1.1 Valoraciones

Relevancia: Sera la necesidad que le otorguemos a este requisito con relación a la prioridad en la implementación en el proyecto, posibles valores: {A=Alta, M=Media y B=Baja}

Necesidad: Sera la categorización que le demos a este requisito con relación a si es necesario o exigible en la implementación en el proyecto, posibles valores: {N=Necesario y P=Prescindible}

2.1.2 Requisitos funcionales

Se detallarán aquellos requisitos funcionales para la implementación de este proyecto, estos tendrán asociados una descripción, relevancia y necesidad de este.

ID	Protección de datos	Relevancia	Necesidad
RBI-F01		A	N
Descripción			
Los datos volcados al almacén de datos no tendrán información sensible ni comercial.			
Detalle			
Se eliminará en el procedimiento de ETL todo dato sensible relacionado con nombres personales y nombres comerciales con el fin de cumplir con la seguridad y protección de datos del sistema.			

ID	Tiempo de datos a volcar	Relevancia	Necesidad
RBI-F02		A	N
Descripción			
Los datos volcados tendrán una antigüedad de máximo 3 años.			
Detalle			
Este tiempo se estima más que suficiente para poder realizar el visualizado de datos.			

ID	Integración de distintas fuentes	Relevancia	Necesidad
RBI-F03		A	N
Descripción			
El sistema permitirá la integración de distintos orígenes de datos.			
Detalle			
Los datos que no se puedan importar directamente de la base de datos, que necesiten ser filtrados o trabajados con otros medios, ya sea en csv, txt u otro formato, se deben poder integrar en el almacén de datos.			

ID	Extracción de los datos	Relevancia	Necesidad
RBI-F04		A	N
Descripción			
El proyecto debe cumplir con la correcta extracción de la data de la base de datos relacional de origen.			
Detalle			
Se deberá hacer un filtrado correcto de las tablas que puedan aportar valor al análisis de negocio, incluyendo validaciones para evitar duplicidad de datos, de manera que la extracción sea lo menos costosa en tiempo y recursos.			

ID	Transformación de los datos	Relevancia	Necesidad
RBI-F05		A	N
Descripción			
El sistema debe cumplir con la correcta transformación de los datos previo a la inserción en el almacén de datos.			
Detalle			
Los datos se deberán tratar correctamente para poder aplicar la transformación de transaccionales a que pueden ser procesados analíticamente en el almacén de datos, este requisito estará asociado al de seguridad de datos (RBI-F01).			

ID	Carga de los datos	Relevancia	Necesidad
RBI-F06		A	N
Descripción			
El sistema debe cumplir con la correcta carga de la información en el almacén de datos.			
Detalle			
La información aun sin trabajar se debe cargar en los cubos correctamente de manera que los datos queden correctamente ubicados en la en la estructura de datos multidimensional.			

ID	Visualización centralizada de la información	Relevancia	Necesidad
RBI-F07		A	N

Descripción	El sistema debe mostrar la información lo más centralizada posible, evitar cambios de pantalla innecesariamente.
Detalle	La información en conjunto es una herramienta muy potente y aumenta la percepción e interpretación de esta. De lo contrario se pierde parte de la visión de los datos cuando se tiene que mover la barra de desplazamiento o cambiar a otra pantalla para ver información que está relacionada entre sí.

ID	Evitar la duplicidad de la información	Relevancia	Necesidad
RBI-F08		A	N
Descripción	El sistema debe evitar tener información duplicada, ya sea a nivel de registros, tablas o cubos multidimensionales.		
Detalle	Evitar estas duplicidades hará más óptimo el uso de recursos como espacio.		

2.1.3 Requisitos no funcionales

Se detallarán aquellos requisitos no funcionales asociados a este proyecto, estos serán los asociados a la usabilidad, tecnología y software usado y tendrán asociados una descripción, relevancia para el mismo.

ID	Visualización centralizada de la información	Relevancia	Necesidad
RBI-NF01		A	N
Descripción	El sistema debe mostrar la información lo más centralizada posible, evitar cambios de pantalla innecesariamente.		
Detalle	La información en conjunto es una herramienta muy potente y aumenta la percepción e interpretación de esta. De lo contrario se pierde parte de la visión de los datos cuando se tiene que mover la barra de desplazamiento o cambiar a otra pantalla para ver información que está relacionada entre sí.		

ID	Uso de paleta de colores accesible	Relevancia	Necesidad
RBI-NF02		M	P
Descripción	El sistema deberá en tener en cuenta usar una paleta de colores accesibles.		
Detalle	Siendo el color unos de los aspectos más relativos cuando estamos visualizando información, se tendrá que, en caso se		

	pueda, tener en cuenta el color al momento de diseñar las presentaciones para cuidar la accesibilidad de estas.
--	---

ID	El coste cero de los recursos usados.	Relevancia	Necesidad
RBI-NF03		A	P
Descripción	El sistema en lo posible no deberá usar licencias de pago.		
Detalle	En la medida de lo posible todo el sistema se tendrá que hacer con licencias libres o software cuya licencia venga incluida en alguna suscripción ya existente.		

2.2 Análisis de Indicadores e Informes

Este punto será fundamental para obtener los resultados esperados del proyecto, y estará ligado directamente con la explotación de la información que será parte central del proyecto, el acceso a los datos si bien se centrara en ser obtenida del repositorio central de incidencias, puede también darse por otros ficheros distintos y externos a esta.

Del conocimiento técnico con el que se cuenta finalmente se decide catalogar la información resultante en las siguientes tipificaciones y se describirán los indicadores principales para cada una de estas.

2.2.1 Datos y contexto de los indicadores

Es necesario puntualizar el contexto con el que se analizaran la información de los indicadores y algunos términos que se usaran dentro de la definición de estos, a continuación, los comentaremos:

Característica	Valores
Granularidad	Día Semana Mes Trimestre Semestre Año
Rango de tiempo de los datos a cargar	3 años.
Se permitirá comparativa entre años	Sí.
Actualización de los datos	2 tiempos, carga inicial y una actualización final.
Criticidad	Crítica Alta Media Baja
Estados	Abierto Pendiente

	Resuelta Cerrada Rechazada
SLA	Crítica: 2 horas Alta: 5 horas Media: 5 semanas Baja: No Tiene
Nodos	Tarificador: Encargado de todo lo relacionado con la tarificación y el saldo en tiempo real. Gestor: Encargado de todo lo relacionado con el ciclo de vida del cliente y actividad de compras en el sistema.
Operadoras	Argentina Uruguay Colombia Perú Venezuela Ecuador Costa Rica México El Salvador Alemania

2.2.2 Indicadores relacionados con criticidad/estado de las incidencias

2.2.2.1 Indicador de Incidencias por Criticidad/Estado Global

IND-IN-INC001	
Objeto	Conocer la evolución en cuanto a cantidad de las distintas criticidades/estados que se manejan en la plataforma.
Objetivo	Analizar globalmente los picos enfocados en la cantidad de incidencias críticas que son las que menos tiempo de SLA (2 horas) tienen.
Dimensiones	Agrupado y sumatoria (conteo de incidencias) sin filtro de nodos, sin filtro de Operadora, sin filtro de estado
Granularidad	Anual, Mensual, Trimestral
Nivel Optimo	Lo más cercano a cero.
Fuente de datos	Base de datos relacional

2.2.2.2 Indicador de Incidencias por Criticidad/Estado filtrado por Operadora

IND-IN-INC002	
Objeto	Conocer la evolución en cuanto a cantidad de las distintas criticidades/estados por operadora que se manejan en la plataforma.

Objetivo	Analizar globalmente los picos enfocados en la cantidad de incidencias relacionándolas directamente por cada operadora de esta plataforma.
Dimensiones	Agrupado y sumatoria (conteo de incidencias) sin filtro de nodos, con filtro de Operadora, sin filtro de estado
Granularidad	Anual, Mensual, Trimestral
Nivel Optimo	Lo más cercano a cero.
Fuente de datos	Base de datos relacional

2.2.2.3 Indicador de Incidencias por Criticidad/Estado filtrado por Nodo

IND-IN-INC003	
Objeto	Conocer la evolución en cuanto a cantidad de las distintas criticidades/estados por nodo que se manejan en la plataforma.
Objetivo	Analizar globalmente los picos enfocados en la cantidad de incidencias relacionándolas directamente por cada nodo de esta plataforma.
Dimensiones	Agrupado y sumatoria (conteo de incidencias) con filtro de nodo, sin filtro de operadora, sin filtro de estado
Granularidad	Anual, Mensual, Trimestral
Nivel Optimo	Lo más cercano a cero.
Fuente de datos	Base de datos relacional

2.2.2.4 Indicador Incidencias Reabiertas

IND-IN-INC004	
Objeto	Conocer la evolución en cuanto a cantidad de las incidencias reabiertas.
Objetivo	Analizar globalmente posibles fallos en el proceso de atención de incidencias.
Dimensiones	Agrupado y sumatoria (conteo de incidencias) con filtro de operadora.
Granularidad	Anual, Mensual, Trimestral
Nivel Optimo	Lo más cercano a cero.
Fuente de datos	Base de datos relacional

2.2.2.5 Indicador Incidencias que han generado parche de la plataforma

IND-IN-INC005	
Objeto	Conocer la cantidad de incidencias con parche para solventar el problema

Objetivo	Analizar globalmente y filtrando los fallos de la plataforma con coste implícito a desarrollo.
Dimensiones	Agrupado y sumatoria (conteo de incidencias) con filtro de operadora.
Granularidad	Anual, Mensual, Trimestral
Nivel Optimo	Lo más cercano a cero.
Fuente de datos	Base de datos relacional

2.2.3 Indicadores relacionados con origen de las incidencias

2.2.3.1 Indicador Incidencias Causa Raíz no asociado a la Plataforma

IND-EX-INC001	
Objeto	Conocer la evolución en cuanto a cantidad de incidencias no asociadas al software de la plataforma
Objetivo	Analizar globalmente las incidencias cuya causa no es un problema de la plataforma
Dimensiones	Agrupado y sumatoria (conteo de incidencias) con filtro de operadora
Granularidad	Anual, Mensual, Trimestral
Nivel Optimo	Lo más cercano a cero.
Fuente de datos	Base de datos relacional

2.2.4 Indicadores relacionados el tiempo de solución

2.2.2.5 Indicador de Incidencias sin cerrar en lapsos de tiempo (>1 mes)

IND-TI-INC001	
Objeto	Conocer el comportamiento de aquellas incidencias que no se han cerrado en más de un mes.
Objetivo	Analizar globalmente la tendencia de aquellas incidencias con más de 1 mes en el sistema y que no han sido cerradas.
Dimensiones	Agrupado y sumatoria (conteo de incidencias) sin filtro de nodos, con filtro de país, con filtro de estado
Granularidad	Anual, Trimestral, Mensual
Nivel Optimo	Lo más cercano a cero.
Fuente de datos	Base de datos relacional

2.2.3.1 Indicador Incidencias sobrepasado el SLA⁵ (ANS)

IND-TI-INC002	
Objeto	Conocer la evolución de las incidencias en las que no se ha cumplido con el SLA.
Objetivo	Analizar globalmente las incidencias las cuales no se cumple el tiempo de solución de acuerdo con su criticidad.
Dimensiones	Agrupado y sumatoria (conteo de incidencias) con filtro de operadora y con filtro de criticidad.
Granularidad	Anual, Mensual, Trimestral
Nivel Optimo	Lo más cercano a cero.
Fuente de datos	Base de datos relacional

2.2.3.1 Indicador Incidencias tiempo promedio de resolución del problema

IND-TI-INC003	
Objeto	Conocer tiempos de respuesta de incidencia.
Objetivo	Analizar globalmente el tiempo de respuesta de las incidencias de acuerdo con su criticidad y operadora.
Dimensiones	Agrupado y promedio con filtro de operadora y con filtro de criticidad.
Granularidad	Mensual
Nivel Optimo	Lo esperado de acuerdo con cada SLA.
Fuente de datos	Base de datos relacional

2.4 Análisis Inicial del Conjunto de Datos de Origen

Para este punto del proyecto y para entender el contexto de la estructura de la base de datos que vamos a explotar y después analizar, describiremos brevemente la plataforma que la genera.

Esta está compuesta de dos grandes módulos que forman el núcleo de toda la plataforma:

- El módulo **Gestor (1)** que es el que se encarga de llevar el ciclo de vida, adquisición de productos (recarga, bono, promoción de acuerdo con ciertos criterios) y repositorio de transacciones hechas de los suscriptores de telefonía móvil.
- El módulo **Tarificador (2)** que es el que se encarga de llevar la tarificación y recarga en tiempo real de los suscriptores de telefonía móvil.

Las incidencias que sean iniciadas por las operadoras, sin importar el módulo **(1 o 2)**, se abren desde una interfaz web y esta es la finalmente es la que alimentara la base de datos en la que se basara este proyecto.

5 Acuerdo de nivel de servicio. (2022). En Wikipedia, la enciclopedia libre. https://es.wikipedia.org/w/index.php?title=Acuerdo_de_nivel_de_servicio&oldid=146797771

A continuación, veremos gráficamente la arquitectura que se ha mencionado previamente.

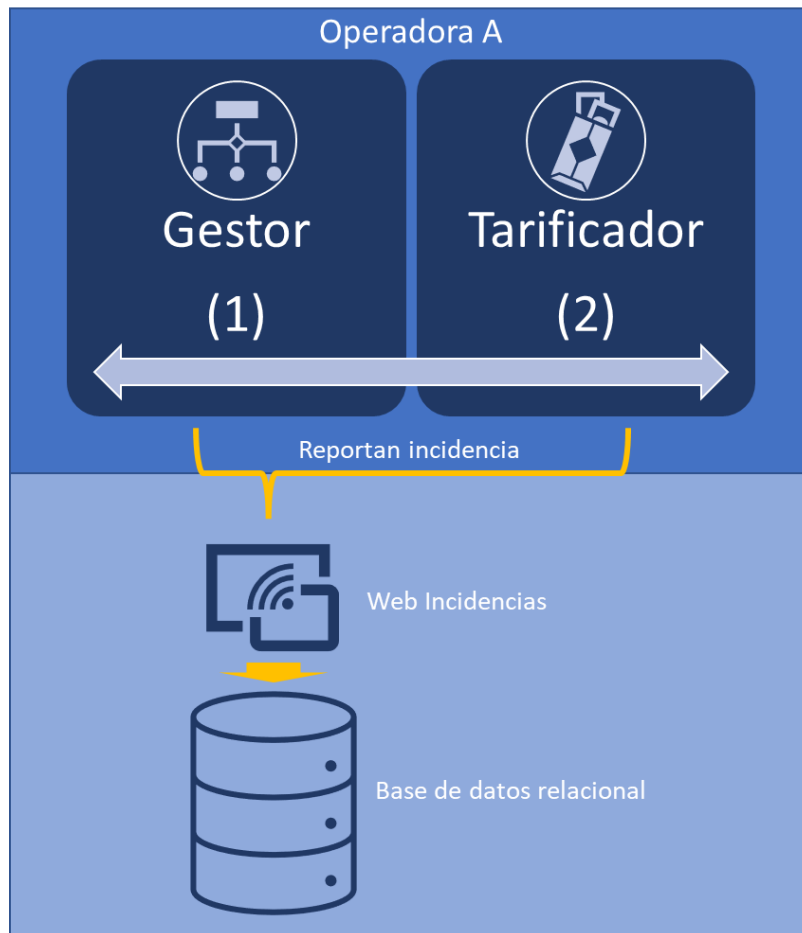


Figura 4. Arquitectura Plataforma. Fuente: Elaboración Propia

La Web de Incidencias, no solo se encarga de la gestión y ciclo de vida de las incidencias, sino también con módulos internos que se encargan de la gestión de consultas generadas por la operadora móvil, y otros módulos como el de seguridad y gestión de usuarios y roles, pero a efectos de este proyecto solo se enfocara el análisis en el módulo de incidencias, cabe recalcar que es donde más puedo entender el funcionamiento debido a los conocimientos adquiridos.

A partir de esta introducción se pasará a explicar la estructura de la base de datos que soporta la data que se almacena una vez reportada una incidencia por parte de una operadora móvil, esto servirá de contexto para obtener las tablas principales a extraer y que luego serán llevadas al almacén de datos que se generara en local.

A continuación, se plasmará una breve descripción de las tablas que, en base al conocimiento de la plataforma, serán las necesarias de exportar para obtener la información que luego visualizaremos en los informes:

2.4.1 Tablas necesarias para obtener el estatus de incidencias

Tabla	INCIDENCIAS
<i>Descripción y campos importantes</i>	Encargada de registrar la incidencia en primera instancia y la que más relaciones tendrá, se listarán algunos campos importantes de esta tabla: Id de la incidencia. Id del usuario responsable. Id de la operadora móvil.

Tabla	CICLO_DE_VIDA
<i>Descripción y campos importantes</i>	Encargada de llevar el ciclo de vida de las incidencias, se listarán algunos campos importantes de esta tabla: Id del ciclo de vida. Id del estado. Id del usuario responsable. Id de la incidencia.

Tabla	ANALISIS
<i>Descripción y campos importantes</i>	Encargada de registrar algunos cambios y relaciones con estados y si esta incidencia ha sido elevada de categoría por ser prioritaria, se listarán algunos campos importantes de esta tabla: Id de la incidencia. Id de análisis de incidencia. Valor si se ha escalado. Valor si se ha repetido antes.

2.4.2 Tablas complementarias para los KPI e Informes

Tabla	USUARIO
<i>Descripción y campos importantes</i>	Encargada de tener la información de los usuarios que podrán interactuar con las incidencias, se listarán algunos campos importantes de esta tabla: Id del usuario responsable. Id de la operadora móvil.

Tabla	GRUPO
<i>Descripción y campos importantes</i>	Encargada de tener la información de los grupos generadoras y receptores o encargados de las incidencias, se listarán algunos campos importantes de esta tabla: Id del grupo. Nombre del grupo.

Tabla	VERSION
--------------	----------------

<i>Descripción y campos importantes</i>	Encargada de tener la información de la versión del aplicativo a la que se asociara la incidencia, se listarán algunos campos importantes de esta tabla: Id del grupo. Nombre del grupo.
---	--

Tabla	CAT_ESTADO
<i>Descripción y campos importantes</i>	Sirve de apoyo para asociar los distintos estados por los que pasara la incidencia, se listarán algunos campos importantes de esta tabla: Id del estado. Nombre del estado.

Tabla	CAT_SUBESTADO
<i>Descripción y campos importantes</i>	Sirve de apoyo para asociar los distintos sub-estados por los que pasara la incidencia, se listarán algunos campos importantes de esta tabla: Id del sub-estado. Id del grupo al que puede pertenecer este sub-estado.

Tabla	CAT_PAIS
<i>Descripción y campos importantes</i>	Sirve de apoyo para asociar las distintas operadoras móviles a las que se pueden asociar la incidencia, se listarán algunos campos importantes de esta tabla: Id del país. Código del país. Nombre del país.

Tabla	SUBCOMPONENTE
<i>Descripción y campos importantes</i>	Sirve de apoyo para asociar los distintos componentes a las que se pueden asociar la incidencia, se listarán algunos campos importantes de esta tabla: Id del país. Código del componente. Nombre del componente.

Tabla	CAT_SUBCOMPONENTE
<i>Descripción y campos importantes</i>	Sirve de apoyo para asociar los distintos subcomponentes a las que se pueden asociar la incidencia, se listarán algunos campos importantes de esta tabla: Id del país. Código del subcomponente. Nombre del subcomponente.

Tabla	CAT_DETECTADO
<i>Descripción y campos importantes</i>	Sirve de apoyo para asociar en donde ha sido detectada esta incidencia, si pertenece a entornos prueba, producción, desarrollo, se listarán algunos campos importantes de esta tabla: Id de entorno detectado. Nombre de entorno detectado.

Tabla	PLATAFORMA
<i>Descripción y campos importantes</i>	Sirve de apoyo para asociar en donde ha sido detectada esta incidencia, si pertenece a entornos prueba, producción, desarrollo, se listarán algunos campos importantes de esta tabla: Id de entorno detectado. Nombre de entorno detectado.

2.4.3 Tablas complementarias para asociar parches

Con estas se obtendrá las relaciones entre los datos necesarios para asociar la incidencia con su respectiva entrega del grupo de Desarrollo (parche).

Tabla	PARCHES_DATOS_COMP
<i>Descripción y campos importantes</i>	Esta tendrá las asociaciones entre incidencias y entrega de parches asociado a esta, se listarán algunos campos importantes de esta tabla: Id de entrega. Id de Incidencia.

Tabla	PARCHES_DATOS
<i>Descripción y campos importantes</i>	Esta tendrá las asociaciones entre incidencias y entrega de parches asociado a esta, se listarán algunos campos importantes de esta tabla: Id de entrega. Id de Incidencia.

2.4.4 Tablas complementarias para la causa raíz

Con estas se obtendrá las relaciones entre los datos necesarios para asociar la incidencia con la posible causas raíz que ya están siendo tratadas o en estatus "cerrado".

Tabla	ACEPTACION
<i>Descripción y campos importantes</i>	Esta tendrá las asociaciones entre incidencias y entrega de parches asociado a esta, se listarán algunos campos importantes de esta tabla: Id de incidencia.

Id de aceptación.

Tabla

CAT_ACEPTACION

Descripción y campos importantes

Esta tendrá la categorización de los tipos de causas, se listarán algunos campos importantes de esta tabla:
Id de aceptación.
Categoría de causa raíz.

2.4.5 Tablas complementarias para medir la afectación

Con estas se obtendrá las relaciones entre los datos necesarios para asociar la incidencia con la posible afectación de las incidencias, estas una vez hayan sido analizadas

Tabla

CAT_TIPO_SERVICIO_AFECTADO

Descripción y campos importantes

Esta tendrá las asociaciones entre incidencias y entrega de parches asociado a esta, se listarán algunos campos importantes de esta tabla:
Id de Tipo de pérdida de servicio.
Descripción de pérdida de servicio.

2.4.6 Tablas complementarias para asociar MOP⁶

Con estas se obtendrá las relaciones entre los datos necesarios para asociar la incidencia con el MOP asociado para resolver o analizar esta.

Tabla

INCIDENCIA_PROCEDIMIENTO

Descripción y campos importantes

Guardara la información relacionada con el MOP, se listarán algunos campos importantes de esta tabla:
Id del MOP.
Id de incidencia.
Fechas de aceptación o rechazo.

Tabla

ESTADO_PROCEDIMIENTO

Descripción y campos importantes

Esta categorizara los posibles estados del MOP, se listarán algunos campos importantes de esta tabla:
Id de estado MOP.
Descripción de estado MOP.

⁶ What is a Method Of Procedure (MOP)? - Definition from Techopedia. (s. f.). Techopedia.Com. Recuperado 11 de noviembre de 2022, de <http://www.techopedia.com/definition/31833/method-of-procedure-mop>

Tabla**RECHAZO_PROCEDIMIENTO***Descripción y campos importantes*

Esta categorizara los posibles motivos de rechazo del MOP, se listarán algunos campos importantes de esta tabla:
Id de motivo rechazo MOP.
Descripción de motivo rechazo MOP.

Tabla**CAT_PERIODO***Descripción y campos importantes*

Esta categorizara la información del tipo de MOP y su estatus con respecto al tiempo de respuesta de la Operadora, se listarán algunos campos importantes de esta tabla:
Id de Tipo Periodo MOP.
Descripción de Tipo Periodo MOP.
Periodo de evaluación.

A continuación, se plasmará el Diagrama de Entidad Relación de dos escenarios que me parece importante tener en cuenta sobre todo para el desarrollo a posteriori de las tablas de hechos y dimensiones, el primer diagrama será el de las tablas que interaccionan en el flujo del ciclo de vida de la incidencia, y el segundo será el del flujo de las incidencias cuando tienen un MOP asociado y su interacción con este.

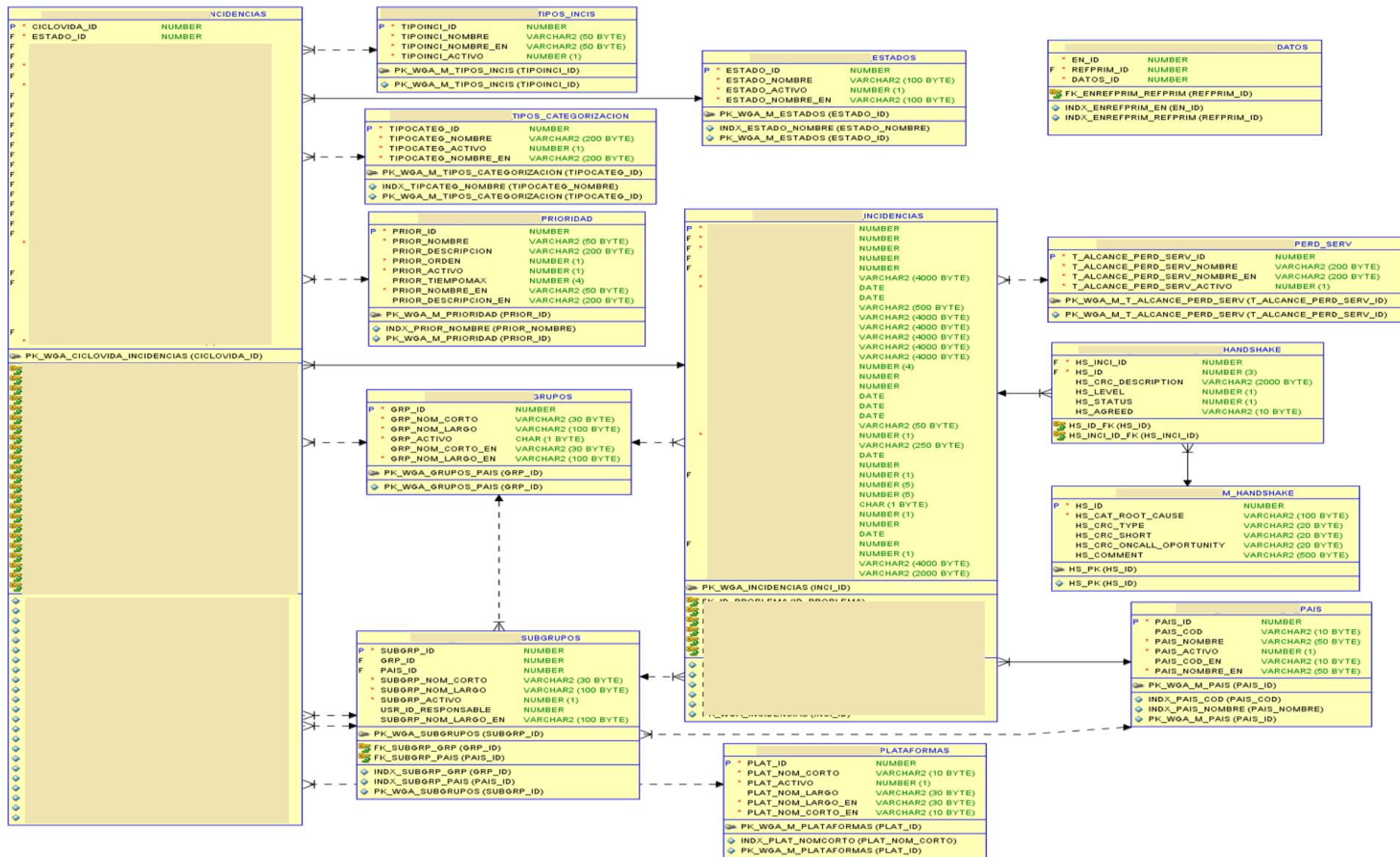


Figura 5. Diagrama Entidad Relación (Flujo Incidencias). Fuente: Elaboración Propia

2.5 Análisis de Modelo de Datos

Este punto del proyecto sentara la base del futuro diseño del almacén de datos que desarrollaremos en local, para esto lo que tendremos que entender es la necesidad que nos plantea nuestro proyecto, se tendrá que entender que las bases de este son, analizar el comportamiento de las incidencias en base a una serie de necesidades a desarrollar y el cómo cubrir estas viene dentro de la base de datos relacional de la cual extraeremos la data, estos parámetros son:

<i>Necesidad 1</i>	La relación de las incidencias con el estado del ciclo de vida en el que se encuentran.
<i>Necesidad 2</i>	La relación de las incidencias con el nodo al que están asociadas.
<i>Necesidad 3</i>	La relación de las incidencias con la criticidad con la son generadas.
<i>Necesidad 4</i>	La relación de las incidencias con la operadora (referencia directa a un país o zona geográfica) que ha generado o sobre el cual inciden estas.
<i>Necesidad 5</i>	La relación de las incidencias con la versión del aplicativo al que se asocian estas.
<i>Necesidad 6</i>	La relación de las incidencias con la causa raíz que las esté generando.
<i>Necesidad 7</i>	Analizar el cumplimiento de los SLA.

Asociado a estas necesidades estarán los siguientes parámetros que entran en la ecuación y que tendrán que ser tomados en cuenta para poder obtener los parámetros antes mencionados, en este caso son:

<i>Tiempo (granularidad)</i>	Se deberá tener un rango para poder agrupar los datos comprendido entre días, semanas, meses, trimestres, semestres hasta el año.
------------------------------	---

<i>Detalle Incidencias</i>	Se contará con los detalles relacionados a las incidencias.
<i>Ciclo de Vida Incidencias</i>	Se contará con los detalles relacionados al ciclo de vida de las incidencias.
<i>Operadora</i>	Se contará con los datos relacionados a que operadora (asociada a un país o zona geográfica) que genera una incidencia.
<i>Parche</i>	Se contará con los datos relacionados entre incidencia y parche de aplicativo.
<i>Causa Raíz</i>	Se contará con los datos relacionados al motivo asociado como causa raíz de la incidencia.
<i>Afectación</i>	Se contará con los datos relacionados la afectación que haya podido generar la incidencia en la Operadora.

Si bien hay muchas formas diferentes de organizar los datos para su análisis, el modelo de datos por el que optaremos es el **esquema de estrella**⁷. A continuación, se explicará de que consta este modelo, cómo se utilizará y las ventajas de usarlo.

2.5.1 Modelo de Estrella

El modelo de esquema en estrella organiza los datos en uno o varios hechos centrales rodeados de dimensiones. Se trata de un tipo de estructura de base de datos que se utiliza habitualmente para los almacenes de datos y la inteligencia de negocios. En este caso una única tabla de gran tamaño llamada tabla de "hechos" almacena todos los datos transaccionales o medibles, mientras que múltiples tablas dimensionales más pequeñas almacenan atributos sobre los datos.

Se llama esquema en estrella porque en el centro del diagrama están la tabla de hechos y sus tablas más pequeñas ramificadas, formando lo que parece una estrella.

