



UCAV

www.ucavila.es

UNIVERSIDAD CATÓLICA DE ÁVILA

Facultad de Ciencias y Artes

TRABAJO FIN DE GRADO

Correspondiente a la titulación de

Título del Trabajo Fin de Grado

Implementación de un Data Warehouse
con tecnologías de Microsoft SQL Server

ALUMNO/A: DIEGO TORIBIO RODRÍGUEZ.

DIRECTOR/A: SORAYA ANDALUZ DELGADO.

CONVOCATORIA

**CURSO
ACADÉMICO**

RESUMEN

El Business Intelligence es un tema de mucho interés para las empresas porque propone soluciones software para aprovechar mejor los datos que generan y almacenan. En este Trabajo de Fin de Grado se estudia un componente importante de estos sistemas: el Data Warehouse.

Un Data Warehouse es un tipo de almacenamiento de datos que tiene un papel fundamental dentro de un sistema de Business Intelligence. Su desarrollo conlleva conocer como se plantea el diseño de su estructura y el modelo de datos a almacenar.

Además del almacenamiento de los datos, este proyecto también contempla el uso de Datos Abiertos como fuente de información, el desarrollo de procesos ETL para trasladar los datos entre diferentes bases de datos y la creación de visualizaciones gráficas para representar los datos con sentido y en contexto. Todo el desarrollo está realizado con los servicios de SQL Server de Microsoft.

ÍNDICE GENERAL

RESUMEN.....	2
ÍNDICE GENERAL.....	3
ÍNDICE DE FIGURAS.....	5
ÍNDICE DE TABLAS.....	8
GLOSARIO DE TÉRMINOS.....	9
CAPÍTULO 1.....	11
1.1. INTRODUCCIÓN.....	11
1.2. JUSTIFICACIÓN.....	11
1.3. OBJETIVOS.....	14
CAPÍTULO 2.....	15
2.1. CONTEXTO DEL PROYECTO.....	15
2.1.1. BIG DATA Y BUSINESS INTELLIGENCE.....	16
2.2. DATA WAREHOUSE.....	18
2.3. PROCESAMIENTO OLTP Y OLAP.....	18
2.3.1. OLTP.....	19
2.3.2. OLAP.....	21
2.4. MODELADO DIMENSIONAL.....	24
2.4.1. PROCESO DE MODELADO DIMENSIONAL.....	26
2.4.2. CICLO DE VIDA DIMENSIONAL DEL NEGOCIO.....	27
2.5. PROCESOS ETL.....	29
2.6. ARQUITECTURA CON TECNOLOGÍAS DE MICROSOFT.....	30
2.7. DATOS ABIERTOS.....	34
2.8. DATOS ABIERTOS Y DATA WAREHOUSE.....	41
CAPÍTULO 3.....	43
3.1. DESCRIPCIÓN DEL PROYECTO.....	43
3.1.1. SELECCIÓN DE DATOS ABIERTOS.....	44
3.1.2. MODELADO DIMENSIONAL: TABLAS FACT Y TABLAS DIM.....	45

3.2. ALCANCE DEL PROYECTO.....	50
3.3. PLANIFICACIÓN.....	50
CAPÍTULO 4.....	52
4.1. ENTORNO DE TRABAJO.....	52
4.2. ANÁLISIS DE LOS CONJUNTOS DE DATOS SELECCIONADOS.....	52
4.3. IMPLEMENTACIÓN DE LOS DATAMARTS.....	61
4.3.1. CREACIÓN DE LAS TABLAS DIMENSIONALES.....	63
4.3.2. CREACIÓN DE LAS TABLAS DE HECHOS.....	65
4.4. IMPLEMENTACIÓN DE LA ESTRUCTURA DEL DATAWAREHOUSE.....	75
4.4.1. CREACIÓN DE LAS TABLAS DIMENSIONALES.....	76
4.4.2. CREACIÓN DE LAS TABLAS DE HECHOS.....	78
4.5. IMPLEMENTACIÓN DE LOS PROCESOS ETL.....	81
4.5.1. CREACIÓN DE UN PROYECTO SSIS.....	81
4.5.2. CREACIÓN DE UN PAQUETE SSIS CON FLUJO DE DATOS.....	84
4.6. VISUALIZACIÓN DE LOS DATOS.....	95
4.6.1. CARGA DE DATOS.....	96
4.6.2. MODELO DE DATOS.....	98
4.6.3. CONSTRUCCIÓN DEL DASHBOARD.....	100
CAPÍTULO 5.....	105
5.1. POSIBLES LÍNEAS FUTURAS DE TRABAJO.....	105
5.2. CONCLUSIONES.....	105
BIBLIOGRAFÍA Y REFERENCIAS.....	108

ÍNDICE DE FIGURAS

Figura 1: tabla normalizada.....	20
Figura 2: tabla denormalizada.....	21
Figura 3: esquema en estrella (star design).....	22
Figura 4: esquema copo de nieve (snowflake design).....	23
Figura 5: cubo OLAP.....	24
Figura 6: Diferencias entre las metodologías Inmon y Kimball.....	25
Figura 7: Fases de la metodología Kimball.....	27
Figura 8: Esquema de un sistema Business Intelligence.....	29
Figura 9: proceso ETL.....	29
Figura 10: esquema de Data Warehouse con Microsoft Azure.....	31
Figura 11: sistema Business Intelligence con Microsoft SQL Server.....	32
Figura 12: sistema detallado Business Intelligence con Microsoft Azure.....	33
Figura 13: Indicadores por temas del ISTAC.....	34
Figura 14: ejemplo de uso de Mapnificent.....	35
Figura 15: portal de LicitaLio.....	36
Figura 16: dashboard del sistema de Observación Meteorológica de Canarias.....	37
Figura 17: Portal de Datos Abiertos de Canarias.....	42
Figura 18: Esquema del proyecto.....	43
Figura 19: Conjunto de datos sobre Turismo del portal de Datos Abiertos de Canarias....	44
Figura 20: Esquema de estrella del Data Warehouse.....	46
Figura 21: Página de un conjunto de datos del portal de Datos Abiertos de Canarias.....	53
Figura 22: Panel de consultas del ISTAC de un conjunto de datos.....	54
Figura 23: Resultado de la consulta del ISTAC de un conjunto de datos.....	55
Figura 24: Información adicional de un conjunto de datos del portal de Datos Abiertos de Canarias.....	56

Figura 25: Archivos JSON.....	56
Figura 26: Estructura del archivo Datos7.json.....	59
Figura 27: Estructura del archivo Datos7.json.....	60
Figura 28: Ejemplo de variables del archivo Datos7.json.....	61
Figura 29: Ejemplo de datos del archivo Datos7.json.....	62
Figura 30: Datamarts creados en SQL Server Management.....	74
Figura 31: Esquema de estrella del Data Warehouse.....	75
Figura 32: Data Warehouse creado en SQL Server Management.....	80
Figura 33: Creación de un proyecto de Integration Services en Visual Studio.....	82
Figura 34: Creación de un proyecto de Integration Services en Visual Studio.....	82
Figura 35: Entorno de SQL Integration Services.....	83
Figura 36: Flujo de control de SQL Integration Services.....	84
Figura 37: Flujo de datos de SQL Integration Services.....	84
Figura 38: Editor de origen de OLE DB.....	85
Figura 39: Creación de conexiones de OLE DB.....	86
Figura 40: Lista de conexiones de OLE DB.....	87
Figura 41: Columnas en el editor de origen de OLE DB.....	88
Figura 42: Flujo de datos entre origen y destino OLE DB.....	89
Figura 43: Columnas en el editor de origen de OLE DB.....	90
Figura 44: Asignación de columnas en el editor de destino de OLE DB.....	91
Figura 45: Ejecucción de un paquete SSIS.....	92
Figura 46: Lista de paquetes SSIS.....	93
Figura 47: Datos de la tabla FactTurista.....	94
Figura 48: Diagrama de como funciona Power BI.....	95
Figura 49: Selección de tablas a cargar en Power BI Desktop.....	96
Figura 50: Tablas cargadas en Power BI Desktop.....	97
Figura 51: Tipos de datos de las columnas seleccionadas de las tablas en Power BI Desktop.	97

Figura 52: Modelo de datos en Power BI Desktop.....	98
Figura 53: Modelo de datos.....	99
Figura 54: Escritorio de Power BI Desktop.....	100
Figura 55: Gráfico de anillos.....	101
Figura 56: Gráfico circular.....	101
Figura 57: Gráficos circulares.....	102
Figura 58: Mapa interactivo de las Islas Canarias.....	102
Figura 59: Gráfico de columnas agrupadas.....	103
Figura 60: Selector de periodos de tiempo.....	103
Figura 61: Dashboard con los datos del Data Warehouse.....	104

ÍNDICE DE TABLAS

Tabla FactTurista.....	47
Tablas Dimensionales.....	49
Tabla de relaciones en archivos json y datasets	57

GLOSARIO DE TÉRMINOS

API: es una interfaz de programación de aplicaciones con un conjunto de subrutinas, funciones y procedimientos que ofrece una capa de abstracción para ser utilizado por otro software.

Array: es una estructura de datos en programación para representar una colección de objetos. Puede ser desde una fila de elementos, hasta una tabla con filas y columnas, o una colección con más de dos dimensiones.

Business Intelligence: conjunto de estrategias, aplicaciones, datos, productos, tecnologías y arquitectura técnicas, los cuales están enfocados a la administración y creación de conocimiento sobre el medio, a través del análisis de los datos existentes en una organización o empresa.

Big Data: conjuntos de datos tan grandes y complejos que precisan de aplicaciones informáticas no tradicionales de procesamiento de datos para tratarlos adecuadamente.

Data Warehouse: Sistema de almacenamiento de datos enfocado al informe y análisis de datos.

Datasets: es un archivo digital que contiene una colección de datos.

Datos Abiertos (Open Data): son iniciativas de entidades que publican sus datos para que puedan ser reutilizados.

Dashboard: panel de visualizaciones donde se representa de manera gráfica medidores e indicadores.

ETL: proceso de extracción, carga y transformación de datos utilizados en el traslado de datos de un sistema de almacenamientos de datos a otro.

OLE DB: Object Linking and Embedding for Databases. Tecnología desarrollada por Microsoft usada para tener acceso a diferentes fuentes de información, o bases de datos, de manera uniforme.

URI: Un identificador de recursos uniforme o URI es una cadena de caracteres que identifica los recursos de una red de forma unívoca .

Revenue management: La gestión de ingresos es la aplicación de herramientas analíticas que predicen el comportamiento del consumidor en un nivel de micromercado, y optimiza la disponibilidad y precio del producto para maximizar el crecimiento de los ingresos.

SaaS: Software as Services. Denominación para software que se accede a través de Internet desde un cliente y que se aloja en los servidores de las compañías que lo desarrolla.

SQL: lenguaje para administrar, modificar y realizar consultas en bases de datos.

CAPÍTULO 1

1.1. INTRODUCCIÓN

Este trabajo de fin de grado (en adelante TFG) consiste en describir la instalación e implementación de un sistema de almacenamiento llamado Data Warehouse con los servicios que ofrece Microsoft SQL Server. La construcción de un Data Warehouse permite tener una colección de datos centralizados que provienen de varias fuentes y servir para un posterior modelado, análisis y visualización de informes basados en esos datos. En el desarrollo de este TFG, se usan datos abiertos y públicos como fuente de datos para construir el Data Warehouse.

A lo largo de este documento, se describe el contexto en el que se desarrolla este proyecto, los conceptos teóricos relacionados, la tecnología utilizada y su implementación. Este documento se divide en varios capítulos y secciones. El capítulo 1 es una breve introducción al proyecto del TFG, su justificación y sus objetivos. En el capítulo 2 se hace una descripción del contexto del proyecto, la definición de Data Warehouse, las herramientas para desarrollarlo y la descripción de los Datos Abiertos. La explicación, la planificación y el alcance del proyecto se detalla en el capítulo 3. El desarrollo del propio proyecto se especifica en el capítulo 4. Y por último, el documento lo finaliza el capítulo 5 con conclusiones y líneas futuras del proyecto, además de la bibliografía utilizada.

1.2. JUSTIFICACIÓN

Este TFG surge de mi curiosidad e interés por el desarrollo de sistemas de Business Intelligence y el uso de fuentes de Open Data.

En estos dos últimos años he tenido la oportunidad de trabajar con un sistema de información que utilizaba varias bases de datos. Uno de los puntos pendientes de la empresa para la que trabajaba era, precisamente, explotar mejor los datos de ese sistema. Las técnicas y tecnologías que permiten esa explotación de los datos internos de una empresa se engloban bajo el término Business Intelligence.

Un sistema de Business Intelligence es una implementación de varias aplicaciones y procesos cuyo objetivo es aprovechar los datos que produce y almacena una empresa. Este aprovechamiento consiste en recolectar datos de interés, tratarlos, realizar informes y sacar conclusiones relevantes para la empresa. Y dependiendo de las particularidades de las bases de datos del sistema de cada empresa, el tipo de almacenamiento más adecuado y más común es el Data Warehouse. Este tipo de almacenamiento de datos es posible realizarlos con los diferentes servicios y aplicaciones de Microsoft SQL Server, un sistema de gestión de bases de datos con el que estoy familiarizado.

Usar un Data Warehouse es una decisión técnica importante. Dependiendo de las necesidades de la organización que decida implantar este sistema y del éxito de la ejecución de dicha implantación, tener un Data Warehouse conlleva una serie de beneficios:

1. **Mejora la inteligencia empresarial.** El uso de diferentes fuentes de datos, su tratamiento y su almacenamiento en un sitio centralizado, permite que los datos tengan menos errores, sean más consistentes y también sean información fundamental para apoyar la toma de decisiones de una organización con un conocimiento mayor de su entorno y de su actividad. La información almacenada también ayuda a estudiar y analizar periodos de tiempo y tendencias de la actividad de una organización.
2. **Mejora la calidad y la consistencia de los datos.** Un Data Warehouse es, en esencia, un registro histórico de datos. La acumulación y la integración constante de una gran cantidad de datos, reduce posibles errores al ser tratados continuamente. Estas integraciones hacen que los datos sean confiables y consistentes. La calidad de los datos ayuda a los analistas a aumentar la eficiencia de su trabajo, proporcionando información empresarial más precisa y a quienes toman las decisiones a basar sus decisiones en hechos concretos.
3. **Alto retorno de la inversión.** En términos de generación de ingresos, el almacenamiento de datos es caro pero produce resultados de calidad con el tiempo. Poner en marcha un sistema de Data Warehouse conlleva un esfuerzo considerable de personal especializado, infraestructura hardware y software adecuado. Sin embargo, la productividad mejora con el tiempo. Con su uso, la información recolectada y analizada facilita la comprensión de la actividad de la

organización, lo que permite adoptar medidas que contribuyan a mejorar la competitividad.

4. **Tiempo eficiente.** El almacenamiento de datos centralizado en un Data Warehouse permite a los responsables de la toma de decisiones acceder rápidamente a datos críticos de diferentes fuentes. Se pueden tomar decisiones eficientes en un intervalo de tiempo rápido, sin perder el tiempo recopilando datos de múltiples fuentes.
5. **Rendimiento mejorado en la consulta de datos.** Los Data Warehouse están hechos especialmente para recopilar y analizar los datos. Estos sistemas facilitan el acceso rápido a los datos y la creación de informes personalizados para necesidades específicas. Si se utiliza con funcionalidades SQL avanzadas, ayuda a optimizar el rendimiento de la base de datos.
6. **BI de fuentes heterogéneas.** En un Data Warehouse, la integración de estos datos se realiza y se hace accesible en un solo lugar. La acumulación de todos los datos de una organización en un solo contenedor disminuye la duplicación de datos, lo que hace que el almacén de datos sea una vista única de un historial en lugar de múltiples historiales con múltiples desarrollos.

Cualquier organización que necesite centralizar información de varias fuentes, es susceptible de usar un sistema de Business Intelligence con un Data Warehouse. La compañía canaria de navieras Fred Olsen S.A. utiliza un Data Warehouse para el departamento de Revenue Management (Gestión de ingresos). Este área analiza el comportamiento de la competencia y la actividad de los clientes de la empresa. David Falcón Molina, Revenue Manager en Fred Olsen.SA lo explica en [la primera parte de esta charla](#).

Otro motivo por el que planteé este TFG es por la cantidad de ofertas laborales relacionadas con el área de Business Intelligence (BI) y las posibilidades de aspirar a nuevas perspectivas laborales. Hay bastante demanda de perfiles con conocimientos de BI, de desarrollo ETL, de especialistas en programas como Pentaho y Power BI, y de administradores de bases de datos. Estos roles están ligados al área de BI y a la gestión de datos, temas que están en auge en el mundo empresarial y laboral.

Y por otro lado, tengo la curiosidad por explorar las posibilidades que ofrecen los Datos Abiertos. Las iniciativas de Open Data también son una fuente de datos muy interesantes con las que explorar el uso de herramientas de datos.

1.3. OBJETIVOS

Los objetivos del desarrollo de este TFG son varios. El objetivo principal es **implementar un sistema de almacenamiento Data Warehouse**. Este desarrollo implica conocer los fundamentos de este tipo de almacenamiento de datos, su arquitectura y su diseño para el caso práctico de este proyecto.

Los objetivos específicos del proyecto son los siguientes:

- Desarrollar y estudiar el papel que juega un Data Warehouse dentro de un sistema de Business Intelligence, ya que es uno de los elementos principales y fuente de la información que se utiliza para hacer modelos de datos e informes.
- Analizar y realizar los procesos de extracción, transformación y carga de datos (ETL) para alimentar el sistema Data Warehouse de información. Es la manera de crear un flujo de datos hacia este sistema de almacenamiento.
- Explorar las iniciativas de Datos Abiertos (también Open Data), que son una fuente de datos muy interesante para combinar con el desarrollo de este proyecto. Profundizaremos en aspectos como quienes son sus proveedores, como se estructuran y como pueden ser utilizadas estas fuentes de datos en un caso práctico.

Este proyecto es la oportunidad perfecta para avanzar mis conocimientos en el campo del Business Intelligence y aprender a utilizar los Datos Abiertos. En el siguiente capítulo, se describe el contexto en el que se ubica este TFG y se definen tecnicismos ya nombrados como Data Warehouse, Business Intelligence, ETL y Datos Abiertos.

CAPÍTULO 2

2.1. CONTEXTO DEL PROYECTO

En la última década, los datos y la información han cobrado un valor muy importante en nuestra sociedad, cada vez más sumergida en el uso de las tecnologías. El uso de redes sociales, aplicaciones web y móviles por parte de millones de personas en todo el mundo genera una ingente cantidad de información diaria que es recogida y procesada. Las organizaciones y entidades empresariales también recopilan y generan cada vez más información, ya sea de sus clientes, de sus actividades, del entorno, etc.

En 2009, el periodista norteamericano [Stephen Baker](#) publicó el libro [Numerati](#), un trabajo de investigación en el que el autor relata como los matemáticos ya estaban utilizando los datos para realizar modelos en los que simulan situaciones para predecir resultados. El libro es un recorrido por los diversos roles en los que la sociedad puede ser modelada matemáticamente por sus datos: como consumidores, como empleados, como votantes, etc. A lo largo del libro, Stephen Baker narra las entrevistas con especialistas de empresas como IBM y el director matemático de la Agencia Nacional de Seguridad norteamericana, entre otros. En sus relatos, hablan de ejemplos prácticos sobre como hacen uso de los datos en sus respectivas áreas de trabajo.

Otro libro más actual, [Alquimia, como los datos se están transformando en oro](#) de Juan Manuel López Zafra y Ricardo A. Queralt Sánchez de las Matas, ambos doctores y profesores de ciencia de datos en el prestigioso centro de estudios financieros [CUNEF](#), describen largo y tendido sobre como se están utilizando los datos en diversas áreas como en las finanzas, los seguros, el turismo y los deportes, e incluso, vaticinan la necesidad de un nuevo perfil profesional dentro del sector de la ciencia de datos: el Data Translator. Como curiosidad, entre los dos libros hay 10 años de diferencia en el momento de sus respectivas publicaciones.

2.1.1. BIG DATA Y BUSINESS INTELLIGENCE

La gestión de grandes cantidades de datos sigue y seguirá planteando nuevos problemas que necesitan nuevas soluciones, nuevas tecnologías y nuevos roles profesionales. El **Big Data** es un concepto amplio que engloba la gestión y el análisis de grandes cantidades de datos de diversas fuentes. Una definición más formal del término Big Data de la web www.powerdata.es, se refiere a *conjuntos de datos o combinaciones de conjuntos de datos cuyo tamaño (volumen), complejidad (variabilidad) y velocidad de crecimiento (velocidad) dificultan su captura, gestión, procesamiento o análisis mediante tecnologías y herramientas convencionales*.

En el marco del Big Data, el análisis de los datos genera a su vez información que puede ser muy útil: sacar conclusiones que de otra manera no se podrían evidenciar y tomarlas como base para tomar decisiones, establecer estrategias, anticiparse a eventos que se podrían producir, etc.

El uso de esos datos por parte de una organización es lo que se denomina **Business Intelligence**. Una definición más concreta y [extraída de la Wikipedia](#), es *el conjunto de estrategias, aplicaciones, datos, productos, tecnologías y arquitectura técnicas, los cuales están enfocados a la administración y creación de conocimiento sobre el medio, a través del análisis de los datos existentes en una organización o empresa*. El objetivo del Business Intelligence es optimizar y mejorar el proceso de toma de decisiones en los negocios. Pero para llegar hasta el punto de que los datos puedan ser interpretados en esta área, es necesario hacer un gran esfuerzo en recolectarlos, almacenarlos y organizarlos.

Las ventajas de usar un sistema de Business Intelligence, de manera resumida, son:

- Obtener un mayor conocimiento al analizar los datos.
- Brindar herramientas de análisis como apoyo para la toma de decisiones.
- Revelar tendencias pasadas y predecir resultados futuros.
- Hacer accesibles los datos a través de informes dinámicos, personalizados y actualizados.
- Automatizar la carga y transformación de datos.

Tradicionalmente, los datos de una organización se almacenaban en papel, hojas de cálculo y bases de datos de una manera básica: historial de compras, stock de productos, datos de los clientes, etc. Cada vez más, las organizaciones fueron implantando sistemas que permitían recoger más información, hacerla más accesible y aprovecharla de diversas maneras. También fueron creciendo las fuentes de datos, en diversidad y cantidad. En este contexto, surge el fenómeno del Big Data.

Como se ha mencionado antes, la gestión de grandes cantidades de datos ha requerido de nuevas tecnologías que se agrupan en el concepto de Big Data. Una organización que quiera concentrar toda su información, dependiendo de la cantidad, puede encontrarse con varios problemas. Uno puede ser la manera de canalizar toda esa información. Necesitará de una arquitectura que soporte la gran cantidad de datos a gestionar. Otro puede ser la heterogeneidad de los datos. Estos pueden ser desde documentos de texto, bases de datos hasta archivos de audios, vídeos y otros formatos diversos.

Además, la gestión de datos necesita de sistemas de almacenamiento. La elección de dicho sistema es una decisión muy importante y clave en el diseño de una arquitectura software de este ámbito. Y esta elección dependerá del modelo de almacenamiento que mejor se ajuste a nuestras necesidades. Luego, dentro del modelo de almacenamiento, se elige un almacén determinado de datos en función de la forma de estructurar los datos, los tipos de operaciones que admiten, el conjunto de características, el costo y la facilidad de administración. En [la documentación oficial de Microsoft Azure](#) se hace una descripción de algunos modelos de almacenamiento de datos y los servicios de Microsoft con los que se puede implementar cada uno de ellos.

El sistema de almacenamiento principal de un sistema de Business Intelligence se denomina **Data Warehouse**. Es un repositorio de datos unificado, los cuales han sido modelados y estructurados para su uso. Conviene explicar también que otros sistemas de almacenamientos relacionados con los sistemas de Business Intelligence y Big Data como Data marts y Data Lake.

Un **Data Mart** tiene la misma funcionalidad que un Data Warehouse, pero en lugar de ser un repositorio más general, es específico de departamentos o áreas de una empresa. Según la arquitectura del sistema de Business Intelligence, la información de un Data Warehouse puede provenir o fluir hacia uno o varios Data Marts.

Un **Data Lake** está formado por datos no estructurados, de distintos formatos y que provienen de diversas fuentes. Es *un repositorio de almacenamiento que contienen una gran cantidad de datos en bruto y que se mantienen allí hasta que sea necesario*, según la web www.powerdata.es.

Mientras que el objetivo de un Data Lake es centralizar toda la información que se generan de diversas fuentes, la finalidad de un Data Warehouse es tener esa información, además de centralizada, disponible de una manera organizada para ser utilizada en análisis, informes, etc.

2.2. DATA WAREHOUSE

¿Qué es un Data Warehouse? Hay varias formas de definir este concepto. Según Cuellar (2011), *“un Data Warehousing es un almacén único, completo y consistente de datos obtenidos de una variedad de fuentes y puestos a disposición de los usuarios finales de una manera que puedan entender y utilizar en un contexto empresaria”*. Otra definición de otro autor, Devlin (2011), *“el Data Warehousing es un proceso, no un producto, para ensamblar y administrar datos de diversas fuentes con el fin de obtener una visión única y detallada de una parte o de toda una empresa”*.

Un Data Warehouse es un sistema de almacenamiento de datos estructurados y homogéneos. Y también es el componente principal y crítico de un sistema de Business Intelligence. Este sistema de almacenamiento permite centralizar los datos de varias fuentes. Estos datos pueden manipularse acorde a requisitos y medidas que la empresa necesite para realizar análisis e informes. Desde otro punto de vista, un Data Warehouse funciona como un histórico de datos.

2.3. PROCESAMIENTO OLTP Y OLAP

Para entender cómo se implementa un Data Warehouse y, por lo tanto, un sistema de Business Intelligence, es necesario explicar los tipos de sistemas de almacenamiento que se utilizan en este ámbito.