A continuación, veremos gráficamente lo que comenta este párrafo

⁷ Star schema. (2022). En Wikipedia. https://en.wikipedia.org/w/index.php?title=Star_schema&oldid=1112456372

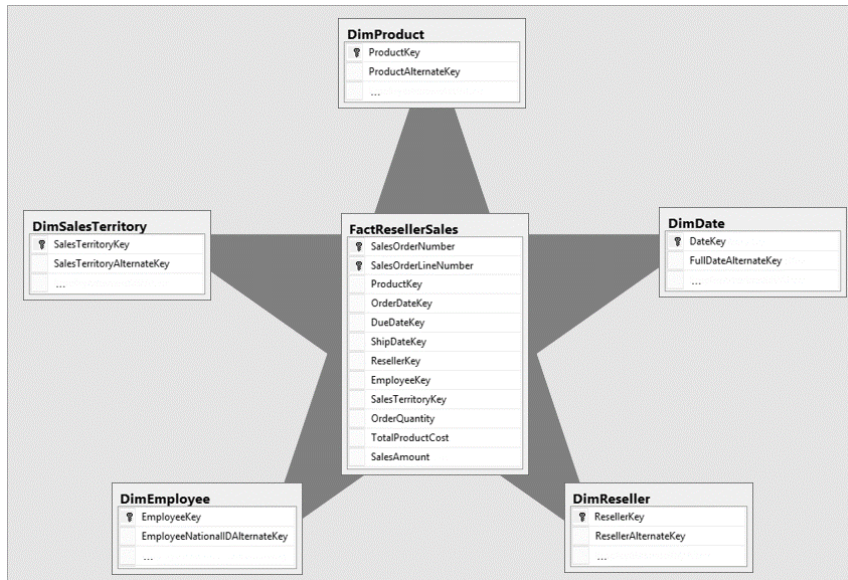


Figura 7. Esquema del Modelo en Estrella. Fuente: <https://learn.microsoft.com/en-us/power-bi/guidance/star-schema>

2.5.2 Funcionamiento del modelo

Lo que se aprecia en el centro del gráfico sería la tabla de hechos, y es donde realmente se encuentran los datos específicos que se quiere analizar, teniendo en cuenta que, dentro de esta tabla, tendremos dos tipos de datos, numéricos y atributos. A diferencia de otros tipos de tablas, cada base de datos de esquema en estrella tiene una sola tabla de hechos y el contenido de esta habrá que centrarlo en lo que queremos analizar.

Por otro lado, tendremos los valores de los atributos de las dimensiones, que no almacenan directamente los datos, sino que contienen el valor de la clave externa en una tabla dimensional relacionada. Muchas filas de la tabla de hechos hacen referencia a este tipo de información, como podrían ser los ID de las incidencias, los valores de las fechas, los ID de los componentes o los ID de las plataformas.

Las tablas de dimensiones almacenan información de apoyo que usará directamente la tabla de hechos, es decir, cada base de datos de esquema en estrella tendrá al menos una tabla de dimensiones, aunque en nuestro proyecto serán muchas. Cada tabla de dimensiones se relacionará con una columna de la tabla de hechos con un valor de dimensión, y almacenará información adicional sobre ese valor.

2.5.3 Ventajas de usar este modelo

Uno de los puntos a favor que encontraremos usando este modelo será que es un esquema desnormalizado, lo que significa que las reglas de normalización aplicadas a las bases de datos transaccionales son menos duras durante la fase de diseño e implementación de estos esquemas.

Esta relajación de la normalización se convertirá en:

- Consultas más sencillas: la lógica de unión del esquema en estrella es generalmente más sencilla que la lógica de unión necesaria para recuperar datos de un esquema transaccional altamente normalizado.
- Simplificación de la lógica de los informes: en comparación con los esquemas altamente normalizados, el esquema en estrella simplifica la lógica de los informes empresariales más comunes, como los informes por periodos.
- Aumento del rendimiento de las consultas: los esquemas en estrella pueden mejorar el rendimiento de las aplicaciones de información de sólo lectura en comparación con los esquemas altamente normalizados.
- Agregaciones rápidas: siendo las consultas más sencillas en un esquema en estrella esto puede influir en mejorar el rendimiento de las operaciones de agregación.

Si bien en la siguiente etapa del proyecto (DISEÑO) se tendrá el detalle de las dimensiones y las tablas de hecho que vayan a formar parte de la implantación final, sí que me gustaría esbozar el primer diseño lógico, en su versión inicial, de la que será el esquema principal

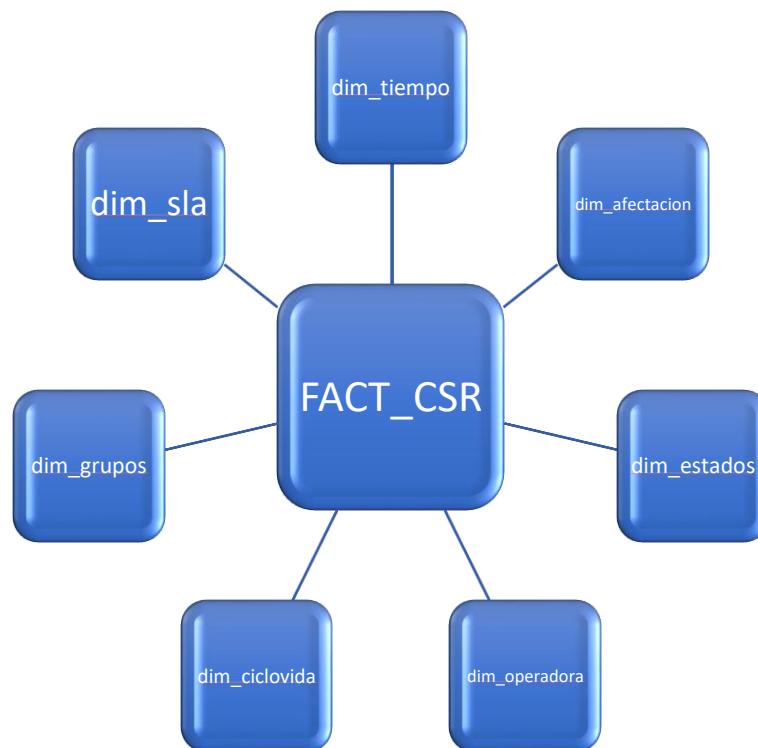


Figura 8. Esquema Inicial Modelo Estrella: Elaboración propia.

3. Diseño de la Solución

3.1 Diseño de la estructura de la base de datos

Este apartado está directamente relacionado con el siguiente subcapítulo asociado al “Diseño del almacén de datos”, debido a que, para poder diseñar el DWH necesitamos tener claramente definido el diseño de nuestra estructura de base de datos.

También cabe retomar momentáneamente el punto 2.5.3, donde se presentaba un esbozo, sin entrar en detalles de la estructura principal que adoptara nuestra base de datos, es a partir de esta que procederemos a desarrollar el diseño de la estructura de datos que aplicaremos en este proyecto.

3.1.1 Estructura OLTP

A continuación, se mostrará el modelo OLTP (actual) y como se relacionan las tablas en las agrupaciones que iremos definiendo y que posteriormente serán las dimensiones o visualmente las puntas del modelo estrella final.

Diseño OLTP de la dimensión CicloVida



Figura 9. OLTP CicloVida. Elaboración propia

La relación con la tabla CICLO_DE_VIDA es de una a muchas y se relaciona por FK con las otras tablas, se entiende la relación como categórica, es decir obtiene datos textuales y es por ello por lo que tiene esta cardinalidad debido a los distintos cambios de estado por los que fluye una incidencia durante su vida.

Diseño OLTP de la dimensión Tiempo

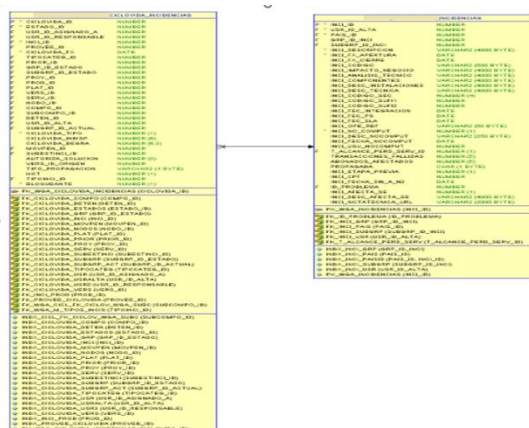


Figura 10. OLTP Tiempo. Elaboración propia

La relación con la tabla CICLO_DE_VIDA es de una a muchas y se relaciona por FK (campo INCI_ID), se entiende que tenga esta cardinalidad ya que el ciclo de vida de una incidencia puede llevar a un mismo INCI_ID a tener varias fechas y tiempos debido a los distintos cambios de estado por los que fluye una incidencia durante su vida.

Diseño OLTP de la dimensión DGrupo

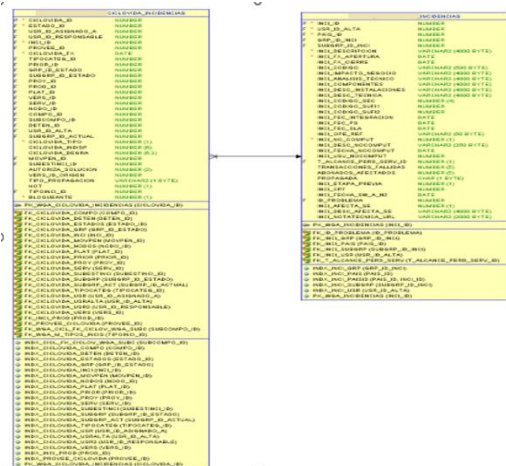


Figura 11. OLTP DGrupo. Elaboración propia

La relación con la tabla CICLO_DE_VIDA es de una a muchas y se relaciona por FK (campo INCI_ID), se entiende que tenga esta cardinalidad ya que, durante el ciclo de vida de una incidencia, esta va a estar en posesión de más de un grupo internamente, desde la generación hasta la resolución de esta.

Diseño OLTP de la dimensión DOperadora



Figura 12. OLTP DOperadora. Elaboración propia

La relación con la tabla CICLO_DE_VIDA es de una a muchas y se relaciona por FK (campo INCI_ID), se entiende que tenga esta cardinalidad ya que, durante el ciclo de vida de una incidencia, esta regresara a la operadora en caso de necesitar más info o programar ventanas de revisión hasta la resolución de esta.

Diseño OLTP de la dimensión ROpen



Figura 13. OLTP ROpen. Elaboración propia

La relación con la tabla CICLO_DE_VIDA es de una a muchas y se relaciona por FK (campo INCI_ID), se entiende que tenga esta cardinalidad ya que, durante el ciclo de vida de una incidencia, esta tendrá la capacidad de ser reabierto, importante para analizar la calidad de resolución de una incidencia.

Diseño OLTP de la dimensión Patch



Figura 14. OLTP Patch. Elaboración propia

No existe relación directa entre la tabla INCIDENCIAS y la de parches ya que son tablas que pertenecen a funcionalidades distintas, pero para obtener el tipo de parche los podremos relacionar por el ID de incidencia para obtener la cantidad de incidencias que han generado impacto/coste en el equipo de desarrollo.

Diseño OLTP de la dimensión RootCause

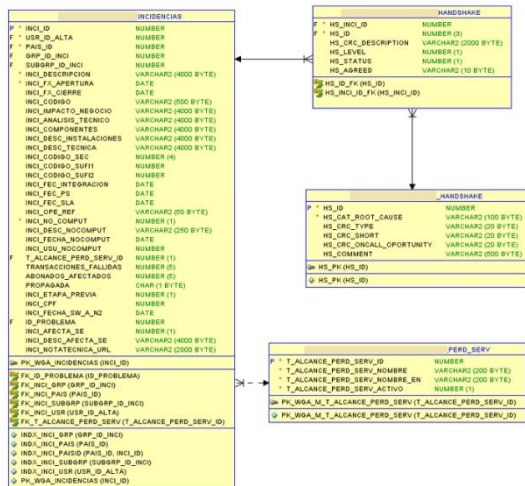


Figura 15. OLTP RootCause. Elaboración propia

La tabla principal será la INCIDENCIAS, se relacionará en una cardinalidad de uno a muchos con la tabla de Handshakes, que será la aceptación de las causas raíz entre el proveedor principal y el grupo de soporte, esta última también necesitará de una tabla categórica para obtener definiciones. Y por otro lado la tabla principal de relaciona otra categórica para darle criticidad a la perdida de servicio.

Diseño OLTP de la dimensión Extra

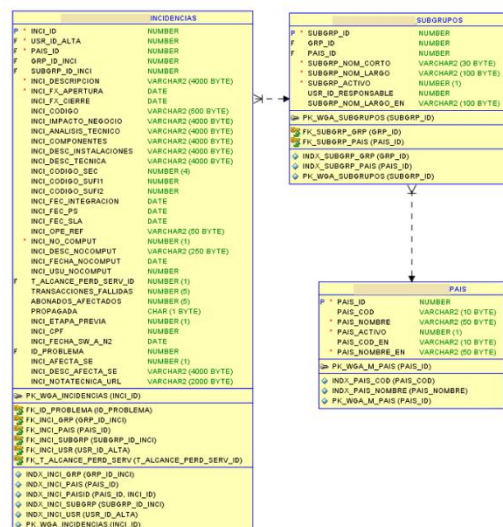


Figura 16. OLTP Extra. Elaboración propia

La tabla principal será la INCIDENCIAS, se relacionará en una cardinalidad de uno a muchos con la tabla de Subgrupos y esta última se relacionada con la de descripción de país, necesaria esta dimensión para obtener la información final de la tabla de hechos y la relación con los países.

3.1.1 Estructura OLAP

Del esbozo del punto 2.5.3 hemos evolucionado el diseño OLAP a este final que mostremos a continuación:

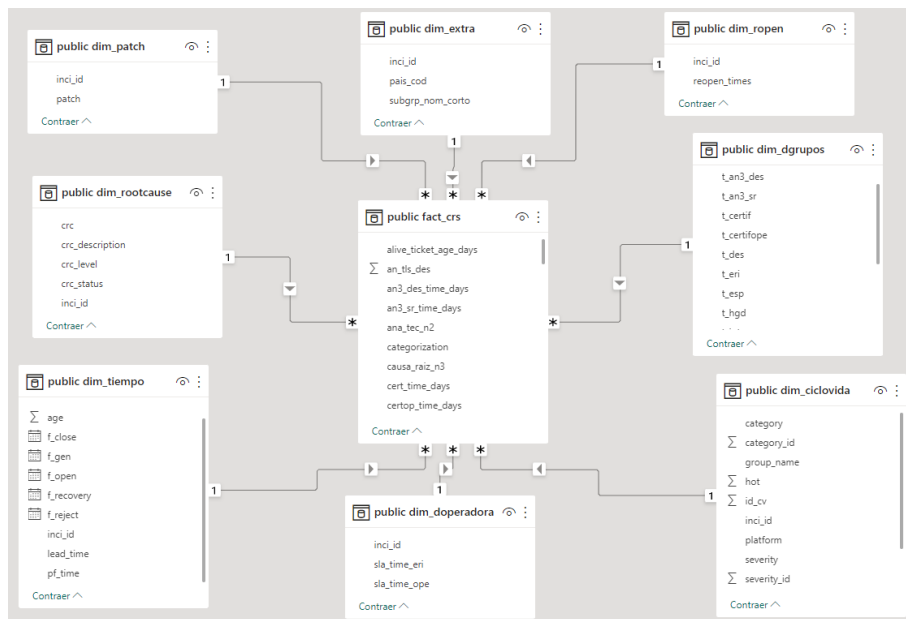


Figura 17. Estructura OLAP. Elaboración propia

De esta estructura obtendremos las distintas dimensiones con las que trabajaremos en nuestra tabla de hechos para poder extraer la data que se analizara en la herramienta Power BI.

Dimensión	Funcionalidad
<i>Dim_Ciclovida</i>	Tendremos el flujo de las incidencias por todos los estados, criticidad, grupo responsable, nodo y fecha de este flujo.
<i>Dim_DGrupos</i>	Tendremos la cantidad de días que ha estado cada incidencia en un grupo específico.
<i>Dim_DOperadora</i>	Tendremos la cantidad de días que ha estado una incidencia entre el grupo que lleva el soporte y la operadora.
<i>Dim_ROpen</i>	Tendremos la relación incidencia y cantidad de veces que se tuvo que reabrir, importante para medir la calidad del servicio de soporte o al menos analizar este punto.
<i>Dim_Patch</i>	Tendremos la relación incidencia y si tiene o no asociado un parche, importante para medir el coste relacionado al grupo de desarrollo, impacta económicamente.
<i>Dim_Rootcause</i>	Tendremos la relación con la causa raíz asociada a cada incidencia, esto aportara a tener una visión de donde realmente se está generando el problema asociado a cada incidencia.

Dim_Extra

Nos dará la relación incidencia operadora, es una dimensión categórica.

Dim_Tiempo

Nos dará las distintas fechas relacionadas con el flujo de la incidencia por su ciclo de vida.

Finalmente, el modelo de estrella quedaría de la siguiente manera:

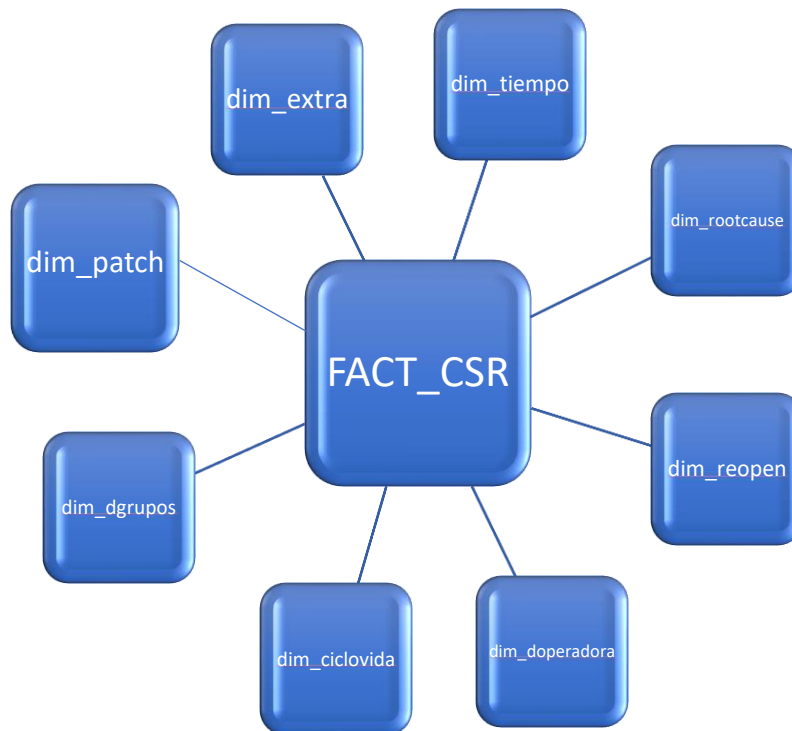


Figura 18. Esquema Final Proyecto Modelo Estrella: Elaboración propia.

3.2 Diseño del Almacén de Datos (DWH)

Un almacén de datos (DWH) es una base de datos diseñada para la consulta y el análisis de datos. Para este proyecto usaremos la metodología desarrollada por Ralph Kimball [\[1\]](#).

Esta se basa en la idea de que los almacenes de datos deben diseñarse para dar soporte a las necesidades empresariales específicas de la organización, y que el diseño del almacén de datos debe estar impulsado por los requisitos de los usuarios empresariales. A continuación, se explicará brevemente los puntos claves de esta metodología y asociarlos a la evolución de este proyecto:

Definir los requisitos empresariales: El primer paso de la metodología Kimball consiste en definir los requisitos de negocio para el almacén de datos. Esto implica trabajar estrechamente con los usuarios de negocio para entender sus necesidades y requisitos, y definir un conjunto claro de objetivos de negocio para la futura explotación de la información obtenida, este punto fue obtenido en la definición de los requisitos descritos en el capítulo 2.

Diseñar el modelo dimensional: El siguiente paso es diseñar el modelo dimensional del almacén de datos. En asociación con nuestro proyecto se optó

por usar el diseño en estrella ya que esté optimizado para un rápido rendimiento de las consultas y con la finalidad de estar alineado con los requisitos.

Poblar el almacén de datos: Una vez diseñado el modelo dimensional, el siguiente paso es llenar el almacén de datos. Para nuestro proyecto esto estará plasmado en el procedimiento que se realizará en el punto 5.1 de este documento, para luego estos datos transformarlos en el formato adecuado y cargarlos en el almacén de datos.

Crear informes y cuadros de mando: El último paso de la metodología Kimball consiste en crear informes y cuadros de mando que permitan a los usuarios acceder a los datos del almacén y analizarlos. Esto implicara el uso de la herramienta de inteligencia empresarial (PowerBI) para crear informes y cuadros de mando que proporcionen información de cómo se está llevando el manejo de las incidencias analizadas.

En general, la metodología Kimball es un enfoque estructurado del almacenamiento de datos que se centra en aportar valor. Siguiendo esta metodología, lo que se busca es garantizar que el almacén de datos está diseñado para satisfacer estas necesidades y que seremos capaces de extraer información valiosa de los datos.

3.2.1 Esquema del almacén de datos

En base a lo expuesto en el capítulo previo, veremos a continuación la estructura que tendrá el almacén de datos:

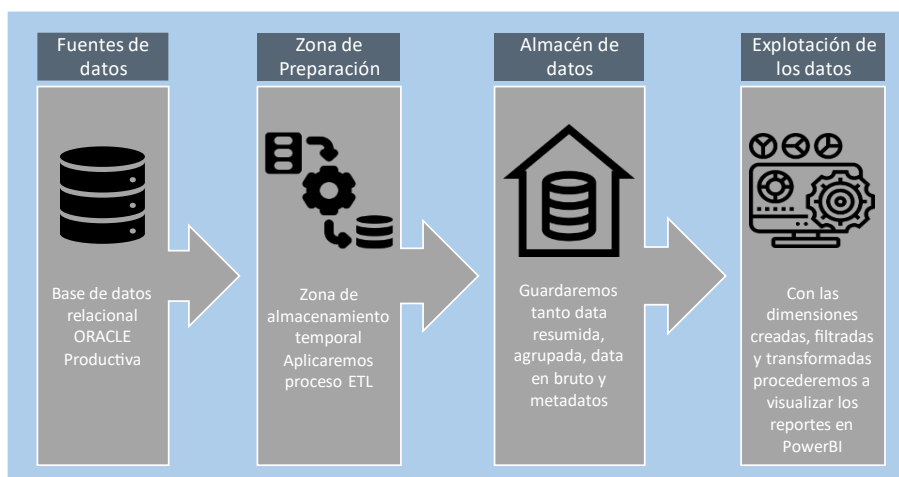


Figura 19. Arquitectura Almacén de Datos. Elaboración propia

3.3 Diseño presentación grafica.

En este apartado intentaremos generar los precedentes asociados al diseño que tendrá que seguir todo lo relacionado con la parte visual de nuestro proyecto.

Para intentar guiar este proceso de diseño tratare de responder las siguientes preguntas en base a la necesidad de lo que se quiere presentar en los reportes,

teniendo siempre como objetivo principal el brindar una fácil lectura y comprensión de la información mostrada.

Por lo que nuestro diseño tendrá que ayudar a mostrar la información clara y concisa y mejorar la experiencia de quien tenga en frente nuestro reporte.

3.3.1 Diseño Reportes.

Siguiendo con el desarrollo de este capítulo, nuestro reporte tendrá ciertos principios que deberá cumplir para estar acorde al objetivo de este proyecto, estos son:

- **Simplicidad** en los informes esto nos permitirá que los datos estén claros y que los usuarios puedan comprender y actuar rápidamente en base a lo mostrado.
- **Coherencia** en los informes, intentando mantener tipos de letras, tamaños y gama de colores en todos los reportes. Todo lo que se muestre debe tener una razón.

A continuación, se presentan los prototipos de la distribución que llevaremos a la implementación final, al ser solo un bosquejo están sujetos de algunos cambios y están representados en una paleta de colores en escala de grises.

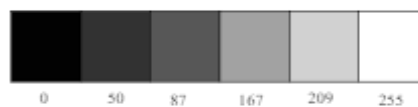


Figura 20. Escala de grises DEMO. Elaboración propia

Los prototipos serán los siguientes:

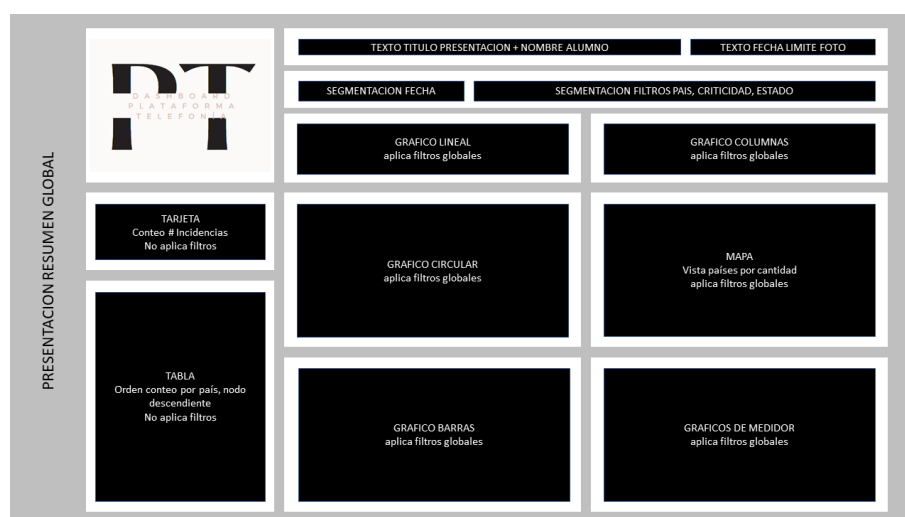


Figura 21. Prototipo Resumen Reporte. Elaboración Propia

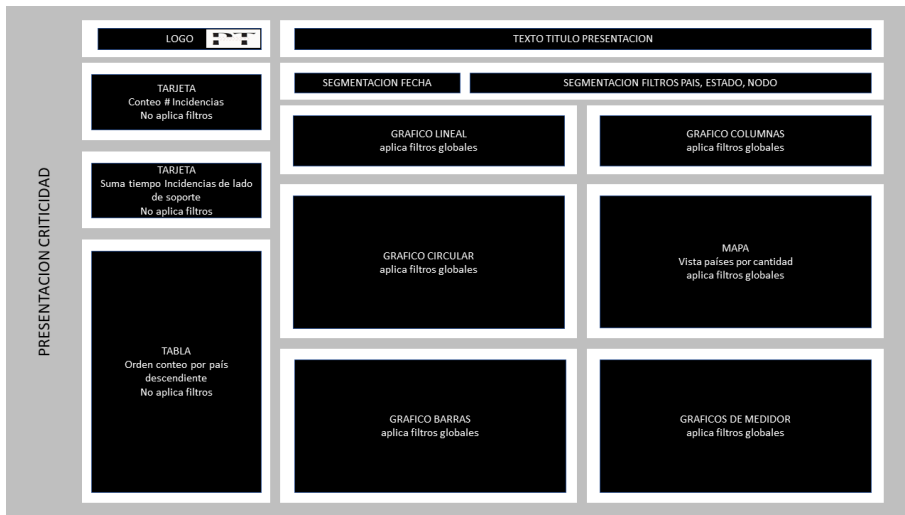


Figura 22. Prototipo Reporte Criticidad. Elaboración Propia

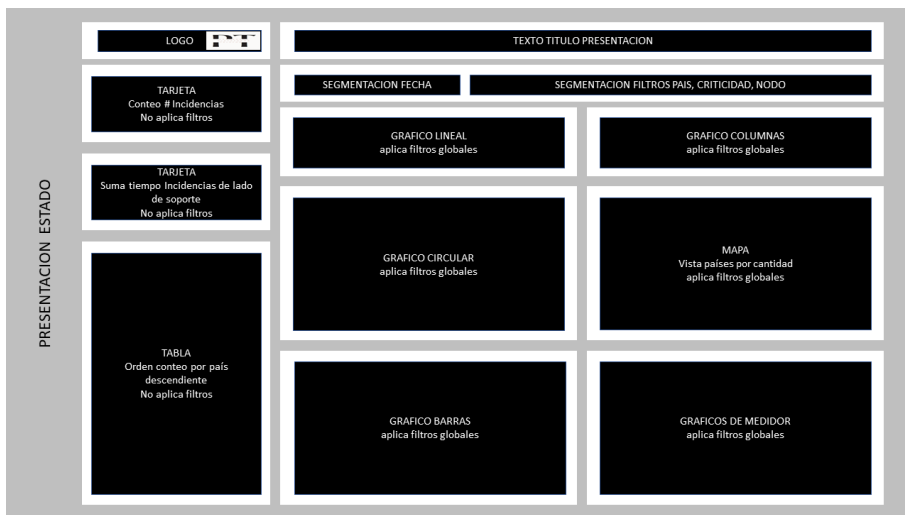


Figura 23. Prototipo Reporte Estado. Elaboración Propia

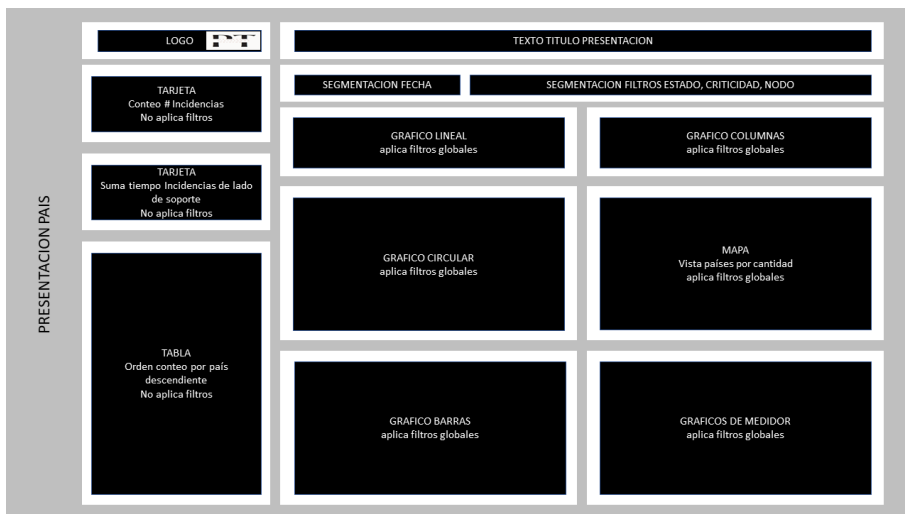


Figura 24. Prototipo Reporte País. Elaboración Propia

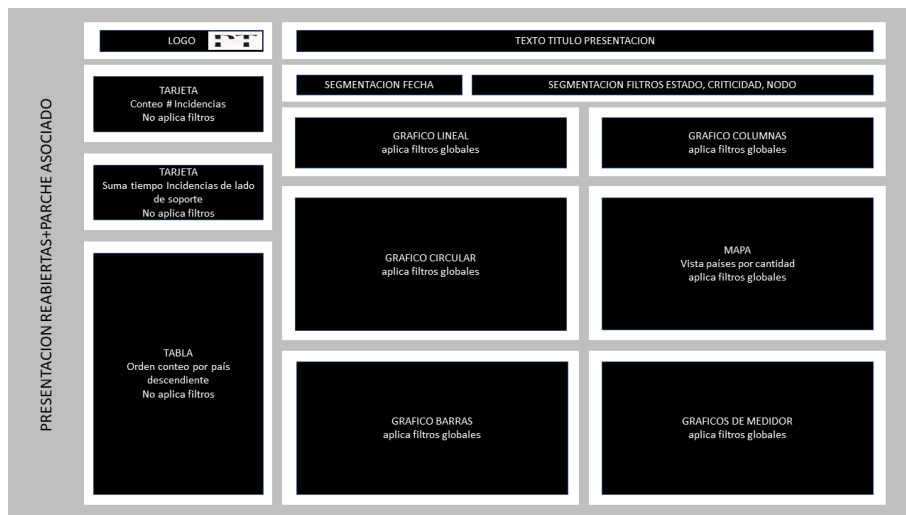


Figura 25. Prototipo Reporte Reabiertas-Parches. Elaboración Propia

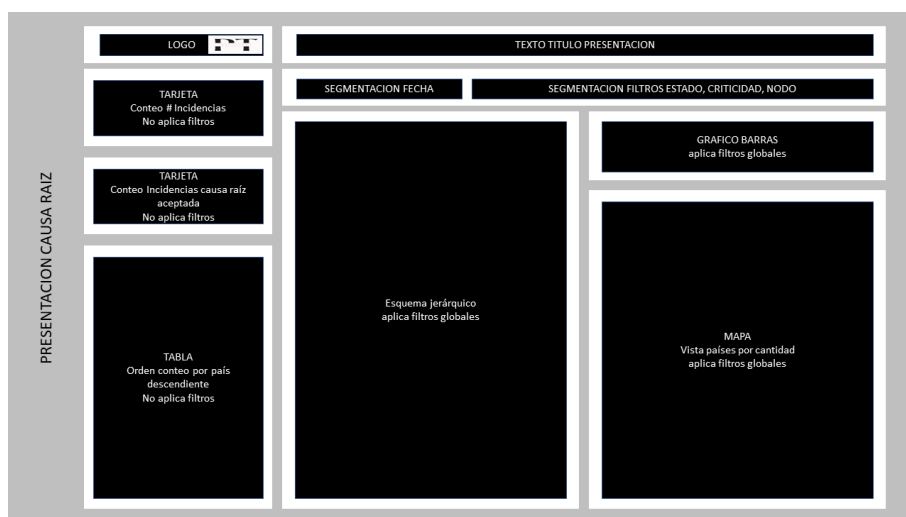


Figura 26. Prototipo Reporte Causa Raíz. Elaboración Propia

Para la implementación del cuadro de mando y reportes usaremos las siguientes características:

Tipo de fuente: Segoe UI

Paleta de colores:

Código de color	Visualización del color
#01B8AA	Color de la fuente
#374649	A usar en gráficos
#FD625E	A usar en gráficos
#F2C80F	Color títulos y rótulos
#5F6B6D	Color relleno cuadros interiores
#8AD4EB	A usar en gráficos
#FE9666	A usar en gráficos
#A66999	A usar en gráficos
#FFFFFF	Fondo de lienzo y pagina

4. Elección de las herramientas de la Solución

Este apartado del proyecto servirá para mostrar las comparativas y justificaciones que se hicieron para la elección de las herramientas que nos serán necesarias para poder desarrollar toda la implementación del proyecto

4.1 Criterios de elección de herramientas.

Los criterios de elección tendrán que cubrir al menos tres aspectos para ser tomados en cuenta, todos ellos serán criterios contrastables, es decir estarán apoyados por información en base a su posicionamiento en el mercado, adecuación técnica, cantidad de información que se encuentre de esta herramienta y dado que el desarrollado es un proyecto de fin de grado de investigación empírica, un criterio será el que se puede usar libremente o se tenga licencia estudiantil, mas no necesariamente que sea de código abierto.