Según [la documentación oficial de Microsoft sobre los distintos tipos de almacenamiento de datos](#), hay almacenamientos que se utilizan en función de la naturaleza de los datos y su procesamiento. En la documentación se especifica cada almacenamiento con su descripción, los productos de Microsoft que los proveen, para qué tipo de datos se utilizan y que cargas de trabajo soporta, además de ejemplos prácticos en los que se puede utilizar cada uno de ellos.

Para seguir profundizando, nos centraremos en los sistemas de administración de bases de datos relacionales (RDBMS) y almacenes de análisis de datos. En estos sistemas, se utilizan procesamiento de transacciones en línea (OLTP) y procesamiento analítico en línea (OLAP).

2.3.1. OLTP

Una base de datos cuya actividad se basa en actualizar, crear y borrar datos de manera frecuente es una **base de datos OLTP** (Online Transactional Processing). Son las bases de datos más comunes y se definen con las siguientes particularidades:

- Se centran en el procesamiento de transacciones, es decir, mediante operaciones online que se realizan desde aplicaciones usadas por usuarios.
- Son eficientes para operaciones frecuentes lectura y escritura, como las compras en una tienda online, las operaciones bancarias de un banco, etc.
- Las operaciones suelen ser muy específicas y registran un número relativamente pequeño de datos.
- Son utilizadas para sistemas de alta disponibilidad.
- Pueden ser utilizadas por un número alto de usuarios.
- Son bases de datos normalizadas.

La normalización es el conjunto de normas que sigue el diseño de bases de datos relacionales para reducir la repetición de datos y mejorar su integridad. Según [la documentación de Microsoft sobre normalización de bases de datos](#), *esto incluye la creación de tablas y el establecimiento de relaciones entre esas tablas de acuerdo con las reglas diseñadas tanto para proteger los datos como para que la base de datos sea más flexible mediante la eliminación de la redundancia y la dependencia incoherente.*

Este conjunto de normas se resume principalmente en 3 formas normales:

1. Primera forma normal:

- Eliminar grupos de repetición en tablas individuales.
- Crear una tabla independiente para cada conjunto de datos relacionados.
- Identificar cada conjunto de datos relacionados con una clave principal.

2. Segunda forma normal:

- Crear tablas independientes para conjuntos de valores que se aplican a varios registros.
- Relacionar estas tablas con una clave externa.

3. Tercera forma normal:

- Eliminar los campos que no dependen de la clave.

Además de estas tres formas, existen una cuarta forma normal llamada forma normal de Boyce-Codd (BCNF) y, en algunos artículos, se describen hasta una quinta y una sexta forma normal.

En la imagen se muestra un ejemplo de tablas de una base de datos normalizada:

Cientes

CienteID	Nombre	Apellidos
1	Ángel	López Rodríguez

Facturas

FacturalID	CienteID	TarjetaCrédito	Caducidad
34	1	Xxxx xxxx xxxx 2345	03/2024

Direcciones de envío

DirecciónID	CienteID	Dirección	Ciudad	País	CP
897	1	Calle Romanones, N.º 3	Arrecife	España	35500

Figura 1: tabla normalizada.

Como podemos ver, las tablas se relacionan entre sí mediante los valores de las columnas ID (claves primarias y claves foráneas), evitando repetir la información que se identifica por una ID determinada en otras tablas.

Este es el tipo de bases de datos que utilizaría cualquier e-commerce, aplicación web y cualquier software que requiera de almacenamiento usado por usuarios comunes.

2.3.2. OLAP

Una **base de datos OLAP** está orientada a facilitar información a analistas, gestores y ejecutivos. Son fundamentales en el Business Intelligence porque están optimizadas para hacer lecturas de grandes cargas de trabajo y pocas operaciones de escritura. Se definen con las siguientes características:

- Se centran en facilitar el análisis.
- Realizan pocas transacciones.
- Los datos son cargados en paquetes de tareas automatizadas, normalmente cuando los servidores tienen poca actividad.
- Los datos son denormalizados y reestructurados.

La denormalización de una base de datos OLAP consiste en romper con los principios de normalización típico de las bases de datos relacionales, explicados en el apartado anterior. Esta denormalización se debe a que la transformación y reestructuración de las bases de datos resultantes, da lugar a datos redundantes en las tablas, a información calculada de operaciones y datos agregados de otras tablas.

La imagen es un ejemplo de tabla de un base de datos OLAP, concretamente de un Data Warehouse:

Cientes

WarehouseID	CienteID	Nombre	Apellidos	FacturaID	TarjetaCrédito	Caducidad
1000	1	Ángel	López Rodríguez	34	Xxxx xxxx xxxx 2345	03/2024
1001	1	Ángel	López Rodríguez	35	Xxxx xxxx xxxx 7896	01/2026

Caducidad	DirecciónID	Dirección	Ciudad	País	CP
03/2024	897	Calle Romanones, N.º 3	Arrecife	España	35500
01/2026	897	Calle Romanones, N.º 3	Arrecife	España	35500

Figura 2: tabla denormalizada.

A diferencia de una tabla normalizada, la imagen anterior muestra tablas donde la información es redundante.

Este tipo de tablas se denominan Fact Tables. Como se puede deducir de su traducción al español, una Fact Table consiste en una tabla que registra un hecho concreto. Estos hechos son indicadores de negocio, valores que sirven de medidas en un determinado contexto. En el ejemplo, cada fila de datos es una transacción donde se detalla el cliente, su tarjeta de crédito y su dirección.

Una Fact Table se produce por la intersección de otras tablas. Estas tablas son llamadas Dimension Tables, o tablas de dimensión. Su nombre se debe a que este tipo de tablas representan una dimensión de la tabla de los hechos. Explicado de otra manera, una tabla de dimensión almacena información descriptiva de la tabla de hechos.

En la imagen del ejemplo anterior, nuestra tablas de dimensión son aquellas tablas cuyas Claves Primarias se encuentran en la Fact Table, como podrían ser unas hipotéticas tablas Cliente, Factura y Dirección. Estas tablas de dimensión extienden la información de la tabla de hechos. Este planteamiento de una Fact Table formada por las claves primarias de varias Dimension Tables da lugar a un **esquema de estrella (Star Design)**.

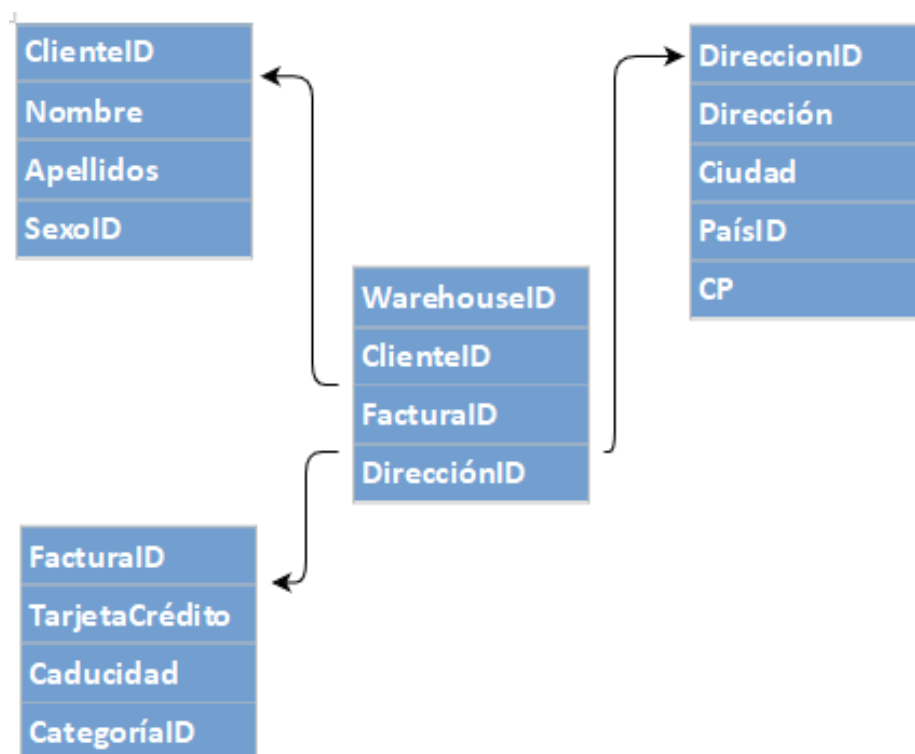


Figura 3: esquema en estrella (star design).

El esquema puede ser más complejo si, a su vez, las tablas de dimensiones tienen sus propias tablas de dimensiones. A este esquema se le llama **copo de nieve (Snowflake Design)**.

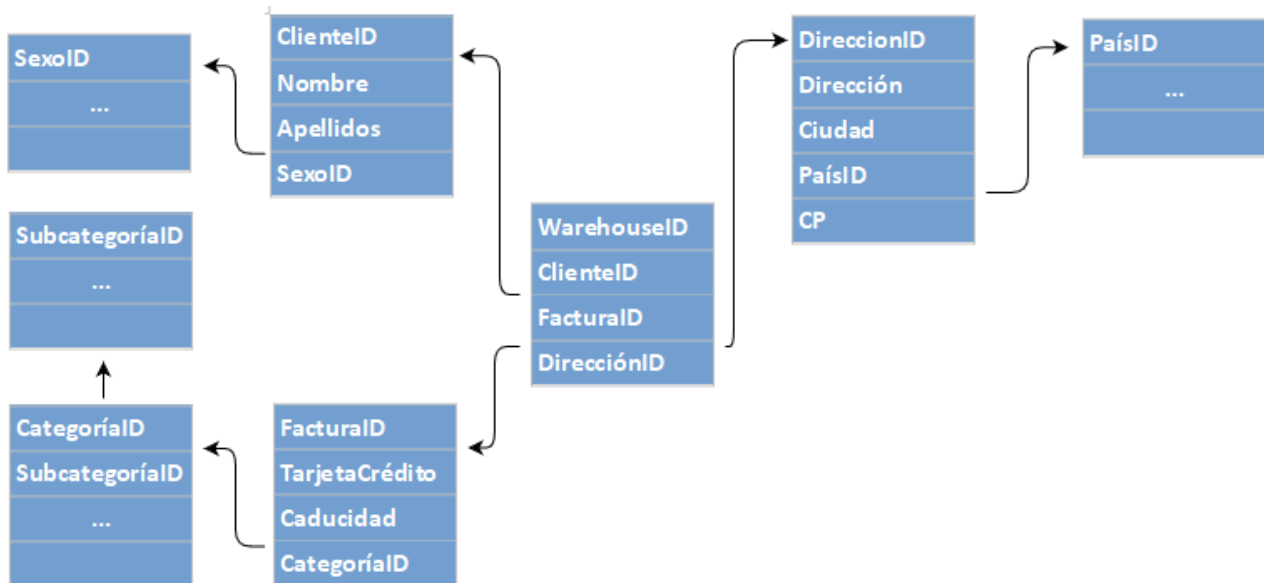


Figura 4: esquema copo de nieve (snowflake design).

Si el esquema se compone de más de una tabla Fact, el esquema se denomina **galaxia (Galaxy Design)**. En este caso, las tablas fact estarían relacionadas entre sí, además de las tablas de dimensiones.

Un cubo OLAP es una herramienta más del análisis de datos y se genera de los datos almacenados de un Data Warehouse. En este proyecto no se va a profundizar en su uso, pero es necesario resaltar que la finalidad de un sistema con Data Warehouse es el posterior análisis de sus datos y la representación gráfica para cumplir con los objetivos de todo sistema de Business Intelligence: proveer de información útil para tomar mejores decisiones.

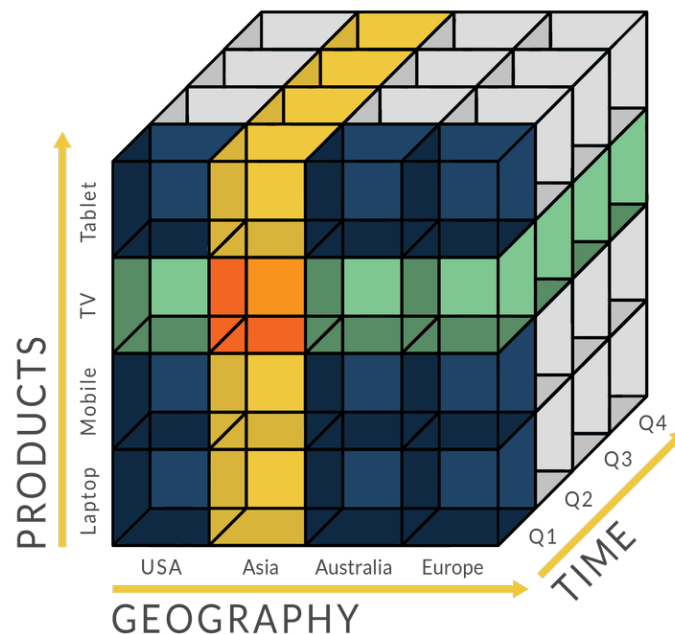


Figura 5: cubo OLAP.

Fuente: <https://www.itconsultors.com/el-ascenso-y-la-caida-del-cubo-olap>

2.4. MODELADO DIMENSIONAL

El modelado es el primer paso para construir un Data Warehouse. Según [su definición en la Wikipedia](#), *el modelado dimensional es el conjunto de técnicas y conceptos utilizados para diseñar almacenes de datos*. También se puede definir el modelo dimensional como *una técnica de diseño lógico que tiene como objetivo presentar los datos [...] para permitir su acceso con un alto rendimiento* (León, Ciclo de vida de Ralph Kimball, 2014).

En este primer paso se realiza en función de los requisitos proporcionados por los propietarios y/o clientes del sistema. También se diseña el esquema y la relación entre los

componentes que formarán el Data Warehouse como las bases de datos, las tablas y los datos que se almacenarán.

Entre las diferentes metodologías que existen para implementar sistemas Data Warehouse, **el modelo Kimball , también llamado modelo dimensional**, es uno de los más usados. De hecho, los términos que ya se han descrito como la tabla dimensional, la tabla de hechos y la manera de relacionarse en distintos esquemas vienen del modelo propuesto por Ralph Kimball.

El modelo Kimball tiene considerables ventajas frente a otras metodologías. Otro modelo llamado Inmon, tiene un enfoque distinto al propuesto por Kimball. El **modelo Inmon** persigue la integración de todos los datos de la compañía, estando **orientado hacia el almacenaje de grandes volúmenes de datos**, por lo que su estructura interna normalizada se diseña para evitar la redundancia de datos, simplificar las labores de mantenimiento, etc. cuestiones que complican las consultas de la información, requiriendo que los usuarios finales estén mucho más especializados. Por el contrario, el **modelo Kimball** está **orientado a la consulta de la información**, por lo que su estructura interna está especialmente diseñada para garantizar una explotación de los datos rápida y sencilla, no requiriendo usuarios especializados para ello.

Las ventajas de usar el modelo Kimball frente al modelo Inmon se resumen en la siguiente tabla:

	Inmon	Kimball
Presupuesto	Coste inicial alto	Coste inicial bajo
Plazos	Requiere más tiempo de desarrollo	Tiempo de desarrollo inferior
Expertise	Equipo con especialización alta	Equipo con especialización media
Alcance	Toda la compañía	Departamentos individuales
Mantenimiento	Fácil mantenimiento	Mantenimiento más complejo

Figura 6: Diferencias entre las metodologías Inmon y Kimball.

Fuente: <https://blog.bi-geek.com/arquitectura-comparativa-inmon-y-kimball/>

2.4.1. PROCESO DE MODELADO DIMENSIONAL

El proceso de modelado dimensional, según su definición en la Wikipedia, se divide en un método de cuatro pasos que ayuda a asegurar la facilidad del uso del modelo y el uso del Data Warehouse. Los fundamentos del diseño se basan en el proceso de negocio real que debe cubrir el Data Warehouse.:

1. **Escoger el proceso de negocio.** El primer paso en el modelo es describir el proceso de negocio en el que se basa el modelo. Esto podría ser por ejemplo una situación de ventas en una tienda al por menor. Para describir el proceso de negocio, se puede optar por hacer esto en texto plano o utilizar Notación de Modelado de Procesos de Negocio (BPMN) u otras guías de diseño, como el Lenguaje Unificado de Modelado (UML).
2. **Declarar el "grain".** El "grain" del modelo es la descripción exacta de lo que el modelo dimensional debería concentrarse. Para aclarar lo que significa el "grain", se escoge el proceso central y se describe con una sola oración. Además el "grain" (oración) es a lo que va a construir las tablas de dimensiones y las tabla de hechos. Puede que resulte necesario volver a este paso para alterar el "grain" debido a nueva información que se aporte al modelo.
3. **Identificar las dimensiones:** Las dimensiones son la base de la tabla de hechos, y es donde se recogen los datos de la tabla de hechos. Normalmente las dimensiones son sustantivos, como fecha, tienda, inventario, etc. Estas dimensiones son donde se almacenan todos los datos. Por ejemplo, la dimensión fecha podría contener datos tales como año, mes y día de la semana.
4. **Identificar los hechos:** Este paso es identificar los hechos numéricos que poblarán cada fila de la tabla de hechos. Este paso está estrechamente relacionado con los usuarios de negocio del sistema, ya que es donde consiguen el acceso a los datos almacenados en el Data Warehouse. Por lo tanto la mayor parte de las filas de la tabla de hecho son cifras numéricas, aditivos tales como cantidad o costo por unidad, etc.

2.4.2. CICLO DE VIDA DIMENSIONAL DEL NEGOCIO

La metodología de Kimball se basa en lo que se denomina el Ciclo de Vida Dimensional del Negocio y se refleja en la siguiente imagen dividido en fases:

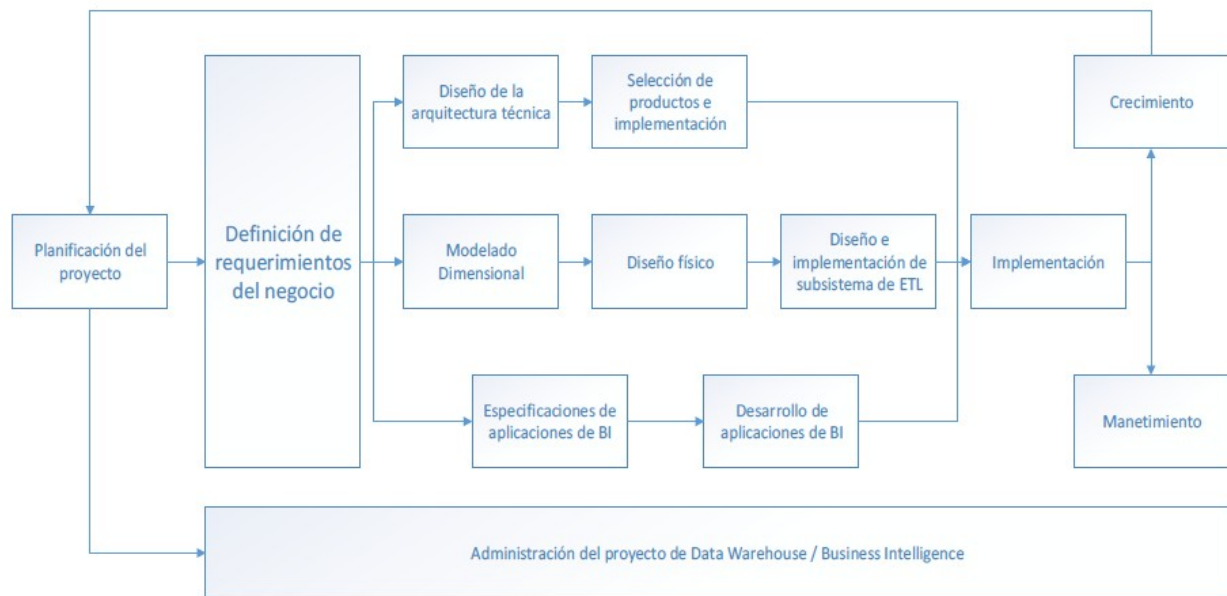


Figura 7: Fases de la metodología Kimball.
Fuente: <https://cienciadigital.org/> (Muirillo, 2008)

Las fases establecidas por Kimball se pueden desarrollar en paralelo o de manera secuencial y su objetivo es asegurar la calidad del desarrollo del Data Warehouse en cada una de sus fases:

- **Planificación del Proyecto:** La planificación implica la definición y el alcance del proyecto. En esta fase se identifica las partes involucradas en el proyecto como los recursos, los perfiles y las tareas a realizar, su duración y su secuencialidad.
- **Definición de los Requerimientos del Negocio:** Los requerimientos del negocio establecen el alcance del Data Warehouse. En esta fase se determina qué datos debe contener, cómo debe estar organizado, cada cuánto debe actualizarse, qué usuarios accederán y desde dónde, etc. Según Kimball, los requerimientos del negocio son el centro de todo lo relacionado con el Data Warehouse.
- **Modelado Dimensional:** A partir de los requerimientos del negocio se obtienen los requerimientos analíticos de los usuarios. Con esta información, se diseña el modelo de datos.

- **Diseño Físico:** El diseño físico de las bases de datos se centra en seleccionar las estructuras necesarias para soportar el diseño del modelo dimensional .
- **Diseño y Desarrollo de subsistema de ETL:** Se definen los procesos de extracción, transformación y carga de los datos (ETL). Estos procesos son los que rellenan el Data Warehouse de información desde las fuentes de datos reflejadas en el diseño físico.
- **Diseño de la Arquitectura Técnica:** El entorno de un Data Warehouse requiere de la integración de varias tecnologías. Se debe tener en cuenta los requerimientos del negocio, los sistemas informáticos actuales y las directrices técnicas.
- **Selección de Productos e Implementación:** El diseño de arquitectura técnica es un marco para seleccionar y evaluar componentes específicos de la arquitectura como la plataforma de hardware, el motor de base de datos, la herramienta de ETL y cualquier desarrollo necesario.
- **Especificaciones y desarrollo de aplicaciones de BI:** En esta etapa se identifican los diferentes perfiles de usuarios del Data Warehouse según el nivel de análisis. Cada perfil de usuario puede tener un alcance y un acceso distinto a los datos. Además, es posible el uso de diferentes tipos de aplicaciones.
- **Implementación:** En el desarrollo se unen las tecnologías seleccionadas, los datos y las aplicaciones de BI para ser usadas por los usuarios según los requerimientos del negocio. Hay otros factores que aseguran el éxito de la implementación: la capacitación de los usuarios finales, el soporte técnico, la comunicación y el feedback.
- **Mantenimiento y crecimiento:** La implementación de un Data Warehouse es un proceso iterativo con etapas bien definidas. Este proceso evoluciona al igual que la organización que lo implementa. El crecimiento y el mantenimiento del Data Warehouse debe ser continuo. Su evolución debe ser visto como un signo de éxito.
- **Administración del Proyecto:** La administración del proyecto asegura que las fases del ciclo de vida dimensional se ejecuten bien y de manera sincronizada. Tal y como indica el diagrama de la figura 7, la administración del proyecto supervisa todo el ciclo de vida controlando el estado del proyecto.

Después de todo lo explicado, se afirma que un Data Warehouse implementado por el modelo Kimball está compuesto por una o varias bases de datos OLAP. Y también se puede entender que los sistemas de bases de datos OLTP son la fuente de datos que se transforman y se almacenan en un Data Warehouse. Este traslado de datos de un sistema a otro se realiza con procesos ETL.

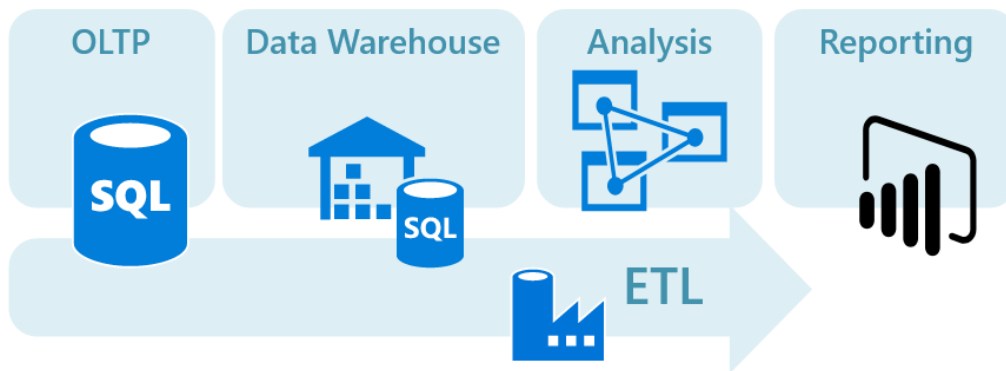


Figura 8: Esquema de un sistema Business Intelligence.

Fuente: <https://docs.microsoft.com/es-es/azure/architecture/data-guide/>

2.5. PROCESOS ETL

Un proceso de extracción, transformación y carga (ETL), tal y como su denominación ya nos indica, consiste en extraer información de varias fuentes, transforma dicha información según las reglas del negocio y se carga en un almacén de destino como un Data Warehouse y/o un Datamart. En otras palabras, un proceso ETL es una canalización de datos (data pipeline) de varias fuentes a un único repositorio.



Figura 9: proceso ETL.

La **extracción** de datos es el primer paso de un proceso ETL y se puede realizar sobre diversas fuentes: bases de datos, archivos en formato XML, JSON, CSV, por nombrar algunos y otras estructuras de datos. Todo dependerá de la herramienta empleada para el proceso ETL.

La **transformación** es el paso que detecta datos incongruentes, valores nulos, errores e incompatibilidades. En este paso se aplican operaciones como filtrado, ordenación, agregación, combinación de datos, limpieza de datos, deduplicación y validación de datos. De esta manera, se asegura una calidad de los datos en la fase de **carga**.

También existe el proceso ELT, cuya diferencia está en el orden de los pasos anteriores, en el cual la transformación de los datos se realizan después de cargarse en el almacén de destino.

2.6. ARQUITECTURA CON TECNOLOGÍAS DE MICROSOFT

Un sistema de Business Intelligence con su Data Warehouse puede ser implementado en diferentes tecnologías, productos software de marcas como Oracle, IBM o Microsoft, por nombrar algunas de las más conocidas.

Un sistema de Business Intelligence se compone de varias etapas:

- Las **fuentes de datos** es el punto de partida. Son todos aquellos elementos que almacenan datos y que alimentan al sistema de Business Intelligence. Estas fuentes de datos pueden ser tanto internas como externas, y pueden ser desde bases de datos, archivos en diversos formatos hasta datos que provengan de sensores, redes sociales y otras fuentes externas.
- Estas fuentes de datos se trasladan y se transforman mediante **procesos ETL**, como ya se ha explicado anteriormente. Esta etapa prepara los datos con los requisitos establecidos para cumplir con las reglas del negocio y del contexto del sistema de Business Intelligence.
- El **almacenamiento de los datos**, ya sea en Data Warehouse y/o Data Marts, es el destino de los datos tras ser extraídos, transformados y cargados por los procesos ETL.

- Una vez almacenados y según las necesidades de los analistas de datos, se crean **modelos con los datos recolectados**. Estos modelos son la base para análisis e informes.
- Por último, la **visualización de informes**, compuesto por gráficas, estadísticas y multitud de elementos visuales, se basa en el almacenamiento de los datos.

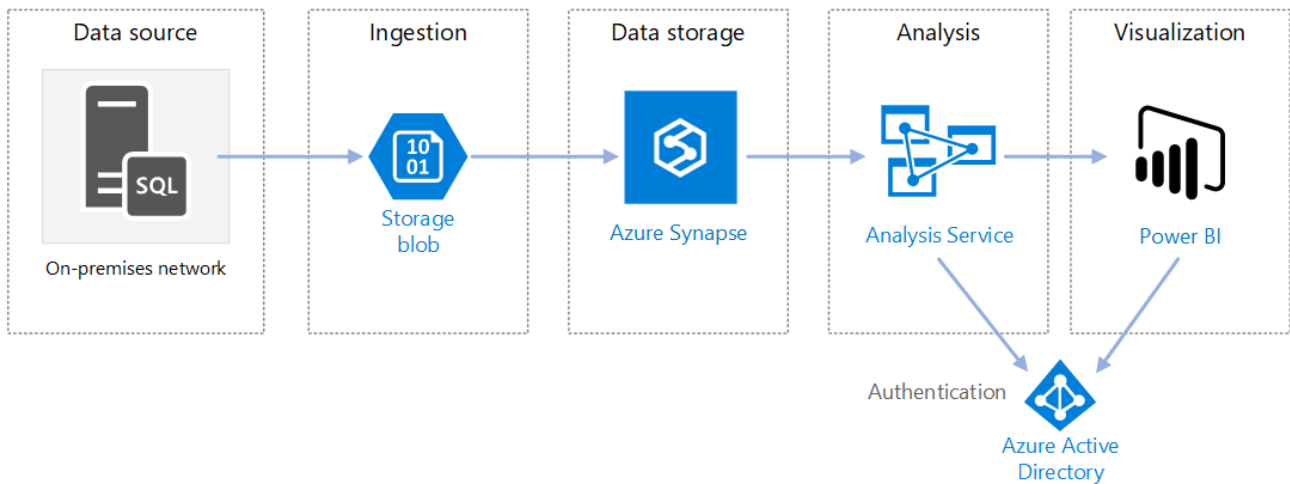


Figura 10: esquema de Data Warehouse con Microsoft Azure.

Fuente: <https://docs.microsoft.com/en-us/azure/architecture/data-guide/relational-data/data-warehousing>

Microsoft tiene un potente motor de bases de datos denominado SQL Server. Este producto se puede complementar con varios servicios para aumentar sus funcionalidades con el tratamiento de bases de datos. Para el objetivo de este TFG, me centraré en hablar de los servicios que me permitirán construir un sistema de almacenamiento Data Warehouse con los servicios de Microsoft SQL Server.

Esquema conceptual de un sistema de Business Intelligence con tecnologías de SQL Server de Microsoft

Por Diego Toribio Rodríguez

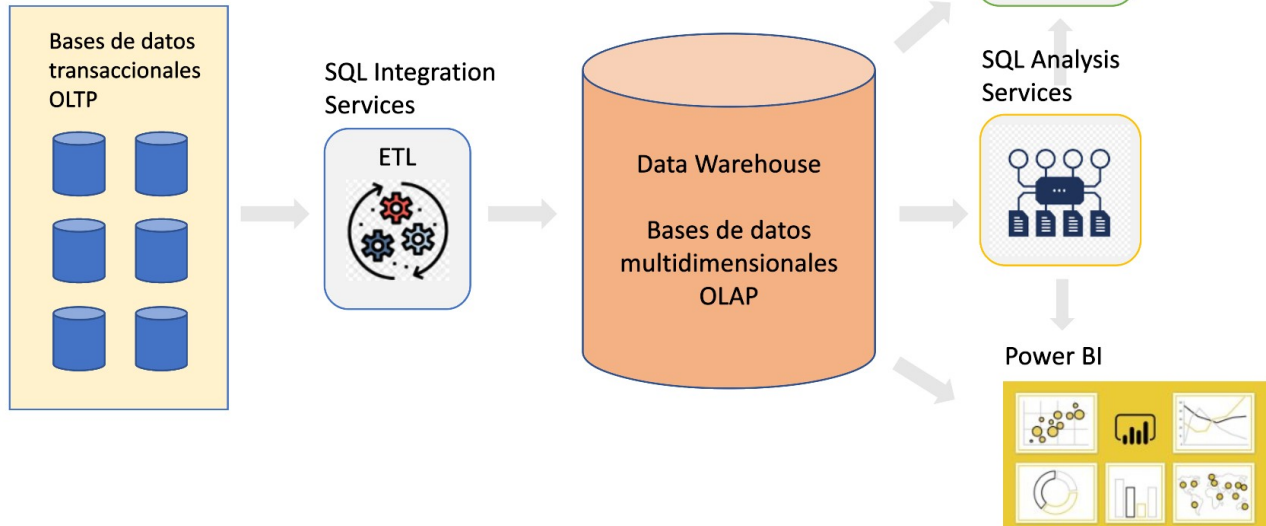


Figura 11: sistema Business Intelligence con Microsoft SQL Server.

A continuación describo cada componente del esquema conceptual:

- **Bases de datos transaccionales OLTP:** Las bases de datos transaccionales, como ya se ha explicado, son aquellas de uso frecuente. En este proyecto están gestionadas con el motor de bases de datos SQL Server.
- **SQL Integration Services:** es el servicio que realiza operaciones ETL: extracción, transformación, integración y minería de datos.
- **Data Warehouse, base de datos OLAP:** son bases de datos relacionales que, al igual que las bases de datos OLTP, están gestionadas con el motor de bases de datos SQL Server.
- **SQL Analysis Services:** es un motor de datos analíticos que ofrece modelos de datos para herramientas de visualización de datos, como SQL Reporting Services y Power BI.
- **SQL Reporting Services y Power BI** son herramientas de visualización de datos, cada una con sus diferencias pero con la misma finalidad.

Este conjunto de tecnologías de Microsoft lleva años usándose para el análisis y modelado de datos. Sin embargo, con el cambio de dirección de la empresa, Microsoft y su actual CEO, Satya Nadella, han apostado muy fuerte por ofrecer una amplia gama de servicios en la nube. Estos servicios están en lo que se denomina Microsoft Azure.

Microsoft Azure es un servicio de computación en la nube que se utiliza para construir, desplegar y administrar aplicaciones utilizando los centros de datos y servidores de Microsoft. Y para ello, Azure ofrece una gama muy grande y variada de productos. Con relación a este TFG, se puede realizar un sistema de Data Warehouse con los servicios de Azure, tal y como se muestra en la imagen:

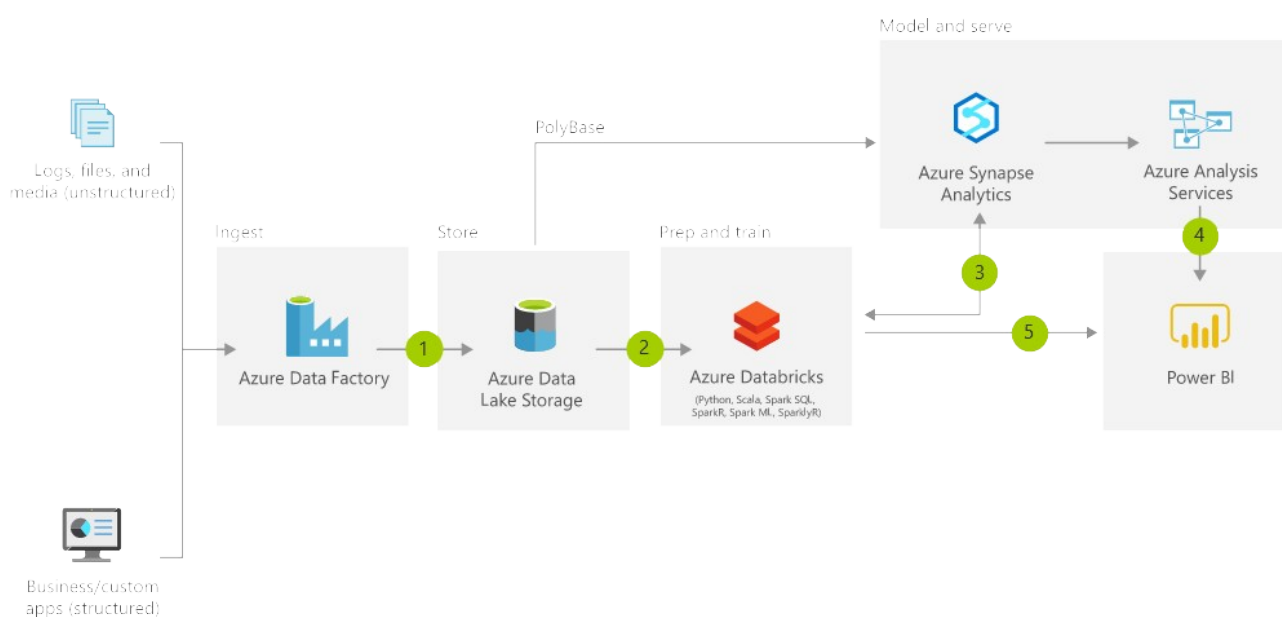


Figura 12: sistema detallado Business Intelligence con Microsoft Azure.

Fuente: <https://docs.microsoft.com/es-es/azure/architecture/solution-ideas/articles/enterprise-data-warehouse>.

La diferencia principal de la implementación de este proyecto entre los servicios de Microsoft SQL Server y Microsoft Azure radica en que toda la instalación y administración del sistema de SQL Server recae en el profesional encargado de esa tarea, mientras que Microsoft Azure ofrece una infraestructura cloud donde la única preocupación es la implementación del proyecto. Pero esta implementación en Microsoft Azure no es objeto de este documento.

2.7. DATOS ABIERTOS

Los Datos Abiertos, también llamados Open Data, forman parte de una iniciativa a nivel mundial que persigue que determinados datos e información de las Administraciones Públicas estén disponibles para todo el mundo. Estos recursos públicos en forma de datos hacen posible su redistribución, reutilización y aprovechamiento por parte de los ciudadanos y empresas.

Los Datos Abiertos recolectados por las Administraciones Públicas provienen de áreas tan dispares como turismo, economía, educación, empleo, gasto presupuestario de la propia Administración, medio ambiente, etc. En la imagen, un ejemplo de los datos en forma de indicadores estadísticos que ofrece el Instituto Estadístico de Canarias, clasificados por temas:



Figura 13: Indicadores por temas del ISTAC.

Fuente: <http://www.gobiernodecanarias.org/istac/datos-abiertos/galerias/visor/indicadores.html>

El valor de estos datos es importante, ya que provienen de fuentes oficiales y pueden servir para generar nuevas oportunidades de negocio, para que nuestros órganos de gobiernos sean más transparentes, para detectar el impacto de las políticas aplicadas y ampliar el conocimiento con dichos datos. Al ser una iniciativa muy reciente y al estar la ciencia de datos en pleno auge, según se relata en la guía web Open Data Handbook [<https://opendatahandbook.org/guide/es/why-open-data/>], *todavía no sabemos qué cosas serán posibles en el futuro. Nuevas combinaciones de datos pueden crear nuevos conocimientos e ideas, que pueden llevar a nuevos campos de aplicación.*

A continuación, enumero algunos ejemplos de aplicaciones basadas en datos abiertos:

- [Findtoilet.dk](#) es una aplicación danesa que muestra la ubicación de los baños públicos en Dinamarca. Esta aplicación está pensada para personas con problemas de salud que necesitan tener siempre un baño localizado.
- [Mapnificent.net](#) es una aplicación que permite calcular el tiempo que se tarda en llegar en transporte público desde un punto de una ciudad a un área, en función del tiempo estimado. Este servicio muestra un gran número de ciudades de todo el mundo y sirve para calcular distancias en tiempo entre lugares de una misma ciudad.

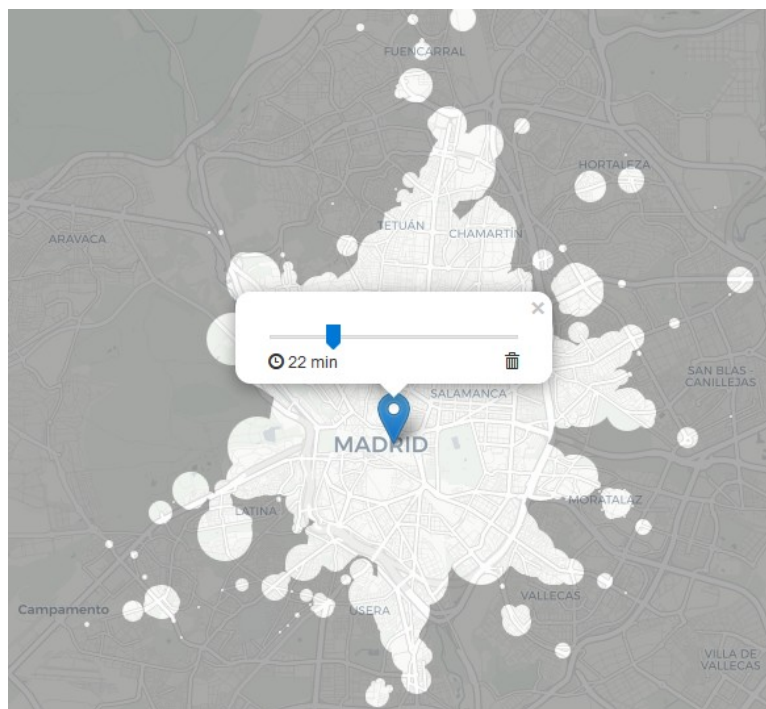


Figura 14: ejemplo de uso de Mapnificent.

Fuente : <https://www.mapnificent.net/>

- LicitaLio es un comparador de contratos públicos donde se muestra información de los contratos de determinadas comunidades autónomas españolas y sus licitadores. La finalidad de esta aplicación es hacer una mejor selección de los licitadores y hacer transparente la información de los contratos públicos.

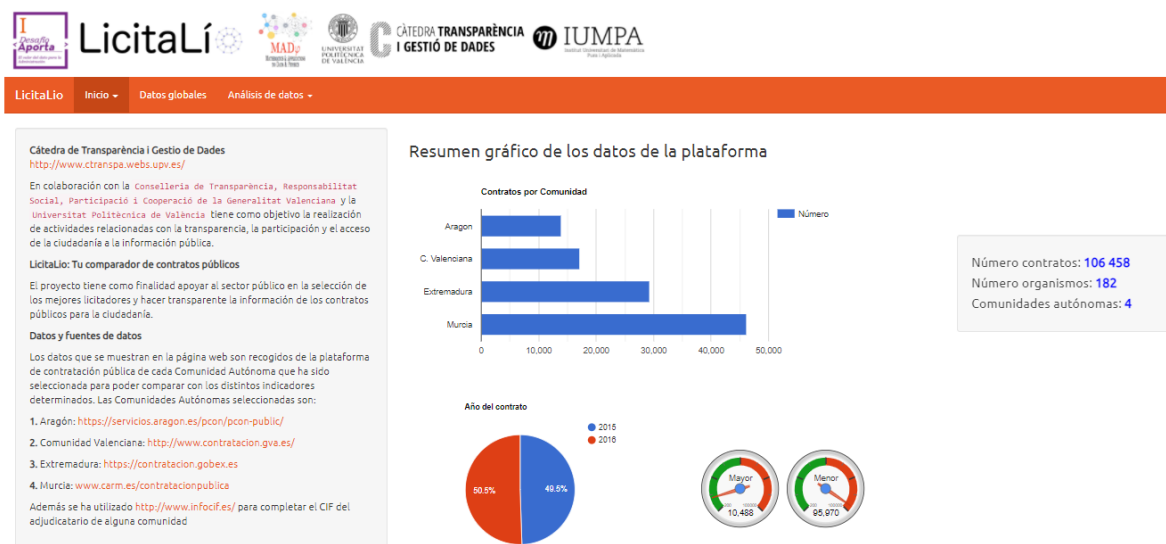


Figura 15: portal de LicitaLio.

Fuente: <https://indicame.upv.es/LicitaLio/>

- Rutasenbici.net es un sitio web para los aficionados y deportistas que buscan rutas de ciclismo por toda la geografía española. Las rutas están detalladas con una descripción, con imágenes y localizadas con coordenadas GPS.

- El Sistema de Observación Meteorológica de Canarias ofrece una plataforma para consultar datos de las estaciones meteorológicas repartidas por el archipiélago canario. Podemos monitorizar en tiempo real los indicadores de las estaciones como la temperatura, la humedad y la velocidad del viento.

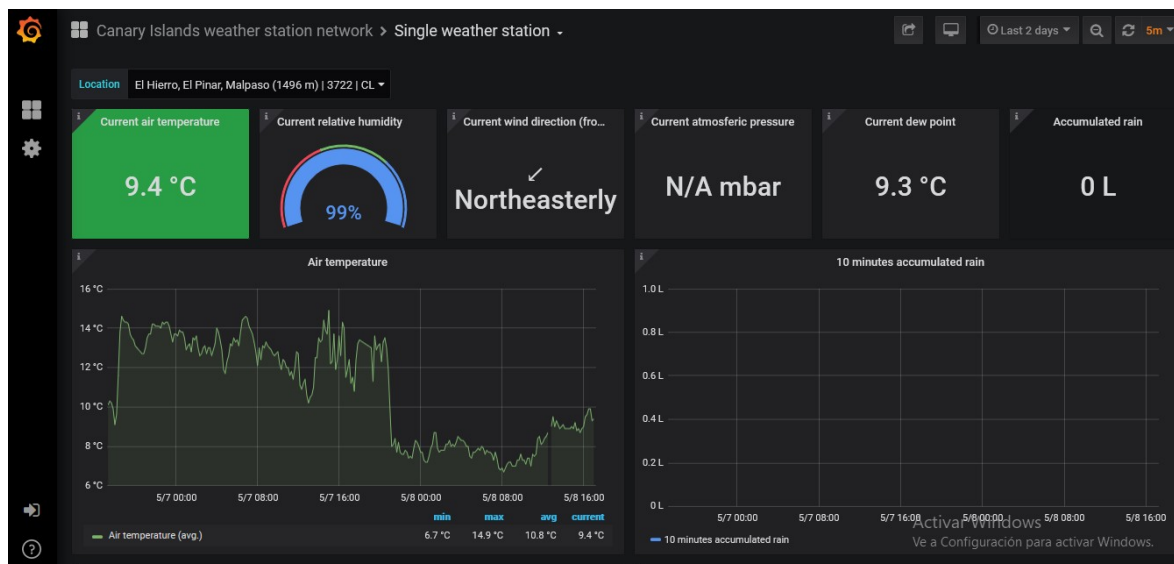


Figura 16: dashboard del sistema de Observación Meteorológica de Canarias.

Fuente: <https://sensores.grafcan.es/>

Estas son solo una muestra de todas las aplicaciones que utilizan Datos Abiertos. Algunas de ellas las podemos encontrar en los sitios web de Datos Abiertos, como [el catálogo de aplicaciones que usan datos del Gobierno de España](#).

Los Datos Abiertos se publican en diferentes formatos de archivo para su descarga y ampliar sus posibilidades de uso. Los formatos más comunes son los siguientes:

- [CSV](#) (Comma- Separated Values): es un formato simple para el almacenamiento de datos tabulares en texto plano que se utiliza frecuentemente para el intercambio de información entre distintas hojas de cálculo que utilizan otros formatos propietarios.
- [GeoJSON \(Geographic JavaScript Object Notation\)](#): es un formato estándar abierto diseñado para representar elementos geográficos sencillos, junto con sus atributos no espaciales, basado en JavaScript Object Notation.
- [GPKG \(GeoPackage\)](#): Formato de archivo universal para datos espaciales vectoriales. Es abierto, basado en estándares, e independiente de plataformas o aplicaciones. Se ha construido sobre la base de [SQLite](#).

- [HTML](#) (HyperText Markup Language): lenguaje utilizado para describir la estructura y el contenido en forma de texto, así como para complementar el texto con objetos tales como imágenes, en la elaboración de páginas web.
- [JSON](#) (JavaScript Object Notation): es un formato utilizado para intercambio de datos, siendo un subconjunto de la notación literal de objetos de JavaScript, pero independiente de lenguaje. Su empleo evita el uso de XML y se hizo muy popular sobre todo entre desarrollos AJAX.
- [KML](#) (Keyhole Markup Language) y [KMZ](#) (Keyhole Markup Language compressed): formato XML para la creación de modelos y almacenamiento de funciones geográficas como puntos, líneas, imágenes, polígonos y modelos. Se emplean principalmente para compartir mapas e información geográfica. Es el estándar del Open Geospatial Consortium y puede emplearse a través de Google Earth. Los ficheros KML habitualmente se distribuyen comprimidos como archivos KMZ.
- [ODS](#) (OpenDocument Spreadsheet): formato de datos abierto y estándar, se emplea para el almacenamiento de hojas de cálculo que muestran información en celdas organizadas en filas y columnas. Cada una de esas celdas contiene datos o fórmulas con referencias – relativas o absolutas – a otras celdas.
- [PDF](#) (Portable Document Format): formato de tipo compuesto (imagen vectorial, mapa de bits y texto) para el almacenamiento de documentos, desarrollado por la empresa Adobe Systems, muy popular e implementado por diversos fabricantes de software.
- [RDF](#) (Resource Description Framework): no es un formato en sí mismo, sino que define un framework de metadatos, como lenguaje para representar la información en la web. Es un modelo universal que permite intercambiar y enlazar a través de diferentes aplicaciones datos y recursos sin que pierdan su significado, lo que facilita la reutilización y el enriquecimiento de los recursos en la web. RDF se representa a través de diversas serializaciones, siendo la más habitual la que emplea la notación XML/RDF.
- [RSS](#) (Really Simple Syndication): es un vocabulario que permite la catalogación de información de manera que sea posible encontrar información precisa adaptada a las preferencias de los/as usuarios/as. Los archivos RSS contienen metadatos sobre fuentes de información especificadas por los/as usuarios/as cuya función

principal es avisarles de que los recursos que ellos/as seleccionaron para formar parte de ese RSS fueron actualizados sin necesidad de comprobar directamente la fuente, es decir, notifican de forma automática cualquier cambio que se realice en esos recursos.

- [SHP \(Shapefile\)](#), [SHX \(Shape Index\)](#), [DBJ \(Base de datos en formato dBASE\)](#) y [PRJ \(Project description\)](#): El formato ESRI Shapefile es un formato de archivo informático propietario de datos espaciales desarrollado por la compañía ESRI, quien crea y comercializa software para Sistemas de Información Geográfica como Arc/Info o ArcGIS.
- [SOAP](#) (Simple Object Access Protocol): es uno de los protocolos abiertos utilizados en los servicios web, definiendo cómo dos objetos en diferentes procesos pueden comunicarse intercambiando datos XML.
- [WFS](#) (Web Feature Service): servicio web que permite interactuar con los mapas servidos por WMS utilizando el lenguaje GML, derivado de XML. Permite recuperar datos vectoriales y la información alfanumérica ligada a los mismos y realizar consultas tanto espaciales como alfanuméricas. Su especificación está recogida en OGC (Open Geospatial Consortium).
- [WMS](#) (Web Map Service): servicio web en formato abierto estándar que produce vía http mapas de datos referenciados espacialmente, de forma dinámica a partir de información geográfica. Su especificación está recogida en OGC (Open Geospatial Consortium).
- [XLS y XLSX](#) son formatos en propiedad de Microsoft y utilizados en la popular aplicación de hojas de cálculo Microsoft Excell. Al igual que el formato ODS, estos formatos también almacenan hojas de cálculos basadas en celdas y fueron los antecesores del formato ODS.
- [XML](#) (eXtensible Markup Language): es un metalenguaje extensible de etiquetas, entendido como una manera de definir lenguajes para diferentes necesidades, y particularmente para el intercambio de información estructurada entre diferentes plataformas. Describe los datos de forma que, empleando etiquetas, sea posible estructurarlos, tal como lo hace HTML, pero que en este caso no están predefinidas, sino que podemos definir nuevos vocabularios para cubrir nuestras necesidades; por ejemplo, KML (Keyhole Markup Language) para la creación de

modelos y el almacenamiento de funciones geográficas, que utilizarán principalmente en aplicaciones de geoposicionamiento y mapas.

Estos archivos pueden ser descargados de manera manual de los sitios web donde están alojados o a través de una API (Application Programming Interfaces). Se denomina API a un conjunto de operaciones con el que un software puede ser utilizado por otro sistema ajeno. La API funciona como una capa intermedia entre un sistema software y el exterior. Esta capa permite la comunicación mediante funciones y procedimientos entre softwares diferentes. En este contexto de Datos Abiertos, una API es el mecanismo por el cual podemos descargar los datos a través de operaciones que podemos automatizar. De esta forma, no dependemos de procesos manuales y podemos hacer extracciones de datos de manera periódica. El uso de una API suele estar documentado con todas las operaciones que se pueden realizar, como es el caso de la API del portal de datos del Gobierno de España [<https://datos.gob.es/es/accessible-apidata>] y el del Instituto Canario de Estadística [<https://datos.canarias.es/api/estadisticas/>]. En ambos casos, las operaciones de consultas se realizan a través de operaciones GET, un método de peticiones del protocolo HTTP. Y las mencionadas operaciones GET se utilizan localizando los datos a través de su URI (Identificador Uniforme de Recursos). La forma de utilizar la operación GET con los URI concretos se detallan en la documentación de cada API.