4.1.1 Elección de herramienta de análisis e inteligencia empresarial

Para la selección de los candidatos de esta categoría recurriremos a una herramienta que, para los que usamos tecnologías de información y comunicación, es el referente por sus predicciones y posicionamiento de estas, me estoy refiriendo al **Cuadrante Mágico de Gartner**⁸, de este cuadrante obtendremos entre los líderes, *es decir aquellas tecnologías y herramientas que tienen un buen desempeño actualmente y que a futuro se encuentran bien posicionados*, a 3 plataformas que se han mantenido en esta área del cuadrante por 3 años consecutivos, estas serán:

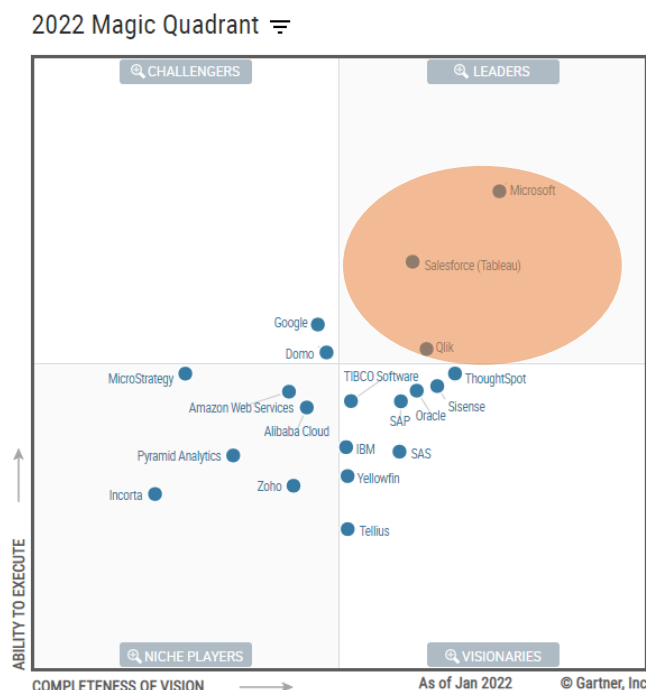


Figura 27. Cuadrante Mágico de Gartner para plataformas de análisis e inteligencia de negocio. Fuente: <https://www.gartner.com/document/4012759?ref=solrAll&refval=347522631>

Breve descripción de las herramientas en el sector líder:

⁸ Magic Quadrant. (s. f.). Recuperado 25 de noviembre de 2022, de <https://www.gartner.com/document/4012759?ref=solrAll&refval=347469340>

- **Tableau⁹**: Esta herramienta de visualización permite a los usuarios combinar grandes fuentes de datos y acceder a ellas con rapidez. También brinda acceso a las visualizaciones a través de aplicaciones seguras y nativas tanto en Android como en iOS.

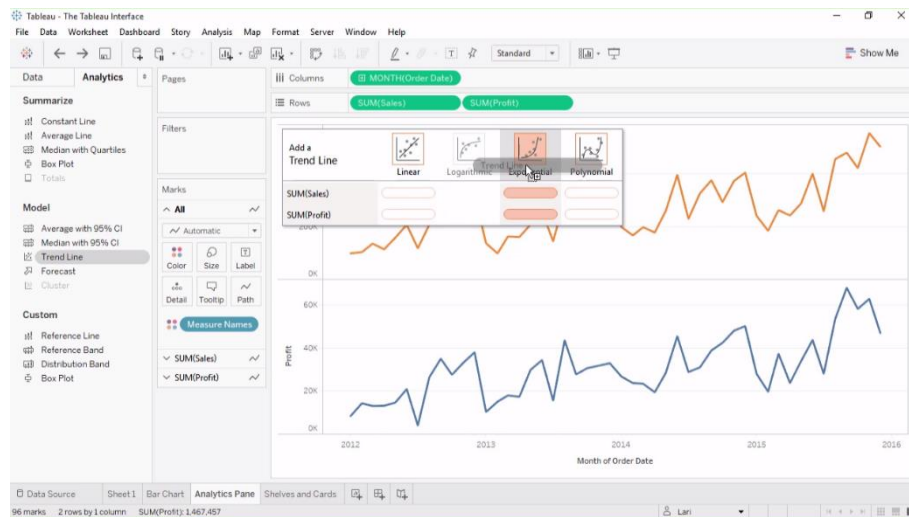


Figura 28. Interface Tableau Desktop. Fuente: Internet

- **Qlik¹⁰**: Herramienta de visualización, que permite interactuar con fuentes de Big Data, obteniendo también una solución totalmente integrada y preparada para la nube. Usa el motor patentado QIX este está basado en la indexación asociativa.

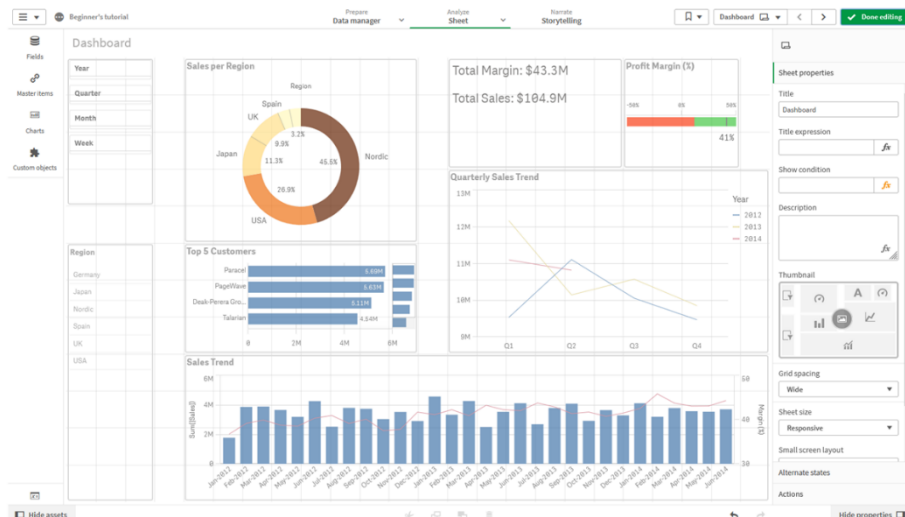


Figura 29. Interface Qlik. Fuente: Internet

- **MS Power BI¹¹**: Herramienta de visualización propia de Microsoft, esto le da una serie de compatibilidades con todo el entorno Azure y Cloud de MS tan usado últimamente, también brinda la posibilidad de interactuar con Excel para importar datos y crear cuadros de mando de datos personalizados.

9 mihart. (s. f.). What is a Power BI «business user»? - Power BI. Recuperado 26 de noviembre de 2022, de <https://learn.microsoft.com/en-us/power-bi/consumer/end-user-consumer>

10 ¿Qué es Tableau? (s. f.). Tableau. Recuperado 26 de noviembre de 2022, de <https://www.tableau.com/es-es/why-tableau/what-is-tableau>

11 Qlik | Help. (s. f.). Recuperado 26 de noviembre de 2022, de <https://help.qlik.com/es-ES/>

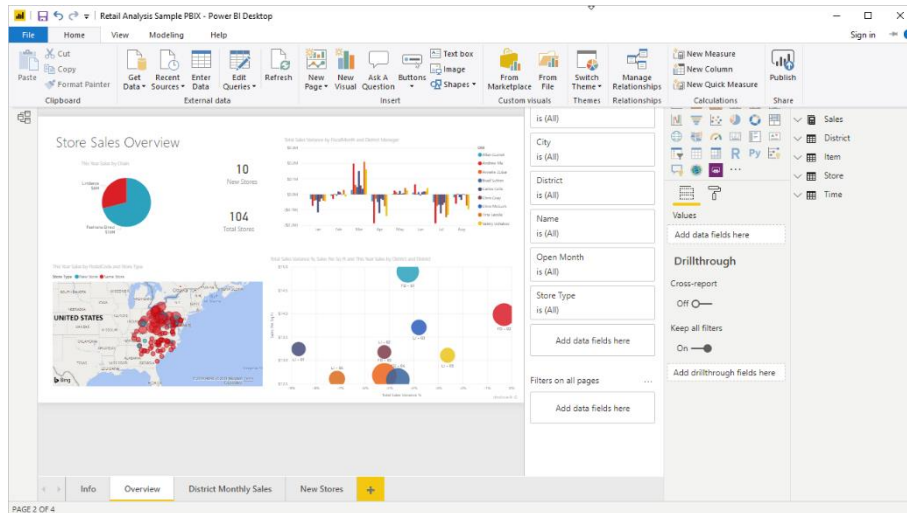


Figura 30. Interface Power BI. Fuente: Internet

A continuación, veremos algunas características a modos de comparación entre estas plataformas:



	Tableau	Qlik	MS Power BI
<i>Capacidad de visualización y fuente de datos.</i>	Conocido por sus perfectas capacidades gráficas y de visualización. Además, es un software fácil de usar. Permite la integración de datos de diversas fuentes.	Es una herramienta de analítica de autoservicio con un motor de almacenamiento de datos en memoria. Proporciona buenas visualizaciones que son dinámicas debido al motor en memoria.	Permite a los usuarios importar datos de fuentes variadas y utilizarlos con tablas, gráficos, cuadros para visualizarlos.
<i>Capacidad de análisis integrado</i>	Sus análisis incrustados pueden utilizarse en una variedad de aplicaciones y procesos, pero los usuarios sólo podrán incrustar cuadros de mando completos.	Los objetos disponibles para incrustar incluyen métricas, valores individuales, cuadros de	No sólo ofrecen cuadros de mando integrados, sino también otros objetos.

	mando y números, y funcionan sin problemas en una amplia gama de escenarios.	
<i>Coste de licenciamiento</i>	Versión básica gratuita, pero funcionalidades muy limitadas. A partir de ahí la versión de pago inicia con \$70 ¹² por usuario	Versión básica gratuita, pero funcionalidades muy limitadas. A partir de ahí tiene un coste de \$30 ¹³ por mes y usuario.
<i>Nivel de complejidad</i>	Requiere de cierto nivel de conocimientos de ciencia de datos para poder explotar todas las características de esta herramienta.	Requiere de conocimientos de programación asociados a ciencia de datos para poder trabajar con esta herramienta.
		La versión pro tiene licenciamiento gratuito para estudiantes, esta versión contiene todas las funcionalidades necesarias para empezar a trabajar la data. A partir de ahí tienen un coste de \$10 ¹⁴ por usuario y mes para el uso de ciertas funcionalidades.
		Su funcionamiento suena más conocido porque usa muchas de las funcionalidades que previamente se han visto en Excel, también tiene una interfaz similar.

Tabla 5. Tabla comparativa herramientas BI. Fuente: Elaboración Propia

De las comparaciones anteriores, nos encontramos que **se elegirá la herramienta de MS Power BI:**

1. Por las continuas actualizaciones que realiza, haciéndole destacar de herramientas más costosas.
2. MS Power BI tiene la opción de venir por defecto con el licenciamiento de estudiante del Office y no por ser “casi” gratis es menos robusto que las otras opciones,
3. Pertenece al ecosistema desarrollado por Microsoft por lo que tiene la capacidad de integrarse y nutrirse de este, a diferencia de las otras dos opciones que se especializan solo en el área de análisis de datos.

12 Pricing & Product Comparison | Microsoft Power BI. (s. f.). Recuperado 26 de noviembre de 2022, de <https://powerbi.microsoft.com/en-us/pricing/>

13 Pricing for data people. (s. f.). Tableau. Recuperado 26 de noviembre de 2022, de <https://www.tableau.com/pricing/teams-orgs>

14 Qlik Pricing | Qlik Sense – Compare All Editions. (s. f.). Qlik. Recuperado 26 de noviembre de 2022, de <https://www.qlik.com/us/pricing>

4. Siendo uno de los líderes en este sector existe muchísima información de apoyo y formación que servirán para poder sacar adelante este proyecto.

4.1.2 Elección de herramienta de ETL

Entre las opciones que tenemos para usar como herramienta de ETL están Pentaho Data Integration y Talend Data, ambas se han comprobado que tienen un buen nivel de aceptación de acuerdo con la **visión del cliente**¹⁵ que recoge Gartner en su sección **Peer Insight**.

Breve descripción de las herramientas comentadas previamente

- **Pentaho DI**¹⁶: Esta herramienta es parte de un conjunto que brinda la posibilidad de análisis de datos, cuenta con una rica biblioteca de transformación con más de 150 objetos de mapeo listos para usar, además soporta una amplia gama de fuentes de datos que incluye más de 30 plataformas de bases de datos de código abierto y propietarias, archivos planos.

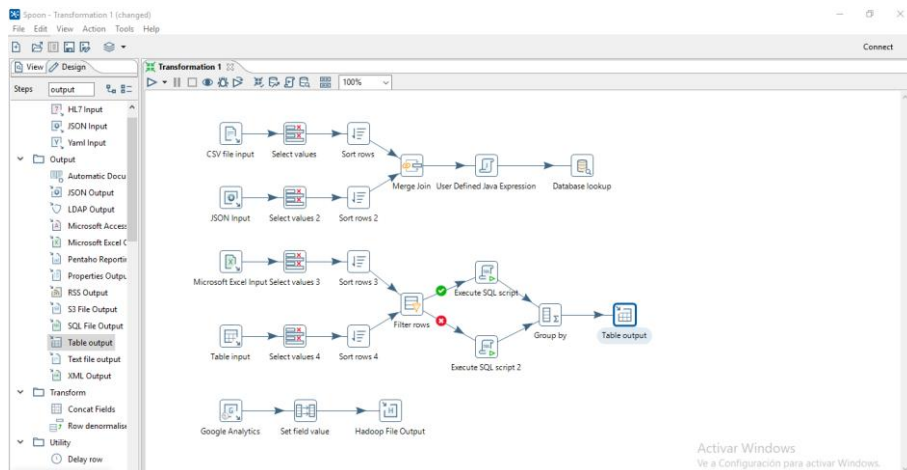


Figura 31. Interface Pentaho Data Integration. Fuente: Internet

- **Talend DI**¹⁷: Es una herramienta de integración de datos de código abierto. Ofrece varios programas y servicios para la integración de datos. Con esta tecnología, los datos son más accesibles, su calidad mejora y pueden trasladarse rápidamente a los sistemas de destino.

15 Inc, G. (s. f.). Pentaho Data Integration (PDI) vs Talend Data Fabric 2022 | Gartner Peer Insights. Gartner. Recuperado 2 de diciembre de 2022, de <https://www.gartner.com/market/data-integration-tools/compare/product/pentaho-data-integration-pdi-vs-talend-data-fabric>

16 Pentaho 9.4. (2022, septiembre 9). Hitachi Vantara Lumada and Pentaho Documentation. <https://help.hitachivantara.com/Documentation/Pentaho/9.4>

17 Data Integration Solutions: A Unified View for Trusted Data. (s. f.). Talend - A Leader in Data Integration & Data Integrity. Recuperado 2 de diciembre de 2022, de <https://www.talend.com/products/data-integration/>

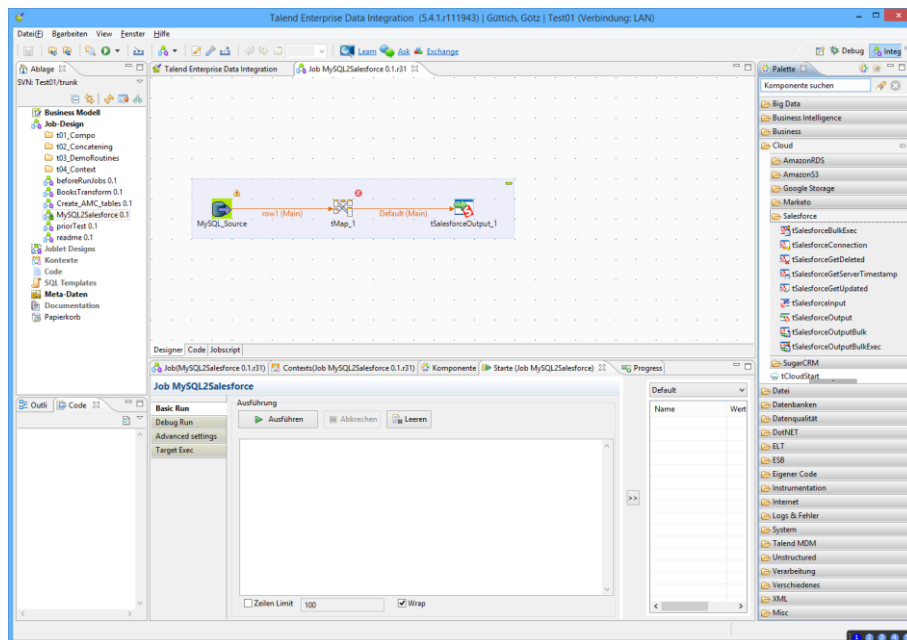


Figura 32. Interface Talend Data Integration. Fuente: Internet

A continuación, veremos algunas características a modos de comparación entre estas plataformas:



	Pentaho	Talend
<i>Enfoque</i>	Orientado en los metadatos.	Orientado en la generación de código.
<i>Despliegue</i>	Java.	Java.
<i>Velocidad</i>	Rápido.	Más lento comparado con Pentaho.
<i>Información-Comunidad de apoyo</i>	Mucha información y un foro de apoyo muy grande	Poca información, foro de apoyo grande.
<i>Interfaz</i>	Interfaz gráfica bastante intuitiva.	Interfaz gráfica difícil de entender.
<i>Coste</i>	Versión Community gratuita.	Versión Open Studio gratuita.

Tabla 6. Tabla comparativa herramientas ETL. Fuente: Elaboración Propia

De las comparaciones anteriores, nos encontramos que **se elegirá la herramienta de Pentaho:**

1. Si bien ambas son herramientas de código abierto robustas, fáciles de usar y fiables, Pentaho Kettle es tiene la interfaz más amigable y fácil de usar.
2. Pentaho cuenta con una gran cantidad de información a la que recurrir cuando se tengan dudas en la implementación del proyecto.

4.1.3 Elección de herramienta de soporte del almacén de datos

Entre las opciones que se tendrán en cuenta para elegir la herramienta que de soporte al almacén de datos son Maria DB, PostgreSQL y MySQL. Para poder realizar una correcta elección compararemos cual se adapta mejor al proyecto.

Breve descripción de las herramientas comentadas previamente

MariaDB¹⁸: Es un DBMS de código abierto, que fue creado como una bifurcación de software de MySQL, lo que los hace altamente compatibles, por desarrolladores que desempeñaron papeles clave en la construcción de la base de datos original; está basada en SQL y soporta **ACID¹⁹**.

PostgreSQL²⁰: Es un DBMS de código abierto, que puede manejar una enorme variedad de casos de uso, desde máquinas individuales hasta almacenes de datos y servicios web con muchos usuarios simultáneos, este utiliza SQL (de ahí su nombre), si bien soporta consultas SQL (relacionales) también soporta JSON (no relacionales).

MySQL²¹: Es un DBMS de código abierto, basado en SQL, que gana una gran aceptación al ser parte de la tecnología LAMP (que consiste en un sistema operativo basado en Linux, el servidor web Apache, una base de datos MySQL y PHP para el procesamiento).

A continuación, veremos algunas características a modos de comparación entre estas plataformas:



	MariaDB	PostgreSQL	MySQL
<i>Tratamiento de los datos</i>	El tratamiento de los datos es lapso, es decir permite ciertos errores de tipo de dato y los cataloga como advertencias.	Es riguroso en la inserción de datos, en caso de encontrar un error de tipo de datos, te rechaza el comando.	El tratamiento de los datos es lapso, es decir permite ciertos errores de tipo de dato y los cataloga como advertencias.

18 MariaDB en resumen. (s. f.). MariaDB.Org. Recuperado 4 de diciembre de 2022, de <https://mariadb.org/es/>

19 ACID. (2022). En Wikipedia, la enciclopedia libre. <https://es.wikipedia.org/w/index.php?title=ACID&oldid=146054033>

20 PostgreSQL: About. (s. f.). Recuperado 4 de diciembre de 2022, de <https://www.postgresql.org/about/>

21 MySQL :: MySQL Documentation. (s. f.). Recuperado 4 de diciembre de 2022, de <https://dev.mysql.com/doc/>

<i>Licenciamiento</i>	GPL versión 2, suscripción comercial para empresas disponible	GPL versión 2. Existen licencias comerciales con funcionalidad ampliada	Berkeley Software Distribution (BSD)
<i>GUI</i>	Existen diferentes interfaces graficas que pueden interactuar con el motor de base de datos.	Existen diferentes interfaces graficas que pueden interactuar con el motor de base de datos.	Viene con su propia interfaz gráfica y es muy intuitiva.

Tabla 7. Tabla comparativa soporte almacén de datos. Fuente: Elaboración Propia

De las comparaciones anteriores, encontramos que son productos muy similares, por lo que **la elección la basaremos en el tipo de licenciamiento y es PostgreSQL** la que no tiene ningún tipo de atadura al momento de usar la herramienta, además también tendremos en cuenta estos dos aspectos:

1. Es la herramienta con más uso en el mercado sobre todo debido a su enfoque en la integridad y fiabilidad de datos.
2. Gran cantidad de información en la comunidad de PostgreSQL, que nos servirá de apoyo a la consecución de los objetivos del proyecto
3. Oportunidad personal de aprender a usar una nueva herramienta.

5. Implementación de la Solución

5.1 ETL

Para este veo necesario explicar que este proyecto, de acuerdo con cómo se ha planteado necesitara de la ejecución de dos procesos de ETL, cada una marcara una fase distinta del proyecto.

A continuación, se describirá ambas etapas de ETL:

5.1.1 ETL Primera etapa

Debido a que la base de datos que contiene la información es productiva y usada diariamente para generar y alojar datos asociados a las incidencias, consultas, gestión de seguridad propia de la plataforma de gestión y tarificación, es que se ha decidido traernos una copia a local.

5.1.1.1 Creación base de datos local en PostgreSQL

Creamos la base de datos que almacenara la copia de los datos del repositorio original.

Para esto abriremos el pgAdmin 4, la consola grafica de gestión de PostgreSQL

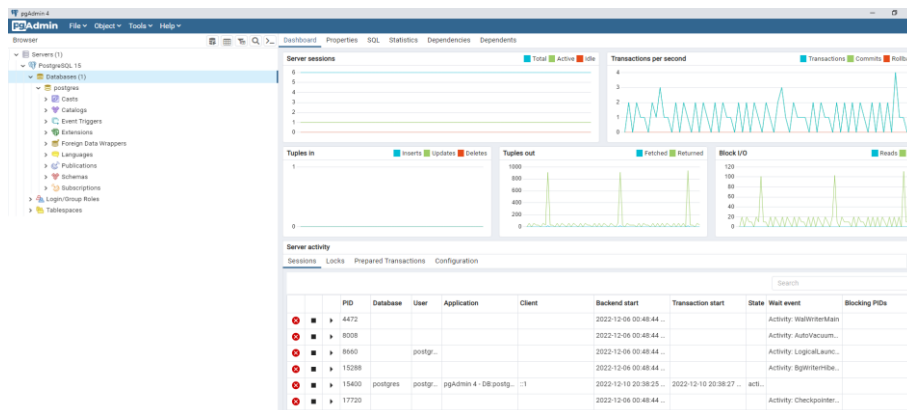


Figura 33. ETL1 Creación BD PG Interfaz pgAdmin. Fuente: Elaboración propia

Posterior a esto procederemos a crear una base de datos.

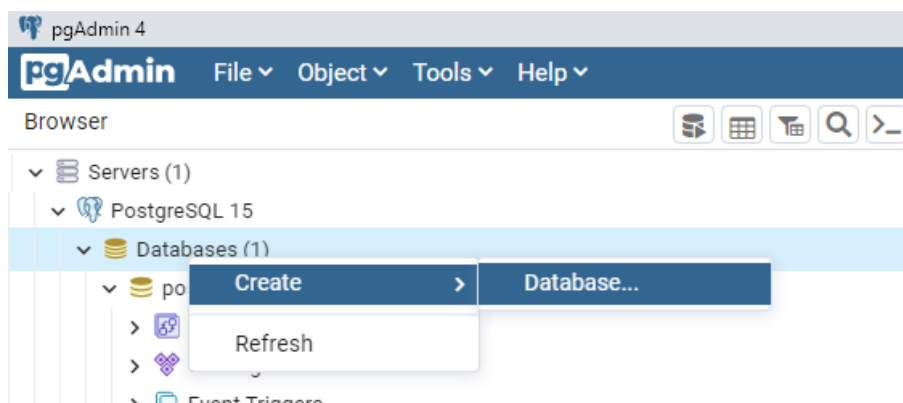


Figura 34. ETL1 Creación BD PG Menú. Fuente: Elaboración propia

Usaremos el usuario por defecto que se crea con la instalación “postgres” y le asignaremos todos los permisos sobre esta base de datos.

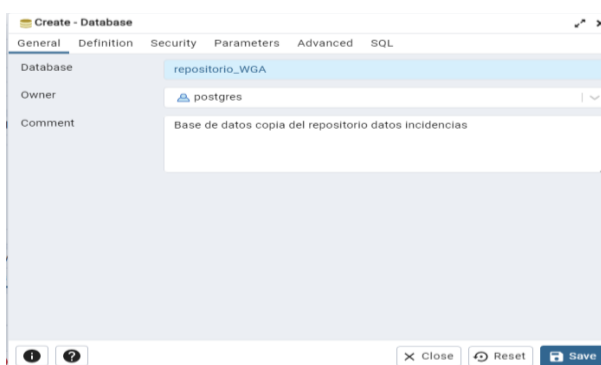


Figura 35. ETL1 Creación BD PG Creación Paso 1. Fuente: Elaboración propia

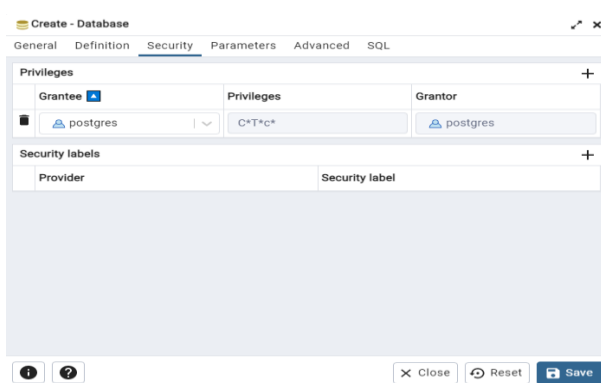


Figura 36. ETL1 Creación BD PG Creación Paso 2. Fuente: Elaboración propia

Y le damos a “**Save**” para crear.

5.1.1.2 Creación trabajo para poblar tablas en Pentaho

Para poblar la base de datos copia (destino) usaremos Pentaho Data Integration, en base a investigación sobre cómo hacerlo encontré la opción de copiar íntegramente un esquema (usuario) de Oracle a PostgreSQL, a continuación, se mostrará el paso a paso:

1. Abrir Spoon

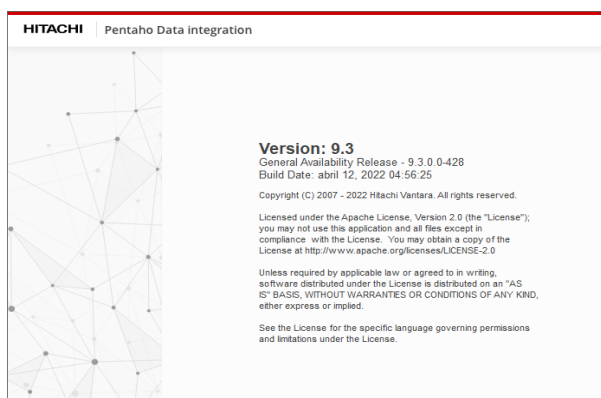


Figura 37. ETL1 Creación BD Carga Spoon. Fuente: Elaboración propia

2. Crearemos un nuevo trabajo.

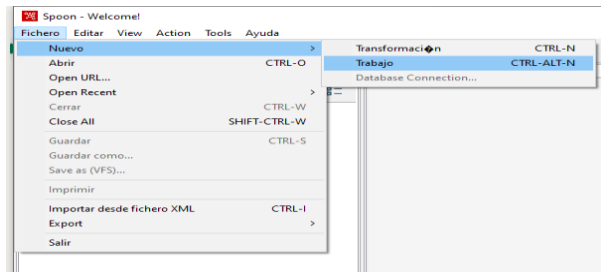


Figura 38. ETL1 Creación Pentaho Menú Trabajo. Fuente: Elaboración propia

- Nos vamos a la pestaña View para crear las dos conexiones a base de datos tanto de origen como de destino.