Para finalizar este apartado, muestro una breve selección de los portales de Datos Abiertos disponibles en Internet:

- La Unión Europea publica datos de ámbito internacional, europeo, nacional, regional y local en el portal [Data.Europa.eu](https://data.europa.eu).
- Los gobiernos de países como Estados Unidos [<https://www.data.gov/>] y Reino Unido [<https://data.gov.uk/>] tienen páginas web con acceso a sus Datos Abiertos. También el gobierno de España se suma a esta iniciativa con el portal <https://datos.gob.es/es>.
- Los gobiernos autonómicos también han apostado por la iniciativa de publicar sus datos, como el [Gobierno de Navarra](#) y el [Gobierno de Canarias](#).
- También se han sumado algunos ayuntamientos como [el de Madrid](#) y [el de Barcelona](#).
- [La plataforma Kaggle](#) es un punto de encuentro para estudiantes y profesionales de la ciencias de datos y del aprendizaje automático. Esta plataforma permite

compartir trabajos sobre conjuntos y modelos de datos. Para este cometido, la plataforma ofrece un catálogo de datasets con los que poder trabajar.

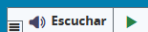
- Google, a través de su servicio Google Cloud, ofrece datasets de datos públicos para ser utilizados directamente en [su plataforma](#).
- En España también existen entidades privadas que recolectan datos y los ofrecen en sus páginas webs. Es el caso de la [Fundación CTIC Centro Tecnológico de Gijón](#) que ofrece datos propios y la empresa Esri España que recopila datos públicos que han sido tratados por ellos y por [tecnología Esri](#).

2.8. DATOS ABIERTOS Y DATA WAREHOUSE

Las fuentes de Datos Abiertos pueden ser muy útiles si se saben utilizar. En la sección anterior se mencionan algunos de los muchos proyectos que existen basados en Datos Abiertos. Sin embargo, este TFG pretende explorar, de manera velada, la combinación entre la utilización de Datos Abiertos y la construcción de un Data Warehouse. Esta combinación podría potenciar el estudio y el análisis de la información de esos datos.

Supongamos posibles usos de estos dos campos de manera conjunta: un partido político que quiere elaborar un programa electoral con medidas sacadas de un estudio serio basado en indicadores como empleo, paro, políticas municipales y presupuesto ejecutado; un grupo de inversores que quieren establecerse en un lugar determinado y usan los Datos Abiertos para realizar un estudio sobre la rentabilidad de sus inversiones en esa zona; una entidad que quiera evaluar las políticas y ofertas de formación y contrastarlo con el nivel de paro. Estos son solo algunos supuestos casos de todas las posibilidades que se presentan si se profundiza en el análisis de los Datos Abiertos.

Para este TFG, se realiza un almacenamiento de datos sobre el tipo de turistas que visitan las Islas Canarias durante los tres últimos años. Esta información nos permitirá tener un conocimiento profundo sobre los distintos perfiles de turistas que visitan el archipiélago. Los datos utilizados provienen del [portal de Datos Abiertos del Gobierno de Canarias](#), un catálogo de datos centralizados y públicos de las administraciones públicas canarias y del Instituto Canario de Estadística (ISTAC).



Datos Abiertos de Canarias

Punto de encuentro para la publicación de datos del sector público canario en formatos abiertos, gratuitos y reutilizables

Buscar



7.655 conjuntos de datos



15 organizaciones

Figura 17: Portal de Datos Abiertos de Canarias.

Fuente: <https://datos.canarias.es/>

CAPÍTULO 3

3.1. DESCRIPCIÓN DEL PROYECTO

El proyecto objeto de este Trabajo de Fin de Grado consiste en la creación de un Data Warehouse con información pública de los Datos Abiertos del Gobierno de Canarias. Este Data Warehouse particular va a almacenar datos sobre un tema concreto: el número de turistas que viajan a las Islas Canarias desde el año 2018 hasta el 2020 y clasificados por varios parámetros: nacionalidad, isla que visitan, sexo, edad, nivel de ingresos, etc. Estos datos se han descargado del portal donde se encuentran publicados y se almacenan en el Data Warehouse mediante procesos ETL. Los procesos ETL juegan un papel fundamental porque habilitan un flujo de datos y un tratamiento sobre los mismos desde su origen hasta los diferentes Datamart, y desde estos hasta el Data Warehouse.

En la siguiente figura se muestra de manera gráfica las partes de este proyecto:

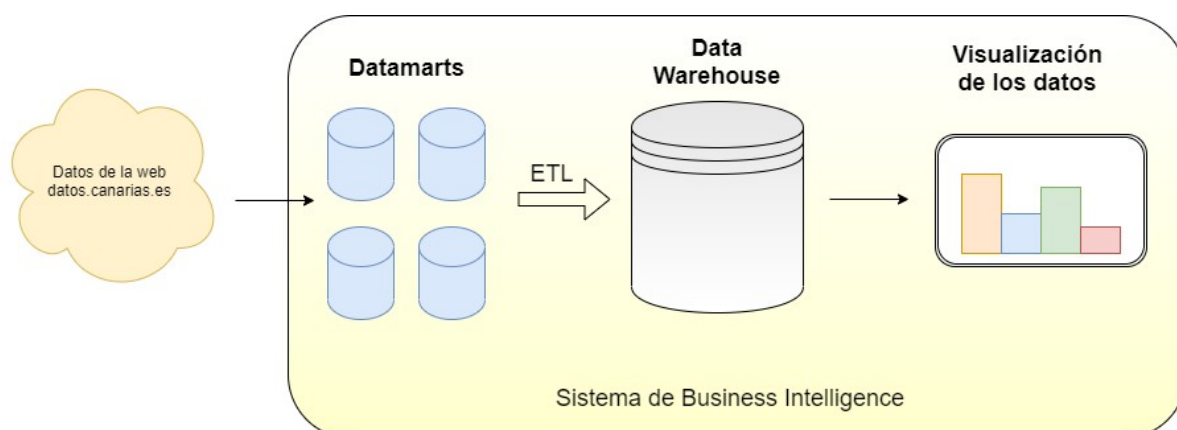


Figura 18: Esquema del proyecto.

- La fuente de datos serán los datos públicos de la web datos.canarias.es
- Los Datamarts, el Data Warehouse y la visualización de los datos forman un sistema de Business Intelligence.
- Las bases de datos que forman los Datamarts y el Data Warehouse son servidas por SQL Server y gestionadas a través del entorno de SQL Server Management Studio.
- Los procesos ETL están desarrollados con SQL Server Integration Services.

- La visualización de datos está creada en Power BI Desktop.

Antes de empezar con el desarrollo de este proyecto, hay que realizar la selección de los datos que se van a utilizar, diseñar el esquema del Data Warehouse con un modelo de datos y especificar los procesos ETL necesarios.

3.1.1. SELECCIÓN DE DATOS ABIERTOS

El objetivo del Data Warehouse de este proyecto es almacenar información sobre los turistas que viajan a las Islas Canarias. Esta información se encuentra publicada en la sección de Turismo de la web de Datos Abiertos del Gobierno de Canarias.



Figura 19: Conjunto de datos sobre Turismo del portal de Datos Abiertos de Canarias.

Fuente: <https://datos.canarias.es/catalogos/general/group/turismo>

La selección de los Datos Abiertos se centra en información relacionada con parámetros y características de los turistas. Y de toda la información disponible, se han seleccionado los siguientes conjuntos de datos:

- [Turistas de 16 y más años según las personas que le acompañan en el viaje por NUTS1 de residencia y periodos.](#)
- [Turistas según niveles de ingresos por tipos de alojamiento. Canarias y periodos.](#)

- [Turistas según grupos de edad y sexos por nacionalidades. Canarias y periodos](#)
- [Turistas de 16 y más años según sexos por NUTS1 de residencia, islas de Canarias y periodos.](#)
- [Turistas según grupos de edad y sexos por tipos de alojamiento. Canarias y periodos.](#)
- [Turistas según niveles de ingresos por tipos de alojamiento. Islas de Canarias y periodos.](#)
- [Turistas según situación laboral por países de residencia. Canarias y periodos.](#)
- [Turistas según niveles de ingresos por países de residencia. Islas de Canarias y periodos.](#)
- [Turistas según grupos de edades y sexos por nacionalidades. Islas de Canarias y periodos.](#)
- [Turistas según canales de información para organizar el viaje por países de residencia. Municipios turísticos de Canarias y periodos.](#)

Estos conjuntos de datos están disponibles en varios formatos como HTML, PC-axis y JSON. Y de estos datos se define el modelo dimensional que albergará nuestro Data Warehouse.

3.1.2. MODELADO DIMENSIONAL: TABLAS FACT Y TABLAS DIM

Como ya se ha introducido en el capítulo 2 sobre el modelo de datos y los tipos de tablas, establecemos la estructura de las tablas que conforman las bases de datos del Data Warehouse.

El turista es el objeto de estudio de nuestro Data Warehouse, por lo que la tabla Fact Tourist almacena los datos de los diferentes tipos de turistas que visita las Islas Canarias.

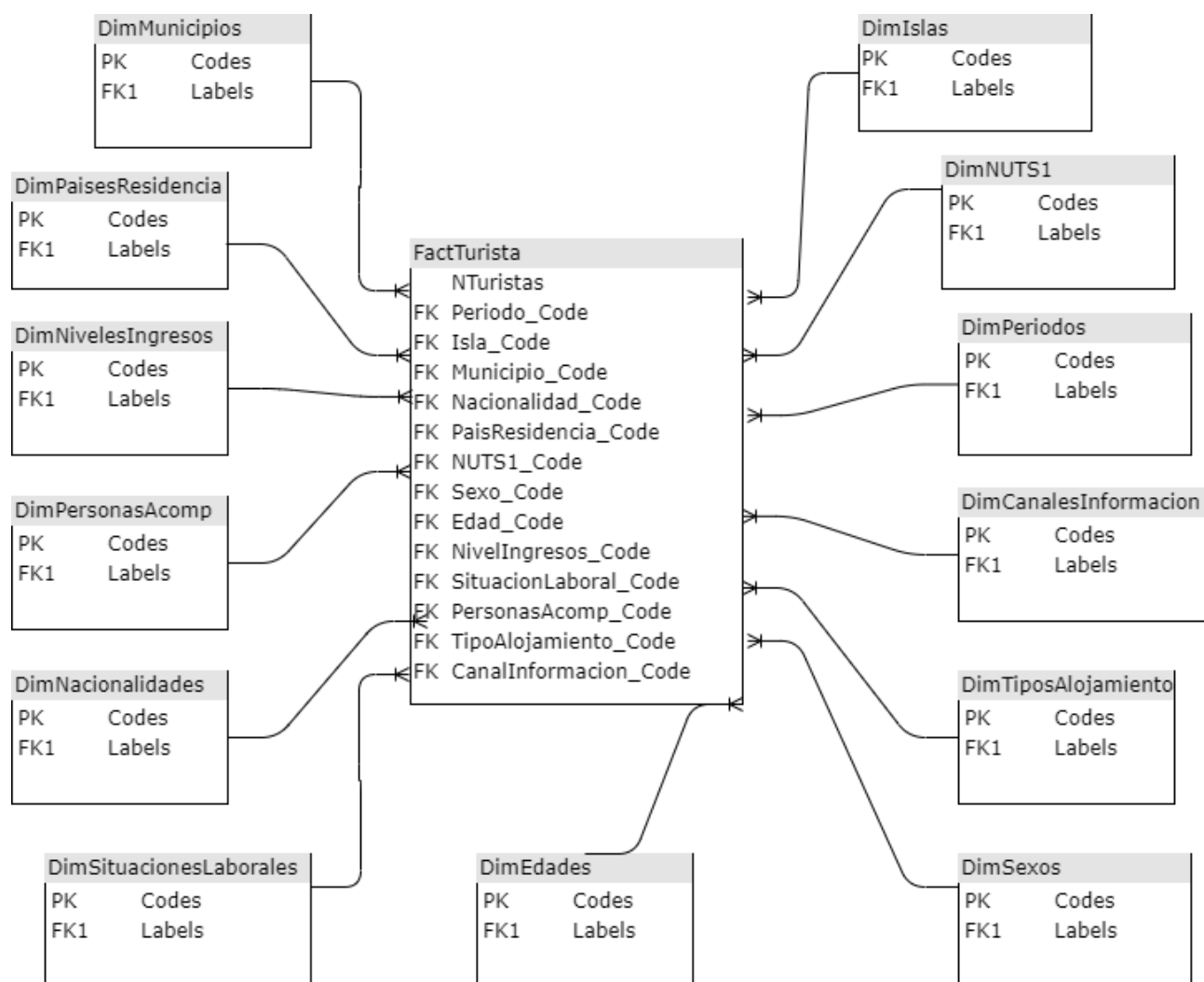


Figura 20: Esquema de estrella del Data Warehouse.

La tabla fact del turista contiene, además de los diferentes perfiles de turistas que visitan las Islas Canarias, la cantidad de cada tipo de perfil de turista, el periodo en el que se ha realizado el viaje y la isla visitada. Las características de cada tipo de turistas vienen dadas por la combinación de los valores que hacen referencias a la tablas dimensionales.

FactTurista	Tabla de turista (tabla fact)	Tabla Dim de referencia
NTuristas	Cantidad de turistas analizados.	-
Periodo_Code	Periodo en año y/o cuatrimestre en el que ha viajado el turista a Canarias.	DimPeriodos
Isla_Code	Isla visitada por el turista.	DimIslas
Municipio_Code	Municipio donde se aloja el turista.	DimMunicipios
Nacionalidad_Code	Nacionalidad del turista.	DimNacionalidades
PaisResidencia_Code	País de procedencia del turista.	DimPaíses
NUTS1_Code	Código de Nomenclatura de las Unidades Territoriales Estadísticas.	DimNUTS1
Sexo_Code	Género del turista.	DimSexos
Edad_Code	Grupo de edad del turista.	DimEdades
NivelIngresos_Code	Nivel del poder adquisitivo del turista.	DimNivelesIngresos
SituacionLaboral_Code	Ocupación del turista.	DimSituacionesLaborales
PersonasAcomp_Code	Tipo de relación de los acompañantes con el turista.	DimPersonasAcomp
TipoAlojamiento_Code	Tipo y categoría del alojamiento donde se hospeda el turista.	DimTiposAlojamiento
CanalInformacion_Code	Tipo de canal de información por el cual el turista organiza el viaje.	DimCanalesInformacion

Tabla FactTurista.

Las tablas Dim representan las diferentes dimensiones de la tabla Fact. En este caso, las tablas Dim son los atributos que representan los diferentes perfiles de los turistas:

DimIslas	DimPersonasAcomp	DimCanalesInformacion
CANARIAS	TOTAL	Visitas anteriores a Canarias
Fuerteventura	Solo	Otros canales
Gran Canaria	Pareja	Amigos o familiares
La Palma	Hijos	Internet o redes sociales
Lanzarote	Otros familiares	Touroperador o agencia de viajes
Tenerife	Otros acompañantes	
DimNUTS1	DimPeriodos	DimMunicipios
ALEMANIA	2018	CANARIAS
Otras regiones de Alemania	2018 Primer trimestre	FUERTEVENTURA
Baden-Württemberg	2018 Segundo trimestre	Antigua
Baviera	2018 Tercer trimestre	La Oliva
Berlín	2018 Cuarto trimestre	Pájara
Brandeburgo	2019	Puerto del Rosario
Hamburgo	2019 Primer trimestre	Otros municipios no turísticos de Fuerteventura
Hesse	2019 Segundo trimestre	GRAN CANARIA
Baja Sajonia	2019 Tercer trimestre	Mogán
Renania del Norte-Westfalia	2019 Cuarto trimestre	Las Palmas de Gran Canaria
Renania-Palatinado	2020	San Bartolomé de Tirajana
Sajonia	2020 Primer trimestre	Otros municipios no turísticos de Gran Canaria
Sajonia-Anhalt	2020 Segundo trimestre	LA PALMA
Schleswig-Holstein	2020 Tercer trimestre	Breña Baja
REINO UNIDO	2020 Cuarto trimestre	Fuencaliente de La Palma
Otras regiones de Reino Unido		Santa Cruz de La Palma
Noreste	DimNacionalidades	Otros municipios no turísticos de La Palma
Noroeste	TOTAL	LANZAROTE
Yorkshire y Humberside	Belga	Arrecife
Midlands Orientales (Inglaterra)	Suiza	Teguiise
Midlands Occidentales	Alemana	Tías

(Inglaterra)	Nórdica	Yaiza
Este de Inglaterra	Española	Otros municipios no turísticos de Lanzarote
Londres	Francesa	TENERIFE
Sureste (UK)	Británica	Adeje
Suroeste (UK)	Irlandesa	Arona
Gales	Italiana	Puerto de la Cruz
Escocia	Holandesa	Santa Cruz de Tenerife
Irlanda del Norte	Otras nacionalidades	Santiago del Teide
		Otros municipios no turísticos de Tenerife

DimPaíses	DimNivelesIngresos	DimEdades
TOTAL	TOTAL	TOTAL GRUPOS DE EDADES
Alemania	Menos de 25.000€	De 16 a 44 años
España	25.000€ - 49.999€	Mayor de 44 años
Reino Unido	50.000€ - 74.999€	
Otros países	75.000€ o más	
		DimSexos
		TOTAL GRUPOS DE EDADES
		De 16 a 44 años
		Mayor de 44 años
DimTiposAlojamiento	DimSituacionesLaborales	
TOTAL	TOTAL	
Hotel de 5 estrellas y 5 GL	Asalariado, ayuda negocio familiar o prácticas remuneradas	
Hotel de 4 estrellas	Empresario con o sin asalariados	
Hotel de 1, 2, 3 estrellas	Estudiante, desempleado o labores del hogar, cuidado de niños o dependientes	
Apartahotel o villa turística	Jubilado, prejubilado o ha cesado su actividad laboral	
Otros establecimientos colectivos (alojamiento rural, crucero, camping, etc.)	Otros tipos de inactividad (rentista, servicio militar obligatorio, etc.)	
Vivienda o habitación alquilada a un particular		
Vivienda propia o vivienda de amigos o familiares o intercambio gratuito de vivienda u otros alojamientos privados		

Tablas Dimensionales.

3.2. ALCANCE DEL PROYECTO

Como se ha descrito en los capítulos anteriores, el Data Warehouse es el elemento principal de un sistema de Business Intelligence. Sin embargo, este proyecto se centra en el diseño y la construcción de las bases de datos que componen el Data Warehouse, los procesos ETL que alimentarán de información al Data Warehouse y la selección de conjuntos de los Datos Abiertos como fuente de información. También se implementa un dashboard compuesto por varias visualizaciones de los de datos extraídos del Data Warehouse. La figura 18 de la sección 3.1 de la descripción del proyecto muestra de manera visual lo que abarca el desarrollo de este trabajo de fin de grado.

3.3. PLANIFICACIÓN

La realización de este proyecto TFG se ha dividido en los siguientes pasos:

1. **Formación, investigación y documentación.** Para este proyecto he necesitado investigar y documentarme acerca de los sistemas de Business Intelligence, el diseño y la implementación de un Data Warehouse, su desarrollo con los servicios de SQL Server y lo relacionado con las iniciativas públicas de Datos Abiertos. También realicé varios cursos sobre desarrollo de Data Warehouse en la plataforma online de LinkedIn Learning. El capítulo 2 es el resultado de este primer paso, donde se establece un contexto teórico. Todas las fuentes de información se recogen en el apartado de la bibliografía de este documento.
2. **Planteamiento y diseño del proyecto.** Con los conocimientos necesarios asimilados por el paso anterior, en este paso se analiza la idea a implementar, se hace un planteamiento del proyecto con el diseño de las tablas del Data Warehouse y los datos que se van a utilizar. En este paso se concreta y se describe el proyecto, explicado en el capítulo 3.
3. **Desarrollo del proyecto.** Se describe con detalle cada una de las fases del desarrollo. Estas fases componen el capítulo 4 de este documento y se dividen en análisis de los conjuntos de datos seleccionados, implementación de los Datamart, implementación de la estructura del Data Warehouse, la implementación de los procesos ETL y la visualización de los datos almacenados.

4. **Cierre del proyecto.** El cierre se produce con la finalización del desarrollo del proyecto, la validación de todo el proyecto y la memoria por parte de la tutora y la finalización de esta memoria, redactada en paralelo con los pasos anteriores. La defensa de este Trabajo de Fin de Grado ante un tribunal también forma parte de este último paso.

CAPÍTULO 4

4.1. ENTORNO DE TRABAJO

Para el desarrollo de este proyecto fue necesario un entorno con las siguientes características:

- Un sistema operativo Windows 10.
- Un servidor de bases de datos como SQL Server 2019.
- Un gestor de bases de datos como SQL Server Management Studio.
- Una instalación del servicio de SQL Integration Services.
- Un entorno de desarrollo como Visual Studio y un editor ligero como Visual Studio Code.
- Una programa para modelar y visualizar datos como Power BI Desktop.

4.2. ANÁLISIS DE LOS CONJUNTOS DE DATOS SELECCIONADOS

En el capítulo 3, se enumeraba los datos seleccionados y extraídos de la plataforma de Datos Abiertos del Gobierno de Canarias. Estos datos alimentan al Data Warehouse con la información que buscamos sobre los turistas que visitan las Islas Canarias.

Cada uno de los datos seleccionados tienen una página propia dentro del portal web de Datos Abiertos del Gobierno de Canarias. En esa página se describe información de los datos como una breve descripción, los distintos formatos en los que se encuentran los datos disponibles y enlaces a los repositorios de esos datos.

En la imagen tomada como ejemplo, se muestra la página donde se encuentra el conjunto de datos sobre turistas según niveles de ingresos por tipos de alojamientos y por periodos.



Figura 21: Página de un conjunto de datos del portal de Datos Abiertos de Canarias.

Fuente: <https://datos.canarias.es/catalogos/general/dataset/turistas-segun-niveles-de-ingresos-por-tipos-de-alojamiento-canarias-y-periodos1-1>

En la sección de *Datos y Recursos* están los tres formatos disponibles de este conjunto: PC-axis, HTML y JSON. Para acceder a cada uno de ellos, basta con pulsar el botón azul *Explorar e Ir al recurso*.

Si optamos por el formato PC-axis o el formato JSON, se descarga un archivo con el formato elegido. Ambos formatos presentan el conjunto de datos de manera estructurada en un archivo que podemos manejar.

Con el formato HTML, el conjunto de datos se presenta en un panel con categorías para configurar las consultas sobre el conjunto de datos. Este panel está alojado en la web del Instituto Canario de Estadística (ISTAC).

Está en: Inicio > Estadísticas > Selección de categorías



Turistas según niveles de ingresos por tipos de alojamiento. Canarias y periodos.

Esta tabla se publica en: EGT / Series trimestrales de perfil del turista. Islas de Canarias. 2018-2021 (Metodología 2018)

► **Seleccione categorías a consultar**

[Consejos para seleccionar](#)

Niveles de ingresos

Seleccionadas 0 Total 5

TOTAL
Menos de 25.000€
25.000€ - 49.999€
50.000€ - 74.999€
75.000€ o más

Tipos de alojamiento

Seleccionadas 0 Total 8

TOTAL
Hotel de 5 estrellas y 5 GL
Hotel de 4 estrellas
Hotel de 1, 2, 3 estrellas
Apartahotel o villa turística
Vivienda o habitación alquilada a un particular
Vivienda propia o vivienda de amigos o familiares o inten

Periodos

Seleccionadas 0 Total 16

2021 Primer trimestre
2020
2020 Cuarto trimestre
2020 Tercer trimestre
2020 Segundo trimestre
2020 Primer trimestre
2019

► **Elija cómo quiere visualizar sus datos**

Variables en filas

Niveles de ingresos
Tipos de alojamiento

Variables en columnas

Periodos

Obtendrá como resultado de la consulta 0 celdas

Figura 22: Panel de consultas del ISTAC de un conjunto de datos.

Fuente: <http://www.gobiernodecanarias.org/istac/jaxi-istac/tabla.do?uripx=urn:uuid:c68b558f-69e3-44a3-9a0a-e0aa0ad7660d>

Este panel presenta los diferentes parámetros para acotar la información a mostrar del conjunto de datos. Podemos hacer una selección personalizada o mostrar todos los datos, tal y como se muestra en la siguiente imagen.

istac INSTITUTO CANARIO DE ESTADÍSTICA

Sede electrónica | Contacto

ESTADÍSTICAS EL ISTAC NOTICIAS DATOS ABIERTOS COVID-19

texto de búsqueda BUSCAR

Está en: Inicio > Estadísticas > Selección de categorías > Resultados

Turistas según niveles de ingresos por tipos de alojamiento. Canarias y periodos.

Esta tabla se publica en: EGT / Series trimestrales de perfil del turista. Islas de Canarias. 2018-2021 (Metodología 2018)

Información general de la tabla

Unidad de medida:	Turistas	Periodo de referencia:	Trimestre
Tipo de dato:	-	Periodo base:	-
A precios:	-	Datos ajustados:	-
Ajuste estacional:	-		
Notas:	Mostrar notas a pie de tabla		

Tabla de resultados

	2021 Primer trimestre	2020	2020 Cuarto trimestre	2020 Tercer trimestre	2020 Segundo trimestre	2020 Primer trimestre
TOTAL						
TOTAL	357.223	4.110.603	605.530	763.359
Hotel de 5 estrellas y 5 GL	45.569	404.097	79.518	69.592
Hotel de 4 estrellas	127.128	1.558.951	225.034	306.734
Hotel de 1, 2, 3 estrellas	37.657	466.447	57.483	70.294
Apartahotel o villa turística	57.200	785.520	86.346	103.472
Vivienda o habitación alquilada	25.239	263.744	38.477	63.750
Vivienda propia o vivienda de	47.267	403.191	88.101	108.047
Otros establecimientos colecti	17.163	228.652	30.571	41.470
Menos de 25.000€						
TOTAL	57.459	629.575	99.121	166.200
Hotel de 5 estrellas y 5 GL	3.521	31.817	4.921	9.278
Hotel de 4 estrellas	16.377	194.114	29.995	58.728
Hotel de 1, 2, 3 estrellas	5.616	65.776	9.196	16.395
Apartahotel o villa turística	10.436	121.598	15.184	22.359
Vivienda o habitación alquilada	5.325	50.751	7.011	16.170

Figura 23: Resultado de la consulta del ISTAC de un conjunto de datos.

Fuente: <http://www.gobiernodecanarias.org/istac/jaxi-istac/tabla.do?uripx=urn:uuid:c68b558f-69e3-44a3-9a0a-e0aa0ad7660d>

En la sección *Información Adicional*, tal y como indica el propio título, se especifica más información extra sobre el conjunto de datos como la fecha de creación, la fecha de última actualización, la frecuencia de actualización, el identificador del conjunto, la fecha de modificación y de publicación, entre otras cuestiones.

SOCIAL	Información Adicional	
Twitter	Campo	Valor
Facebook	Última actualización	24 de junio de 2021, 12:03 (UTC+01:00)
LICENCIA	Creado	19 de marzo de 2021, 8:40 (UTC+00:00)
Aviso Legal del ISTAC	Email de contacto	consultas.istac@gobiernodecanarias.org
	Frecuencia	Trimestral
	GUID	https://datos.canarias.es/catalogos/estadisticas/dataset/2398722b-53a7-4c26-b891-4fa844e29ced
	Identificador	http://www.gobiernodecanarias.org/istac/jaxi-istac/tabla.do?uripx=urn:uuid:c68b558f-69e3-44a3-9a0a-e0aa0ad7660d
	Idioma	["es"]
	Modificado	2021-05-17T11:00:00+00:00
	Nombre del publicador	ISTAC
	Publicado	2021-05-18T09:08:45.533000+00:00
	Tema	["http://datos.gob.es/kos/sector-publico/sector/turismo"]
	URI	https://datos.canarias.es/catalogos/estadisticas/dataset/2398722b-53a7-4c26-b891-4fa844e29ced
	URI del publicador	http://datos.gob.es/recurso/sector-publico/org/Organismo/A05003423

Figura 24: Información adicional de un conjunto de datos del portal de Datos Abiertos de Canarias.

Fuente: <https://datos.canarias.es/catalogos/general/dataset/turistas-segun-niveles-de-ingresos-por-tipos-de-alojamiento-canarias-y-periodos1-1>

Para este proyecto, he optado por descargar los archivos en formato JSON y renombrarlos para hacer más fácil su uso.

Nombre	Fecha de modificación	Tipo	Tamaño
Datos1.json	23/05/2021 12:39	Archivo JSON	376 KB
Datos2.json	23/05/2021 12:39	Archivo JSON	56 KB
Datos3.json	23/05/2021 16:54	Archivo JSON	173 KB
Datos4.json	23/05/2021 16:54	Archivo JSON	170 KB
Datos5.json	23/05/2021 16:54	Archivo JSON	109 KB
Datos6.json	23/05/2021 17:06	Archivo JSON	387 KB
Datos7.json	23/05/2021 17:06	Archivo JSON	111 KB
Datos8.json	23/05/2021 17:12	Archivo JSON	245 KB
Datos9.json	23/05/2021 17:12	Archivo JSON	476 KB
Datos10.json	23/05/2021 17:12	Archivo JSON	262 KB

Figura 25: Archivos JSON.

La siguiente tabla explica que archivo JSON se corresponde con cada uno de los conjuntos de datos abiertos seleccionados para crear el Data Warehouse:

Nombre del archivo	Dataset relacionado con enlace a la fuente original
Datos1.json	<u>Turistas de 16 y más años según las personas que le acompañan en el viaje por NUTS1 de residencia y periodos.</u>
Datos2.json	<u>Turistas según niveles de ingresos por tipos de alojamiento. Canarias y periodos.</u>
Datos3.json	<u>Turistas según grupos de edad y sexos por nacionalidades. Canarias y periodos</u>
Datos4.json	<u>Turistas de 16 y más años según sexos por NUTS1 de residencia, islas de Canarias y periodos.</u>
Datos5.json	<u>Turistas según grupos de edad y sexos por tipos de alojamiento. Canarias y periodos.</u>
Datos6.json	<u>Turistas según niveles de ingresos por tipos de alojamiento. Islas de Canarias y periodos.</u>
Datos7.json	<u>Turistas según situación laboral por países de residencia. Canarias y periodos.</u>
Datos8.json	<u>Turistas según niveles de ingresos por países de residencia. Islas de Canarias y periodos.</u>
Datos9.json	<u>Turistas según grupos de edades y sexos por nacionalidades. Islas de Canarias y periodos.</u>
Datos10.json	<u>Turistas según canales de información para organizar el viaje por países de residencia. Municipios turísticos de Canarias y periodos.</u>

Tabla de relaciones en archivos json y datasets .

Los archivos JSON albergan los datos de manera estructurada. Algunos de los campos más importantes son:

- Title: título que denomina el conjunto de datos del archivo JSON.
- Categories: es una colección de variables. Estas variables son las que definen el conjunto de datos. En la siguiente imagen vemos que las variables son “Situaciones laborales”, “Países de residencia” y “Periodos”. Cada variable tiene dos campos:
 - Codes: es el identificador único del valor de la variable.
 - Labels: es la etiqueta que identifica de manera textual al valor de la variable.
- Data: es una colección de datos. Aquí es donde se estructura los valores de los datos.
 - Valor: es el dato que da sentido a las variables y las relaciona entre sí.
 - DimCodes: son los identificadores de las variables, es decir, los valores del campo “codes” de cada variable.

Estos archivos comparten la siguiente estructura de datos:

```

1  {
2    "uuid": "69d1c086-fd06-4418-a8d2-2e40beed1d63",
3    "title": "Turistas según situación laboral por países de residencia. Canarias y periodos.",
4    "uriPx": "urn:uuid:3be13249-f561-496f-82cf-9450e3099d3d",
5    "stub": [
6      "Situaciones laborales",
7      "Países de residencia"
8    ],
9    "heading": [
10     "Periodos"
11   ],
12   "categories": [
13     {
14       "variable": "Situaciones laborales",
15       "codes": [
16         "0",
17         "1",
18         "2",
19         "3",
20         "4",
21         "5"
22       ],
23       "labels": [
24         "TOTAL",
25         "Asalariado, ayuda negocio familiar o prácticas remuneradas",
26         "Empresario con o sin asalariados",
27         "Estudiante, desempleado o labores del hogar, cuidado de niños o dependientes",
28         "Jubilado, prejubilado o ha cesado su actividad laboral",
29         "Otros tipos de inactividad (rentista, servicio militar obligatorio, etc.)"
30       ]
31     },
32     {
33       "variable": "Países de residencia",
34 >     "codes": [ ...
47     ],
48 >     "labels": [ ...
61     ]
62   },
63   {
64     "variable": "Periodos",
65 >     "codes": [ ...
82     ],
83 >     "labels": [ ...
100    ]
101   }
102 ],

```

Figura 26: Estructura del archivo Datos7.json

```

102 → ],
103 → "temporals": [
104 →   "Periodos"
105 → ],
106 → "spatials": [
107 →   "Países de residencia"
108 → ],
109 → "notes": [
110 →   "(*) Dato estimado con menos de 20 observaciones muestrales"
111 → ],
112 → "source": "Instituto Canario de Estadística (ISTAC).",
113 → "surveyCode": "C00028A",
114 → "surveyTitle": "Encuesta sobre el Gasto Turístico",
115 → "publishers": [
116 →   "Instituto Canario de Estadística (ISTAC)"
117 → ],
118 → "data": [
119 →   {
120 →     "Valor": "357223",
121 →     "dimCodes": [
122 →       "0",
123 →       "0",
124 →       "2021Q1"
125 →     ]
126 →   },
127 →   {
128 →     "Valor": "4110603",
129 →     "dimCodes": [
130 →       "0",
131 →       "0",
132 →       "2020"
133 →     ]
134 →   },
135 →   {
136 →     "Valor": "605530",
137 →     "dimCodes": [
138 →       "0",
139 →       "0",
140 →       "2020Q4"
141 →     ]
142 →   },

```

Figura 27: Estructura del archivo Datos7.json

4.3. IMPLEMENTACIÓN DE LOS DATAMARTS

Los Datamarts suelen ser bases de datos más específicas de un área o un tema concreto. En un sistema de Business Intelligence, estas bases de datos juegan un papel de apoyo al Data Warehouse, ya que alimentan al Data Warehouse de información o viceversa, dependiendo de la arquitectura planteada en el sistema.

En el contexto de este proyecto, los Datamarts son las bases de datos que almacenan los datos de cada archivo JSON. Y son los Datamart los que proveen de datos al Data Warehouse a través de los procesos ETL.

Un Datamart tiene la misma estructura que un Data Warehouse, es decir, un esquema de estrella. Este diseño es fácilmente aplicable porque la estructura de los archivos JSON predispone los datos para dividirlos entre tablas dimensionales y tablas de hechos. Sabiendo esto, el esquema de estrella de las tablas está compuesto de la siguiente manera:

- Las tablas dimensionales representan las variables del conjunto de datos. Estas tablas almacenan los códigos de las variables (codes) y las etiquetas de los códigos (labels). Los códigos son las claves primarias de la tabla.

```
"categories": [
  {
    "variable": "Situaciones laborales",
    "codes": [
      "0",
      "1",
      "2",
      "3",
      "4",
      "5"
    ],
    "labels": [
      "TOTAL",
      "Asalariado, ayuda negocio familiar o prácticas remuneradas",
      "Empresario con o sin asalariados",
      "Estudiante, desempleado o labores del hogar, cuidado de niños o dependientes",
      "Jubilado, prejubilado o ha cesado su actividad laboral",
      "Otros tipos de inactividad (rentista, servicio militar obligatorio, etc.)"
    ]
  },
  {
    "variable": "Países de residencia",
    "codes": [ ...
  ],
  "labels": [ ...
  ]
},
]
```

Figura 28: Ejemplo de variables del archivo Datos7.json

- La tabla de hechos es la que almacena los datos (Valor) y las claves primarias de las tablas dimensionales (dimCodes). De esta manera, se establece las relaciones entre las tablas.

```

"data": [
  {
    "Valor": "357223",
    "dimCodes": [
      "0",
      "0",
      "2021Q1"
    ]
  },
  {
    "Valor": "4110603",
    "dimCodes": [
      "0",
      "0",
      "2020"
    ]
  },
  {
    "Valor": "605530",
    "dimCodes": [
      "0",
      "0",
      "2020Q4"
    ]
  },
]

```

Figura 29: Ejemplo de datos del archivo Datos7.json

La creación de cada Datamart se realizó a través de scripts en lenguaje SQL. El proceso de creación descrito paso a paso es el siguiente:

1. Crear la base de datos DatamartN con SQL Server Management, siendo N el número correspondiente del archivo JSON.
2. Crear las tablas dimensiones necesarias. Por cada tabla dimensión:
 - 2.1. Se almacenan los datos del archivo JSON en una variable.
 - 2.2. Se crean dos tablas provisionales, una para almacenar todos los campos Codes de las variables y otra para todos los campos labels.
 - 2.3. Se combinan las dos tablas dimensionales para crear la tabla dimensión definitiva.
 - 2.4. Se establece las claves primarias.

- 2.5. Se borran las tablas provisionales.
3. Crear la tabla de hechos.
 - 3.1. Se extraen los valores del JSON.
 - 3.2. Se añade un valor autogenerado como clave primaria para cada fila de datos.
 - 3.3. Se establecen las claves foráneas relacionándolas con su respectiva tabla dimensión.

Este proceso se repitió diez veces, una vez por cada archivo JSON para generar su correspondiente Datamart.

4.3.1. CREACIÓN DE LAS TABLAS DIMENSIONALES

Para la creación de las tablas dimensionales, usé el siguiente script de SQL.

```
-- Creación de las tablas Dim desde los datos de los archivos JSON de  
Datos Abiertos ---
```

```
USE [Datamart10]
```

```
-- Se almacena los datos del archivo json en la variable @json.
```

```
DECLARE @json nvarchar(max);
```

```
SELECT @json = BulkColumn FROM OPENROWSET (
```

```
    BULK 'C:\Program Files\Microsoft SQL Server\MSSQL15.MSSQLSERVER\  
MSSQL\DATA\JSON_DatosAbiertosCanarias\Datos10.json',
```

```
    SINGLE_CLOB) AS [Json];
```

```
-- Comprueba si el archivo json está bien formateado.
```

```
IF ISJSON(@json) = 1
```

```
BEGIN
```

```
    -- Crea la tabla provisional ProvDim1.
```

```
    CREATE TABLE ProvDim1 (
```

```
        ID INT IDENTITY(1,1),
```

```
        Codes nvarchar(10) NOT NULL PRIMARY KEY)
```

```
    -- Inserta los valores Codes del json en la tabla ProvDim1.
```

```

INSERT INTO ProvDim1(Codes)
SELECT [value]
FROM OPENJSON(@json, '$.categories[3].codes')
    -- El índice [n] nos permite elegir los datos de la tabla
dimensión en el array "categories" del json.

-- Crea la tabla provisional ProvDim2.
CREATE TABLE ProvDim2 (
    ID INT IDENTITY(1,1),
    Labels nvarchar(200))

-- Inserta los valores Codes del json en la tabla ProvDim2.
INSERT INTO ProvDim2(Labels)
SELECT [value]
FROM OPENJSON(@json, '$.categories[3].labels')
    -- El índice [n] nos permite elegir los datos de la tabla
dimensión en el array "categories" del json.

-- Combina las tablas provisionales y crea la tabla dimensión
correspondiente.
SELECT ProvDim1.Codes, ProvDim2.Labels INTO DimPeriodos
    -- Nombre de la tabla Dim a crear: Dim_____
FROM ProvDim1
INNER JOIN ProvDim2
ON ProvDim1.ID = ProvDim2.ID;

-- Establece los valores de la columna Codes como primary keys.
ALTER TABLE DimPeriodos -- Nombre de la tabla Dim a crear: Dim_____
ADD PRIMARY KEY (Codes);

-- Borrar las tablas provisionales.
DROP TABLE ProvDim1, ProvDim2;

END

```


Con este script, creé todas las tablas dimensionales de los Datamart. Los textos resaltados en **amarillo** son los datos que cambiaba para crear todas las tablas implicadas en cada Datamart, como el número del nombre de la base de datos, el índice del array de categories y el nombre de la tabla dimensional.

4.3.2. CREACIÓN DE LAS TABLAS DE HECHOS

Para la creación de las tablas de hechos, usé el siguiente script de SQL.

----- Creación de la tabla fact de Datos1.json en Datamart1. -----

```
USE [Datamart1]
```

---- Crea la tabla fact con los valores del archivo json, insertados en su columna correspondiente.

```
SELECT data.* INTO FactDatos1 FROM OPENROWSET (
    BULK 'C:\Program Files\Microsoft SQL Server\MSSQL15.MSSQLSERVER\
MSSQL\DATA\JSON_DatosAbiertosCanarias\Datos1.json',
    SINGLE_CLOB) AS [Json]
CROSS APPLY OPENJSON ( BulkColumn, '$.data' )
WITH (
    NTuristas nvarchar(200) '$.Valor',
    PersonasAcomp_Code nvarchar(10) '$.dimCodes[0]',
    NUTS1_Code nvarchar(10) '$.dimCodes[1]',
    Periodo_Code nvarchar(10) '$.dimCodes[2]',
    Isla_Code nvarchar(50) '$.dimCodes[3]'
) AS [data]
```

-- Se añade la columna con el valor autogenerado IDFact como clave primaria y se establece el resto de campos como claves ajenas.

```
ALTER TABLE dbo.FactDatos1
    ADD IDFact INT IDENTITY(1,1) NOT NULL PRIMARY KEY,
    FOREIGN KEY (PersonasAcomp_Code) REFERENCES DimPersonasAcomp(Codes),
    FOREIGN KEY (NUTS1_Code) REFERENCES DimNUTS1(Codes),
    FOREIGN KEY (Periodo_Code) REFERENCES DimPeriodos(Codes),
    FOREIGN KEY (Isla_Code) REFERENCES DimIslas(Codes);
```

Este script es para crear la tabla de hechos del Datamart 1. Sin embargo, los scripts para crear las tablas de hechos del resto de Datamarts son similares. Los textos resaltados en **amarillo** son los datos que cambiaba para crear todas las tablas implicadas en cada Datamart, como el número del nombre de la base de datos, las variables del array categories y el nombre de la tabla dimensional. A continuación, los scripts para crear el resto de tablas de hechos:

----- Creación de la tabla fact de Datos2.json en Datamart2. -----

USE [Datamart2]

---- Crea la tabla fact con los valores del archivo json, insertados en su columna correspondiente.

```
SELECT data.* INTO FactDatos2 FROM OPENROWSET (
    BULK 'C:\Program Files\Microsoft SQL Server\MSSQL15.MSSQLSERVER\
MSSQL\DATA\JSON_DatosAbiertosCanarias\Datos2.json',
    SINGLE_CLOB) AS [Json]
CROSS APPLY OPENJSON ( BulkColumn, '$.data' )
WITH (
    NTuristas nvarchar(200) '$.Valor',
    NivelIngresos_Code nvarchar(10) '$.dimCodes[0]',
    TipoAlojamiento_Code nvarchar(10) '$.dimCodes[1]',
    Periodo_Code nvarchar(10) '$.dimCodes[2]'
) AS [data]
```

-- Se añade la columna con el valor autogenerado IDFact como clave primaria y se establece el resto de campos como claves ajenas.

```
ALTER TABLE dbo.FactDatos2
    ADD IDFact INT IDENTITY(1,1) NOT NULL PRIMARY KEY,
    FOREIGN KEY (NivelIngresos_Code) REFERENCES
DimNivelesIngresos(Codes),
    FOREIGN KEY (TipoAlojamiento_Code) REFERENCES
DimTiposAlojamiento(Codes),
    FOREIGN KEY (Periodo_Code) REFERENCES DimPeriodos(Codes);
```

----- Creación de la tabla fact de Datos3.json en Datamart3. -----

```
USE [Datamart3]
```

---- Crea la tabla fact con los valores del archivo json, insertados en su columna correspondiente.

```
SELECT data.* INTO FactDatos3 FROM OPENROWSET (
    BULK 'C:\Program Files\Microsoft SQL Server\MSSQL15.MSSQLSERVER\
MSSQL\DATA\JSON_DatosAbiertosCanarias\Datos3.json',
    SINGLE_CLOB) AS [Json]
CROSS APPLY OPENJSON ( BulkColumn, '$.data' )
WITH (
    NTuristas nvarchar(200) '$.Valor',
    Sexo_Code nvarchar(10) '$.dimCodes[0]',
    Edad_Code nvarchar(10) '$.dimCodes[1]',
    Nacionalidad_Code nvarchar(50) '$.dimCodes[2]',
    Periodo_Code nvarchar(10) '$.dimCodes[3]'
) AS [data]
```

-- Se añade la columna con el valor autogenerado IDFact como clave primaria y se establece el resto de campos como claves ajenas.

```
ALTER TABLE dbo.FactDatos3
    ADD IDFact INT IDENTITY(1,1) NOT NULL PRIMARY KEY,
    FOREIGN KEY (Sexo_Code) REFERENCES DimSexos(Codes),
    FOREIGN KEY (Edad_Code) REFERENCES DimEdades(Codes),
    FOREIGN KEY (Nacionalidad_Code) REFERENCES
DimNacionalidades(Codes),
    FOREIGN KEY (Periodo_Code) REFERENCES DimPeriodos(Codes);
```

----- Creación de la tabla fact de Datos4.json en Datamart4. -----

```
USE [Datamart4]
```

---- Crea la tabla fact con los valores del archivo json, insertados en su columna correspondiente.

```
SELECT data.* INTO FactDatos4 FROM OPENROWSET (
```

```

BULK 'C:\Program Files\Microsoft SQL Server\MSSQL15.MSSQLSERVER\
MSSQL\DATA\JSON_DatosAbiertosCanarias\Datos4.json',
SINGLE_CLOB) AS [Json]
CROSS APPLY OPENJSON ( BulkColumn, '$.data' )
WITH (
    NTuristas nvarchar(200) '$.Valor',
    Sexo_Code nvarchar(10) '$.dimCodes[0]',
    NUTS1_Code nvarchar(10) '$.dimCodes[1]',
    Periodo_Code nvarchar(10) '$.dimCodes[2]',
    Isla_Code nvarchar(50) '$.dimCodes[3]'
) AS [data]

```

-- Se añade la columna con el valor autogenerado IDFact como clave primaria y se establece el resto de campos como claves ajenas.

```

ALTER TABLE dbo.FactDatos4
    ADD IDFact INT IDENTITY(1,1) NOT NULL PRIMARY KEY,
    FOREIGN KEY (Sexo_Code) REFERENCES DimSexos(Codes),
    FOREIGN KEY (NUTS1_Code) REFERENCES DimNUTS1(Codes),
    FOREIGN KEY (Periodo_Code) REFERENCES DimPeriodos(Codes),
    FOREIGN KEY (Isla_Code) REFERENCES DimIslas(Codes);

```

----- Creación de la tabla fact de Datos5.json en Datamart5. -----

```

USE [Datamart5]

```

---- Crea la tabla fact con los valores del archivo json, insertados en su columna correspondiente.

```

SELECT data.* INTO FactDatos5 FROM OPENROWSET (
    BULK 'C:\Program Files\Microsoft SQL Server\MSSQL15.MSSQLSERVER\
MSSQL\DATA\JSON_DatosAbiertosCanarias\Datos5.json',
    SINGLE_CLOB) AS [Json]
CROSS APPLY OPENJSON ( BulkColumn, '$.data' )
WITH (
    NTuristas nvarchar(200) '$.Valor',
    Sexo_Code nvarchar(10) '$.dimCodes[0]',

```

```

        Edad_Code nvarchar(10) '$.dimCodes[1]',
        TipoAlojamiento_Code nvarchar(10) '$.dimCodes[2]',
        Periodo_Code nvarchar(10) '$.dimCodes[3]'
    ) AS [data]

```

-- Se añade la columna con el valor autogenerado IDFact como clave primaria y se establece el resto de campos como claves ajenas.

```

ALTER TABLE dbo.FactDatos5
    ADD IDFact INT IDENTITY(1,1) NOT NULL PRIMARY KEY,
    FOREIGN KEY (Sexo_Code) REFERENCES DimSexos(Codes),
    FOREIGN KEY (Edad_Code) REFERENCES DimEdades(Codes),
    FOREIGN KEY (TipoAlojamiento_Code) REFERENCES
DimTiposAlojamiento(Codes),
    FOREIGN KEY (Periodo_Code) REFERENCES DimPeriodos(Codes);

```

----- Creación de la tabla fact de Datos6.json en Datamart6. -----

```

USE [Datamart6]

```

---- Crea la tabla fact con los valores del archivo json, insertados en su columna correspondiente.

```

SELECT data.* INTO FactDatos6 FROM OPENROWSET (
    BULK 'C:\Program Files\Microsoft SQL Server\MSSQL15.MSSQLSERVER\
MSSQL\DATA\JSON_DatosAbiertosCanarias\Datos6.json',
    SINGLE_CLOB) AS [Json]
CROSS APPLY OPENJSON ( BulkColumn, '$.data' )
WITH (
    NTuristas nvarchar(200) '$.Valor',
    NivelIngresos_Code nvarchar(10) '$.dimCodes[0]',
    TipoAlojamiento_Code nvarchar(10) '$.dimCodes[1]',
    Isla_Code nvarchar(50) '$.dimCodes[2]',
    Periodo_Code nvarchar(10) '$.dimCodes[3]'
) AS [data]

```

-- Se añade la columna con el valor autogenerado IDFact como clave primaria y se establece el resto de campos como claves ajenas.

```
ALTER TABLE dbo.FactDatos6
    ADD IDFact INT IDENTITY(1,1) NOT NULL PRIMARY KEY,
    FOREIGN KEY (NivelIngresos_Code) REFERENCES
DimNivelesIngresos(Codes),
    FOREIGN KEY (TipoAlojamiento_Code) REFERENCES
DimTiposAlojamiento(Codes),
    FOREIGN KEY (Isla_Code) REFERENCES DimIslas(Codes),
    FOREIGN KEY (Periodo_Code) REFERENCES DimPeriodos(Codes);
```

----- Creación de la tabla fact de Datos7.json en Datamart7. -----

```
USE [Datamart7]
```

---- Crea la tabla fact con los valores del archivo json, insertados en su columna correspondiente.

```
SELECT data.* INTO FactDatos7 FROM OPENROWSET (
    BULK 'C:\Program Files\Microsoft SQL Server\MSSQL15.MSSQLSERVER\
MSSQL\DATA\JSON_DatosAbiertosCanarias\Datos7.json',
    SINGLE_CLOB) AS [Json]
CROSS APPLY OPENJSON ( BulkColumn, '$.data' )
WITH (
    NTuristas nvarchar(200) '$.Valor',
    SituacionLaboral_Code nvarchar(10) '$.dimCodes[0]',
    PaisResidencia_Code nvarchar(50) '$.dimCodes[1]',
    Periodo_Code nvarchar(10) '$.dimCodes[2]'
) AS [data]
```

-- Se añade la columna con el valor autogenerado IDFact como clave primaria y se establece el resto de campos como claves ajenas.

```
ALTER TABLE dbo.FactDatos7
    ADD IDFact INT IDENTITY(1,1) NOT NULL PRIMARY KEY,
    FOREIGN KEY (SituacionLaboral_Code) REFERENCES
DimSituacionesLaborales(Codes),
```

```

        FOREIGN KEY (PaisResidencia_Code) REFERENCES
DimPaísesResidencia(Codes),
        FOREIGN KEY (Periodo_Code) REFERENCES DimPeriodos(Codes);

----- Creación de la tabla fact de Datos8.json en Datamart8. -----
USE [Datamart8]

---- Crea la tabla fact con los valores del archivo json, insertados en
su columna correspondiente.
SELECT data.* INTO FactDatos8 FROM OPENROWSET (
    BULK 'C:\Program Files\Microsoft SQL Server\MSSQL15.MSSQLSERVER\
MSSQL\DATA\JSON_DatosAbiertosCanarias\Datos8.json',
    SINGLE_CLOB) AS [Json]
CROSS APPLY OPENJSON ( BulkColumn, '$.data' )
WITH (
    NTuristas nvarchar(200) '$.Valor',
    NivelIngresos_Code nvarchar(10) '$.dimCodes[0]',
    PaisResidencia_Code nvarchar(50) '$.dimCodes[1]',
    Isla_Code nvarchar(50) '$.dimCodes[2]',
    Periodo_Code nvarchar(10) '$.dimCodes[3]'
) AS [data]

-- Se añade la columna con el valor autogenerado IDFact como clave
primaria y se establece el resto de campos como claves ajenas.
ALTER TABLE dbo.FactDatos8
    ADD IDFact INT IDENTITY(1,1) NOT NULL PRIMARY KEY,
        FOREIGN KEY (NivelIngresos_Code) REFERENCES
DimNivelesIngresos(Codes),
        FOREIGN KEY (PaisResidencia_Code) REFERENCES
DimPaísesResidencia(Codes),
        FOREIGN KEY (Isla_Code) REFERENCES DimIslas(Codes),
        FOREIGN KEY (Periodo_Code) REFERENCES DimPeriodos(Codes);

```

----- Creación de la tabla fact de Datos9.json en Datamart9. -----

USE [Datamart9]

---- Crea la tabla fact con los valores del archivo json, insertados en su columna correspondiente.

```
SELECT data.* INTO FactDatos9 FROM OPENROWSET (
    BULK 'C:\Program Files\Microsoft SQL Server\MSSQL15.MSSQLSERVER\
MSSQL\DATA\JSON_DatosAbiertosCanarias\Datos9.json',
    SINGLE_CLOB) AS [Json]
CROSS APPLY OPENJSON ( BulkColumn, '$.data' )
WITH (
    NTuristas nvarchar(200) '$.Valor',
    Edad_Code nvarchar(10) '$.dimCodes[0]',
    Sexo_Code nvarchar(10) '$.dimCodes[1]',
    Nacionalidad_Code nvarchar(50) '$.dimCodes[2]',
    Isla_Code nvarchar(10) '$.dimCodes[3]',
    Periodo_Code nvarchar(10) '$.dimCodes[4]'
) AS [data]
```

-- Se añade la columna con el valor autogenerado IDFact como clave primaria y se establece el resto de campos como claves ajenas.