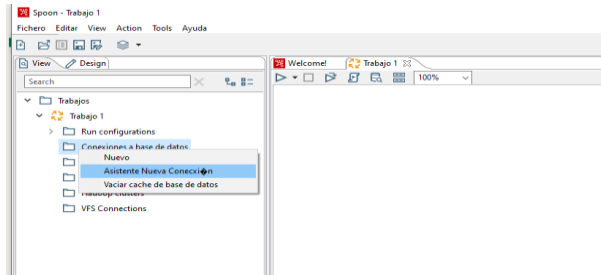


Figura 39. ETL1 Creación Pentaho Nuevo Trabajo. Fuente: Elaboración propia

- Iniciamos creando la conexión que usaremos para obtener la data original, la llamaremos ORIG_Oracle, tipo de base de datos: Oracle y tipo de acceso Native (JDBC) y presionamos Next.

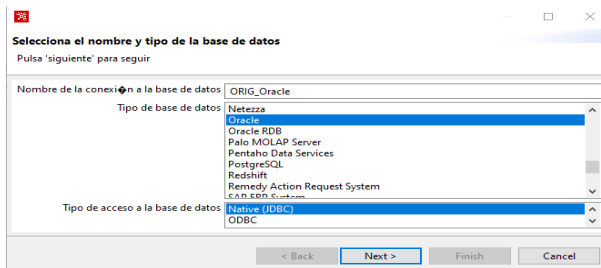


Figura 40. ETL1 Creación Pentaho Conexión Oracle. Fuente: Elaboración propia

- En la siguiente pantalla agregaremos la información de la base de datos original, para proteger los datos la información del nombre del servidor y del nombre de la base de datos, cubriremos y presionamos Next.

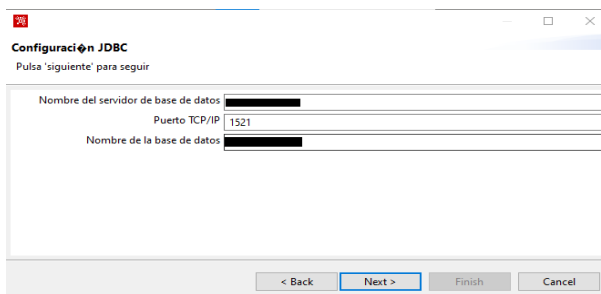


Figura 41. ETL1 Creación Pentaho Conexión Oracle Datos 1. Fuente: Elaboración propia

- No se rellena nada en la siguiente pantalla y presionamos Next.

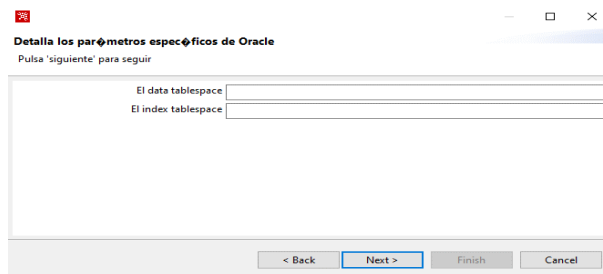


Figura 42. ETL1 Creación Pentaho Conexión Oracle Datos 2. Fuente: Elaboración propia

- Ingresamos usuario y contraseña, para proteger los datos la información se cubrirá, y procedemos antes a probar la conexión para validar antes que todo este correcto, una vez validado esto le daremos a Finish.

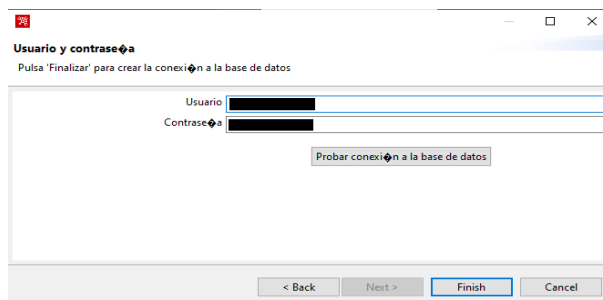


Figura 43. ETL1 Creación Pentaho Conexión Oracle Datos 3. Fuente: Elaboración propia

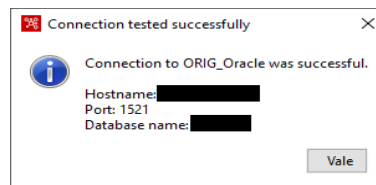


Figura 44. ETL1 Creación Pentaho Conexión Oracle Test. Fuente: Elaboración propia

- Realizaremos el mismo procedimiento con la conexión local de PostgreSQL. Los datos que manejaremos para rellenar serán:
Nombre de la conexión a base de datos: DEST_PostgreSQL_Full
Tipo de base de datos: PostgreSQL
Tipo de acceso: Native (JDBC)
Nombre del servidor de base de datos: localhost
Puerto: Por defecto PostgreSQL.
Nombre de la base de datos: repositorio_EXP
Usuario: postgres
Contraseña: temporal

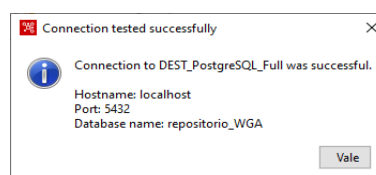


Figura 45. ETL1 Creación Pentaho Conexión Postgres Test. Fuente: Elaboración propia

- Una vez creadas las conexiones de bases de datos de origen y destino, procederemos a usar la herramienta de Copia de Tablas de Pentaho:

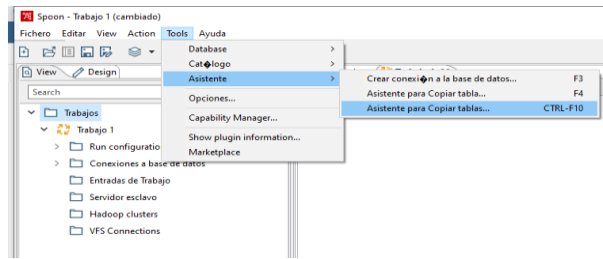


Figura 46. ETL1 Creación Pentaho Copia Tablas 1. Fuente: Elaboración propia

10. Seleccionamos origen y destino y presionamos **Next**.

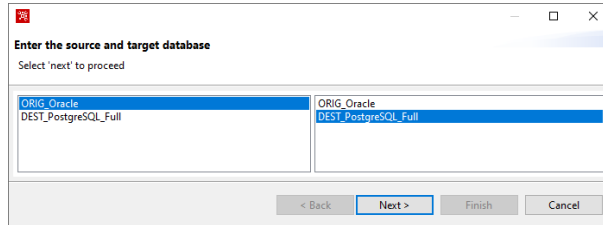


Figura 47. ETL1 Creación Pentaho Agregación Conexiones. Fuente: Elaboración propia

11. Seleccionaremos todas las tablas del esquema, a excepción de las que sean vistas, tablas temporales y tablas que no sean relevantes para nuestro proyecto y presionaremos el icono que nos permite agregar todas las tablas seleccionadas (>).

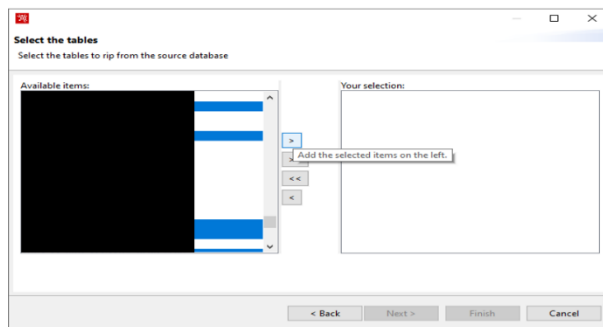


Figura 48. ETL1 Creación Pentaho Tablas Importar. Fuente: Elaboración propia

12. Cargamos las tablas y damos **Next**.

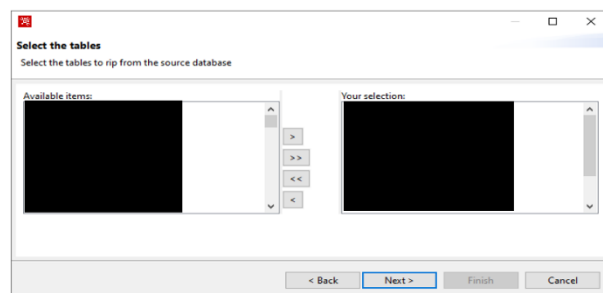


Figura 49. ETL1 Creación Pentaho Listado Tablas. Fuente: Elaboración propia

13. Le damos un nombre al trabajo y un directorio donde se harán las extracciones de origen, conversiones y desde donde se cargará la data a destino, finalmente procederemos a darle a **Finish**.

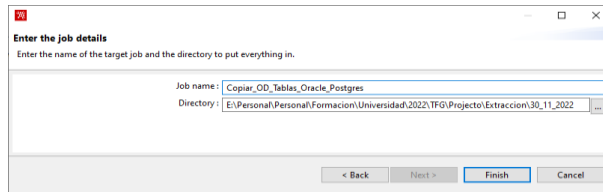


Figura 50. ETL1 Creación trabajo Configuración directorio. Fuente: Elaboración propia

14. Posterior a esto, Pentaho nos generara los pasos y saltos necesarios para poder hacer la copia de los datos de Origen a nuestro base de datos destino en local.

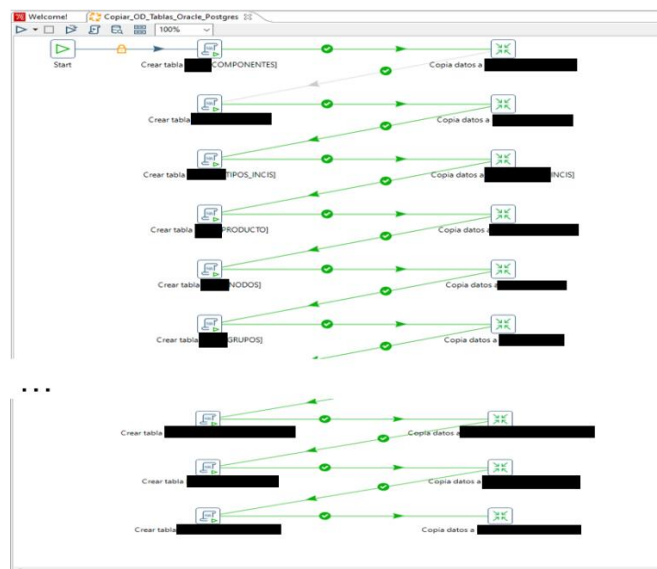


Figura 51. ETL1 Creación Pentaho Panel Tareas. Fuente: Elaboración propia

15. Luego procederemos a darle al Run (F9) y nos mostrará la pantalla con las opciones para ejecutar el trabajo, debido a que tenemos la versión Pentaho Community, esta solo nos permite ejecutarlo en local, lo que si configuraremos será el Nivel de registro a *Row Level (very detailed)* ya que en caso de error esto nos será útil, luego presionaremos *Ejecutar*.

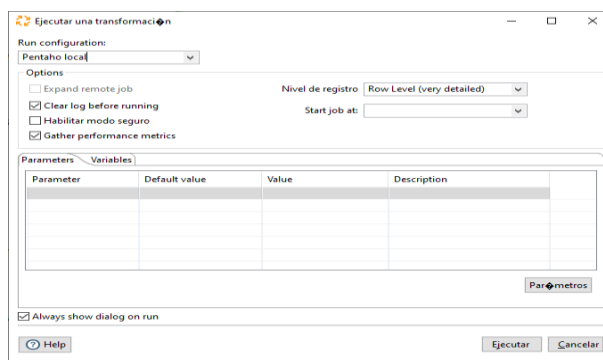


Figura 52. ETL1 Creación Pentaho Ejecución Tarea. Fuente: Elaboración propia

16. Monitorizaremos la base de datos para validar que se está insertando la información correctamente

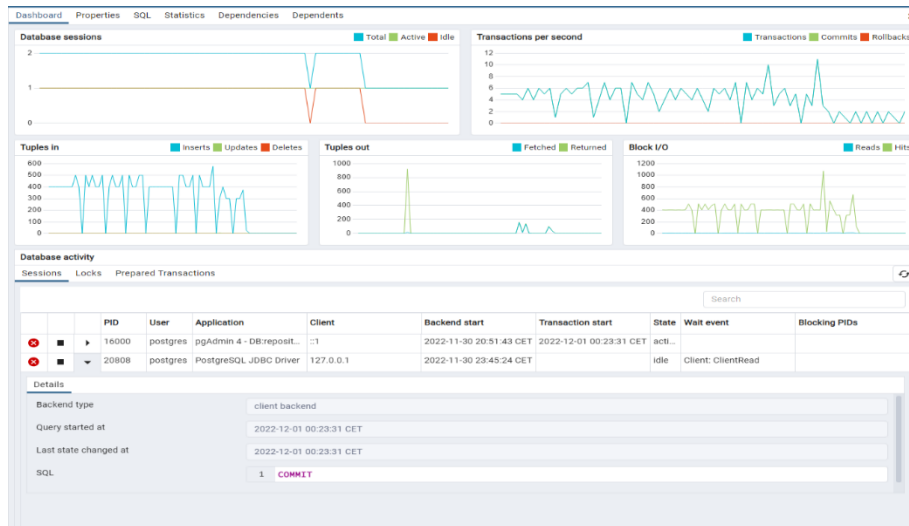


Figura 53. ETL1 Creación Pentaho Monitorización Postgres. Fuente: Elaboración propia

17. Luego de aproximadamente 3 horas se termina de hacer la copia de la información de origen a destino, podemos validar la finalización en la pestaña “Jobs metrics” de la interfaz gráfica de Pentaho.

Comentario	Resultado	Razón	Nombre Fichero
Copiar datos a [WGA_NODOS_PAIS]	Job execution finished	Exitó	file:///E:/Personal/Personal/Formacion/Univ
Crear tabla [WGA_VERSION_COMPONENTES]	Start of job execution	Followed link after success	file:///E:/Personal/Personal/Formacion/Univ
Copiar datos a [WGA_VERSION_COMPONENTES]	Job execution finished	Exitó	file:///E:/Personal/Personal/Formacion/Univ
Copiar datos a [WGA_VERSION_COMPONENTES]	Start of job execution	Followed link after success	file:///E:/Personal/Personal/Formacion/Univ
Copiar datos a [WGA_VERSION_COMPONENTES]	Job execution finished	Exitó	file:///E:/Personal/Personal/Formacion/Univ
Crear tabla [WGA_INCL_FICHMOP]	Start of job execution	Followed link after success	file:///E:/Personal/Personal/Formacion/Univ
Crear tabla [WGA_INCL_FICHMOP]	Job execution finished	Exitó	file:///E:/Personal/Personal/Formacion/Univ
Copiar datos a [WGA_INCL_FICHMOP]	Start of job execution	Followed link after success	file:///E:/Personal/Personal/Formacion/Univ
Copiar datos a [WGA_M_DETECTADO_EN]	Job execution finished	Exitó	file:///E:/Personal/Personal/Formacion/Univ
Copiar datos a [WGA_M_DETECTADO_EN]	Start of job execution	Followed link after success	file:///E:/Personal/Personal/Formacion/Univ
Copiar datos a [WGA_M_DETECTADO_EN]	Job execution finished	Exitó	file:///E:/Personal/Personal/Formacion/Univ
Tablas Copiar_OD_Tablas_Oracle_Postgres	Job execution finished	Exitó	finished

Figura 54. ETL1 Creación Pentaho Jobs Metrics. Fuente: Elaboración propia

18. Validaremos que las tablas asociadas para la realización de este proyecto se hayan cargado correctamente en la base de datos de PostgreSQL, para esto consultaremos vía SQL para mostrar los resultados en una tabla y gráficamente

Esquema_PostgreSQL	Nombre_Tabla	Cantidad_Registros
public	CICLO_DE_VIDA	1206203
public	PARCHES_DATOS_COMP	716916
public	INCIDENCIAS	305900
public	PARCHES_DATOS	291388
public	ACEPTACION	13556
public	SUBCOMPONENTE	11208
public	USUARIO	3957
public	INCIDENCIA_PROCEDIMIENTO	2258
public	VERSION	1355
public	CAT_PAIS	51
public	CAT_SUBESTADO	35
public	CAT_DETECTADO	28
public	CAT_ACEPTACION	25
public	CAT_ESTADO	19
public	GRUPO	14

public	CAT_TIPO_SERVICIO_AFECTADO	9
public	RECHAZO_PROCEDIMIENTO	6
public	CAT_PERIODO	6
public	ESTADO_PROCEDIMIENTO	3

Tabla 8. ETL1 Creación Pentaho Conteo registros tablas. Fuente: Elaboración propia

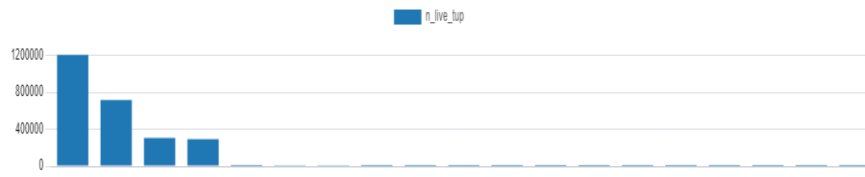


Figura 55. ETL1 Creación Pentaho Grafico Tablas Creadas. Fuente: Elaboración propia

Teniendo una copia casi completa de la base de datos productiva nos aseguramos de que en caso de necesitar algún dato ya sea categórico o numérico podremos obtenerlo sin necesidad de realizar una nueva consulta a la base de datos remota.

5.1.2 ETL Segunda etapa

Para la creación del DWH en PostgreSQL, primero empezaremos creando el usuario **dwh** con contraseña **“dwh”**, que será nuestro usuario de creación, también le daremos acceso y permisos al usuario postgres para que puedan interactuar. Se muestra a continuación la sentencia a ejecutar:

```

1 CREATE ROLE dwh WITH
2 LOGIN
3 SUPERUSER
4 CREATEDB
5 CREATEROLE
6 INHERIT
7 REPLICATION
8 CONNECTION LIMIT -1
9 PASSWORD 'xxxxxxx';
10 GRANT postgres TO dwh WITH ADMIN OPTION;
11 COMMENT ON ROLE dwh IS 'Usuario propietario del DWH.';

```

Figura 56. ETL2 Creación Usuario dwh. Fuente: Elaboración propia

Luego procederemos a crear la base de datos, esta se llamará **dwh_EXP** y es donde volcaremos las dimensiones necesarias para explotarla posteriormente por la herramienta de análisis.

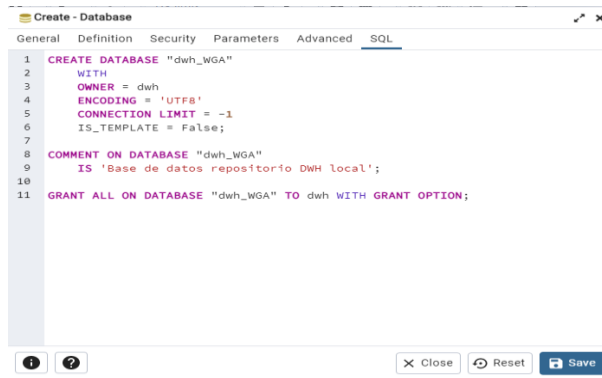


Figura 57. ETL2 Creación BD dwh_EXP. Fuente: Elaboración propia

El siguiente paso será abrir Spoon (Pentaho) para poder crear la tarea que nos ayudará en la creación de las dimensiones en el almacén de datos, le daremos a crear una nueva transformación y procederemos a guardar la transformación

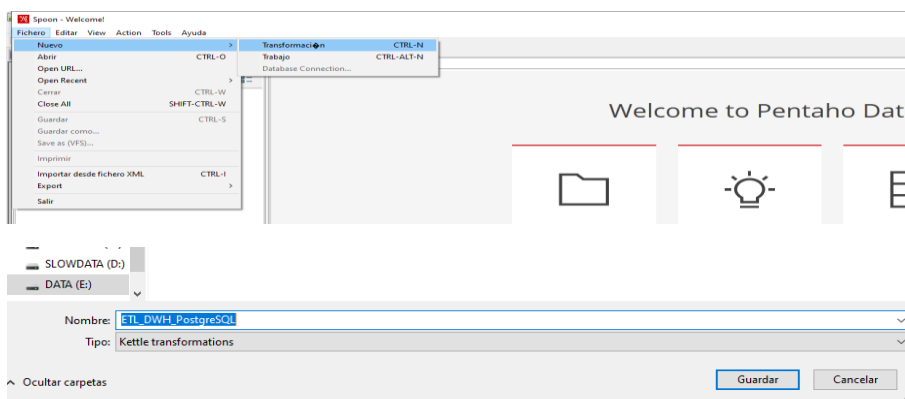


Figura 58. ETL2 Creación tarea Pentaho. Fuente: Elaboración propia

5.1.2.1 Configuración previa dimensión

Configuración previa dimensión DIM_CicloVida

Luego procederemos a agregar las tablas que irán sobre el Almacén de datos, para esto iremos al panel de la derecha y seleccionaremos Entrada=>Entrada Tabla y lo arrastraremos al panel principal

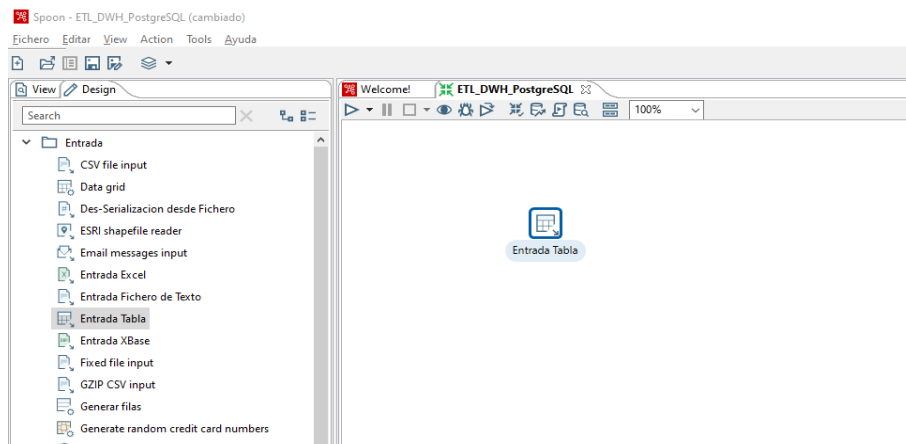


Figura 59. ETL2 Conf DIM_Ciclovida. Fuente: Elaboración propia

Le pulsamos 2 veces sobre el icono arrastrado de “Entrada Tabla” y procederemos a editar las propiedades:

Nombre Paso: Entrada_DIM_CicloVida

Conexión: Crearemos una conexión hacia la instancia de postgres de repositorio de donde obtendremos la data inicial.

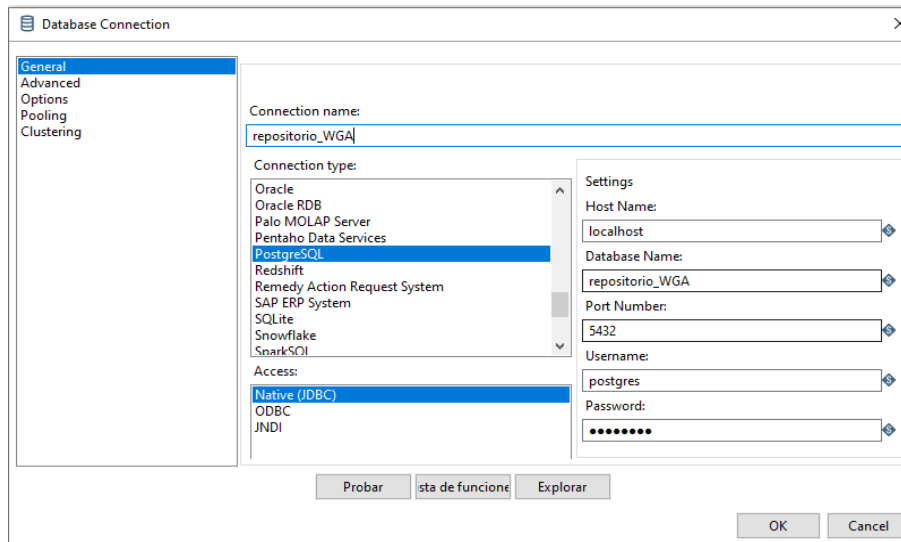


Figura 60. ETL2 Conf DIM_Ciclovida 2. Fuente: Elaboración propia

SQL: Se agregará la siguiente consulta que será la que obtenga la información necesaria para probar la DIM_CicloVida.

```
SELECT C.INCI_ID,
       C.CICLOVIDA_FX STATUS_DATE,
       PR.PRIOR_NOMBRE_EN SEVERITY,
       PR.PRIOR_ID SEVERITY_ID,
       E.ESTADO_NOMBRE_EN STATUS_NAME,
       G.GRP_NOM_CORTO GROUP_NAME,
       PL.PLAT_NOM_CORTO PLATFORM,
       TC.TIPOCATEG_NOMBRE CATEGORY,
       TC.TIPOCATEG_ID CATEGORY_ID,
       TI.TIPOINCI_NOMBRE TICKET_TYPE,
       C.HOT HOT
FROM (SELECT INCI_ID, MAX(CICLOVIDA_ID) ID FROM CICLO_DE_VIDA GROUP BY INCI_ID) a,
CICLO_DE_VIDA c
LEFT OUTER JOIN CAT_PRIORIDAD pr ON (C.PRIOR_ID = PR.PRIOR_ID)
LEFT OUTER JOIN CAT_ESTADO e ON (C.ESTADO_ID = E.ESTADO_ID)
LEFT OUTER JOIN GRUPO g ON (C.GRP_ID_ESTADO = G.GRP_ID)
LEFT OUTER JOIN PLATAFORMAs pl ON (C.PLAT_ID = PL.PLAT_ID)
LEFT OUTER JOIN CAT_CATEGORIZACION tc ON (C.TIPOCATEG_ID = TC.TIPOCATEG_ID)
LEFT OUTER JOIN CAT_TIPOS_INC ti ON (C.TIPOINCI_ID = TI.TIPOINCI_ID)
WHERE C.CICLOVIDA_ID = A.ID AND C.CICLOVIDA_TIPO = '1'
      AND C.PROD_ID IN ('4', '5', '8', '9')
      AND C.ESTADO_ID NOT IN ('0', '10');
```

Esta sería la Entrada Tabla (DIM_CicloVida):



Figura 61. ETL2 Creación Tabla DIM_Ciclovida. Fuente: Elaboración propia

El procedimiento para crear el resto de las dimensiones será igual, a continuación, se agregarán las consultas y el nombre de cada dimensión, que será lo que variará entre los pasos de entrada que se creen.

5.1.2.2 Configuración previa dimensión DIM_Tiempo

Nombre: Entrada_DIM_Tiempo

Consulta:

```
SELECT I.INCI_ID,
       coalesce(GEN.FECHA,I.INCI_FX_APERTURA) F_GEN,
       OPN.FECHA F_OPEN,
       coalesce(OPN.FECHA,LOCALTIMESTAMP)-coalesce(GEN.FECHA,I.INCI_FX_APERTURA)
LEAD_TIME,
       I.INCI_FEC_SLA F_RECOVERY,
       coalesce(CLS.FECHA,I.INCI_FX_CIERRE) F_CLOSE,
       RJC.FECHA F_REJECT,
       CASE WHEN A3DES.FECHA IS NULL THEN NULL ELSE '1' END A3DES,
       ((coalesce(coalesce(CLS.FECHA,I.INCI_FX_CIERRE),RJC.FECHA))-
coalesce(OPN.FECHA,coalesce(GEN.FECHA,I.INCI_FX_APERTURA))) PF_TIME,
       CASE WHEN coalesce(coalesce(CLS.FECHA,I.INCI_FX_CIERRE),RJC.FECHA) IS NULL THEN
EXTRACT(DAY FROM (LOCALTIMESTAMP-coalesce(GEN.FECHA,I.INCI_FX_APERTURA))) ELSE 0 END AGE
       FROM (SELECT INCI_ID, MIN(C.CICLOVIDA_FX) FECHA FROM CICLO_DE_VIDA C WHERE ESTADO_ID='1'
GROUP BY C.INCI_ID) gen, INCIDENCIAS i
LEFT OUTER JOIN (SELECT INCI_ID, MIN(C.CICLOVIDA_FX) FECHA FROM CICLO_DE_VIDA C WHERE
ESTADO_ID='2' GROUP BY C.INCI_ID) opn ON (I.INCI_ID = OPN.INCI_ID)
LEFT OUTER JOIN (SELECT INCI_ID, MIN(C.CICLOVIDA_FX) FECHA FROM CICLO_DE_VIDA C WHERE
ESTADO_ID='4' GROUP BY C.INCI_ID) cls ON (I.INCI_ID = CLS.INCI_ID)
LEFT OUTER JOIN (SELECT INCI_ID, MIN(C.CICLOVIDA_FX) FECHA FROM CICLO_DE_VIDA C WHERE
ESTADO_ID='8' GROUP BY C.INCI_ID) rjc ON (I.INCI_ID = RJC.INCI_ID)
LEFT OUTER JOIN (SELECT INCI_ID, MIN(C.CICLOVIDA_FX) FECHA FROM CICLO_DE_VIDA C WHERE
ESTADO_ID='15' AND GRP_ID_ESTADO='4' GROUP BY C.INCI_ID) a3des ON (I.INCI_ID = A3DES.INCI_ID)
WHERE I.INCI_ID = GEN.INCI_ID;
```

5.1.2.3 Configuración previa dimensión DIM_DGrupos

Nombre: Entrada_DIM_DGrupos

Consulta:

```
SELECT B.INCI_ID,
       SUM(CASE WHEN B.G1='1' THEN CASE WHEN B.E1='4' THEN 0 WHEN B.E1='8' THEN 0 ELSE
EXTRACT(DAY FROM (B.F2-B.F1))::integer END ELSE 0 END ) T_OPE,
```

```

SUM(CASE WHEN B.G1='1' THEN 0 WHEN B.G1='8' THEN 0 ELSE EXTRACT(DAY FROM (B.F2-
B.F1))::integer
END ) T_PROV,
SUM(CASE WHEN B.G1='2' THEN EXTRACT(DAY FROM (B.F2-B.F1))::integer ELSE 0 END
) T_SR,
SUM(CASE WHEN B.G1='2' THEN CASE WHEN B.E1='15' THEN EXTRACT(DAY FROM (B.F2-
B.F1))::integer
ELSE 0 END ELSE 0 END ) T_AN3_SR,
SUM(CASE WHEN B.G1='3' THEN EXTRACT(DAY FROM (B.F2-B.F1))::integer ELSE 0 END
) T_MC,
SUM(CASE WHEN B.G1='4' THEN EXTRACT(DAY FROM (B.F2-B.F1))::integer ELSE 0 END
) T_DES,
SUM(CASE WHEN B.G1='4' THEN CASE WHEN B.E1='15' THEN EXTRACT(DAY FROM (B.F2-
B.F1))::integer
ELSE 0 END ELSE 0 END ) T_AN3_DES,
SUM(CASE WHEN B.G1='5' THEN EXTRACT(DAY FROM (B.F2-B.F1))::integer ELSE 0 END
) T_PS,
SUM(CASE WHEN B.G1='6' THEN EXTRACT(DAY FROM (B.F2-B.F1))::integer ELSE 0 END
) T_CERTIF,
SUM(CASE WHEN B.G1='7' THEN EXTRACT(DAY FROM (B.F2-B.F1))::integer ELSE 0 END
) T_SMaqueta,
SUM(CASE WHEN B.G1='8' THEN EXTRACT(DAY FROM (B.F2-B.F1))::integer ELSE 0 END
) T_CertifOPE,
SUM(CASE WHEN B.G1='9' THEN EXTRACT(DAY FROM (B.F2-B.F1))::integer ELSE 0 END
) T_INT,
SUM(CASE WHEN B.G1='11' THEN EXTRACT(DAY FROM (B.F2-B.F1))::integer ELSE 0 END
) T_ESP,
SUM(CASE WHEN B.G1='12' THEN EXTRACT(DAY FROM (B.F2-B.F1))::integer ELSE 0 END
) T_OPEINS,
SUM(CASE WHEN B.G1='14' THEN EXTRACT(DAY FROM (B.F2-B.F1))::integer ELSE 0 END
) T_HGD
FROM (
SELECT A.INCI_ID, A.F1, A.C1, A.E1, A.G1, A.F2, A.C2, A.E2, A.G2
FROM (
SELECT C1.INCI_ID,
C1.CICLOVIDA_FX F1,
C1.CICLOVIDA_ID C1,
C1.ESTADO_ID E1,
C1.GRP_ID_ESTADO G1,
C2.CICLOVIDA_FX F2,
C2.CICLOVIDA_ID C2,
C2.ESTADO_ID E2,
C2.GRP_ID_ESTADO G2
FROM CICLO_DE_VIDA C1,
CICLO_DE_VIDA C2
WHERE C1.INCI_ID = C2.INCI_ID
AND C1.CICLOVIDA_ID < C2.CICLOVIDA_ID
) A
WHERE A.C2 = (
SELECT MIN( C.CICLOVIDA_ID )
FROM CICLO_DE_VIDA C
WHERE C.INCI_ID = A.INCI_ID
AND C.CICLOVIDA_ID > A.C1)
UNION ALL
(SELECT C3.INCI_ID,
C3.CICLOVIDA_FX F1,
C3.CICLOVIDA_ID C1,
C3.ESTADO_ID E1,
C3.GRP_ID_ESTADO G1,
LOCALTIMESTAMP F2,
C3.CICLOVIDA_ID C2,
C3.ESTADO_ID E2,
C3.GRP_ID_ESTADO G2
FROM INCIDENCIAS I,
CICLO_DE_VIDA C3,
(SELECT INCI_ID, MAX(CICLOVIDA_ID) CICLOVIDA_ID FROM CICLO_DE_VIDA
GROUP BY INCI_ID) C4
WHERE C3.INCI_ID=C4.INCI_ID
AND C3.INCI_ID=I.INCI_ID
AND C3.CICLOVIDA_ID=C4.CICLOVIDA_ID
AND C3.ESTADO_ID NOT IN ('4','8')
AND I.INCI_FX_CIERRE IS NULL)
) B
GROUP BY B.INCI_ID;

```

Configuración previa dimensión DIM_DOperadora

Nombre: Entrada_DIM_DOperadora

Consulta:

```
SELECT TICKET_ID INCI_ID,
       SUM(CASE WHEN OLD_GROUP='1' THEN EXTRACT(DAY FROM ELAPSED) ELSE 0 END )
SLA_TIME_OPE,
       SUM(CASE WHEN OLD_GROUP='1' THEN 0 ELSE EXTRACT(DAY FROM ELAPSED) END )
SLA_TIME_PROV
FROM (
  SELECT A.INCI_ID TICKET_ID,
         A.F1 OLD_DATE,
         A.E1 OLD_STATE,
         A.G1 OLD_GROUP,
         A.F2 NEW_DATE,
         A.E2 NEW_STATE,
         A.G2 NEW_GROUP,
         A.F2-A.F1 ELAPSED
  FROM (
    SELECT C1.INCI_ID,
           C1.CICLOVIDA_FX F1,
           C1.CICLOVIDA_ID C1,
           C1.ESTADO_ID E1,
           C1.GRP_ID_ESTADO G1,
           CASE WHEN TO_CHAR(C2.CICLOVIDA_FX, 'hh24:mi:ss')='00:00:00'
THEN C1.CICLOVIDA_FX ELSE C2.CICLOVIDA_FX END F2,
           C2.CICLOVIDA_ID C2,
           C2.ESTADO_ID E2,
           C2.GRP_ID_ESTADO G2
    FROM CICLO_DE_VIDA C1,
         CICLO_DE_VIDA C2,
         INCIDENCIAS I
    WHERE C1.INCI_ID = C2.INCI_ID
        AND C1.INCI_ID=I.INCI_ID
        AND I.INCI_FEC_SLA IS NOT NULL
        AND C1.CICLOVIDA_FX >= (SELECT MIN(C3.CICLOVIDA_FX) FROM CICLO_DE_VIDA
C3 WHERE C3.INCI_ID=C1.INCI_ID AND C3.ESTADO_ID='2')
        AND C2.CICLOVIDA_FX >= (SELECT MIN(C3.CICLOVIDA_FX) FROM CICLO_DE_VIDA
C3 WHERE C3.INCI_ID=C2.INCI_ID AND C3.ESTADO_ID='2')
        AND C1.CICLOVIDA_FX <= I.INCI_FEC_SLA
        AND C2.CICLOVIDA_FX <= I.INCI_FEC_SLA
        AND C1.CICLOVIDA_ID < C2.CICLOVIDA_ID
    ) A
  WHERE A.C2 = (
    SELECT MIN( C.CICLOVIDA_ID )
    FROM CICLO_DE_VIDA C
    WHERE C.INCI_ID = A.INCI_ID
        AND C.CICLOVIDA_ID > A.C1)
UNION ALL
  SELECT A.INCI_ID TICKET_ID,
         A.CICLOVIDA_FX OLD_DATE,
         A.ESTADO_ID OLD_STATE,
         A.GRP_ID_ESTADO OLD_GROUP,
         I.INCI_FEC_SLA NEW_DATE,
         A.ESTADO_ID NEW_STATE,
         A.GRP_ID_ESTADO NEW_GROUP,
         I.INCI_FEC_SLA-A.CICLOVIDA_FX ELAPSED
  FROM INCIDENCIAS I,
       CICLO_DE_VIDA A
  WHERE A.INCI_ID=I.INCI_ID
        AND A.CICLOVIDA_ID = (SELECT MAX(B.CICLOVIDA_ID) FROM CICLO_DE_VIDA B WHERE
B.INCI_ID=I.INCI_ID AND B.CICLOVIDA_FX <= I.INCI_FEC_SLA)
        AND I.INCI_FEC_SLA IS NOT NULL)
GROUP BY TICKET_ID;
```

Configuración previa dimensión DIM_ROpen

Nombre: Entrada_DIM_ROpen

Consulta:

```
SELECT C.INCI_ID,
       COUNT(*) REOPEN_TIMES
FROM (SELECT CVI1.INCI_ID,
            CVI1.CICLOVIDA_ID C1,
            CVI1.ESTADO_ID E1,
```

```

CUI2.CICLOVIDA_ID C2,
CUI2.ESTADO_ID E2
FROM CICLO_DE_VIDA CUI1, CICLO_DE_VIDA CUI2
WHERE CUI1.INCI_ID = CUI2.INCI_ID
      AND CUI1.CICLOVIDA_ID < CUI2.CICLOVIDA_ID
      AND CUI1.ESTADO_ID = '3'
      AND CUI2.ESTADO_ID = '9') C,
INCIDENCIAS I
WHERE C.INCI_ID = I.INCI_ID
      AND C.C2 = (SELECT MIN(CUI1.CICLOVIDA_ID)
                  FROM CICLO_DE_VIDA CUI1
                  WHERE CUI1.INCI_ID = C.INCI_ID
                  AND CUI1.CICLOVIDA_ID > C.C1)

GROUP BY C.INCI_ID;

```

Configuración previa dimensión DIM_Patch

Nombre: Entrada_DIM_Patch

Consulta:

```

SELECT I.INCI_ID,
       '1' PATCH
FROM INCIDENCIAS i
LEFT OUTER JOIN PARCHES_DATOS_COMP ent_rpd ON (I.INCI_ID = ENT_RPD.DATOS_ID)
WHERE ENT_RPD.REFPRIM_ID = '2' GROUP BY I.INCI_ID;

```

Configuración previa dimensión DIM_RootCause

Nombre: Entrada_DIM_RootCause

Consulta:

```

SELECT IHS.HS_INCI_ID INCI_ID,
       HS.HS_CAT_ROOT_CAUSE CRC,
       IHS.HS_CRC_DESCRIPTION CRC_DESCRIPTION,
       CASE
         WHEN IHS.HS_LEVEL='1' THEN 'Level_1'
         WHEN IHS.HS_LEVEL='2' THEN 'Level_2'
         WHEN IHS.HS_LEVEL='-1' THEN 'Otros'
         ELSE IHS.HS_LEVEL::varchar
       END CRC_LEVEL,
       CASE
         WHEN IHS.HS_STATUS='0' THEN 'Not_agreed'
         WHEN IHS.HS_STATUS='1' THEN 'Agreed'
         WHEN IHS.HS_STATUS='2' THEN 'Pending_agreement'
         WHEN IHS.HS_STATUS='-1' THEN 'Otros'
         ELSE IHS.HS_STATUS::varchar
       END CRC_STATUS
FROM CAT_ACEPTACION hs, INCIDENCIAS i
LEFT OUTER JOIN ACEPTACION ihs ON (I.INCI_ID = IHS.HS_INCI_ID)
WHERE HS.HS_ID = IHS.HS_ID;

```

Configuración previa dimensión DIM_Extra

Nombre: Entrada_DIM_Extra

Consulta:

```

select I.INCI_ID INCI_ID, OPE.SUBGRP_NOM_CORTO SUBGRP_NOM_CORTO, PA.PAIS_COD PAIS_COD
from CAT_PAIS PA, INCIDENCIAS I LEFT OUTER JOIN SUBGRUPOS ON (I.SUBGRP_ID_INCI = OPE.SUBGRP_ID)
where
I.GRP_ID_INCI = '1'
AND I.PAIS_ID=PA.PAIS_ID
AND
PA.PAIS_COD
IN
('ARG', 'URU', 'COL', 'TLC', 'VEN', 'VFLS', 'ECU', 'PER', 'PFLS', 'MEX', 'TMMFLS', 'COS', 'GUA', 'NIC', 'PAN',
'SAL', 'D', 'ESP', 'I', 'UK');

```

5.1.2.2 Creación de las dimensiones

Posterior a estas agregaciones se quedará un esquema de pasos el siguiente:

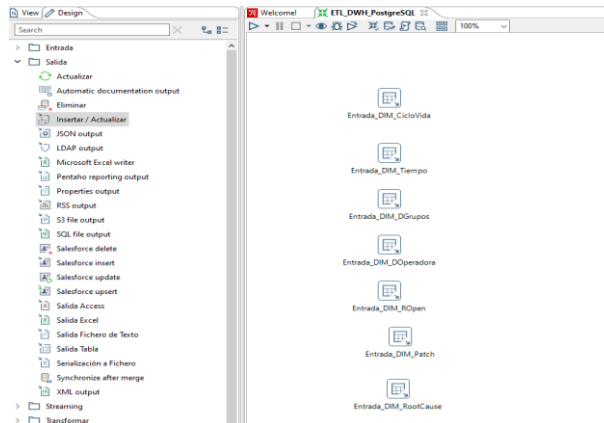


Figura 62. ETL2 Panel Pentaho Pasos 1. Fuente: Elaboración propia

Dado que necesitamos una nueva conexión hacia el almacén de datos, usaremos otra modalidad para crearla que será yendo al panel izquierdo y seleccionando la pestaña de View y haciendo click derecho sobre Conexiones a base de datos para agregar la nueva conexión.

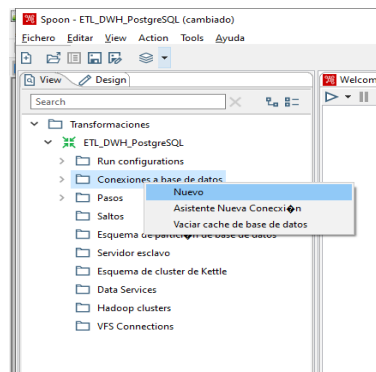


Figura 63. ETL2 Agregar conexión BD dwh. Fuente: Elaboración propia

Luego, tal y como se configuraron las anteriores se hará, aunque en este caso para la dwh_EXP, los datos son los siguientes:

Connection name: dwh_EXP, **Host Name:** localhost, **Database Name:** dwh_EXP, **Username:** dwh, **Password:** dwh.

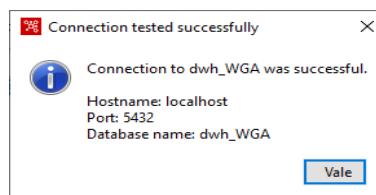


Figura 64. ETL2 Test conexión BD dwh. Fuente: Elaboración propia

La siguiente tarea será crear los pasos que nos permitirán actualizar el almacén de datos, en este caso usaremos el de "Insertar/Actualizar" para cada uno de los pasos ya creados, con este nos llevaremos los datos a la base de datos "dwh_EXP"

En la pestaña derecha, seleccionaremos Design y nos iremos al grupo de pasos de Salida, donde elegiremos el de “Insertar/Actualizar” y lo arrastraremos al panel central.

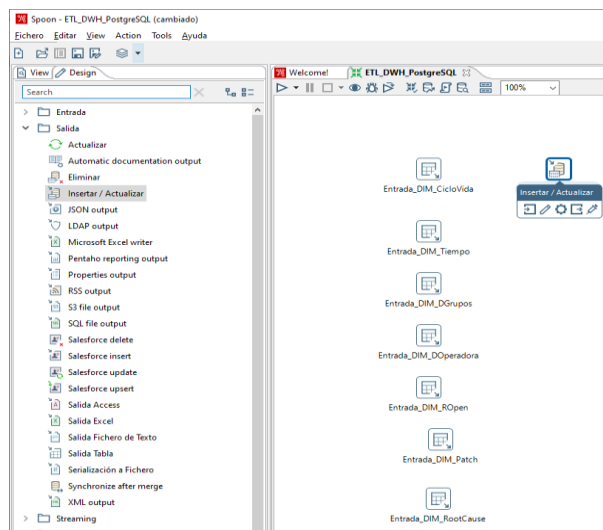


Figura 65. ETL2 Agregar Paso Insertar en DWH. Fuente: Elaboración propia

Le daremos doble click y empezaremos a configurarlo, se agrega nombre de paso, conexión, esquema destino, nombre de la tabla (la crearemos en este paso al final), le daremos click a Obtener Campos y a Obtener Campos de actualización; obteniendo una vista como la que se muestra a continuación:

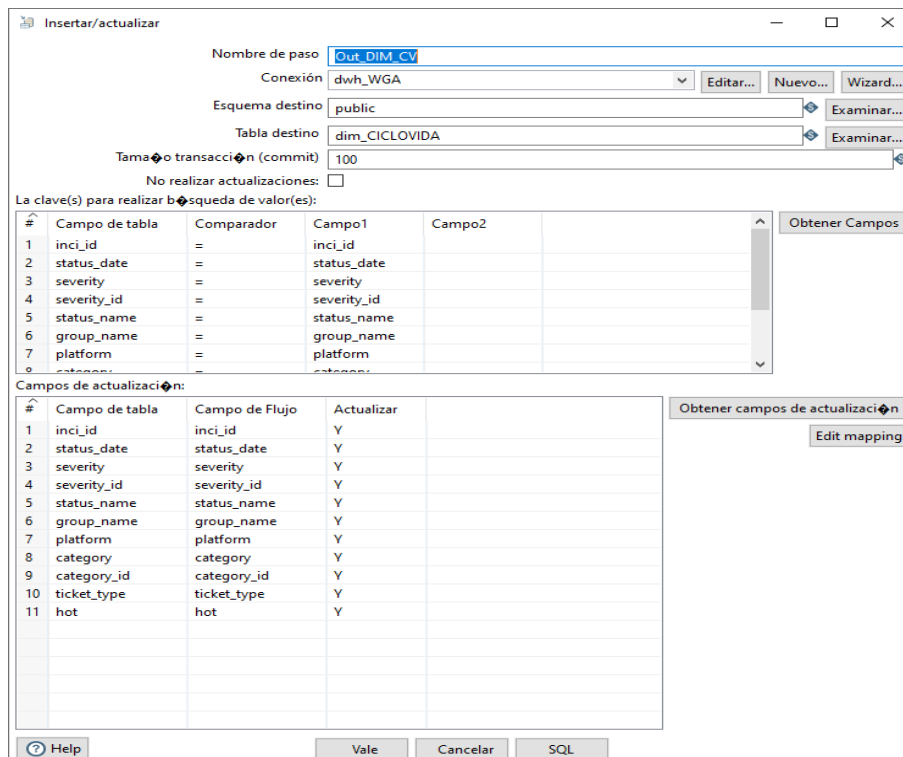


Figura 66. ETL2 Configurar Paso Insertar en DWH. Fuente: Elaboración propia

Dado que no hemos creado la tabla, usaremos la funcionalidad de SQL que parece entre los botones de abajo para crearlo con el mismo Pentaho, a continuación, se muestra el contenido:

```

Simple SQL editor
SQL statements, separated by semicolon ;
CREATE TABLE "public".dim_CICLOVIDA
(
  inci_id BIGINT
  status_date TIMESTAMPTZ
  severity VARCHAR(50)
  severity_id BIGINT
  status_name VARCHAR(100)
  group_name VARCHAR(30)
  platform VARCHAR(30)
  category VARCHAR(200)
  category_id BIGINT
  ticket_type VARCHAR(50)
  hot SMALLINT
)
CREATE INDEX idx_dim_CICLOVIDA_lookup ON "public".dim_CICLOVIDA(inci_id, status_date, severity, severity_id, status_date, severity_id, status_name, group_name, platform, category, category_id, ticket_type, hot)

```

Figura 67. ETL2 Agregar Creación Dimension DWH. Fuente: Elaboración propia

Y procedemos a darle Execute, luego a Vale.

Vamos a la interfaz de pgAdmin para confirmar la creación:

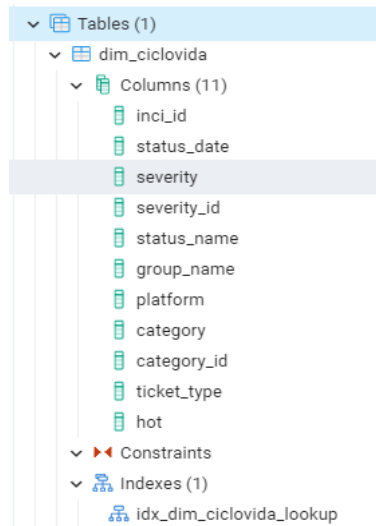


Figura 68. ETL2 Validación Dimension Postgres. Fuente: Elaboración propia

Seguiremos el mismo procedimiento para los siguientes pasos, obteniendo la siguiente foto final:

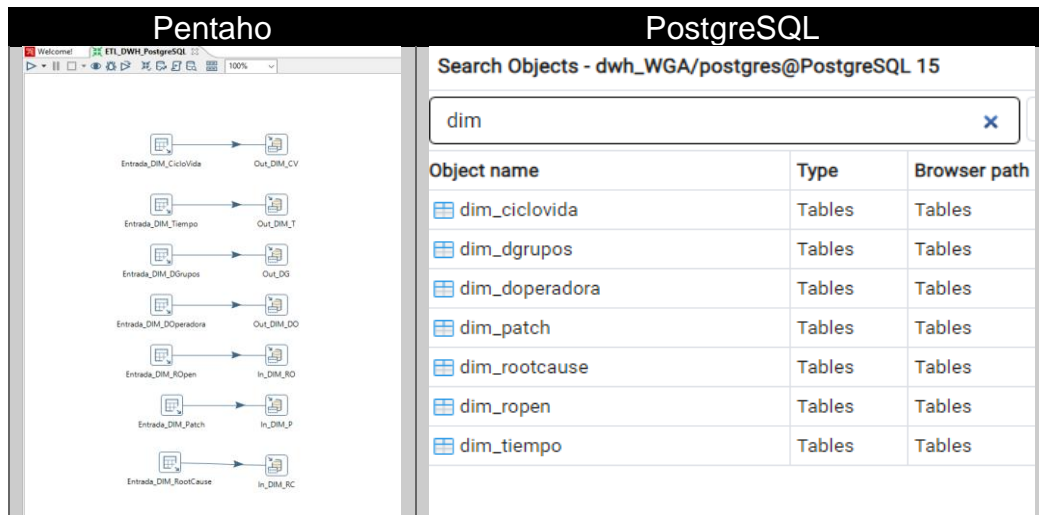


Figura 69. ETL2 Foto Final Dimensiones DWH. Fuente: Elaboración propia

Ahora procederemos a lanzar la transformación para poblar el almacén de datos.

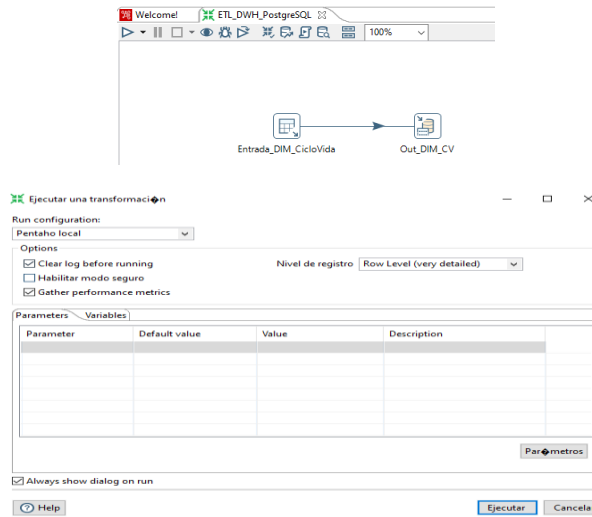


Figura 70. ETL2 Pentaho Ejecutar tarea poblar DWH. Fuente: Elaboración propia

Una vez finalizados veremos todos las confirmaciones en verde en la esquina de cada tarea:

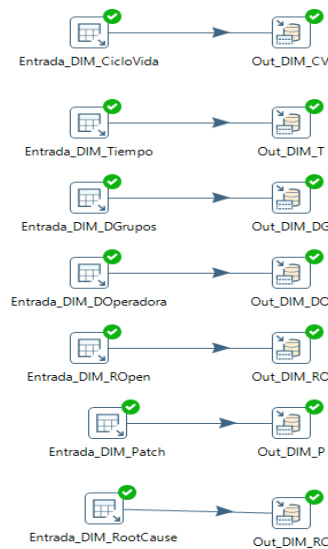


Figura 71. ETL2 Pentaho Fin Ejecución. Fuente: Elaboración propia

Obtendremos el siguiente cuadro con los tiempos:

#	Nombre Paso	Numero Copia	Leido	Escrito	Entrada	Salida	Actualizado	Rejected	Errores	Act ivo	Tiempo	Velocidad (r/s)
1	Entrada_DIM_Tiempo	0	0	305229	305229	0	0	0	0	Finalizado	3mn 27s	1.469
2	Entrada_DIM_Patch	0	0	79288	79288	0	0	0	0	Finalizado	49.6s	1.597
3	Entrada_DIM_DOperadora	0	0	15088	15088	0	0	0	0	Finalizado	13.9s	1.089
4	Entrada_DIM_CicloVida	0	0	43041	43041	0	0	0	0	Finalizado	41.5s	1.036
5	Entrada_DIM_DGrupos	0	0	305780	305780	0	0	0	0	Finalizado	3mn 39s	1.396
6	Entrada_DIM Rope n	0	0	1966	1966	0	0	0	0	Finalizado	1.7s	1.146
7	Entrada_DIM_RootCause	0	0	13556	13556	0	0	0	0	Finalizado	3.2s	4.258
8	Out_DIM_T	0	305229	305229	305229	305.229	0	0	0	Finalizado	3mn 34s	1.426
9	Out_DIM_DG	0	305780	305780	305780	305780	0	0	0	Finalizado	3mn 43s	1.366
10	Out_DIM_DO	0	15088	15088	15088	15088	0	0	0	Finalizado	21.8s	691
11	Out_DIM_RO	0	1966	1966	1966	1966	0	0	0	Finalizado	3.2s	611
12	Out_DIM_P	0	79288	79288	79288	79288	0	0	0	Finalizado	55.9s	1.419
13	Out_DIM_RC	0	13556	13556	13556	13556	0	0	0	Finalizado	11.7s	1.158
14	Out_DIM_CV	0	43041	43041	43041	43041	0	0	0	Finalizado	52.3s	823

Figura 72. ETL2 Pentaho Resumen Ejecución. Fuente: Elaboración propia

5.1.2.3 Validación de la inserción de las tablas

Validamos en el almacén de datos si tenemos la información cargada correctamente, al menos en cantidad.

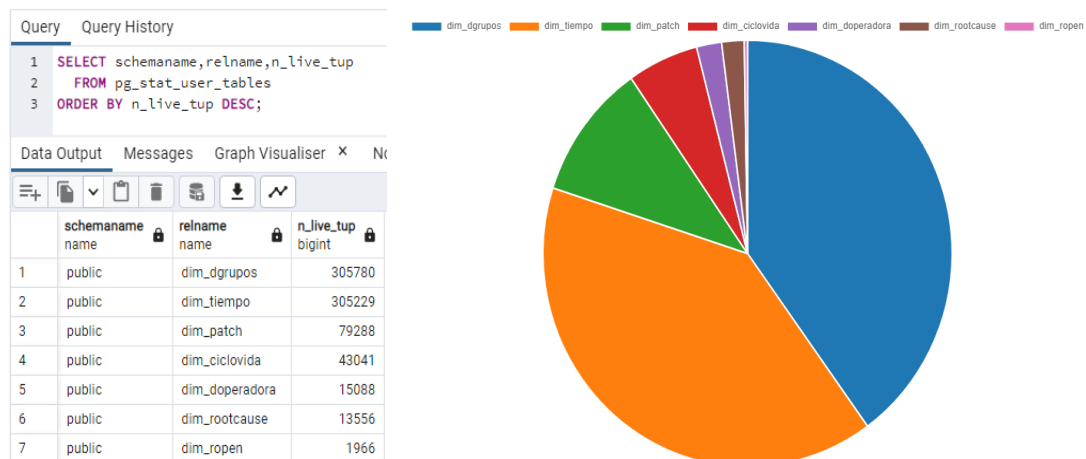


Figura 73. ETL2 Registros Dimensiones y Grafico PostgreSQL. Fuente: Elaboración propia

5.1.2.4 Creación de la tabla de hechos

De estas dimensiones cargadas crearemos nuestra tabla de hechos que llamaremos **fact_CRS** (tablas de hechos **Customer Request Support**), para realizar esta carga sobre nuestra tabla de hechos usaremos directamente la interfaz que nos ofrece pgAdmin (PostgreSQL) para ejecutar la sentencia y crear nuestra tabla de hechos, la sentencia seria la siguiente:

```

create table fact_crs as (SELECT I.INCI_ID TICKET_ID,
CASE WHEN A.PLATFORM='TARIF' THEN 'TARIFICADOR' WHEN A.PLATFORM='GEST' THEN 'GESTOR'
WHEN A.PLATFORM='SIPS' THEN 'IVR' ELSE A.PLATFORM END PLATFORM,
I.PAIS_COD COUNTRY,
CASE WHEN A.SEVERITY='Without Priority' THEN 'Low' ELSE A.SEVERITY END SEVERITY,
A.STATUS_NAME TICKET_STATUS,
A.GROUP_NAME TICKET_GROUP,
F.F_GEN GENERATION_DATE,
TO_CHAR(F.F_GEN,'yyyy') GENERATION_YEAR,
TO_CHAR(F.F_GEN,'mm') GENERATION_MONTH,
F.F_OPEN OPEN_DATE,
(F.LEAD_TIME)::varchar LEAD_TIME_DAYS,
coalesce(F.F_CLOSE,F.F_REJECT) CLOSURE_DATE,
TO_CHAR(coalesce(F.F_CLOSE,F.F_REJECT),'yyyy') CLOSURE_YEAR,
TO_CHAR(coalesce(F.F_CLOSE,F.F_REJECT),'mm') CLOSURE_MONTH,
(F.PF_TIME)::varchar PF_DAYS,
(F.AGE)::numeric ALIVE_TICKET_AGE_DAYS,
(LOCALTIMESTAMP-A.STATUS_DATE)::varchar LAST_UPDATE,
F.F_RECOVERY REMEDY_DATE,
(F.F_RECOVERY-F.F_OPEN)::varchar SLA_TIME_DAYS,
(S.SLA_TIME_OPE)::numeric SLA_TIME_OPE_DAYS,
(S.SLA_TIME_PROV)::numeric SLA_TIME_proveedor_DAYS,
(S.SLA_TIME_PROV)::numeric *24 SLA_TIME_proveedor_HOURS,
A.CATEGORY CATEGORIZATION,
CR.SERVICE_OUTAGE SERVICE_OUTAGE,
(CASE WHEN F.F_RECOVERY IS NULL THEN '0' ELSE CASE WHEN A.CATEGORY_ID='30' THEN '0' WHEN
A.CATEGORY_ID='31' THEN '0' WHEN A.CATEGORY_ID='32' THEN '0' WHEN A.CATEGORY_ID='33' THEN '0'
WHEN A.CATEGORY_ID='34' THEN '0' WHEN A.CATEGORY_ID='35' THEN '0' WHEN A.CATEGORY_ID='43' THEN
'0' ELSE CASE WHEN CR.T_ALCANCE_PERD_SERV_ID='1' THEN '0' ELSE CASE WHEN
SIGN(coalesce(S.SLA_TIME_PROV,T.T_PROV-EXTRACT(DAY FROM (F.LEAD_TIME)))-(2/24))='-1' THEN '0'
WHEN SIGN(coalesce(S.SLA_TIME_PROV,T.T_PROV-EXTRACT(DAY FROM (F.LEAD_TIME)))-(2/24))='0' THEN
'0' ELSE CASE WHEN SIGN(coalesce(S.SLA_TIME_PROV,T.T_PROV-EXTRACT(DAY FROM (F.LEAD_TIME)))-
(4/24))='-1' THEN CASE WHEN A.SEVERITY_ID='1' THEN '1' ELSE '0' END WHEN
SIGN(coalesce(S.SLA_TIME_PROV,T.T_PROV-EXTRACT(DAY FROM (F.LEAD_TIME)))-(4/24))='0' THEN CASE
WHEN A.SEVERITY_ID='1' THEN '1' ELSE '0' END ELSE CASE WHEN
SIGN(coalesce(S.SLA_TIME_PROV,T.T_PROV-EXTRACT(DAY FROM (F.LEAD_TIME)))-(35))='-1' THEN CASE
WHEN A.SEVERITY_ID='1' THEN '1' WHEN A.SEVERITY_ID='4' THEN '1' ELSE '0' END WHEN