ALTER TABLE dbo.FactDatos9

```
ADD IDFact INT IDENTITY(1,1) NOT NULL PRIMARY KEY,
    FOREIGN KEY (Edad_Code) REFERENCES DimEdades(Codes),
    FOREIGN KEY (Sexo_Code) REFERENCES DimSexos(Codes),
    FOREIGN KEY (Nacionalidad_Code) REFERENCES
DimNacionalidades(Codes),
    FOREIGN KEY (Isla_Code) REFERENCES DimIslas(Codes),
    FOREIGN KEY (Periodo_Code) REFERENCES DimPeriodos(Codes);
```

----- Creación de la tabla fact de Datos10.json en Datamart10. -----

USE [Datamart10]

---- Crea la tabla fact con los valores del archivo json, insertados en su columna correspondiente.

```
SELECT data.* INTO FactDatos10 FROM OPENROWSET (
    BULK 'C:\Program Files\Microsoft SQL Server\MSSQL15.MSSQLSERVER\
MSSQL\DATA\JSON_DatosAbiertosCanarias\Datos10.json',
    SINGLE_CLOB) AS [Json]
CROSS APPLY OPENJSON ( BulkColumn, '$.data' )
WITH (
    NTuristas nvarchar(200) '$.Valor',
    CanalInformacion_Code nvarchar(10) '$.dimCodes[0]',
    PaisResidencia_Code nvarchar(50) '$.dimCodes[1]',
    Municipio_Code nvarchar(50) '$.dimCodes[2]',
    Periodo_Code nvarchar(10) '$.dimCodes[3]'
) AS [data]
```

-- Se añade la columna con el valor autogenerado IDFact como clave primaria y se establece el resto de campos como claves ajenas.

```
ALTER TABLE dbo.FactDatos10
    ADD IDFact INT IDENTITY(1,1) NOT NULL PRIMARY KEY,
    FOREIGN KEY (CanalInformacion_Code) REFERENCES
DimCanalesInformacion(Codes),
    FOREIGN KEY (PaisResidencia_Code) REFERENCES
DimPaisesResidencia(Codes),
    FOREIGN KEY (Municipio_Code) REFERENCES DimMunicipios(Codes),
    FOREIGN KEY (Periodo_Code) REFERENCES DimPeriodos(Codes);
```

Una vez realizados y ejecutados todos los scripts, tenemos los Datamarts creados.

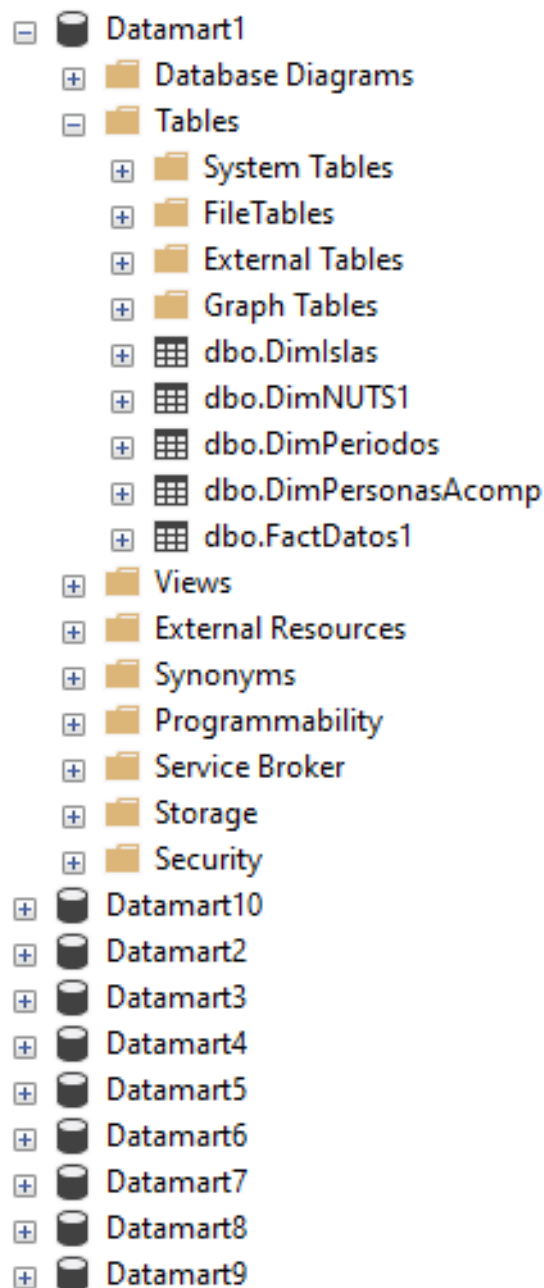


Figura 30: Datamarts creados en SQL Server Management.

4.4. IMPLEMENTACIÓN DE LA ESTRUCTURA DEL DATAWAREHOUSE

Nuestro Data Warehouse está implementado como una base de datos cuyas tablas tienen el esquema de estrella que se describe en la sección 3.1.2 del capítulo 3.

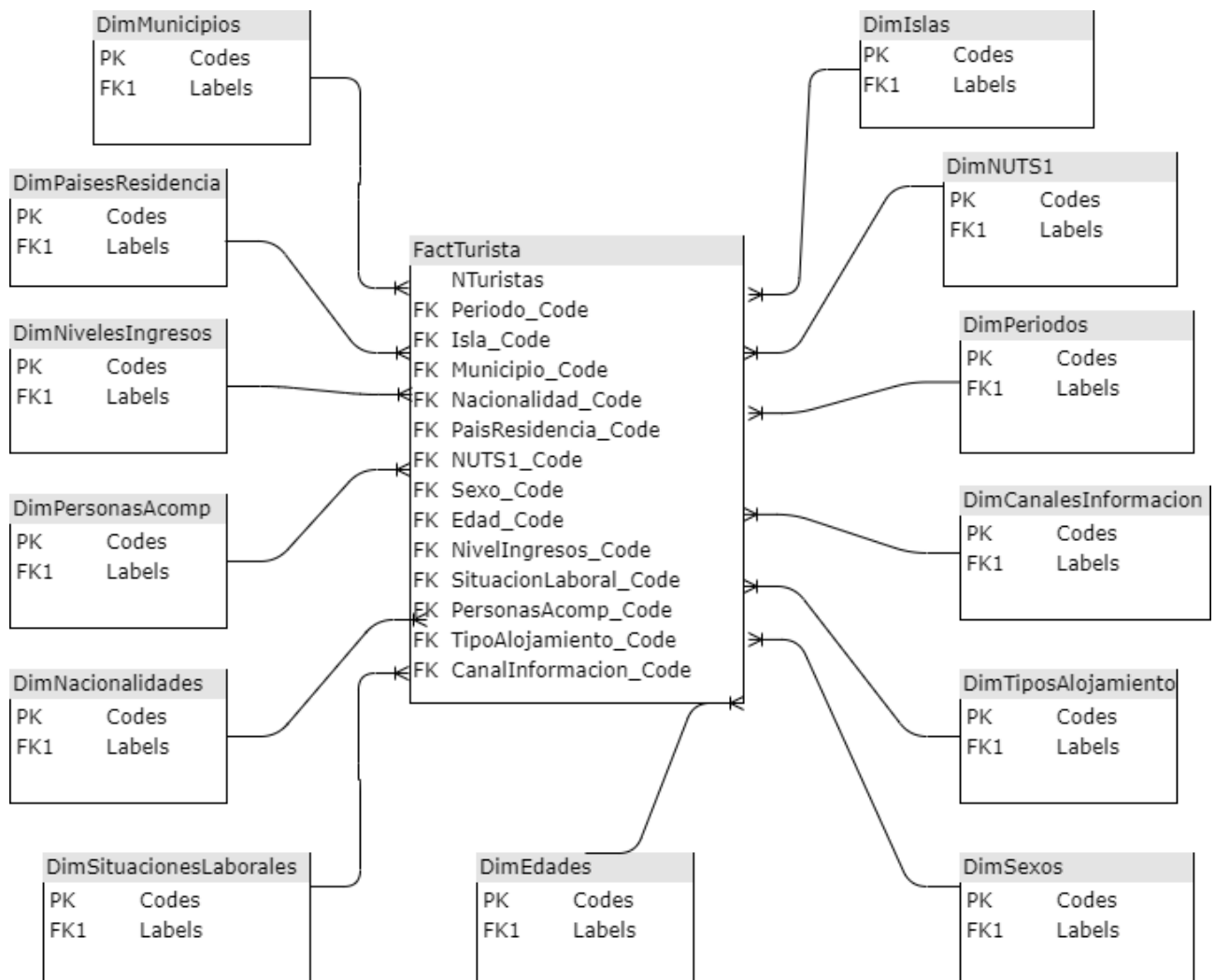


Figura 31: Esquema de estrella del Data Warehouse.

La creación de esta estructura de tablas la realicé con scripts de SQL. La descripción del proceso paso a paso es la siguiente:

1. Crear la base de datos llamada DWTFG2021 desde SQL Server Management.
2. Crear cada una de las tablas dimensionales.
3. Crear la tabla de hechos, cuyas claves foráneas se relacionan con cada una de las tablas dimensionales.

4.4.1. CREACIÓN DE LAS TABLAS DIMENSIONALES

Para la creación de las tablas dimensionales, se copiaron las tablas dimensionales de los Datamarts, dado que son las mismas dimensiones del Data Warehouse. La creación y la copia de las tablas de dimensiones se realizó con el siguiente script de SQL.

```
----- Tabla DimIslas -----  
SELECT * INTO [DWTFG2021].[dbo].[DimIslas] FROM [Datamart1].[dbo].  
[DimIslas];
```

```
ALTER TABLE [DWTFG2021].[dbo].[DimIslas]  
    ADD PRIMARY KEY (Codes);
```

```
----- Tabla DimNUTS1 -----  
SELECT * INTO [DWTFG2021].[dbo].[DimNUTS1] FROM [Datamart1].[dbo].  
[DimNUTS1];
```

```
ALTER TABLE [DWTFG2021].[dbo].[DimNUTS1]  
    ADD PRIMARY KEY (Codes);
```

```
----- Tabla Periodos -----  
SELECT * INTO [DWTFG2021].[dbo].[DimPeriodos] FROM [Datamart2].[dbo].  
[DimPeriodos];
```

```
ALTER TABLE [DWTFG2021].[dbo].[DimPeriodos]  
    ADD PRIMARY KEY (Codes);
```

```
----- Tabla Tipos de Alojamiento -----  
SELECT * INTO [DWTFG2021].[dbo].[DimTiposAlojamiento] FROM [Datamart2].  
[dbo].[DimTiposAlojamiento];
```

```
ALTER TABLE [DWTFG2021].[dbo].[DimTiposAlojamiento]  
    ADD PRIMARY KEY (Codes);
```

```
----- Tabla Niveles Ingresos -----  
SELECT * INTO [DWTFG2021].[dbo].[DimNivelesIngresos] FROM [Datamart2].  
[dbo].[DimNivelesIngresos];
```

```
ALTER TABLE [DWTFG2021].[dbo].[DimNivelesIngresos]  
    ADD PRIMARY KEY (Codes);
```

----- Tabla Personas Acompañantes -----

```
SELECT * INTO [DWTFG2021].[dbo].[DimPersonasAcomp] FROM [Datamart1].  
[dbo].[DimPersonasAcomp];  
ALTER TABLE [DWTFG2021].[dbo].[DimPersonasAcomp]  
    ADD PRIMARY KEY (Codes);
```

----- Tabla Edades -----

```
SELECT * INTO [DWTFG2021].[dbo].[DimEdades] FROM [Datamart3].[dbo].  
[DimEdades];  
ALTER TABLE [DWTFG2021].[dbo].[DimEdades]  
    ADD PRIMARY KEY (Codes);
```

----- Tabla Nacionalidades -----

```
SELECT * INTO [DWTFG2021].[dbo].[DimNacionalidades] FROM [Datamart3].  
[dbo].[DimNacionalidades];  
ALTER TABLE [DWTFG2021].[dbo].[DimNacionalidades]  
    ADD PRIMARY KEY (Codes);
```

----- Tabla Sexos -----

```
SELECT * INTO [DWTFG2021].[dbo].[DimSexos] FROM [Datamart3].[dbo].  
[DimSexos];  
ALTER TABLE [DWTFG2021].[dbo].[DimSexos]  
    ADD PRIMARY KEY (Codes);
```

----- Tabla Países de residencia -----

```
SELECT * INTO [DWTFG2021].[dbo].[DimPaisesResidencia] FROM [Datamart7].  
[dbo].[DimPaisesResidencia];  
ALTER TABLE [DWTFG2021].[dbo].[DimPaisesResidencia]  
    ADD PRIMARY KEY (Codes);
```

----- Tabla Situaciones laborales -----

```
SELECT * INTO [DWTFG2021].[dbo].[DimSituacionesLaborales] FROM  
[Datamart7].[dbo].[DimSituacionesLaborales];  
ALTER TABLE [DWTFG2021].[dbo].[DimSituacionesLaborales]  
    ADD PRIMARY KEY (Codes);
```

----- Tabla Canales de Información -----

```
SELECT * INTO [DWTFG2021].[dbo].[DimCanalesInformacion] FROM
[Datamart10].[dbo].[DimCanalesInformacion];
ALTER TABLE [DWTFG2021].[dbo].[DimCanalesInformacion]
    ADD PRIMARY KEY (Codes);
```

----- Tabla Municipios -----

```
SELECT * INTO [DWTFG2021].[dbo].[DimMunicipios] FROM [Datamart10].[dbo].
[DimMunicipios];
ALTER TABLE [DWTFG2021].[dbo].[DimMunicipios]
    ADD PRIMARY KEY (Codes);
```

4.4.2. CREACIÓN DE LAS TABLAS DE HECHOS

Una vez creadas las tablas dimensionales, se crea la tabla de hecho llamada FactTurista. Esta tabla se hace después de crear las tablas dimensionales para poder relacionar las claves foráneas con cada una de sus respectivas tablas. El script de SQL utilizado para la creación de la tabla de hechos es el siguiente.

----- Creación de la tabla Fact del turista -----

```
USE [DWTFG2021]
CREATE TABLE FactTurista ( -- CREAR LA TABLA FACT CON FK DE TODAS LAS
TABLAS DIM
    IDFactTurista INT IDENTITY(1,1) NOT NULL PRIMARY KEY,
    Periodo_Code nvarchar(10),
    NTuristas nvarchar(200),
    Isla_Code nvarchar(50),
    Municipio_Code nvarchar(50),
    Nacionalidad_Code nvarchar(50),
    PaisResidencia_Code nvarchar(50),
    NUTS1_Code nvarchar(10),
    Sexo_Code nvarchar(10),
    Edad_Code nvarchar(10),
```

```

NivelIngresos_Code nvarchar(10),
SituacionLaboral_Code nvarchar(10),
PersonasAcomp_Code nvarchar(10),
TipoAlojamiento_Code nvarchar(10),
CanalInformacion_Code nvarchar(10),
FOREIGN KEY (Periodo_Code) REFERENCES DimPeriodos(Codes),
FOREIGN KEY (Isla_Code) REFERENCES DimIslas(Codes),
FOREIGN KEY (Municipio_Code) REFERENCES DimMunicipios(Codes),
FOREIGN KEY (Nacionalidad_Code) REFERENCES
DimNacionalidades(Codes),
FOREIGN KEY (PaisResidencia_Code) REFERENCES
DimPaísesResidencia(Codes),
FOREIGN KEY (NUTS1_Code) REFERENCES DimNUTS1(Codes),
FOREIGN KEY (Sexo_Code) REFERENCES DimSexos(Codes),
FOREIGN KEY (Edad_Code) REFERENCES DimEdades(Codes),
FOREIGN KEY (NivelIngresos_Code) REFERENCES
DimNivelesIngresos(Codes),
FOREIGN KEY (SituacionLaboral_Code) REFERENCES
DimSituacionesLaborales(Codes),
FOREIGN KEY (PersonasAcomp_Code) REFERENCES
DimPersonasAcomp(Codes),
FOREIGN KEY (TipoAlojamiento_Code) REFERENCES
DimTiposAlojamiento(Codes),
FOREIGN KEY (CanalInformacion_Code) REFERENCES
DimCanalesInformacion(Codes),);

```

El Data Warehouse está creado y esta es su vista en el entorno de SQL Server Managament.

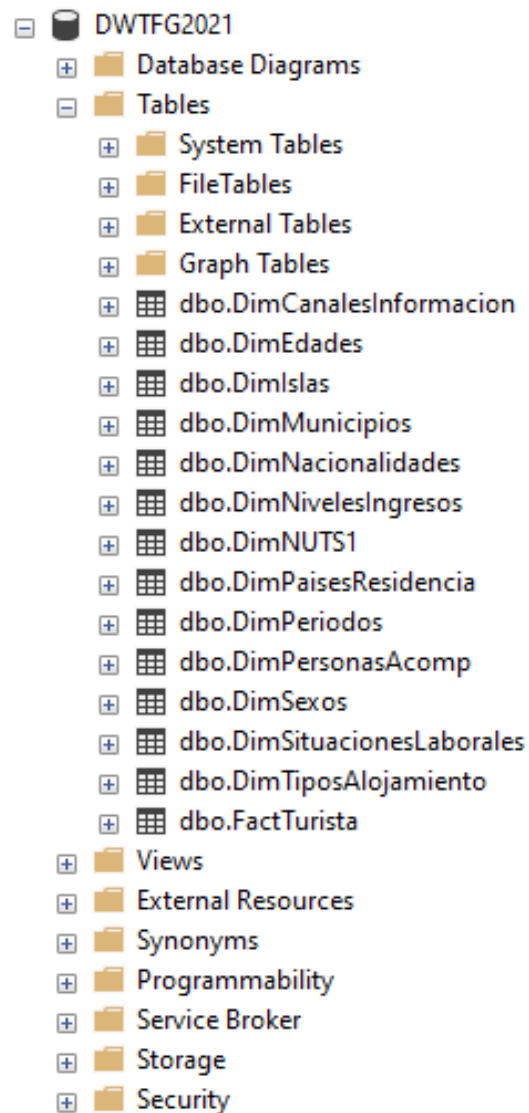


Figura 32: Data Warehouse creado en SQL Server Management.

En este paso del proyecto, el Data Warehouse DWTFG2021 está creado. Las tablas dimensionales tienen la información necesaria pero la tabla de hechos está vacía. El siguiente paso es rellenar la tabla de hechos con procesos ETL.

4.5. IMPLEMENTACIÓN DE LOS PROCESOS ETL

Los procesos ETL nos permiten establecer un flujo de datos entre las bases de datos de los Datamarts con la base de datos del Data Warehouse DWTFG2021. Para este paso, se utilizó SQL Integration Services, un servicio de Microsoft para hacer integraciones y transformaciones de datos.

En este proyecto, los procesos desarrollados con SQL Integration consisten en trasladar los datos de las tablas de hechos de cada Datamart a la tabla de hechos del Data Warehouse. Así se consiguen centralizar todos los datos en el Data Warehouse.

Los pasos para crear los procesos ETL con SQL Integration Services (SSIS en adelante) fueron los siguientes:

1. Crear un proyecto en Visual Studio de SSIS.
2. Crear un paquete SSIS.
3. Crear un flujo de control.
4. Crear un flujo de datos.
5. Crear las conexiones a cada una de las bases de datos.
6. Ejecutar el paquete SSIS y comprobar el traslado de los datos.

4.5.1. CREACIÓN DE UN PROYECTO SSIS

Una vez instalado SQL Integration Services, Visual Studio habilita la creación de proyectos de integración de datos. Creé el proyecto llamado Integration Services TFG que contiene todo los procesos necesarios para este trabajo de fin de grado.

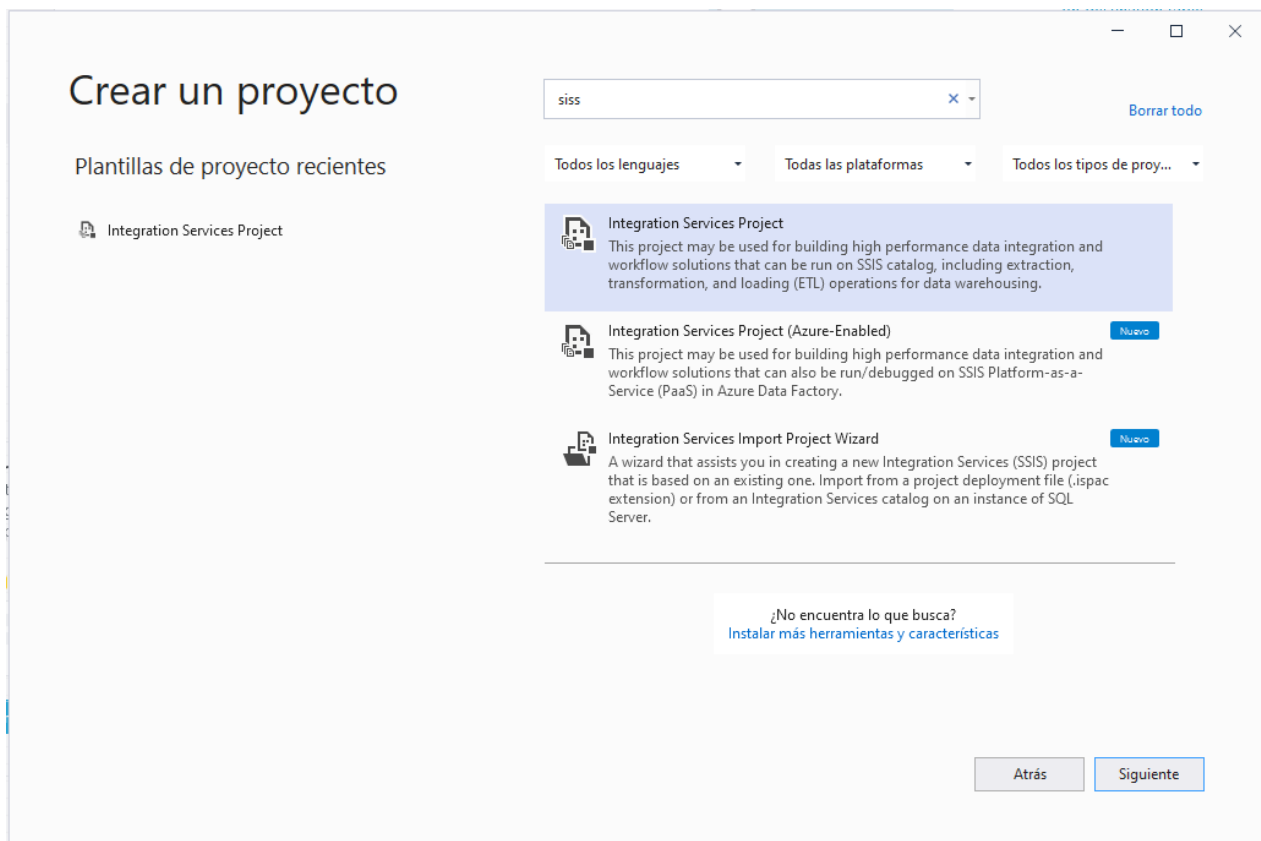


Figura 33: Creación de un proyecto de Integration Services en Visual Studio.