```

```

SIGN(coalesce(S.SLA_TIME_PROV,T.T_PROV-EXTRACT(DAY FROM (F.LEAD_TIME)))-(35))='0' THEN CASE
WHEN A.SEVERITY_ID='1' THEN '1' WHEN A.SEVERITY_ID='4' THEN '1' ELSE '0' END ELSE CASE WHEN
SIGN(coalesce(S.SLA_TIME_PROV,T.T_PROV-EXTRACT(DAY FROM (F.LEAD_TIME)))-(56))='-1' THEN CASE
WHEN A.SEVERITY_ID='1' THEN '1' WHEN A.SEVERITY_ID='4' THEN '1' WHEN A.SEVERITY_ID='2' THEN '1'
ELSE '0' END WHEN SIGN(coalesce(S.SLA_TIME_PROV,T.T_PROV-EXTRACT(DAY FROM (F.LEAD_TIME)))-
(56))='0' THEN CASE WHEN A.SEVERITY_ID='1' THEN '1' WHEN A.SEVERITY_ID='4' THEN '1' WHEN
A.SEVERITY_ID='2' THEN '1' ELSE '0' END ELSE '1' END END END END END END)::DECIMAL
SLA_VIOLATION,
(CASE WHEN S.SLA_TIME_OPE IS NULL THEN T.T_OPE ELSE null END)::numeric
SLA_TIME_OPE_ESTIMATED_DAYS,
(CASE WHEN S.SLA_TIME_PROV IS NULL THEN T.T_PROV-EXTRACT(DAY FROM (F.LEAD_TIME)) ELSE null END
)::numeric SLA_TIME_proveedor_ESTIMATED_DAYS,
(CASE WHEN F.F_RECOVERY IS NULL THEN CASE WHEN A.CATEGORY_ID='30' THEN '0' WHEN
A.CATEGORY_ID='31' THEN '0' WHEN A.CATEGORY_ID='32' THEN '0' WHEN A.CATEGORY_ID='33' THEN '0'
WHEN A.CATEGORY_ID='34' THEN '0' WHEN A.CATEGORY_ID='35' THEN '0' WHEN A.CATEGORY_ID='43' THEN
'0' ELSE CASE WHEN CR.T_ALCANCE_PERD_SERV_ID='1' THEN '0' ELSE CASE WHEN
SIGN(coalesce(S.SLA_TIME_PROV,T.T_PROV-EXTRACT(DAY FROM (F.LEAD_TIME)))-(2/24))='-1' THEN '0'
WHEN SIGN(coalesce(S.SLA_TIME_PROV,T.T_PROV-EXTRACT(DAY FROM (F.LEAD_TIME)))-(2/24))='0' THEN
'0' ELSE CASE WHEN SIGN(coalesce(S.SLA_TIME_PROV,T.T_PROV-EXTRACT(DAY FROM (F.LEAD_TIME)))-
(4/24))='-1' THEN CASE WHEN A.SEVERITY_ID='1' THEN '1' ELSE '0' END WHEN
SIGN(coalesce(S.SLA_TIME_PROV,T.T_PROV-EXTRACT(DAY FROM (F.LEAD_TIME)))-(4/24))='0' THEN CASE
WHEN A.SEVERITY_ID='1' THEN '1' ELSE '0' END ELSE CASE WHEN
SIGN(coalesce(S.SLA_TIME_PROV,T.T_PROV-EXTRACT(DAY FROM (F.LEAD_TIME)))-(35))='-1' THEN CASE
WHEN A.SEVERITY_ID='1' THEN '1' WHEN A.SEVERITY_ID='4' THEN '1' ELSE '0' END WHEN
SIGN(coalesce(S.SLA_TIME_PROV,T.T_PROV-EXTRACT(DAY FROM (F.LEAD_TIME)))-(35))='0' THEN CASE
WHEN A.SEVERITY_ID='1' THEN '1' WHEN A.SEVERITY_ID='4' THEN '1' ELSE '0' END ELSE CASE WHEN
SIGN(coalesce(S.SLA_TIME_PROV,T.T_PROV-EXTRACT(DAY FROM (F.LEAD_TIME)))-(56))='-1' THEN CASE
WHEN A.SEVERITY_ID='1' THEN '1' WHEN A.SEVERITY_ID='4' THEN '1' WHEN A.SEVERITY_ID='2' THEN '1'
ELSE '0' END WHEN SIGN(coalesce(S.SLA_TIME_PROV,T.T_PROV-EXTRACT(DAY FROM (F.LEAD_TIME)))-
(56))='0' THEN CASE WHEN A.SEVERITY_ID='1' THEN '1' WHEN A.SEVERITY_ID='4' THEN '1' WHEN
A.SEVERITY_ID='2' THEN '1' ELSE '0' END ELSE '1' END END END END END END ELSE '0'
END)::decimal SLA_VIOLATION_ESTIMATED,
(T.T_OPE)::numeric OPE_TIME_DAYS,
(T.T_PROV)::numeric PROVEEDOR_TIME_DAYS,
(T.T_SR)::numeric SR_TIME_DAYS,
(T.T_AN3_SR)::numeric AN3_SR_TIME_DAYS,
(T.T_MC)::numeric MC_TIME_DAYS,
(T.T_DES)::numeric DES_TIME_DAYS,
(T.T_AN3_DES)::numeric AN3_DES_TIME_DAYS,
(T.T_PS)::numeric PS_TIME_DAYS,
(T.T_CERTIF)::numeric CERT_TIME_DAYS,
(T.T_SMaqueta)::numeric SMAQ_TIME_DAYS,
(T.T_CertifOPE)::numeric CERTOP_TIME_DAYS,
(T.T_INT)::numeric INT_TIME_DAYS,
(T.T_ESP)::numeric ESPEC_TIME_DAYS,
(T.T_OPEINS)::numeric OPEINS_TIME_DAYS,
(T.T_HGD)::numeric HGD_TIME_DAYS,
(CASE WHEN R.REOPEN_TIMES IS NULL THEN '0' WHEN R.REOPEN_TIMES='0' THEN '0' ELSE '1'
END)::numeric REOPEN_TIMES,
(P.PATCH)::numeric L3_PATCH,
(CASE WHEN A.SEVERITY='Critical' THEN '1' ELSE null END)::numeric CRITICAL_TT,
(CASE WHEN A.SEVERITY='High' THEN '1' ELSE null END)::numeric HIGH_TT,
(CASE WHEN A.SEVERITY='Medium' THEN '1' WHEN A.SEVERITY='Low' THEN '1' WHEN
A.SEVERITY='Without Priority' THEN '1' ELSE null END)::numeric MEDIUM_LOW_TT,
(CASE WHEN A.STATUS_NAME='Closed' THEN '1' WHEN A.STATUS_NAME='Rejected' THEN '1' ELSE
null END)::numeric CLOSED_TT,
(F.A3DES)::numeric AN_TLS_DES,
A.TICKET_TYPE,
A.HOT,
CR.CRC_LEVEL CRC_LEVEL,
CASE WHEN CR.CRC_LEVEL='Level 1' THEN CASE WHEN CR.CRC_STATUS='Pending agreement' THEN
0 WHEN CR.CRC_STATUS='Not agreed' THEN 0 ELSE 1 END ELSE 0 END CRC_ESCALATED_L1,
CASE WHEN CR.CRC_LEVEL='Level 1' THEN 0 ELSE 1 END CRC_ESCALATED_L2,
CR.CRC_STATUS CRC_STATUS,
CASE WHEN CR.CRC_STATUS='Not agreed' THEN 1 ELSE 0 END CRC_NOT_AGREED,
CASE WHEN CR.CRC_STATUS='Pending agreement' THEN 1 ELSE 0 END CRC_PENDING_HANDSHAKE,
CR.CRC_CRC,
CR.CRC_DESCRIPTION CRC_DESCRIPTION
FROM
dim_ciclovida A,
dim_tiempo F,
dim_dgrupos T,
dim_doperadora S,
dim_ropen R,
dim_patch P,
dim_rootcause CR,
dim_extra I

```

```
WHERE coalesce(F.F_CLOSE,coalesce(F.F_REJECT,F.F_GEN)) >= TO_DATE('01/01/2018
00:00:00','dd/mm/yyyy hh24:mi:ss')
AND (A.CATEGORY != 'OTE Error en herramienta ' OR A.CATEGORY IS NULL)
AND I.INCI_ID=A.INCI_ID);
```

Se ejecuta la sentencia y se queda creada la tabla:



Figura 74. Creación tabla de hechos FACT_CRS

5.2 Validaciones Datos Post ETL

En las validaciones realizadas post volcado de las dimensiones y tabla de hechos “fact_CRS”, para limpiar la data hemos procedido a usar la herramienta “**Editor de Power Query**” que pertenece al Power BI, en la data hemos encontrado:

- Una gran cantidad de datos a **null**, en campos que son numéricos, donde usando las opciones de “Reemplazar” se ha procedido a cambiar los **null** por cero en las columnas que lo necesitaban.
- Se han eliminado las columnas que se trajeron con datos de nombres comerciales de las operadoras y se han dejado las iniciales de los países a los que hacen referencia para el análisis asociado a región o zona geográfica.
- Se han reemplazado subnodos que antes eran individuales y se han agregado al conteo de los 2 nodos principales que analizaremos.
- Se han reemplazado nombres comerciales de los nodos por los definidos en este documento “**Gestor**” y “**Tarificador**”.
- Se han cambiado tipos que con el ETL se han transformado de numéricos a textuales, para esto hemos usado las pestaña de “Transformar” y seleccionando el tipo al que queríamos pasar.

A continuación, un pantallazo de los cambios aplicados.

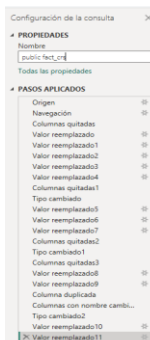


Figura 75. Limpieza PowerBI tabla FACT_CRS. Elaboración propia

Se agregará como **anexo (8.2)** la lista completa de las transformaciones necesarias para dejar la data adaptada para la obtención de los reportes finales.

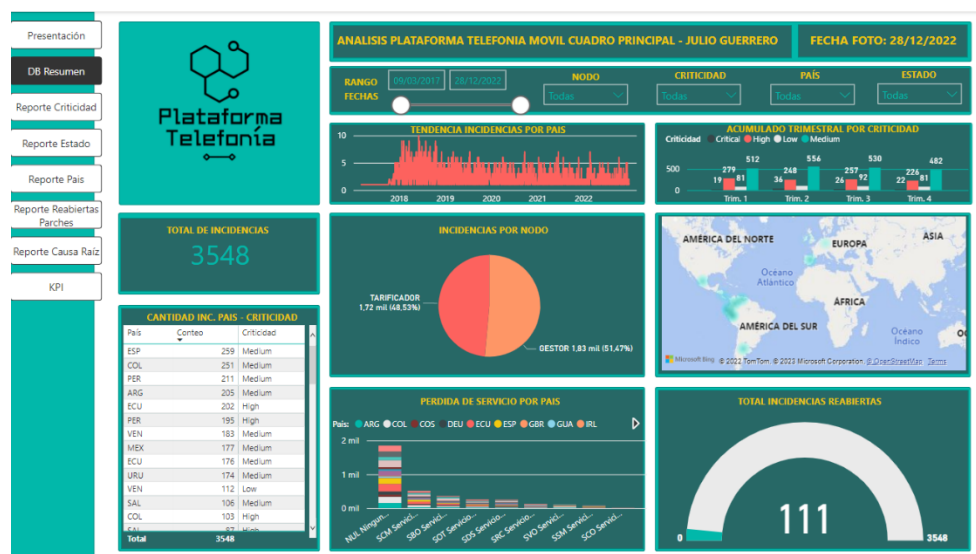
5.3 Implementación Reportes

Para los siguientes reportes que se visualizaran mantendremos una estructura similar es decir todos contarán con:

- Un panel izquierdo con el menú que nos ayudara en la navegación entre los distintos reportes, así como también ir a la presentación y cuadro de mando resumen.
- El siguiente panel del reporte se iniciará en la parte superior con el logo, debajo las tarjetas fijas que nos dará un total de los parámetros asignados y también tendremos las tarjetas que se actualizarán de acuerdo con la selección y en la parte inferior de este tendremos el cuadro fijo con los atributos asociados que perfilarán el reporte a criterios específicos.
- El panel más grande, a la derecha, iniciará en la parte superior con los filtros de fechas y parámetros que nos permitirán seleccionar distintas dimensiones y granularidades, finalmente debajo de este tendremos los gráficos que nos ayudaran a mostrar la información necesaria para ver el comportamiento de las incidencias en base a país, acumulados y porcentajes.

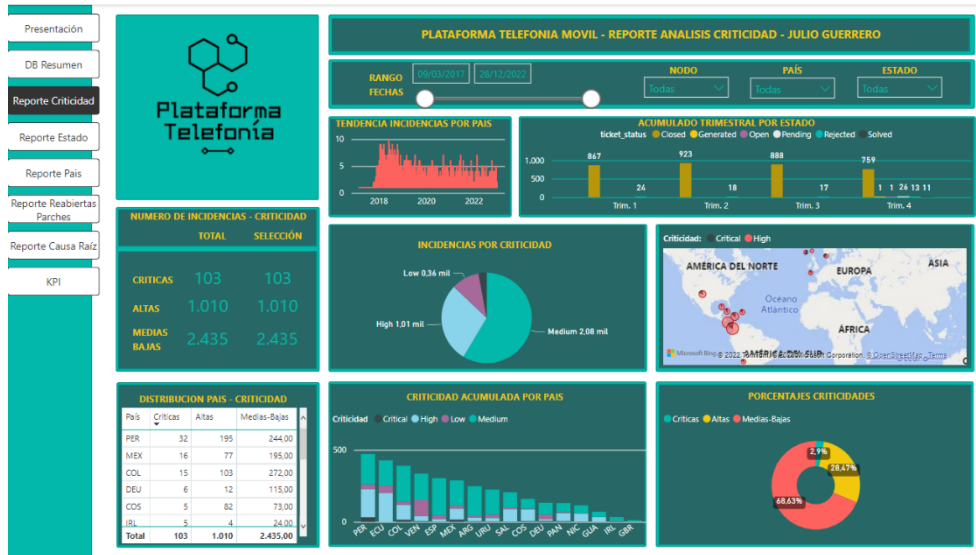
5.3.1 Implementación Dashboard Resumen

Se presenta a continuación la versión final del Dashboard Resumen, en la parte izquierda del diseño se ha agregado un menú para poder ir a los distintos reportes, también se ha suavizado el fondo con un 30% de transparencia para que los colores oscuros de los gráficos no se pierdan, tanto menú como el porcentaje de transparencia se mantendrán en el resto de los reportes:



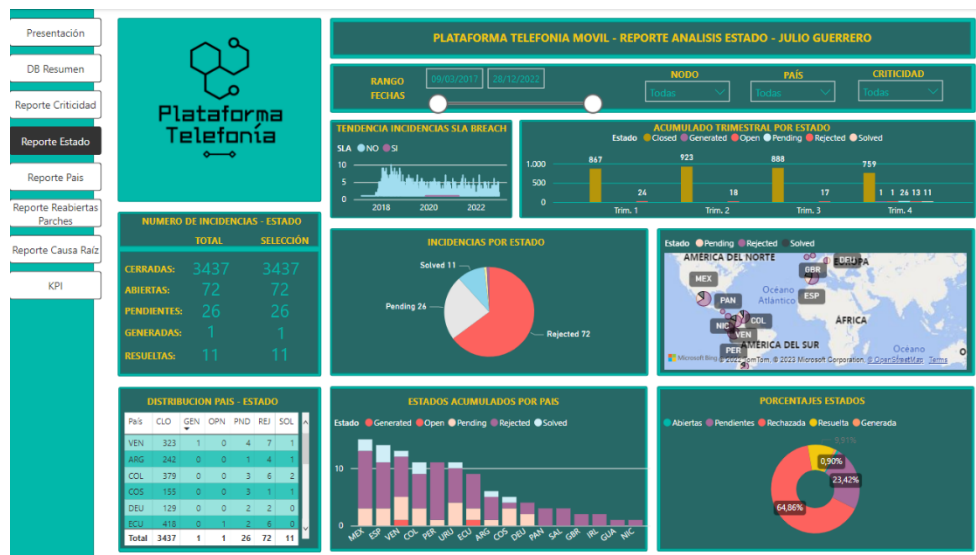
5.3.2 Implementación Reporte Criticidad

A continuación, se mostrará la visualización asociada al reporte enfocado en la criticidad de las incidencias, las más relevantes serán la crítica per se, esta esta asociada a llamadas de guardias, tienen un SLA de 2 horas, alta que tendrá un SLA de 4 horas y las criticidades medias y bajas que suelen tener un SLA lapso y variable de acuerdo con cada operadora.



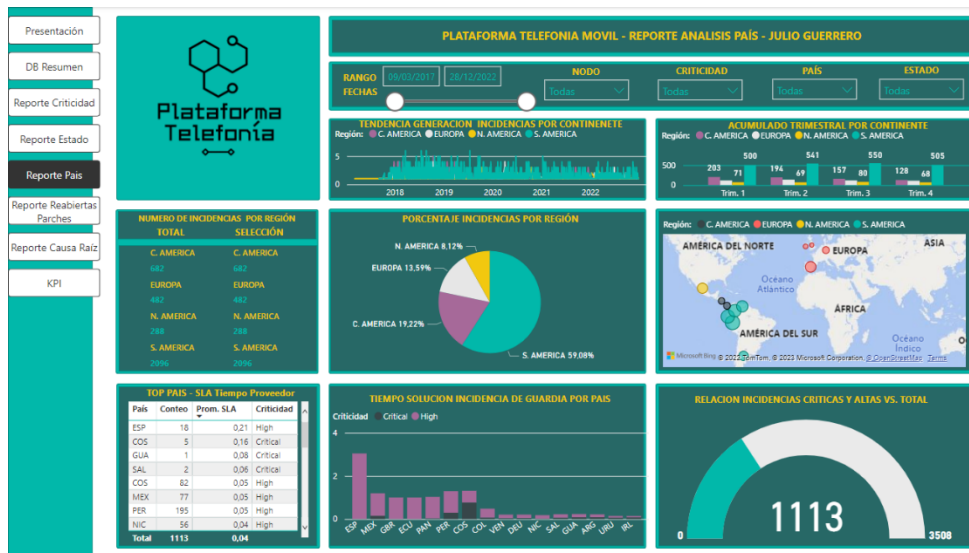
5.3.3 Implementación Reporte Estado

A continuación, se mostrará la visualización asociada al reporte enfocado en el estado de las incidencias, las más importantes serán las cerradas, pendientes, generada, abiertas y resueltas, se podrán observar grandes bloques de cerradas porque es el estado final en el que se dejaron las incidencias, por lo que en algunos se han aplicado filtros para evitar que visualmente nos haga perder el foco en los otros estados.



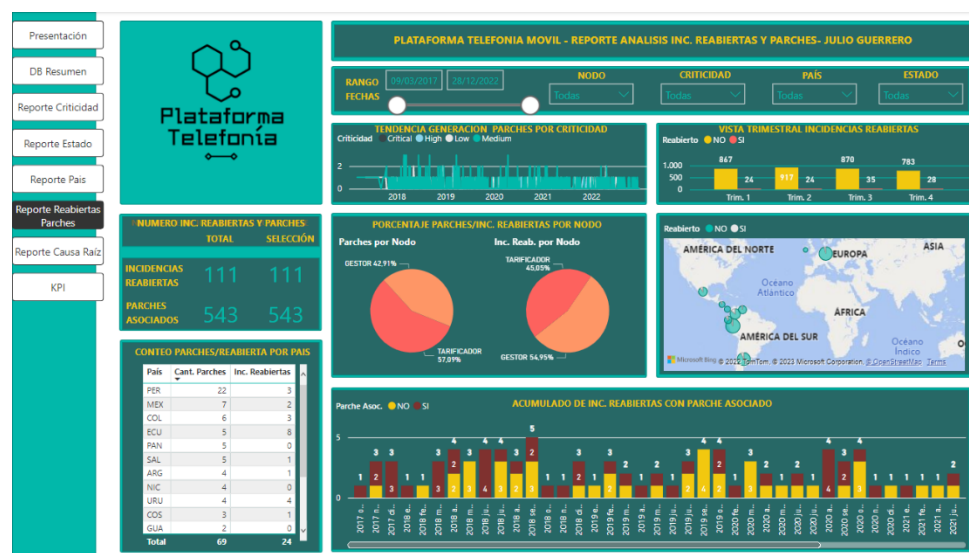
5.3.4 Implementación Reporte País

A continuación, se mostrará la visualización asociada al reporte enfocado en los países y su relación con los otros parámetros o dimensiones definidas, en algunos casos se han usado grupos por regiones (Europa, Norte, Sur y Centro América) para que visualmente nos sea más fácil leer lo que quiere transmitir el grafico o información. Los países son 16 en su versión más extensa, cabe aclarar que algunos países fueron dejando de usar la plataforma como por ejemplo Nicaragua, Irlanda, aun asi si decidimos analizar cómo se comportaron estas operadoras tendremos la información cargada.



5.3.5 Implementación Reporte Reabiertas y Parches

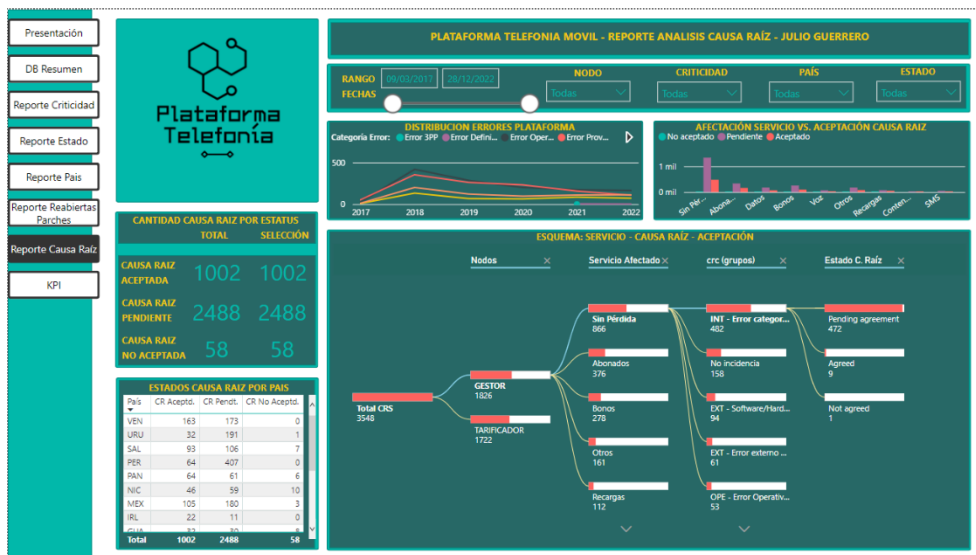
A continuación, se mostrará la visualización asociada al reporte enfocado en las incidencias reabiertas que nos indicara un problema tanto interno como externo ya que por algún motivo no hubo un correcto análisis y el en caso de aquellas incidencias que tengan parche asociado nos indicara la relación entre las operadoras y el coste asociado a otros grupos internos como el de desarrollo de la plataforma (N3).



5.3.6 Implementación Reporte Causa Raíz

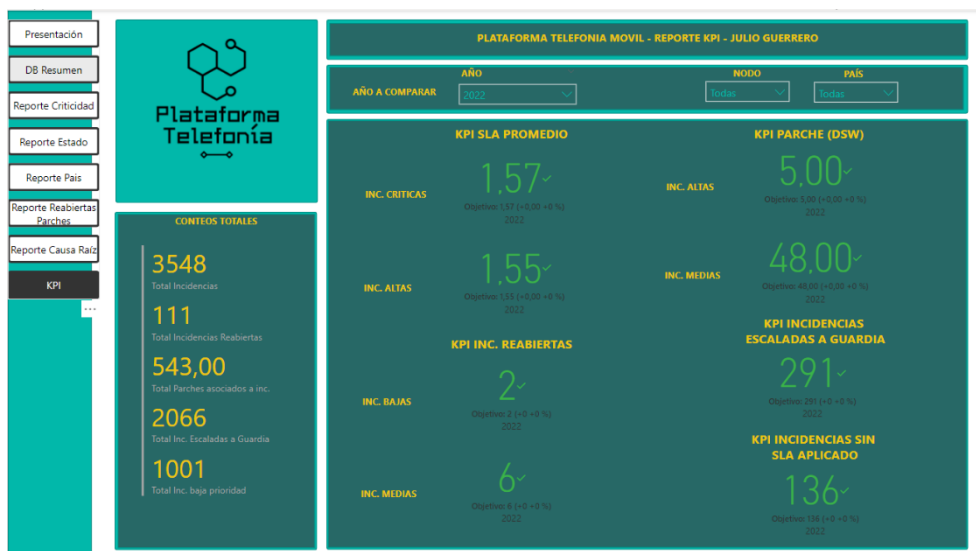
A continuación, se mostrará la visualización asociada al reporte enfocado en analizar servicios afectados, la causa raíz asociada a estos y su estado de aceptación por parte de la operadora, se han transformado los servicios para agruparlos y que sea más interpretable la información.

Al igual que se han hecho grupos también de las causas raíz para categorizarlos y asociarlos a problemas Internos (responsabilidad del proveedor del soporte), Externos (responsabilidad de terceros ya sea hardware o software no asociado a la plataforma) y Operadora (responsabilidad de la operadora, mala categorización, operativa o configuración). Otro parámetro para tener en cuenta será el estado en que se encuentra esta causa raíz que podrá ser pendiente de aceptación, aceptada o no aceptada.



5.3.7 Implementación KPIs

Se observarán KPIs medibles como SLA, Reabiertas, Causa Raíz Aceptación, Incidencias con parche asociado, Incidencias escaladas y baja prioridad.



6. Conclusiones

Este capítulo tendrá como finalidad el desglosar los aspectos más relevantes que se han encontrado durante la elaboración de este proyecto y como se ha desarrollado o resuelto para obtener el resultado esperado.

6.1 Asociadas al desarrollo del proyecto

El planteamiento inicial fue lograr realizar un proyecto que permitiera explotar datos de una base relacional que lleva el flujo completo del ciclo de vida de las incidencias relacionadas con la gestión de abonados y la tarificación de un conjunto de operadoras de telefonía móvil en Europa y América, idea que fue madurando entrega a entrega, pasando por la planificación, análisis, diseño y finalmente implementar la solución asociada al proyecto.

Durante todo el desarrollo se tuvieron que tomar muchas decisiones, como las herramientas idóneas para usar tanto en la extracción de la data, el cómo cargarla en local para poder explotarla, el dónde cargarla, haciendo uso para este punto de un almacén de datos, una vez decidido esto, el saber elegir la estructura que compondría este repositorio para luego saber elegir que datos finales tendríamos para analizar y presentar en los reportes. Y aunque leyendo lo previo pueda sonar como una concatenación de ideas y procedimiento claros, no fue hasta dedicarle muchas horas que se pudo obtener como resultado el proyecto final.

6.2 Asociadas a lo alcanzado en el proyecto

Tal y como se ha desarrollado el proyecto, al día de la entrega final:

- Se puede dar por alcanzado el objetivo general que era analizar el comportamiento de la plataforma centrados siempre en las incidencias y sus interacciones con el ciclo de vida de estas, los parches (entregas generadas por DSW), siempre enfocados en el entorno productivo y en todas las operadoras.
- Se pueden dar por alcanzado los objetivos específicos de manera que se ha logrado conocer y aprender de tres tecnologías de uso en entornos de trabajo real como son Pentaho, PostgreSQL y Power BI, sumado a esto podemos dar por desarrolladas capacidades para poder extraer, transformar y cargar distintos tipos de datos de distintas fuentes y finalmente se ha podido desarrollar un proyecto que permite visualizar datos dándoles a estos la capacidad de convertirse en información.

6.3 Asociadas a una posible mejora

Como ya se ha comentado este proceso se inició con limitada experiencia y conocimiento tanto en las tecnologías como en las metodologías relacionadas al desarrollo del proyecto, por lo que conforme se ha ido empapándose y desarrollando los capítulos he visto por la que se podría seguir mejorando el proyecto.

Esta posibilidad es la de llevar el almacén de datos a la nube, entre las opciones que vi para hacerlo serian:

Utilizar un proveedor de servicios de almacenamiento de datos en la nube, como Amazon Redshift, Google BigQuery o Azure Synapse Analytics. Estos servicios proporcionarían en el caso de ser necesario una plataforma escalable y de altas prestaciones para almacenar y analizar grandes cantidades de datos.

Para realizar esto, se tendrían que migrar los datos a una plataforma en la nube utilizando herramientas de migración de datos, como AWS Data Migration Service o Azure Data Factory. Estas herramientas permitirían transferir los datos desde la base de datos local a un almacén de datos en la nube.

Casi todas las herramientas mencionadas previamente comparten similitudes como:

- Análisis en tiempo real, es decir permiten realizar consultas en tiempo real sobre grandes volúmenes de datos, lo que facilita la toma de decisiones en tiempo real.
- Costos, utilizan un modelo de tarifa por uso, lo que significa que solo se pagan los recursos que se utilizan, lo cual es una ventaja frente a tener una infraestructura propia.
- Seguridad, ofrecen una serie de características de seguridad, como la cifrado de datos lo que garantiza la seguridad de estos.

6.4 Asociadas a lo personal

Un problema que se tuvo fue la falta de expertise en este tipo de tecnologías y metodologías, quizás el más grande de los obstáculos sorteados durante todo el proyecto, pero una vez que se había decidido en que área realizar mi proyecto de fin de grado, realice una inmersión diaria tanto antes de iniciar el semestre como durante, buscando información, tutoriales, foros de cada herramienta y mucho ensayo y error.

Se ha intentado optimizar toda la información que se ha ido recabando, enfocando el aprendizaje en conceptos claves y como estos se relacionaban con la consecución de los objetivos del proyecto, ya que se tenía claro de que por tiempo personal no tendría la capacidad de tener todos los detalles técnicos de las herramientas.

En lo relativo a la planificación inicial se intentado cumplir en tiempo y forma con esta, aunque diversos motivos personales nos han llevado a modificar algunos plazos, aunque estos cambios no han impactado en la consecución de los objetivos, y en la medida de lo posible se ha intentado minimizarlos.

Todas las retroalimentaciones recibidas se han asumido como parte de los objetivos a lograr a corto plazo, para que cada PEC lleven estos implementados o desarrollados.

Una frase que resumiría las conclusiones a nivel personal sería **“El éxito en la vida no se mide por lo que logras sino por los obstáculos que superas”**.

7. Glosario

- BI – Business Intelligence
- DWH/Data Warehouse – Almacen de datos
- Dashboard – Cuadro de mandos
- ETL – Extract, Transform and Load
- TFG – Trabajo de fin de grado
- PBIX – Power BI extensión de fichero
- PEC – Prueba de Evaluación Continua
- OLAP – OnLine Analytical Processing
- OLTP – OnLine Transaction Processing
- BD/BBDD – Base o bases de datos
- SLA – Service Level Agreement

8. Bibliografía

[1] Kimball DW/BI Lifecycle Methodology, sin fecha. *Kimball Group*. en línea. [Accedido 01 diciembre 2022]. Recuperado a partir de:

<https://www.kimballgroup.com/data-warehouse-business-intelligence-resources/kimball-techniques/dw-bi-lifecycle-method/>

[2] MIHART, sin fecha. The Power BI service - basic concepts for beginners - Power BI. en línea. [Accedido 5 diciembre 2022]. Recuperado a partir de:

<https://learn.microsoft.com/en-us/power-bi/consumer/end-user-basic-concepts>

[3] Pentaho 9.4, 2022. *Hitachi Vantara Lumada and Pentaho Documentation*. en línea. [Accedido 5 diciembre 2022]. Recuperado a partir de:

<https://help.hitachivantara.com/Documentation/Pentaho/9.4>

[4] PostgreSQL 15.1 Documentation, 2022. *PostgreSQL Documentation*. en línea. [Accedido 6 diciembre 2022]. Recuperado a partir de:

<https://www.postgresql.org/docs/15/index.html>

9. Anexos

9.1 Introducción al ecosistema de las herramientas

La finalidad de este apartado es brindar una rápida introducción a manera de presentación de las herramientas que se usaran en la implementación de este proyecto, no es un manual de uso de estas, por lo tanto, no se profundizara en detalles, estará enfocado en ser un *overview* (visión general).

El contenido de los siguientes subtítulos ha sido extraído de la documentación oficial, de la cual se ha hecho un resumen y extrado la información que se considera más importante o que se debe tener en cuenta para el desarrollo de este proyecto

9.1.1 Introducción MS Power BI

Se hace referencia en la bibliografía a la URL oficial de Power BI [2] - Versión: 2.111.590.0.

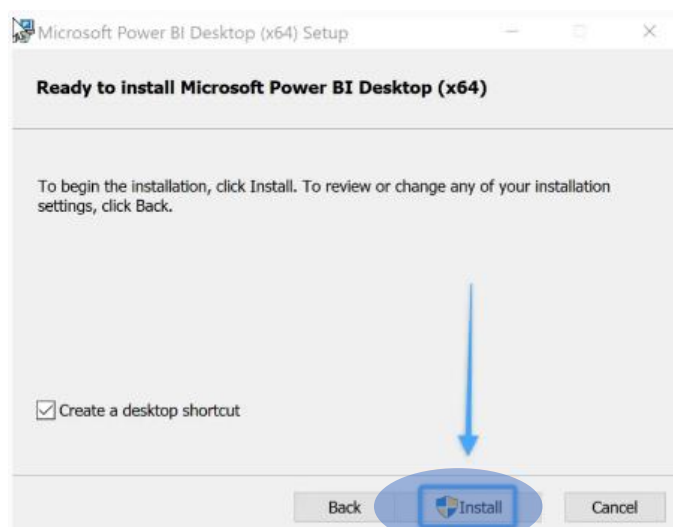
9.1.1.1 Descarga e instalación de la herramienta

Para la descarga iremos al sitio web de Power BI (<https://powerbi.microsoft.com/es-es/downloads/>) y se dará clic en el botón "Descargar".



Esto descargará la aplicación Power BI Desktop en el ordenador.

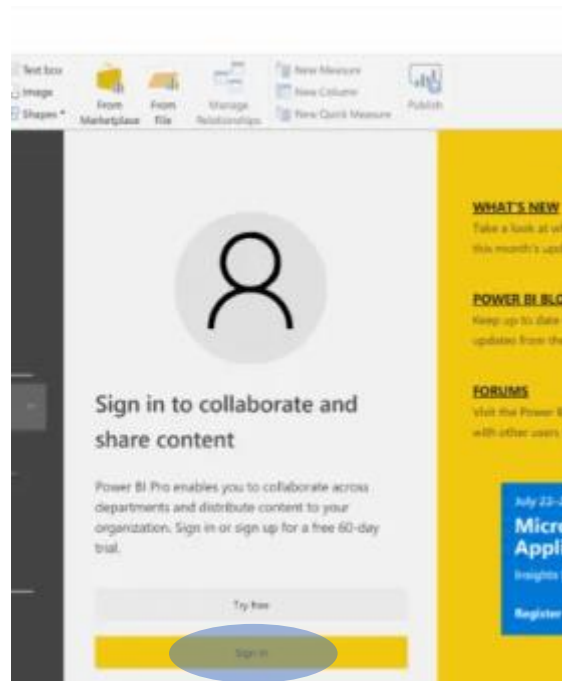
Una vez finalizada la descarga, daremos doble clic en el archivo descargado para iniciar el proceso de instalación.



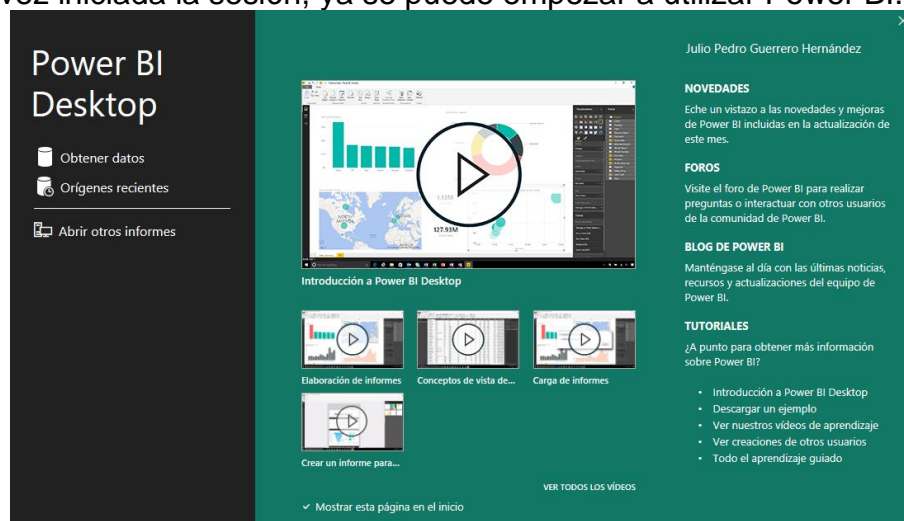
Luego seguimos las instrucciones en pantalla para completar la instalación.

Una vez instalado Power BI, se iniciará la aplicación haciendo clic en el icono de Power BI Desktop en el escritorio o en el menú de inicio.

Cuando se inicie Power BI, iniciaremos sesión con la cuenta de estudiante de la UOC que está asociada a Microsoft. Aunque también he comprobado que, si no se tiene una cuenta Microsoft, puede crear una gratuitamente siguiendo las instrucciones que aparecen en pantalla.



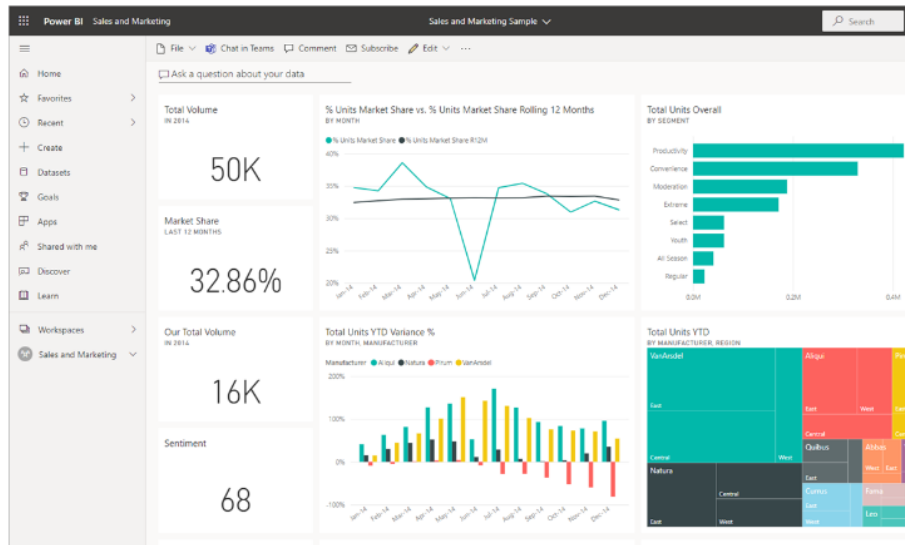
Una vez iniciada la sesión, ya se puede empezar a utilizar Power BI.



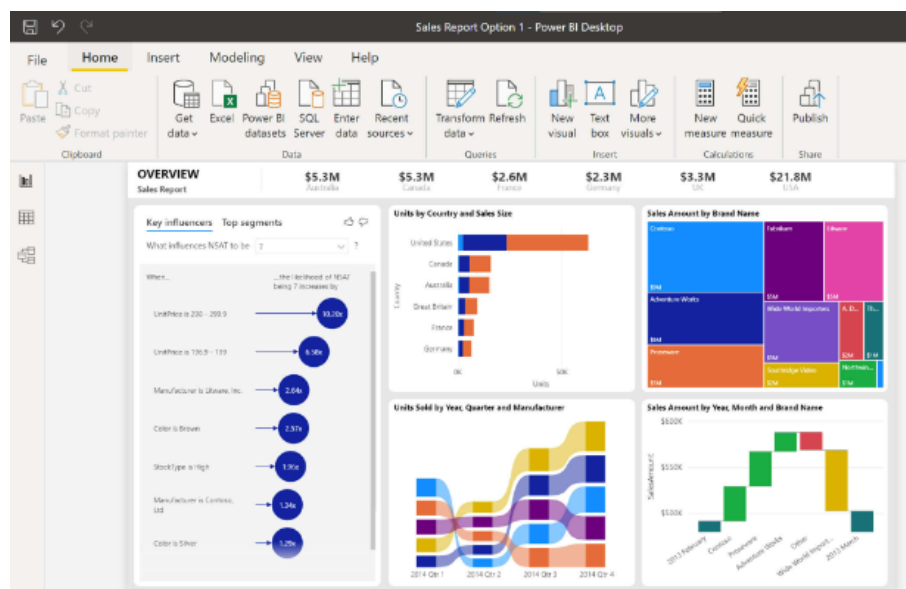
9.1.1.2 Términos a usar debido a esta herramienta

Algunos términos comunes que se utilizaran en Power BI en relación con el proyecto que se desarrolla:

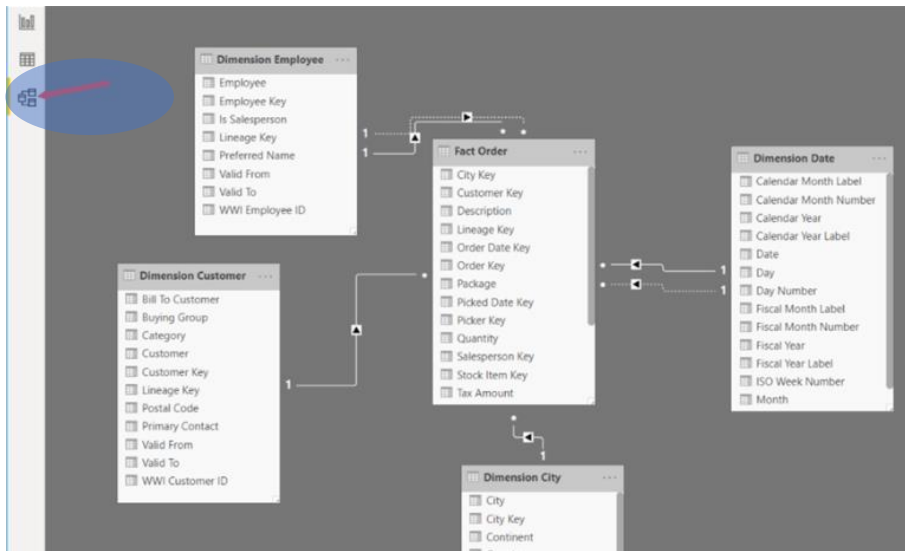
- Cuadro de mando: Una representación visual de datos y métricas que proporciona una visión general de alto nivel de los indicadores clave para una empresa u organización.



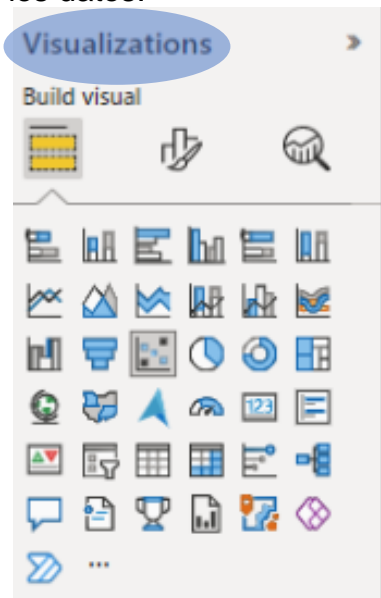
- Informe: Un análisis detallado de los datos que se presenta en un formato visual, como un gráfico o una tabla.



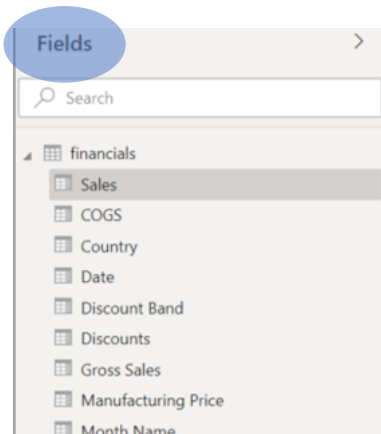
- Conjunto de datos: Una colección de datos que se utiliza para crear informes y cuadros de mando en Power BI.
- Modelado de datos: El modelado de datos es el proceso de organizar y estructurar los datos de forma que sea más fácil analizarlos y comprenderlos. Suele implicar la creación de relaciones entre distintas entidades de datos y la definición de medidas y columnas calculadas.



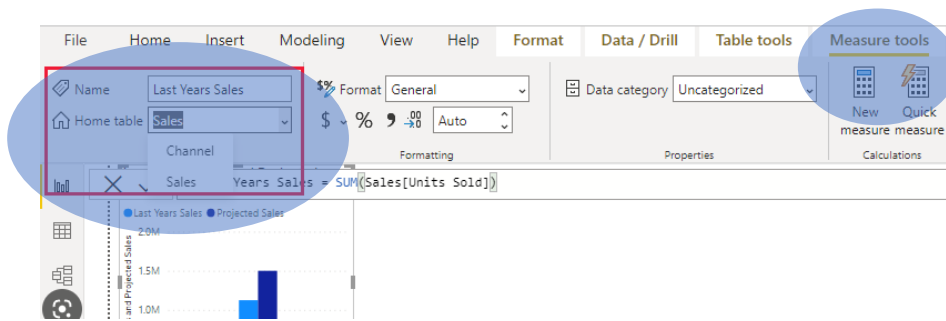
- **Visualización:** Una representación gráfica de los datos, como un gráfico de barras o un gráfico circular, que se utiliza para ayudar a los usuarios a comprender y analizar los datos.



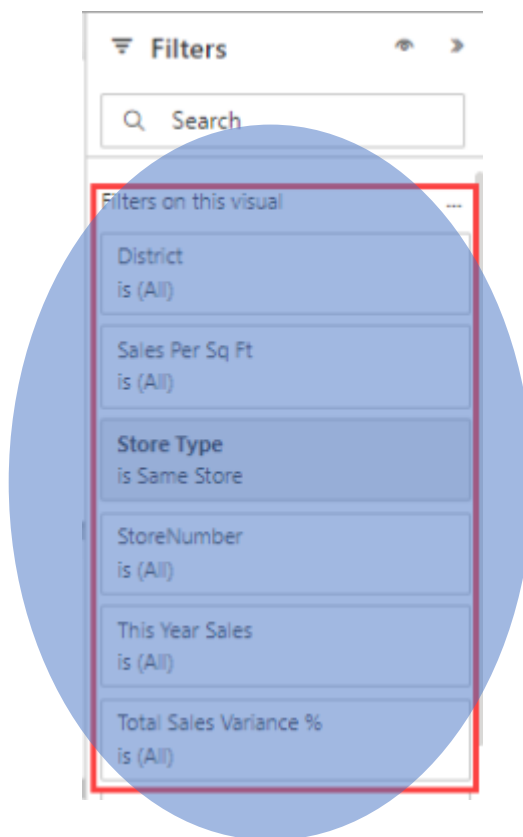
- **Campo:** Columna de un conjunto de datos que contiene un tipo específico de datos, como una fecha, un número o una cadena de texto.



- Medida: Cálculo que se realiza sobre los datos de un conjunto de datos, como una suma o una media.



- Filtro: Función que permite a los usuarios limitar los datos que se muestran en un informe o cuadro de mando, seleccionando criterios específicos.



- Temas: Un tema es un conjunto de opciones de formato que pueden aplicarse a un cuadro de mando o informe para darle un aspecto coherente. Puede incluir colores, fuentes y otros elementos de diseño que ayudan a crear una presentación cohesionada y profesional de los datos.

Estos son sólo algunos de los términos que utilizaremos en Power BI. Hay muchos otros términos y conceptos que es importante comprender para utilizar eficazmente esta herramienta.

9.1.2 Introducción Pentaho Data Integration

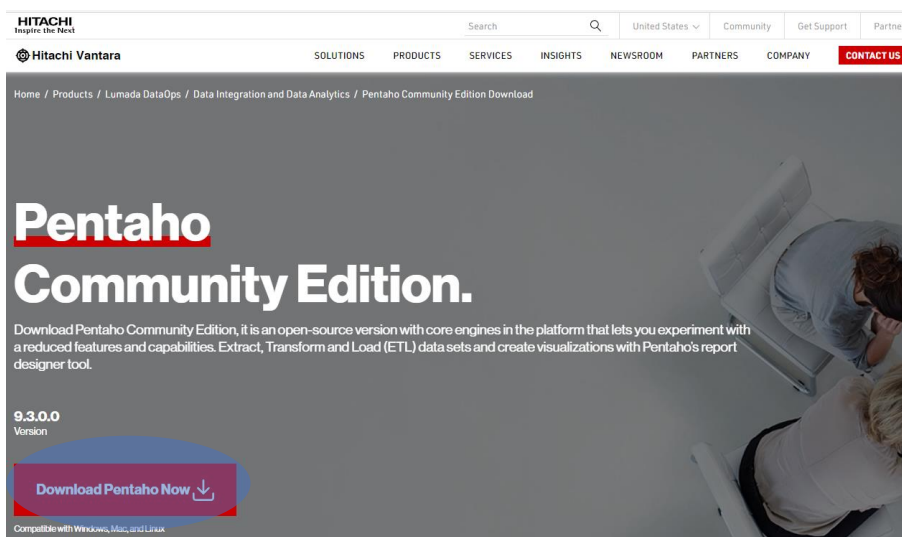
Se hace referencia en la bibliografía a la URL oficial de Pentaho Data Integration [3] - Versión 9.3.

9.1.2.1 Descarga e instalación de la herramienta

Para descargar la herramienta se usará el siguiente enlace:

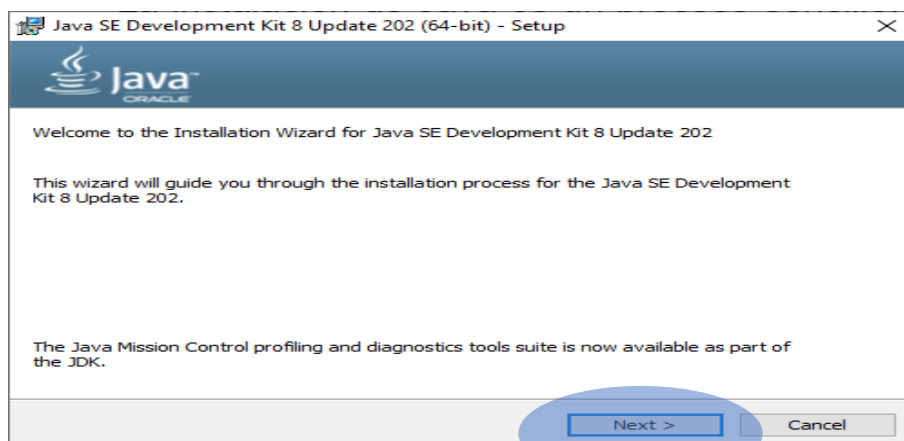
- <https://www.hitachivantara.com/en-us/products/lumada-dataops/data-integration-analytics/pentaho-community-edition.html>

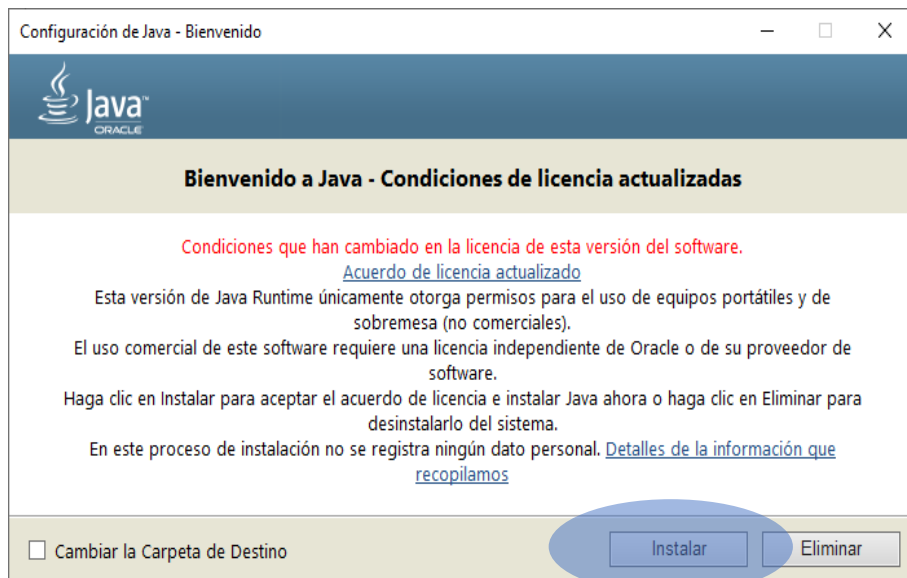
En este caso se ha elegido la versión 9.3.0.0, aclarar que también necesitaremos descargar Java ya que esta herramienta la usa para poder funcionar.



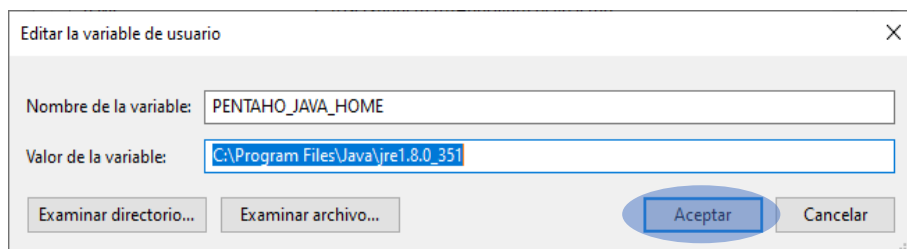
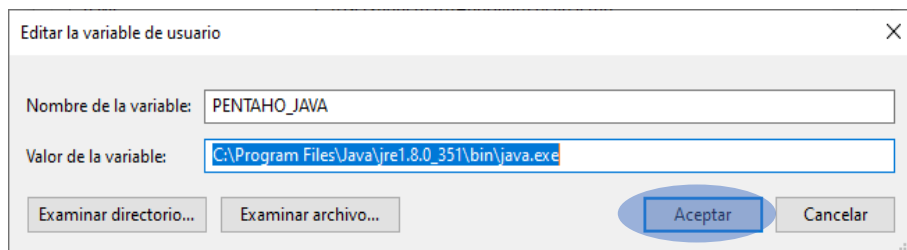
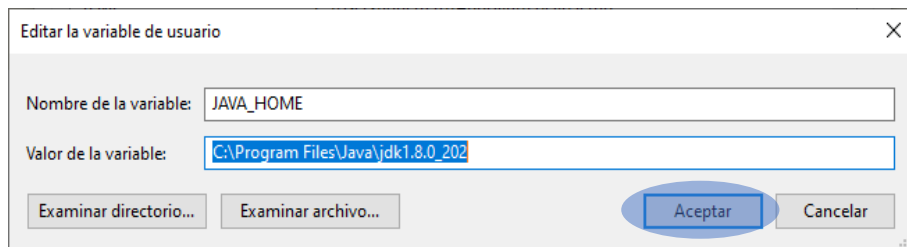
Una vez descargado a local procederemos a elegir un directorio y desempaquetarlo.

La instalación de Java es un proceso sencillo. En este caso necesitaremos instalar el JDK y JRE, en ambos casos basta con hacer doble clic en el archivo .exe del instalador de Windows, aceptar las condiciones y hacer clic varias veces en Siguiente.





Posterior a la instalación de java procederemos a configurar las variables necesarias, para esto entraremos en la Variables de Entorno de Windows y en las variables de usuario agregaremos las siguientes:



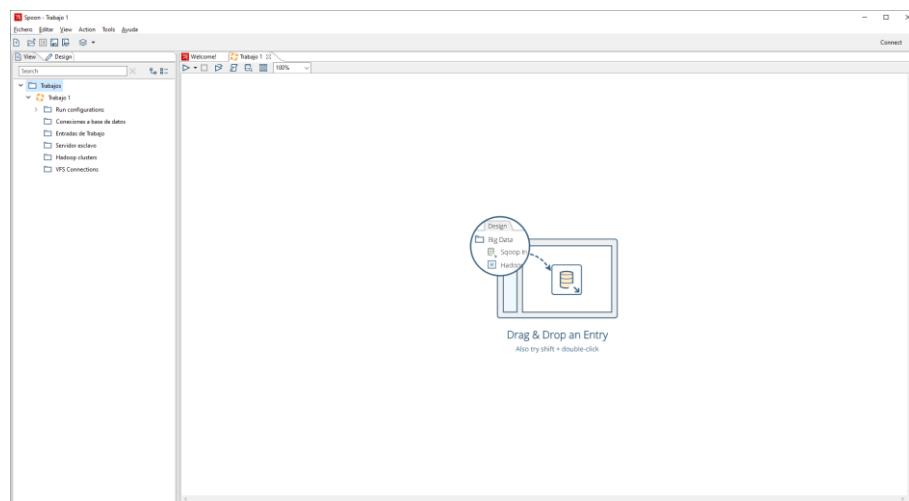
Finalmente abriremos la interfaz de usuario de Spoon, usando el archivo Spoon.bat, gracias a este se nos permitirá crear de forma sencilla transformaciones complejas de datos mediante pasos y saltos de arrastrar y soltar.

9.1.2.2 Términos a usar debido a esta herramienta

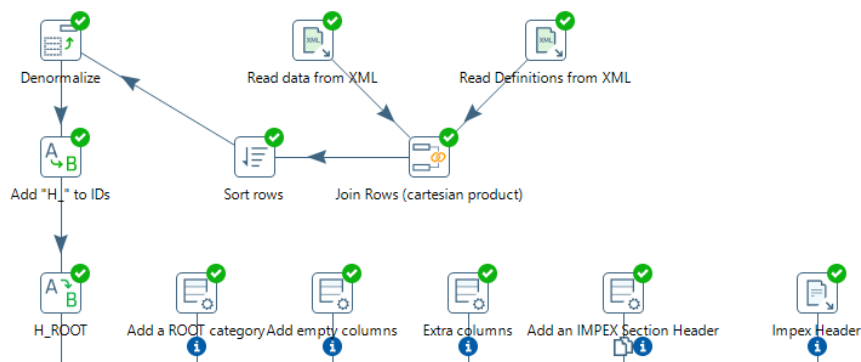
Esta herramienta tendrá como tarea el extraer datos de diversas fuentes, transformarlos en un formato adecuado para el análisis y cargarlos en un sistema de destino, como un almacén de datos.

En Pentaho Data Integration se utilizan varios términos que es importante comprender para utilizar la herramienta con eficacia. Algunos de estos términos son:

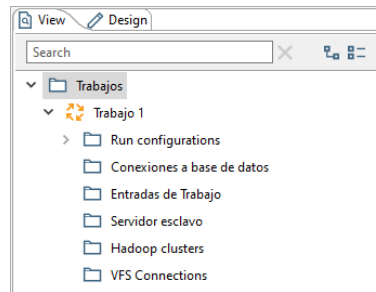
- Spoon: Es la interfaz gráfica de usuario (GUI) para diseñar y ejecutar transformaciones y trabajos en Kettle. Permite a los usuarios arrastrar y soltar pasos para crear transformaciones y enlazar transformaciones para crear trabajos.



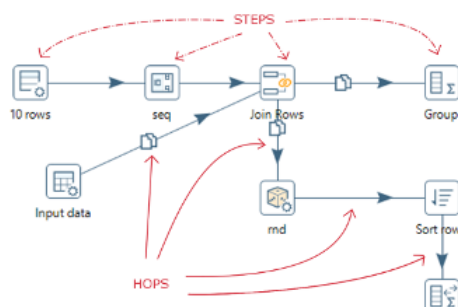
- Transformaciones: Una transformación es un conjunto de instrucciones que definen cómo deben extraerse, transformarse y cargarse los datos en un sistema de destino. Puede incluir diversas operaciones, como filtrar, ordenar y unir datos.



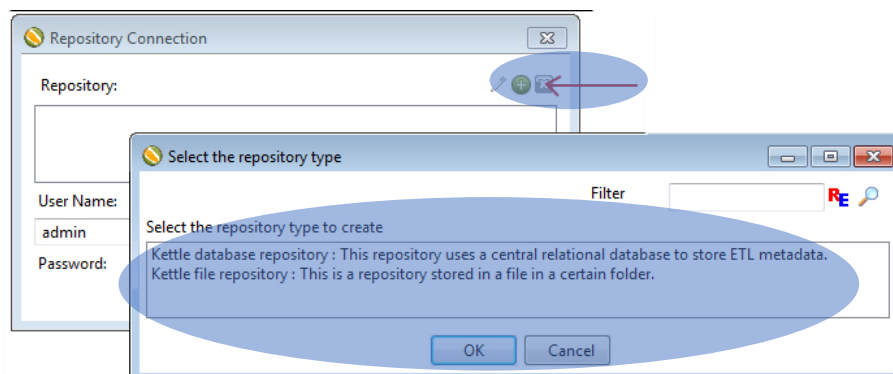
- Trabajos: Un trabajo es una serie de transformaciones que se ejecutan en un orden específico para realizar una tarea concreta. Por ejemplo, una tarea puede extraer datos de varias fuentes, transformarlos y cargarlos en un almacén de datos.



- Pasos: Un paso es una operación específica que se realiza dentro de una transformación. Existe una gran variedad de pasos disponibles en Kettle, incluidos los pasos para leer y escribir datos, transformar datos y ejecutar consultas SQL.
- Saltos: El salto será la dirección que tomará el paso que se vaya a ejecutar.



- Repositorios: Un repositorio es una ubicación central donde se pueden almacenar y gestionar transformaciones y trabajos. Permite a los usuarios compartir y colaborar en proyectos de integración de datos, así como versionar y realizar un seguimiento de los cambios en las transformaciones y los trabajos.