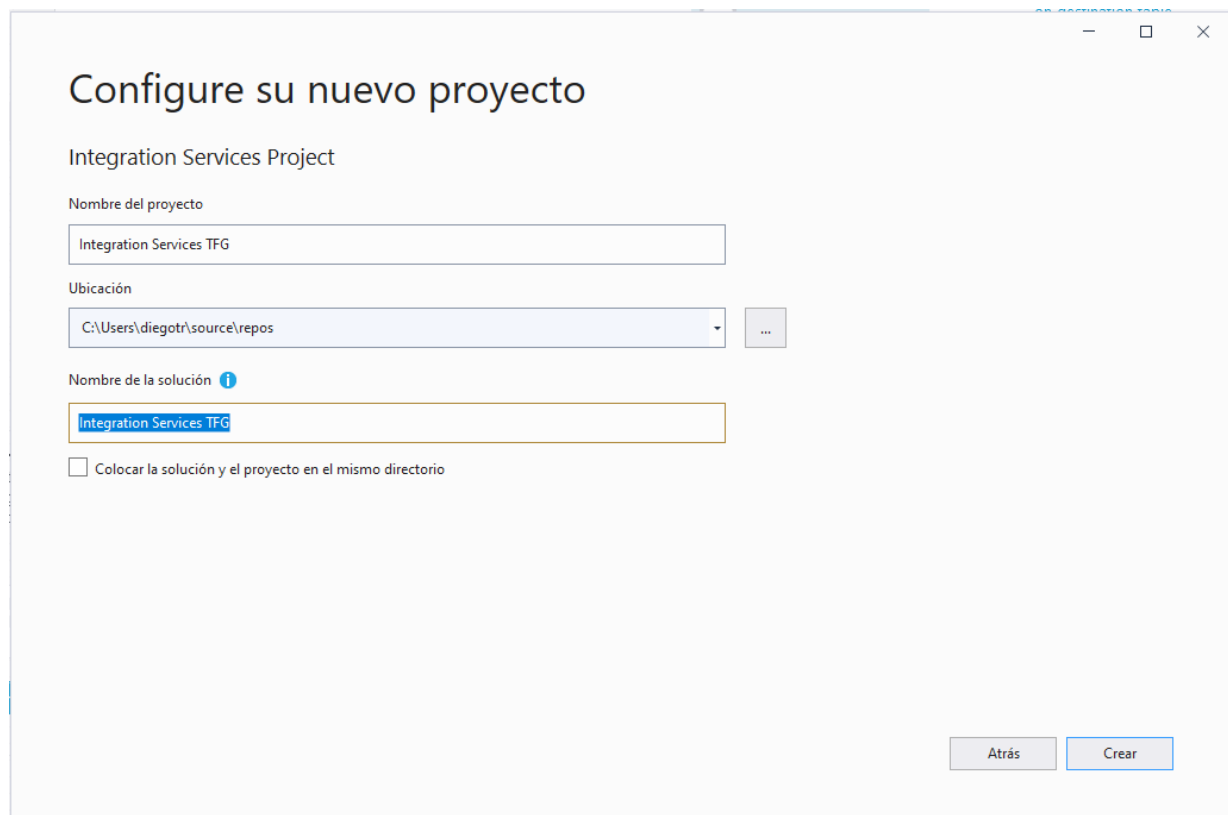


Figura 34: Creación de un proyecto de Integration Services en Visual Studio.

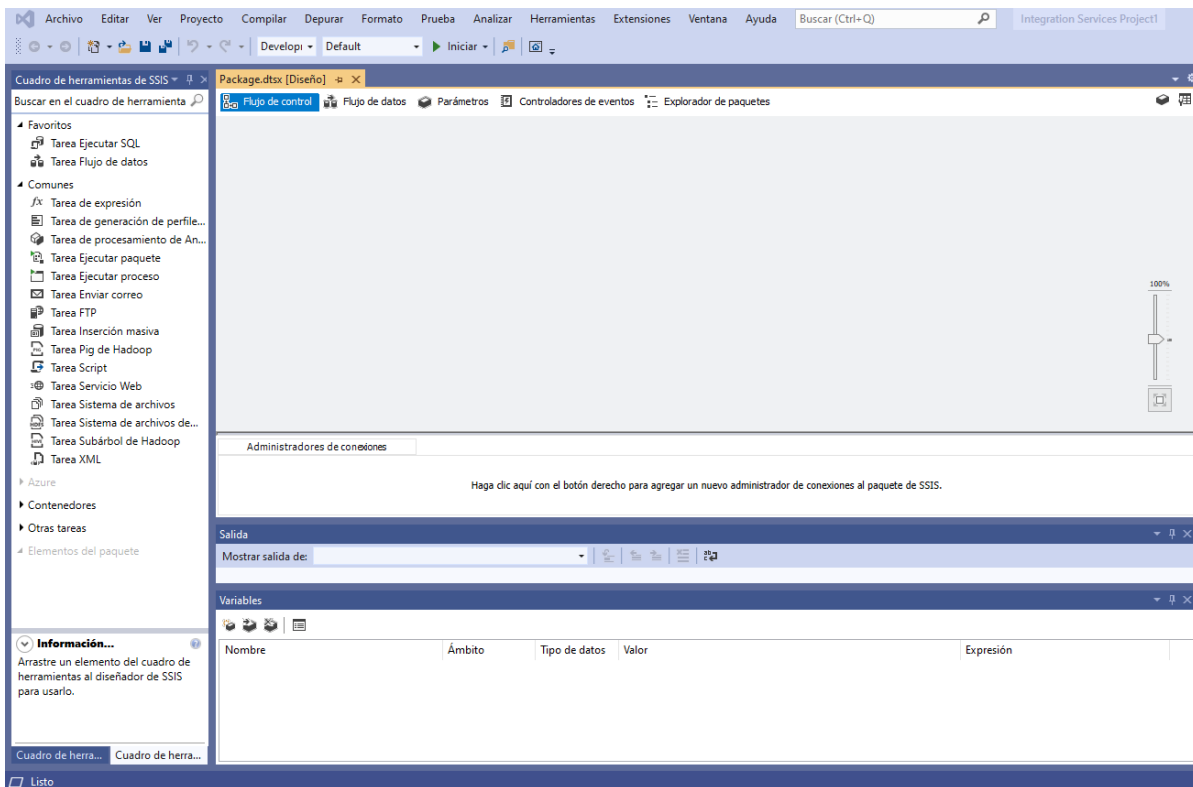


Figura 35: Entorno de SQL Integration Services.

Con el proyecto creado, aparece la ventana abierta de un paquete con formato dtsx. Este es el formato de los paquetes de SSIS. Y es aquí donde creamos el flujo de datos.

4.5.2. CREACIÓN DE UN PAQUETE SSIS CON FLUJO DE DATOS

Primero hay que crear la tarea del flujo de datos, escogiéndola de todos los tipos de tareas disponibles en el cuadro de herramientas SSIS (panel lateral izquierdo).

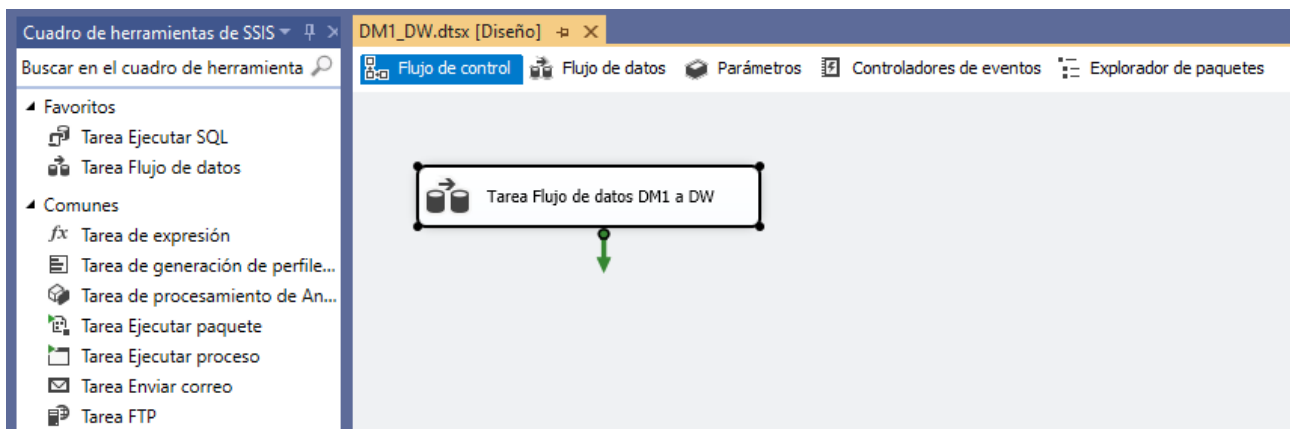


Figura 36: Flujo de control de SQL Integration Services.

La tarea tiene un nombre descriptivo para saber cual es su función. Al igual que el paquete dtsx, guardado con el nombre DM1_DW.dtsx. Con esta nomenclatura se deduce que el proceso se realiza desde el Datamart1 hacia el Data Warehouse.

En la pestaña de flujo de datos, se crea la canalización de los datos de una base de datos a otra. Se seleccionan los objetos “Origen de OLE DB” y “Destino de OLE DB” del cuadro de herramientas SSIS (panel lateral izquierdo).

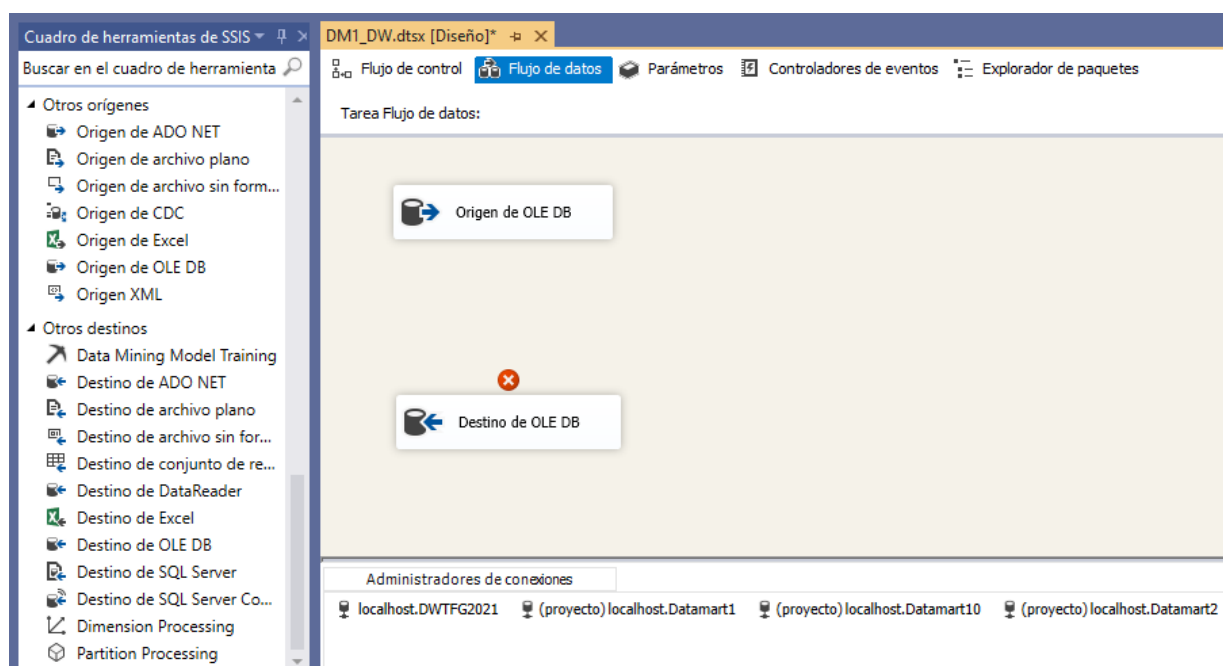


Figura 37: Flujo de datos de SQL Integration Services.

Estos objetos son los que se utilizan para nuestro cometido: extraer y cargar datos de bases de datos relacionales compatibles con OLE DB, una API de Microsoft de acceso a bases de datos. El siguiente paso es configurar ambos objetos.

La configuración del origen de OLE DB consiste en seleccionar la base de datos la tabla que contiene los datos y el modo de acceso:

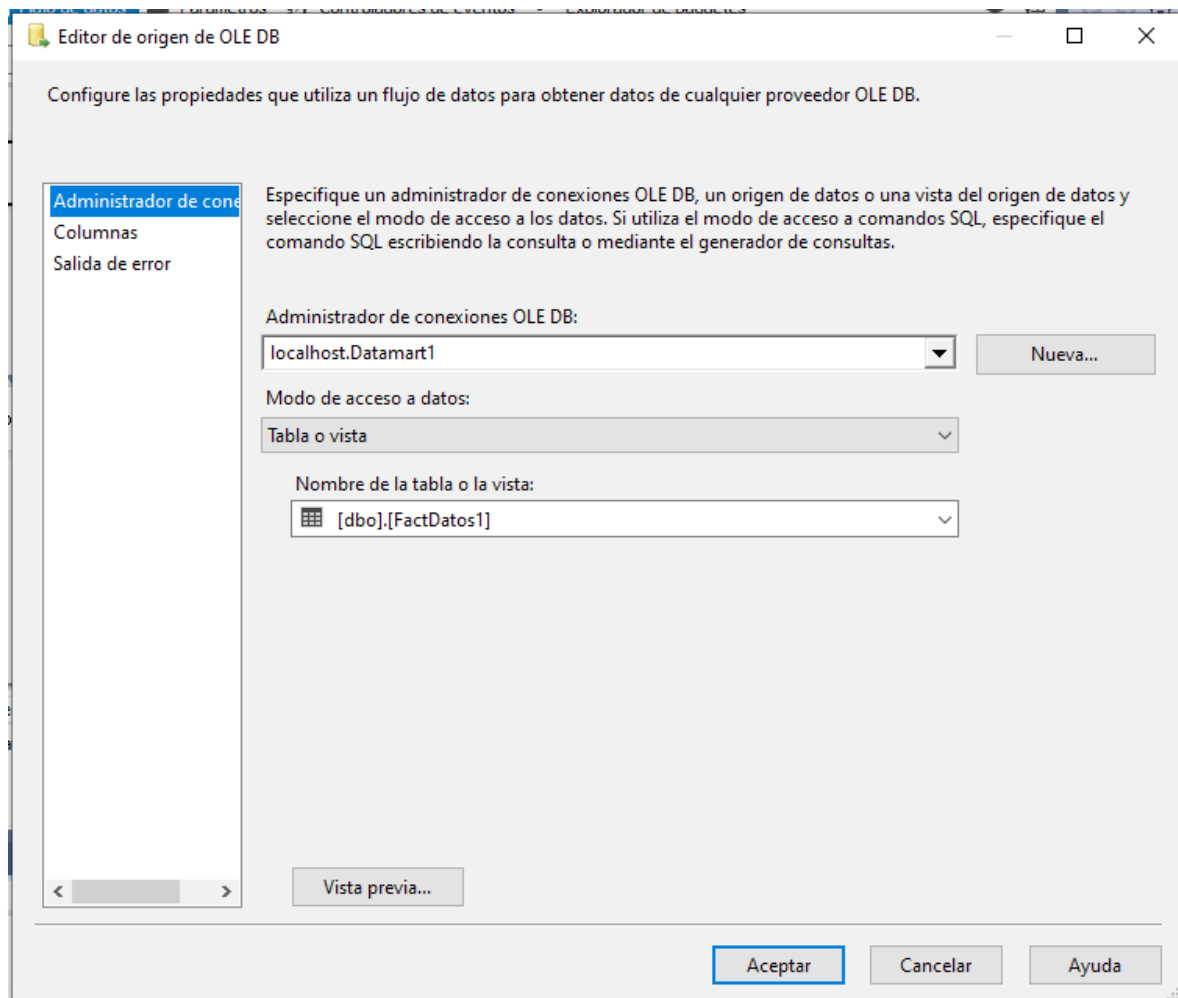


Figura 38: Editor de origen de OLE DB.

Para seleccionar la base de datos de origen, es necesario crear una conexión OLE DB a esa base de datos. En el botón “Nueva ...” accedemos a la siguiente vista:

The screenshot shows the 'Administrador de conexiones' (Connections Administrator) window. At the top, the 'Proveedor' (Provider) is set to 'OLE DB nativo\SQL Server Native Client 11.0'. Below this, the 'Nombre del servidor' (Server name) is 'localhost'. The 'Autenticación' (Authentication) is set to 'Autenticación de Windows'. The 'Establecer conexión con una base de datos' (Establish connection with a database) section has two options: 'Seleccionar o escribir el nombre de la base de datos:' (selected) and 'Adjuntar un archivo de base de datos:'. The selected option has a dropdown menu with 'Datamart1' entered. The 'Aceptar' (Accept) button is highlighted.

Figura 39: Creación de conexiones de OLE DB.

En ella, se elige el nombre del servidor de bases de datos y el nombre de la base de datos. También el modo de autenticación para acceder al servidor de bases de datos.

Es necesario crear una conexión OLE DB por cada una de las bases de datos con las que van a trabajar los flujos de datos.

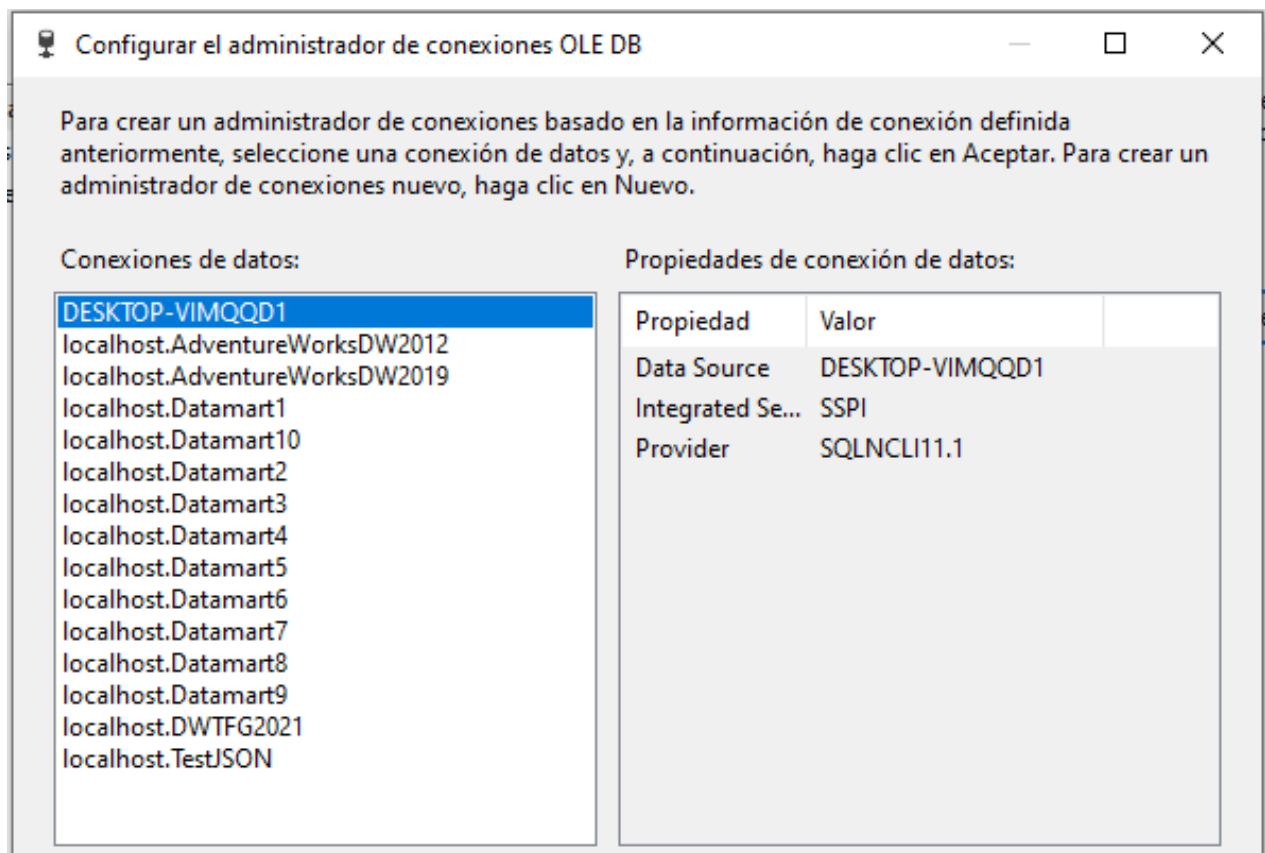


Figura 40: Lista de conexiones de OLE DB.

Recordemos que el objetivo de cada paquete SSIS es trasladar los datos de las tablas de hechos de cada Datamart hasta la tabla de hechos del Data Warehouse DWTFG2021.

Para terminar de configurar el origen, hay que asegurarse de que marcar todas las columnas que se van a trasladar en el flujo de datos.

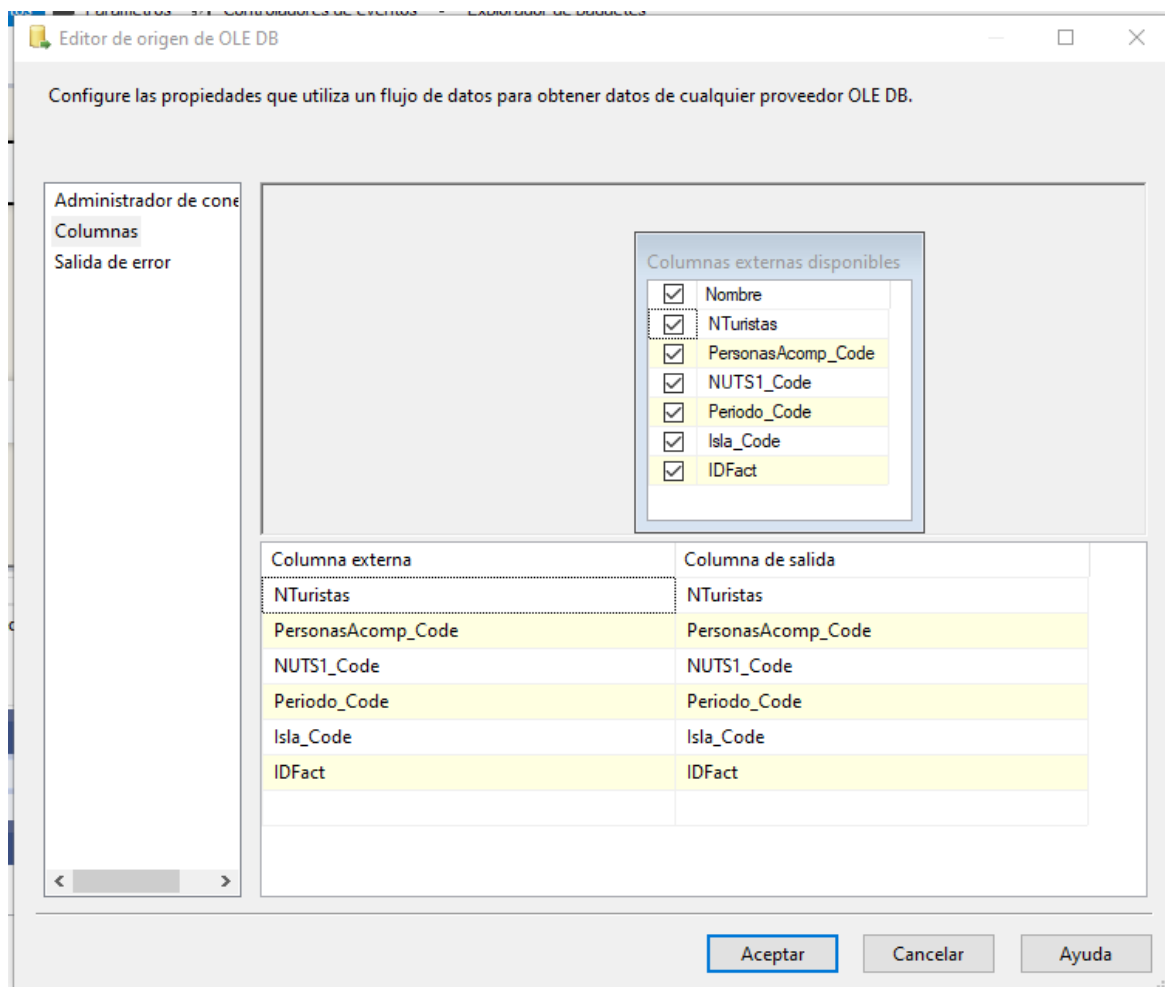


Figura 41: Columnas en el editor de origen de OLE DB.

Después de estas configuraciones, se une el origen de OLE DB al destino por la flecha azul. De esta manera, se establece el flujo de datos.

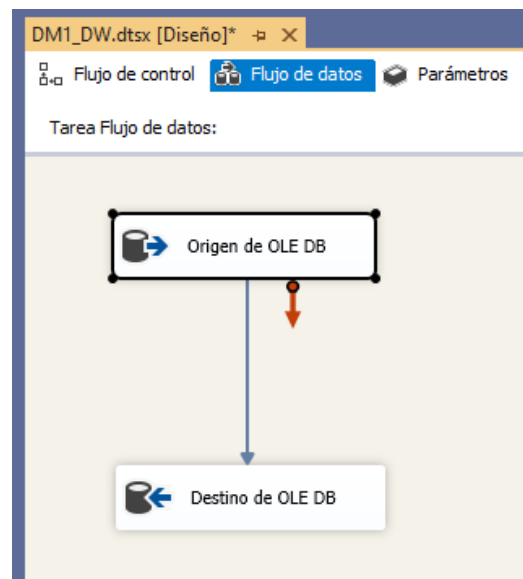


Figura 42: Flujo de datos entre origen y destino OLE DB.

La configuración del destino de OLE DB es parecida a la del origen. También es necesario usar una conexión hacia la base de datos de destino y la tabla que almacenará los datos trasladados.

Editor de destino de OLE DB

Configure las propiedades para insertar datos en una base de datos relacional mediante un proveedor OLE DB.

Administrador de conexiones OLE DB: localhost.DWTFG2021 Nueva...