9.1.3 Introducción PostgreSQL

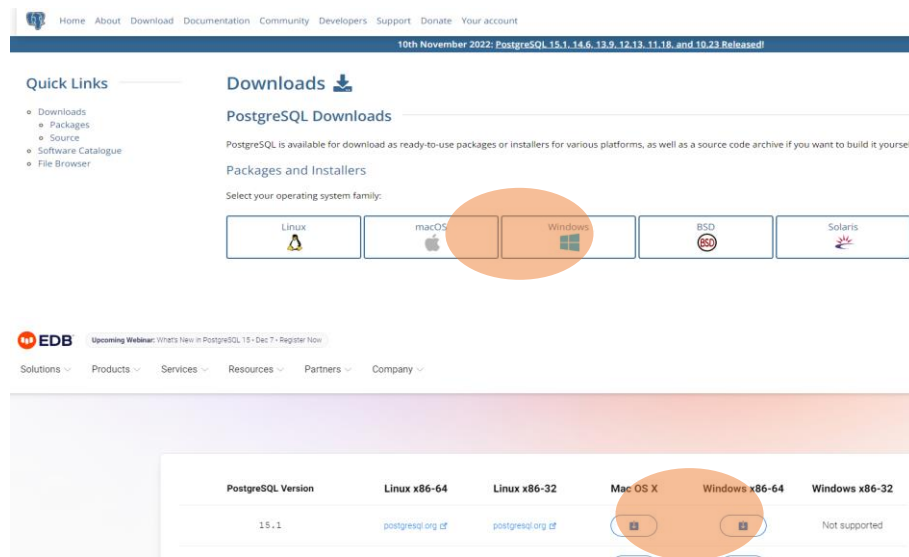
Se hace referencia en la bibliografía a la URL oficial de PostgreSQL [\[4\]](#) – Versión 15.1

9.1.3.1 Descarga e instalación de la herramienta

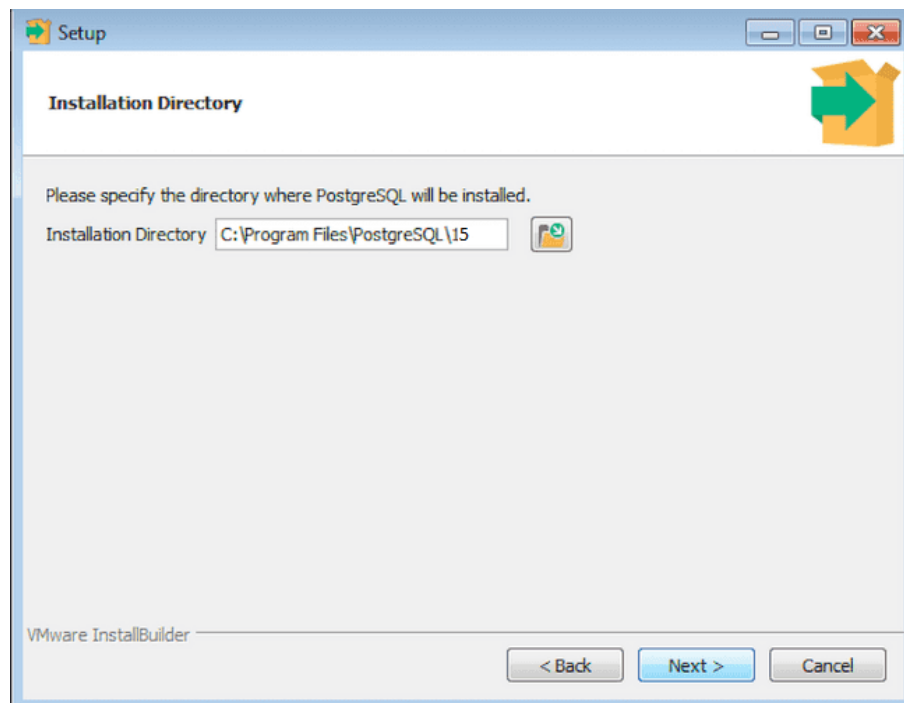
Para descargar la herramienta se usará el siguiente enlace:

- <https://www.postgresql.org/download/>

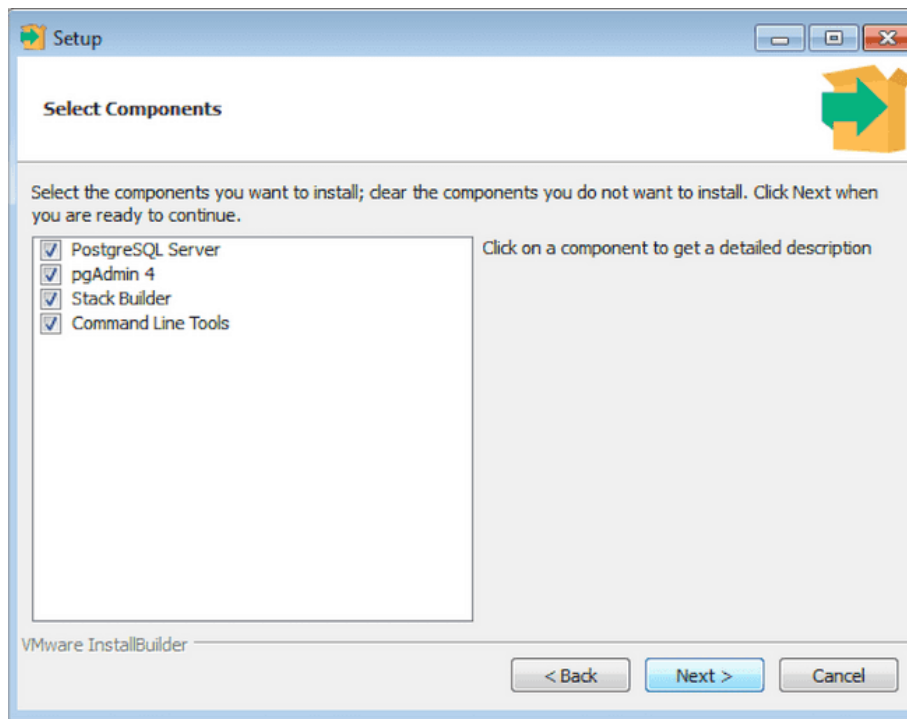
Se elige la versión 15.1 de la plataforma Windows, ya que es el entorno donde se desarrollará el proyecto.



A continuación, iniciaremos la instalación:



En la pantalla siguiente elegiremos los componentes, en este caso, se han elegido todos los que viene con la herramienta por defecto: El servidor de PostgreSQL, la interfaz de usuario, la utilidad de Stack Builder y la línea de comandos para usar la Shell para gestionar la base de datos.



Se dará siguiente hasta llegar al resumen preinstalación:



Al seleccionar el “**Stack Builder**”, proporcionamos una interfaz que simplifica el proceso de descarga e instalación de módulos que complementan su instalación PostgreSQL.

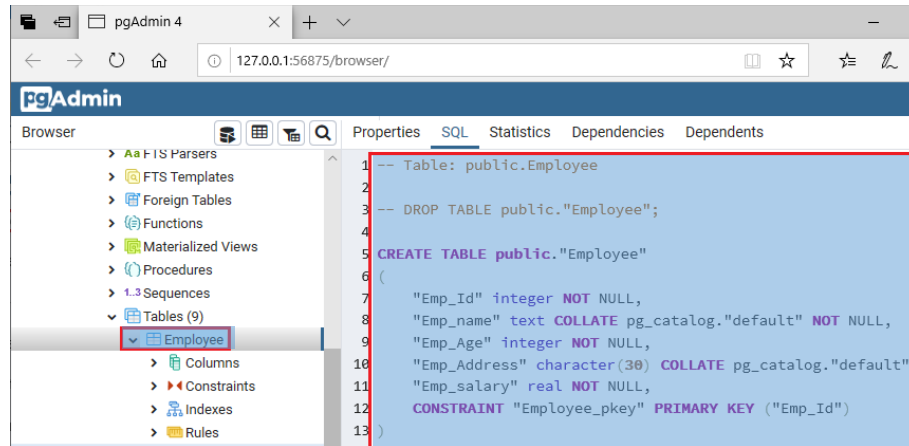
9.1.3.2 Términos a usar debido a esta herramienta

Siendo un sistema de gestión de bases de datos relacionales (RDBMS) de código abierto que utiliza el lenguaje de consulta estructurado (SQL) para

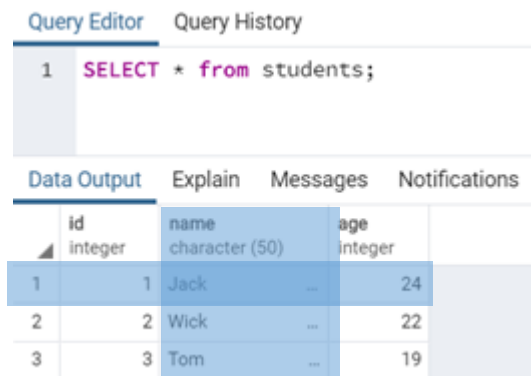
gestionar y manipular los datos almacenados en sus tablas, encontraremos los siguientes términos:

Base de datos: Una colección de datos organizados en tablas, vistas y otros objetos.

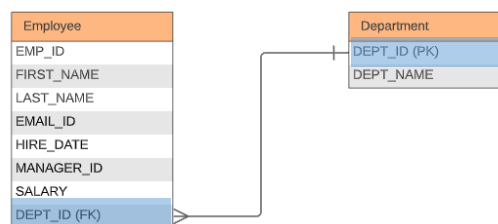
- Tabla: Una colección de datos organizados en filas y columnas, similar a una hoja de cálculo.



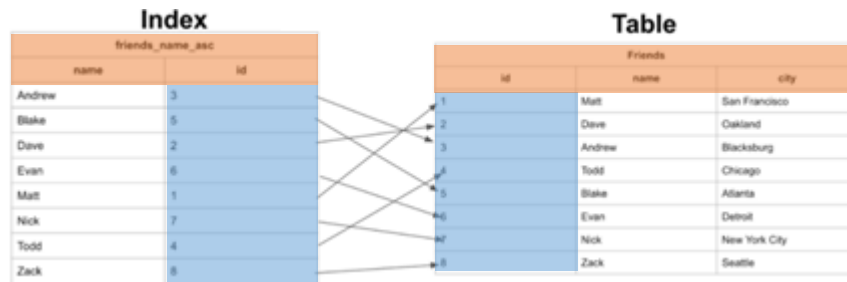
- Fila: Registro único de una tabla, formado por uno o varios valores almacenados en columnas
- Columna: Disposición vertical de los datos en una tabla, utilizada para almacenar una información específica de cada fila de la tabla.



- Clave primaria: Columna o conjunto de columnas que identifican de forma exclusiva cada fila de una tabla.
- Clave foránea: Columna o conjunto de columnas que hacen referencia a la clave primaria de otra tabla, creando una relación entre las dos tablas.



- Índice: Estructura de datos que acelera el rendimiento de las consultas permitiendo a la base de datos encontrar y recuperar rápidamente los datos deseados.



- Consulta: Una solicitud para recuperar datos de la base de datos, escrita en SQL.

Query Editor Query History

```

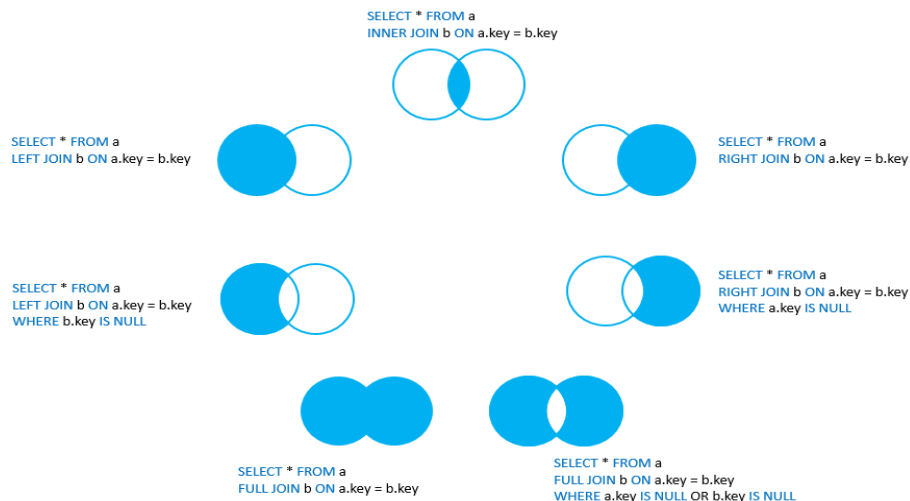
1 SELECT c1.car_name, c1.car_model,
2     (SELECT MIN(car_id)
3     FROM car c2
4     WHERE c1.car_id = c2.car_id) Subquery1
5 FROM car c1;

```

Data Output Messages Explain Notifications

	car_name	car_model	subquery1
1	Acura	NSX	1001
2	Porche	911 Carrera	1002
3	Nissan	Armada	1003

- Join: Operación utilizada para combinar datos de dos o más tablas en un único conjunto de resultados.



9.2 Transformaciones Posteriores al ETL

De manera que se tuviera una data totalmente utilizable fue necesario que posterior a la carga al almacén de datos y una vez cargada la información a Power BI se ejecutaran ciertas transformaciones, algunas de ellas fueron detalladas en el **punto 5.2** de este documento, pero a continuación se presenta el listado de operaciones ejecutadas sobre la data:

```

let
    Origen = PostgreSQL.Database("localhost", "dwh_EXP"),
    public_fact_crs1 = Origen[{Schema="public",Item="fact_crs1"}][Data],
    #"Columnas quitadas" = Table.RemoveColumns(public_fact_crs1,{"ope"}),
    #"Valor reemplazado" = Table.ReplaceValue(#"Columnas quitadas","IVR","TARIFICADOR",Replacer.ReplaceText,{"platform"}),
    #"Valor reemplazado1" = Table.ReplaceValue(#"Valor reemplazado","OCMP","GESTOR",Replacer.ReplaceText,{"platform"}),
    #"Valor reemplazado2" = Table.ReplaceValue(#"Valor reemplazado1","OCS","XXX",Replacer.ReplaceText,{"platform"}),
    #"Valor reemplazado3" = Table.ReplaceValue(#"Valor reemplazado2","OC","GESTOR",Replacer.ReplaceText,{"platform"}),
    #"Valor reemplazado4" = Table.ReplaceValue(#"Valor reemplazado3","XXX","OCS",Replacer.ReplaceText,{"platform"}),
    #"Columnas quitadas1" = Table.RemoveColumns(#"Valor reemplazado4",{"description","ana_tec_n2","causa_raiz_n3"}),
    #"Tipo cambiado" = Table.TransformColumnTypes(#"Columnas quitadas1",{"lead_time_days", type number}, {"pf_days", type
number}, {"last_update", type number}, {"sla_time_days", type number}, {"sla_time_proveedor_days", type number},
{"sla_time_proveedor_hours", type number}, {"ope_time_days", type number}, {"proveedor_time_days", type number},
{"sr_time_days", type number}, {"an3_sr_time_days", type number}, {"des_time_days", type number}, {"an3_des_time_days", type
number}, {"ps_time_days", type number}, {"cert_time_days", type number}, {"smaq_time_days", type number}, {"certop_time_days",
type number}, {"int_time_days", type number}, {"espec_time_days", type number}, {"opeins_time_days", type number},
{"hgd_time_days", type number}, {"reopen_times", Int64.Type}}),
    #"Valor reemplazado5" = Table.ReplaceValue(#"Tipo cambiado",null,0,Replacer.ReplaceValue,{"l3_patch","critical_tt",
"high_tt","medium_low_tt","closed_tt"}),
    #"Valor reemplazado6" = Table.ReplaceValue(#"Valor reemplazado5",null,0,Replacer.ReplaceValue,{"an_tls_des"}),
    #"Valor reemplazado7" = Table.ReplaceValue(#"Valor reemplazado6",null,"Not Categorized",Replacer.ReplaceValue,{"crc"}),
    #"Columnas quitadas2" = Table.RemoveColumns(#"Valor reemplazado7",{"ticket_name"}),
    #"Tipo cambiado1" = Table.TransformColumnTypes(#"Columnas quitadas2",{"alive_ticket_age_days", type number}, {"pf_days",
type number}}),
    #"Columnas quitadas3" = Table.RemoveColumns(#"Tipo cambiado1",{"external_id"}),
    #"Valor reemplazado8" = Table.ReplaceValue(#"Columnas quitadas3","GESTOR","GESTOR",Replacer.ReplaceText,{"platform"}),
    #"Valor reemplazado9" = Table.ReplaceValue(#"Valor reemplazado8","TARIFICADOR","TARIFICADOR",Replacer.ReplaceText,{"platform"}),
    #"Columna duplicada" = Table.DuplicateColumn(#"Valor reemplazado9","country",{"country - Copia"}),
    #"Columnas con nombre cambiado" = Table.RenameColumns(#"Columna duplicada",{"country - Copia",{"countryTextual"}},
{"countryTextual", type text}),
    #"Tipo cambiado2" = Table.TransformColumnTypes(#"Columnas con nombre cambiado",{"countryTextual", type text}),
    #"Valor reemplazado10" = Table.ReplaceValue(#"Tipo cambiado2",null,"ONC Otras Causas",Replacer.ReplaceValue,{"categorization"}),
    #"Valor reemplazado11" = Table.ReplaceValue(#"Valor reemplazado10",null,"NUL Ninguna Pérdida de Servicio",Replacer.ReplaceValue,{"service_outage"}),
    #"Columna duplicada1" = Table.DuplicateColumn(#"Valor reemplazado11","ticket_status",{"ticket_status - Copia"}),
    #"Columna duplicada2" = Table.DuplicateColumn(#"Columna duplicada1","ticket_status - Copia",{"ticket_status - Copia -
Copia"}),
    #"Columna duplicada3" = Table.DuplicateColumn(#"Columna duplicada2","ticket_status - Copia - Copia",{"ticket_status - Copia -
Copia - Copia"}),
    #"Columna duplicada4" = Table.DuplicateColumn(#"Columna duplicada3","ticket_status - Copia - Copia - Copia",{"ticket_status -
Copia - Copia - Copia - Copia"}),
    #"Columnas con nombre cambiado1" = Table.RenameColumns(#"Columna duplicada4",{"ticket_status - Copia",{"tt_status_closed"}},
{"ticket_status - Copia - Copia",{"tt_status_open"}}, {"ticket_status - Copia - Copia - Copia",{"tt_status_reject"}},
{"ticket_status - Copia - Copia - Copia - Copia",{"tt_status_solved"}}),
    #"Columna duplicada5" = Table.DuplicateColumn(#"Columnas con nombre cambiado1",{"tt_status_closed",{"tt_status_closed -
Copia"}),
    #"Columnas con nombre cambiado2" = Table.RenameColumns(#"Columna duplicada5",{"tt_status_closed - Copia",{"tt_status_pending"}},
{"tt_status_pending"}),
    #"Columna duplicada6" = Table.DuplicateColumn(#"Columnas con nombre cambiado2",{"tt_status_closed",{"tt_status_closed -
Copia"}),
    #"Columnas con nombre cambiado3" = Table.RenameColumns(#"Columna duplicada6",{"tt_status_closed - Copia",{"tt_status_generated"}},
{"tt_status_generated"}),
    #"Valor reemplazado12" = Table.ReplaceValue(#"Columnas con nombre cambiado3","Closed","1",Replacer.ReplaceText,{"tt_status_closed"}),
    #"Columnas quitadas4" = Table.RemoveColumns(#"Valor reemplazado12",{"tt_status_open","tt_status_reject",
"tt_status_solved","tt_status_pending","tt_status_generated"}),
    #"Columna condicional agregada" = Table.AddColumn(#"Columnas quitadas4",{"tt_status_closed1", each if [ticket_status] =
"Closed" then 1 else if [ticket_status] <> "Closed" then 0 else null}),
    #"Columnas quitadas5" = Table.RemoveColumns(#"Columna condicional agregada",{"tt_status_closed"}),
    #"Columnas con nombre cambiado4" = Table.RenameColumns(#"Columnas quitadas5",{"tt_status_closed1",{"tt_status_closed"}},
{"tt_status_closed"}),
    #"Tipo cambiado3" = Table.TransformColumnTypes(#"Columnas con nombre cambiado4",{"tt_status_closed", Int64.Type}),
    #"Columnas quitadas6" = Table.RemoveColumns(#"Tipo cambiado3",{"tt_status_closed"}),
    #"Columna condicional agregada1" = Table.AddColumn(#"Columnas quitadas6",{"tt_status_pending", each if [ticket_status] =
"Pending" then 1 else if [ticket_status] = "Analysis_TLS" then 1 else if [ticket_status] = "Pend. L1 Action" then 1 else if not
Text.StartsWith([ticket_status], "Pend") then 0 else null}),
    #"Tipo cambiado4" = Table.TransformColumnTypes(#"Columna condicional agregada1",{"tt_status_pending", Int64.Type}),
    #"Columna condicional agregada2" = Table.AddColumn(#"Tipo cambiado4",{"tt_status_solved", each if [ticket_status] = "Solved"
then 1 else if [ticket_status] <> "Solved" then 0 else null}),
    #"Columna condicional agregada3" = Table.AddColumn(#"Columna condicional agregada2",{"tt_status_rejected", each if
[ticket_status] = "Rejected" then 1 else if [ticket_status] <> "Rejected" then 0 else null}),
    #"Columna condicional agregada4" = Table.AddColumn(#"Columna condicional agregada3",{"tt_status_open", each if
[ticket_status] = "Open" then 1 else if [ticket_status] <> "Open" then 0 else null}),
    #"Columna condicional agregada5" = Table.AddColumn(#"Columna condicional agregada4",{"tt_status_generated", each if
[ticket_status] = "Generated" then 1 else if [ticket_status] <> "Generated" then 0 else null}),
    #"Tipo cambiado5" = Table.TransformColumnTypes(#"Columna condicional agregada5",{"tt_status_solved", Int64.Type},
{"tt_status_rejected", Int64.Type}, {"tt_status_open", Int64.Type}, {"tt_status_generated", Int64.Type}),
    #"Valor reemplazado13" = Table.ReplaceValue(#"Tipo cambiado5",{"Analysis_TLS","Pending",Replacer.ReplaceText,{"ticket_status"}),
    #"Valor reemplazado14" = Table.ReplaceValue(#"Valor reemplazado13",{"Pend. L1 Action","Pending",Replacer.ReplaceText,{"ticket_status"}},
    #"Columna condicional agregada6" = Table.AddColumn(#"Valor reemplazado14",{"tt_status_closed", each if [ticket_status] =
"Closed" then 1 else if [ticket_status] <> "Closed" then 0 else null}),
    #"Valor reemplazado15" = Table.ReplaceValue(#"Columna condicional agregada6",{"Operadora_D","DEU",Replacer.ReplaceText,{"country"}},
    #"Valor reemplazado16" = Table.ReplaceValue(#"Valor reemplazado15",{"Operadora_I","IRL",Replacer.ReplaceText,{"country"}},
    #"Valor reemplazado17" = Table.ReplaceValue(#"Valor reemplazado16",{"Operadora_UK","GBR",Replacer.ReplaceText,{"country"}},

```

```

#Valor reemplazado18 = Table.ReplaceValue("#Valor
reemplazado17","Operadora_D","DEU",Replacer.ReplaceText,{"countryTextual"}),
#Valor reemplazado19 = Table.ReplaceValue("#Valor
reemplazado18","Operadora_I","IRL",Replacer.ReplaceText,{"countryTextual"}),
#Valor reemplazado20 = Table.ReplaceValue("#Valor
reemplazado19","Operadora_UK","GBR",Replacer.ReplaceText,{"countryTextual"}),
#Columna condicional agregada7 = Table.AddColumn("#Valor reemplazado20", "tt_sla_breach", each if [sla_violation] = "0"
then "NO" else if [sla_violation] = "1" then "SI" else null),
#Valor reemplazado21 = Table.ReplaceValue("#Columna condicional agregada7",null,"Level
2",Replacer.ReplaceValue,{"crc_level"}),
#Columna condicional agregada8 = Table.AddColumn("#Valor reemplazado21", "tt_service_outage", each if
Text.StartsWith([service_outage], "NUL") then "Sin Pérdida" else if Text.StartsWith([service_outage], "SBO") then "Bonos" else
if Text.StartsWith([service_outage], "SCM") then "Abonados" else if Text.StartsWith([service_outage], "SCO") then "Contenidos"
else if Text.StartsWith([service_outage], "SDS") then "Datos" else if Text.StartsWith([service_outage], "SOT") then "Otros"
else if Text.StartsWith([service_outage], "SRC") then "Recargas" else if Text.StartsWith([service_outage], "SSM") then "SMS"
else if Text.StartsWith([service_outage], "SVO") then "Voz" else null),
#Columna condicional agregada9 = Table.AddColumn("#Columna condicional agregada8", "tt_grp_categorization", each if
Text.StartsWith([categorization], "CNF") then "No error" else if Text.StartsWith([categorization], "CDU") then "No error" else
if Text.StartsWith([categorization], "COS") then "Error Operadora" else if Text.StartsWith([categorization], "CCF") then "Error
Operadora" else if Text.StartsWith([categorization], "CPC") then "Futura mejora" else if Text.StartsWith([categorization],
"CPH") then "Error Operadora" else if Text.StartsWith([categorization], "CSM") then "Error Operadora" else if
Text.StartsWith([categorization], "CSN") then "Error Operadora" else if Text.Contains([categorization], "3PP") then "Error 3PP"
else if Text.Contains([categorization], "Proveedor") then "Error Proveedor" else if Text.StartsWith([categorization], "EHL")
then "Error Definición OP+Pro" else if Text.StartsWith([categorization], "ESF") then "Error Proveedor" else if
Text.StartsWith([categorization], "ONC") then "Error sin determinar" else if Text.StartsWith([categorization], "ONR") then
"Error sin determinar" else if Text.StartsWith([categorization], "SOPMAQ") then "NO COMPUTABLE" else null),
#Valor reemplazado22 = Table.ReplaceValue("#Columna condicional
agregada9",null,"Otros",Replacer.ReplaceValue,{"crc_status"}),
#Columna condicional agregada10 = Table.AddColumn("#Valor reemplazado22", "crc_pending", each if [crc_status] = "Otros"
then 1 else if [crc_pending_handshake] = 1 then 1 else 0),
#Columnas reordenadas = Table.ReorderColumns("#Columna condicional agregada10",{"ticket_id", "platform", "country",
"severity", "ticket_status", "ticket_group", "generation_date", "generation_year", "generation_month", "open_date",
"lead_time_days", "closure_date", "closure_year", "closure_month", "pf_days", "alive_ticket_age_days", "last_update",
"remedy_date", "sla_time_days", "sla_time_ope_days", "sla_time_proveedor_days", "sla_time_proveedor_hours", "categorization",
"service_outage", "sla_violation", "sla_time_ope_estimated_days", "sla_time_proveedor_estimated_days",
"sla_violation_estimated", "ope_time_days", "proveedor_time_days", "sr_time_days", "an3_sr_time_days", "mc_time_days",
"des_time_days", "an3_des_time_days", "ps_time_days", "cert_time_days", "smaq_time_days", "certop_time_days", "int_time_days",
"espec_time_days", "opeins_time_days", "hgd_time_days", "reopen_times", "l3_patch", "critical_tt", "high_tt", "medium_low_tt",
"closed_tt", "an_tls_des", "ticket_type", "hot", "crc_level", "crc_escalated_l1", "crc_escalated_l2", "crc_status",
"crc_not_agreed", "crc_pending_handshake", "crc_pending", "crc", "crc_description", "countryTextual", "tt_status_pending",
"tt_status_solved", "tt_status_rejected", "tt_status_open", "tt_status_generated", "tt_status_closed", "tt_sla_breach",
"tt_service_outage", "tt_grp_categorization"}),
#Tipo cambiado6 = Table.TransformColumnTypes("#Columnas reordenadas",{"crc_pending", Int64.Type},
{"crc_pending_handshake", Int64.Type}, {"crc_not_agreed", Int64.Type}, {"crc_escalated_l2", Int64.Type}, {"crc_escalated_l1",
Int64.Type}),
#Columnas reordenadas1 = Table.ReorderColumns("#Tipo cambiado6",{"ticket_id", "platform", "country", "severity",
"ticket_status", "ticket_group", "generation_date", "generation_year", "generation_month", "open_date", "lead_time_days",
"closure_date", "closure_year", "closure_month", "pf_days", "alive_ticket_age_days", "last_update", "remedy_date",
"sla_time_days", "sla_time_ope_days", "sla_time_proveedor_days", "sla_time_proveedor_hours", "categorization",
"service_outage", "sla_violation", "sla_time_ope_estimated_days", "sla_time_proveedor_estimated_days",
"sla_violation_estimated", "ope_time_days", "proveedor_time_days", "sr_time_days", "an3_sr_time_days", "mc_time_days",
"des_time_days", "an3_des_time_days", "ps_time_days", "cert_time_days", "smaq_time_days", "certop_time_days", "int_time_days",
"espec_time_days", "opeins_time_days", "hgd_time_days", "reopen_times", "l3_patch", "critical_tt", "high_tt", "medium_low_tt",
"closed_tt", "an_tls_des", "ticket_type", "hot", "crc_level", "crc_escalated_l1", "crc_escalated_l2", "crc_status",
"crc_not_agreed", "crc_pending", "crc_pending_handshake", "crc", "crc_description", "countryTextual", "tt_status_pending",
"tt_status_solved", "tt_status_rejected", "tt_status_open", "tt_status_generated", "tt_status_closed", "tt_sla_breach",
"tt_service_outage", "tt_grp_categorization"}),
#Valor reemplazado23 = Table.ReplaceValue("#Columnas reordenadas1", "Otros", "Pending
agreement",Replacer.ReplaceText,{"crc_status"}),
#Columna condicional agregada11 = Table.AddColumn("#Valor reemplazado23", "crc_agreed", each if [crc_status] = "Agreed"
then 1 else 0),
#Columnas reordenadas2 = Table.ReorderColumns("#Columna condicional agregada11",{"ticket_id", "platform", "country",
"severity", "ticket_status", "ticket_group", "generation_date", "generation_year", "generation_month", "open_date",
"lead_time_days", "closure_date", "closure_year", "closure_month", "pf_days", "alive_ticket_age_days", "last_update",
"remedy_date", "sla_time_days", "sla_time_ope_days", "sla_time_proveedor_days", "sla_time_proveedor_hours", "categorization",
"service_outage", "sla_violation", "sla_time_ope_estimated_days", "sla_time_proveedor_estimated_days",
"sla_violation_estimated", "ope_time_days", "proveedor_time_days", "sr_time_days", "an3_sr_time_days", "mc_time_days",
"des_time_days", "an3_des_time_days", "ps_time_days", "cert_time_days", "smaq_time_days", "certop_time_days", "int_time_days",
"espec_time_days", "opeins_time_days", "hgd_time_days", "reopen_times", "l3_patch", "critical_tt", "high_tt", "medium_low_tt",
"closed_tt", "an_tls_des", "ticket_type", "hot", "crc_level", "crc_escalated_l1", "crc_escalated_l2", "crc_status",
"crc_agreed", "crc_not_agreed", "crc_pending", "crc_pending_handshake", "crc", "crc_description", "countryTextual",
"tt_status_pending", "tt_status_solved", "tt_status_rejected", "tt_status_open", "tt_status_generated", "tt_status_closed",
"tt_sla_breach", "tt_service_outage", "tt_grp_categorization"})
in
#Columnas reordenadas2

```