Modo de acceso a datos: Carga rápida de tabla o vista

Nombre de la tabla o la vista: [dbo].[FactTurista] Nueva...

☒ Mantener valores de identidad ☒ Bloqueo de tabla

☒ Mantener valores NULL ☒ Comprobar restricciones

Filas por lote:

Tamaño máximo de confirmación de inserción: 2147483647

Ver datos

⚠ Asigne las columnas en la página Asignaciones.

Aceptar Cancelar Ayuda

Figura 43: Columnas en el editor de origen de OLE DB.

En este caso, se debe verificar las asignaciones entre las columnas de la tabla de la base de datos de origen con las columnas de la tabla de la base de datos de destino, y corregir cuando proceda.

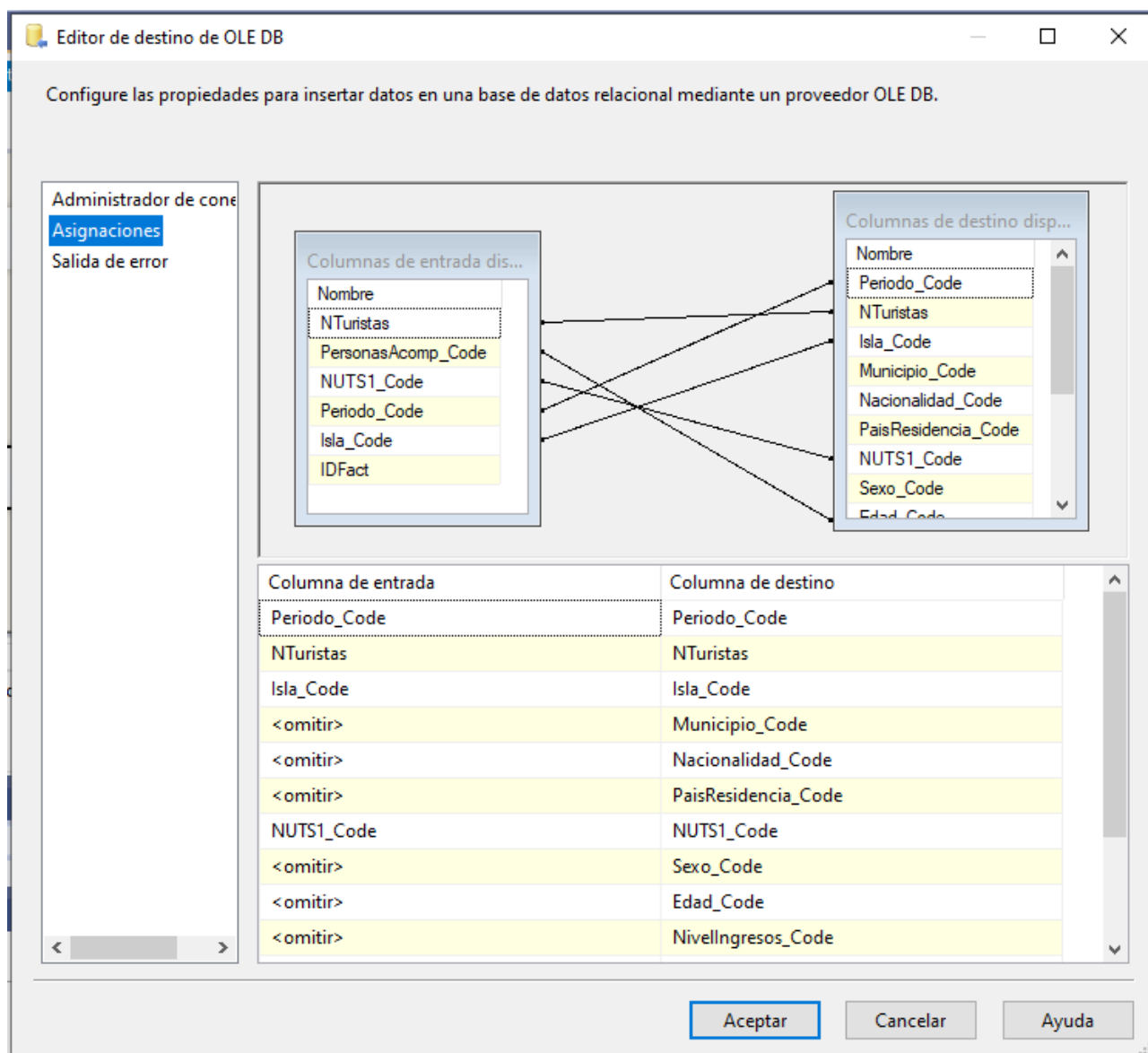
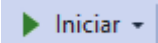


Figura 44: Asignación de columnas en el editor de destino de OLE DB.

Con el flujo de datos configurado, se ejecuta el paquete SSIS en el botón 

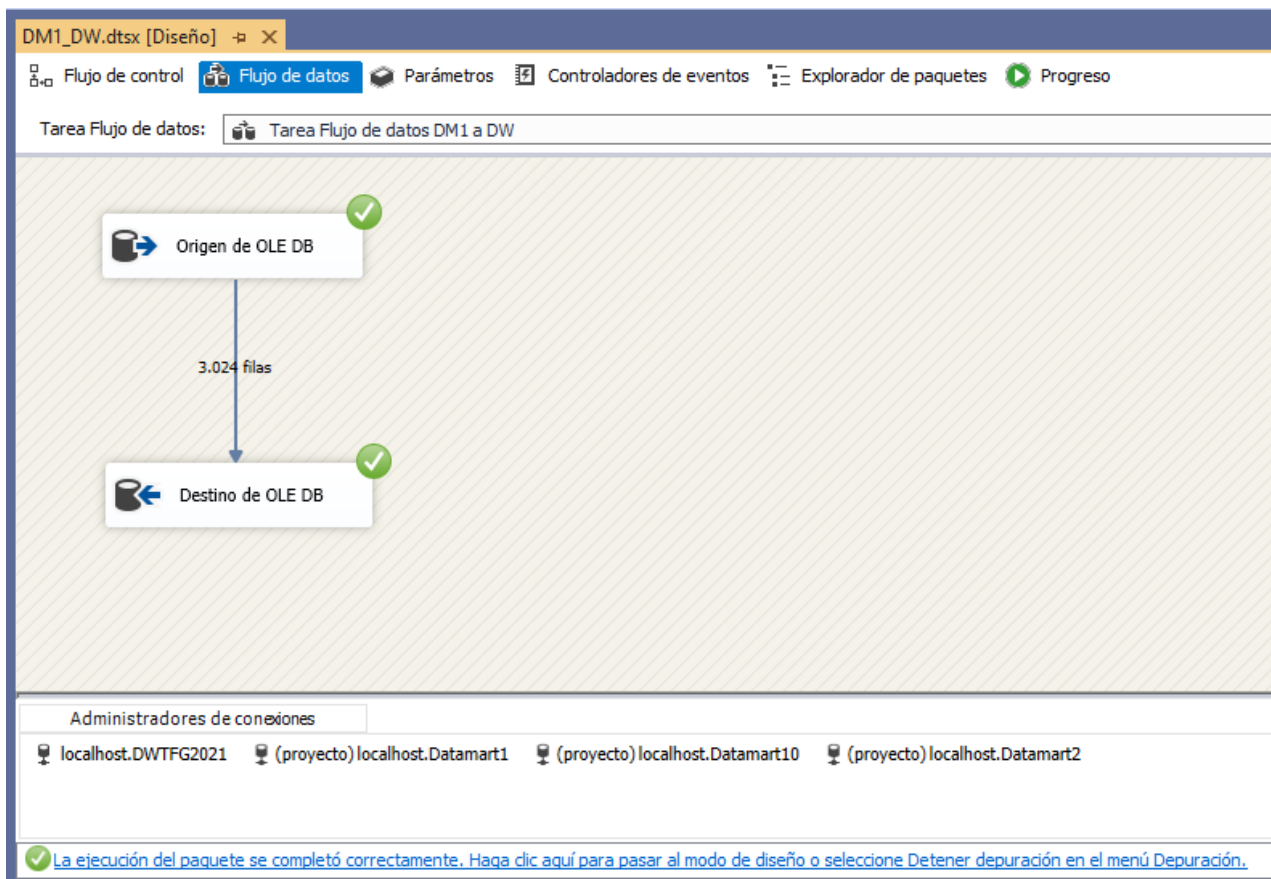


Figura 45: Ejecución de un paquete SSIS.

La ejecución ha tenido éxito y se indica el número de filas que han sido insertadas en la tabla de destino. Este proceso de creación de paquetes SSIS y de flujo de datos se repite diez veces, una por cada Datamart creado.

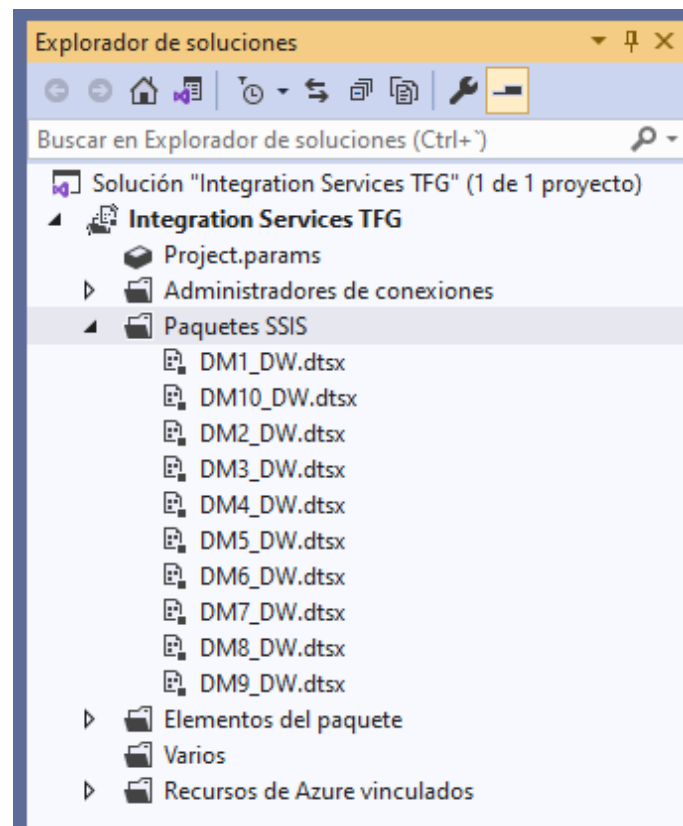


Figura 46: Lista de paquetes SSIS.

Una vez se hayan creado y ejecutado todos los paquetes SSIS, nuestro Data Warehouse estará creado con toda la información extraída de los Datamarts.

SQLQuery1.sql - DE...QOD1\diegotr (55))

```

1  USE [DWTFG2021]
2  GO
3
4  SELECT [Periodo_Code]
5         , [NTuristas]
6         , [Isla_Code]
7         , [Municipio_Code]
8         , [Nacionalidad_Code]
9         , [PaisResidencia_Code]
10        , [NUTS1_Code]
11        , [Sexo_Code]
12        , [Edad_Code]
13        , [NivelIngresos_Code]
14        , [SituacionLaboral_Code]
15        , [PersonasAcomp_Code]
16        , [TipoAlojamiento_Code]
17        , [CanalInformacion_Code]
18  FROM [dbo].[FactTurista]
19
20  GO
21

```

110 %

Results Messages

	Periodo_Code	NTuristas	Isla_Code	Municipio_Code	Nacionalidad_Code	PaisResidencia_Code	NUTS1_Code	Sexo_Code
1	2020Q1	27851	NULL	NULL	ITA380	NULL	NULL	1
2	2019	123496	NULL	NULL	ITA380	NULL	NULL	1
3	2019Q4	33640	NULL	NULL	ITA380	NULL	NULL	1
4	2019Q3	23493	NULL	NULL	ITA380	NULL	NULL	1
5	2019Q2	27358	NULL	NULL	ITA380	NULL	NULL	1
6	2019Q1	39005	NULL	NULL	ITA380	NULL	NULL	1
7	2018	127052	NULL	NULL	ITA380	NULL	NULL	1
8	2018Q4	32539	NULL	NULL	ITA380	NULL	NULL	1
9	2018Q3	28417	NULL	NULL	ITA380	NULL	NULL	1
10	2018Q2	29370	NULL	NULL	ITA380	NULL	NULL	1
11	2018Q1	36725	NULL	NULL	ITA380	NULL	NULL	1
12	2020	149422	NULL	NULL	DNK208_FIN246_NOR578_SWE752	NULL	NULL	1
13	2020Q4	6051	NULL	NULL	DNK208_FIN246_NOR578_SWE752	NULL	NULL	1
14	2020Q3	1161	NULL	NULL	DNK208_FIN246_NOR578_SWE752	NULL	NULL	1
15	2020Q1	142209	NULL	NULL	DNK208_FIN246_NOR578_SWE752	NULL	NULL	1
16	2019	429658	NULL	NULL	DNK208_FIN246_NOR578_SWE752	NULL	NULL	1
17	2019Q4	184215	NULL	NULL	DNK208_FIN246_NOR578_SWE752	NULL	NULL	1
18	2019Q3	23233	NULL	NULL	DNK208_FIN246_NOR578_SWE752	NULL	NULL	1
19	2019Q2	37200	NULL	NULL	DNK208_FIN246_NOR578_SWE752	NULL	NULL	1
20	2019Q1	185009	NULL	NULL	DNK208_FIN246_NOR578_SWE752	NULL	NULL	1
21	2018	434368	NULL	NULL	DNK208_FIN246_NOR578_SWE752	NULL	NULL	1

Query executed successfully.

Figura 47: Datos de la tabla FactTurista.

4.6. VISUALIZACIÓN DE LOS DATOS

Una de las finalidades de disponer de información almacenada en un Data Warehouse es poder aprovechar los datos para realizar visualizaciones con las que tener una mejor comprensión del contexto de los datos.

Para la visualización de los datos de este proyecto, he elegido el software de Power BI.

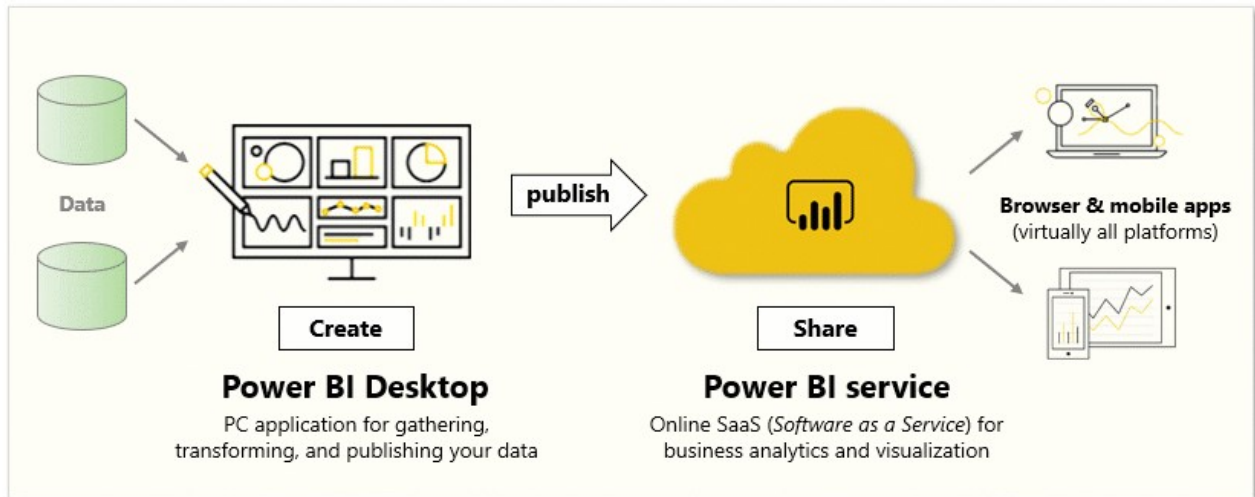


Figura 48: Diagrama de como funciona Power BI.

Fuente: <https://360reports.io/get-data-with-power-bi-desktop/explore-power-bi-desktop/>

Power BI es una solución software de análisis de Microsoft que ofrece visualizaciones interactivas y capacidades de Business Intelligence para que los usuarios finales podamos crear informes y paneles de datos. Y esta solución se compone principalmente de dos partes:

- Power BI Desktop: Una versión de escritorio para almacenar los datos, transformarlos y crear las visualizaciones interactivas.
- Power BI service: Una versión SaaS online para publicar y compartir nuestros informes de datos desde la nube hacia cualquier dispositivo. Para usar este servicio es necesaria una cuenta de pago de Microsoft.

Para el alcance de este proyecto se usa Power BI Desktop para construir un panel con gráficas de datos o también llamado dashboard. Para ello, es necesario hacer una carga de datos en Power BI Desktop, verificar el modelo de datos y construir el dashboard a partir de los datos.

4.6.1. CARGA DE DATOS

Para poder trabajar con los datos, Power BI Desktop necesita tenerlos cargados en su software. Esto no impide que si los datos se modifican en el Data Warehouse, también se sincronicen en Power BI Desktop.

Para cargar los datos, debemos conectar con la base de datos del Data Warehouse y seleccionar las tablas con las que vamos a trabajar. En nuestro caso, usaremos todas las tablas del Data Warehouse DWTFG2021.

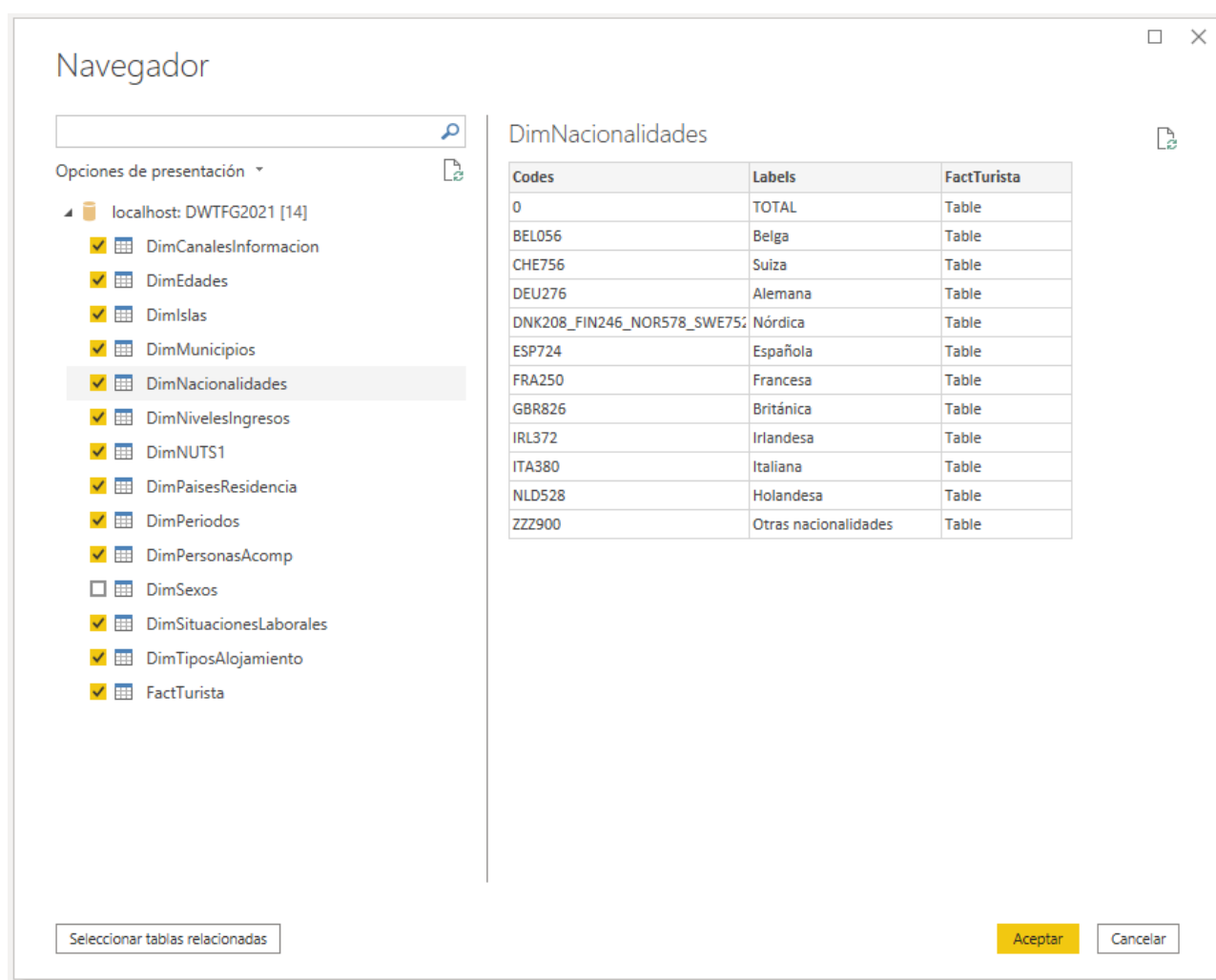


Figura 49: Selección de tablas a cargar en Power BI Desktop.

tourist_dw - Editor de Power Query

Inicio Transformar Agregar columna Vista Herramientas Ayuda

Cerrar y aplicar • Nuevo origen recientes • Especificar datos Configuración de origen de datos Administrar parámetros Actualizar vista previa Editor avanzado • Administrar columnas Eliminar columnas Conservar filas Quitar filas Reducir filas Ordenar Dividir columnas Agrupar por Tipo de datos: Texto • Usar la primera fila como encabezado • Reemplazar los valores Combinar consultas • Anexar consultas • Combinar archivos Text Analytics Visión Azure Machine Learning Conclusiones de IA

Consultas [14] < FactT turista 1.2 Nº Turistas A Isla_Code A Municipio_Code A Nacionalidad_Code A PaísResidencia_Code A NUTSI_Code A Sexo_Code A Edad_Code

		A Isla_Code	A Municipio_Code	A Nacionalidad_Code	A PaísResidencia_Code	A NUTSI_Code	A Sexo_Code	A Edad_Code
DimCanalesInformacion	1	2020Q1	27851	null	ITA800	null	null 1	2
DimEdades	2	2019	123496	null	ITA800	null 1	2	
DimMunicipios	3	2019Q4	33640	null	ITA800	null	null 1	2
DimIslas	4	2019Q3	23493	null	ITA800	null 1	2	
DimNacionalidades	5	2019Q2	27558	null	ITA800	null 1	2	
DimNivelesIngresos	6	2019Q1	39005	null	ITA800	null	null 1	2
FactT turista	7	2018	127052	null	ITA800	null 1	2	
DimTiposAlojamiento	8	2018Q4	32539	null	ITA800	null	null 1	2
DimSituacionesLaborales	9	2018Q3	28417	null	ITA800	null 1	2	
DimSexos	10	2018Q2	29370	null	ITA800	null	null 1	2
DimPersonasAcomp	11	2018Q1	36725	null	ITA800	null	null 1	2
DimPeriodos	12	2020	149422	null	DNK208_FIN246_NOR578_SWE_	null	null 1	2
DimPaísesResidencia	13	2020Q4	60524	null	DNK208_FIN246_NOR578_SWE_	null	null 1	2
DimNUTS1	14	2020Q3	11651	null	DNK208_FIN246_NOR578_SWE_	null	null 1	2
	15	2020Q2	142209	null	DNK208_FIN246_NOR578_SWE_	null	null 1	2
	16	2019	429658	null	DNK208_FIN246_NOR578_SWE_	null	null 1	2
	17	2019Q4	184215	null	DNK208_FIN246_NOR578_SWE_	null	null 1	2
	18	2019Q3	23223	null	DNK208_FIN246_NOR578_SWE_	null	null 1	2
	19	2019Q2	37200	null	DNK208_FIN246_NOR578_SWE_	null	null 1	2
	20	2019Q1	180009	null	DNK208_FIN246_NOR578_SWE_	null	null 1	2
	21	2018	434968	null	DNK208_FIN246_NOR578_SWE_	null	null 1	2
	22	2018Q4	174812	null	DNK208_FIN246_NOR578_SWE_	null	null 1	2
	23	2018Q3	24885	null	DNK208_FIN246_NOR578_SWE_	null	null 1	2
	24	2018Q2	39081	null	DNK208_FIN246_NOR578_SWE_	null	null 1	2
	25	2018Q1	195582	null	DNK208_FIN246_NOR578_SWE_	null	null 1	2
	26	2020	18006	null	CHET56	null	null 1	2
	27	2020Q4	2512	null	CHET56	null	null 1	2
	28	2020Q3	2756	null	CHET56	null	null 1	2
	29	2020Q2	12799	null	CHET56	null	null 1	2
	30	2018	54365	null	CHET56	null	null 1	2
	31	2018Q4	21448	null	CHET56	null	null 1	2
	32	2018Q3	10252	null	CHET56	null	null 1	2
	33	2018Q2	12339	null	CHET56	null	null 1	2
	34	2018Q1	10247	null	CHET56	null	null 1	2
	35	2018	50455	null	CHET56	null	null 1	2
	36	2018Q4	17165	null	CHET56	null	null 1	2
	37	2018Q3	7536	null	CHET56	null	null 1	2
	38	2018Q2	15947	null	CHET56	null	null 1	2
	39	2018Q1	12707	null	CHET56	null	null 1	2
	40							

27 COLUMNAS, 999+ FILAS Generación de perfiles de columnas basada en las 1000 primeras filas

VISTA PREVIA DESCARGADA A LAS 11:15

Configuración de la consulta

- PROPIEDADES
 - NOMBRE
 - FactT turista
 - Todas las propiedades
- PASOS APLICADOS
 - Origen
 - Navegación
 - Tipo cambiado
 - Valor reemplazado
 - Transformado 1
 - + Columnas con nombre cambi...

En este paso, Power BI Desktop nos da herramientas para transformar los datos. Es aquí donde verificamos cada columna de las tablas para comprobar que el tipo de datos se corresponde con el valor que representa.

97

4.6.2. MODELO DE DATOS

Con los datos cargados y verificados, Power BI Desktop nos construye un diagrama con las relaciones entre tablas. Esto es la representación del modelo de datos.

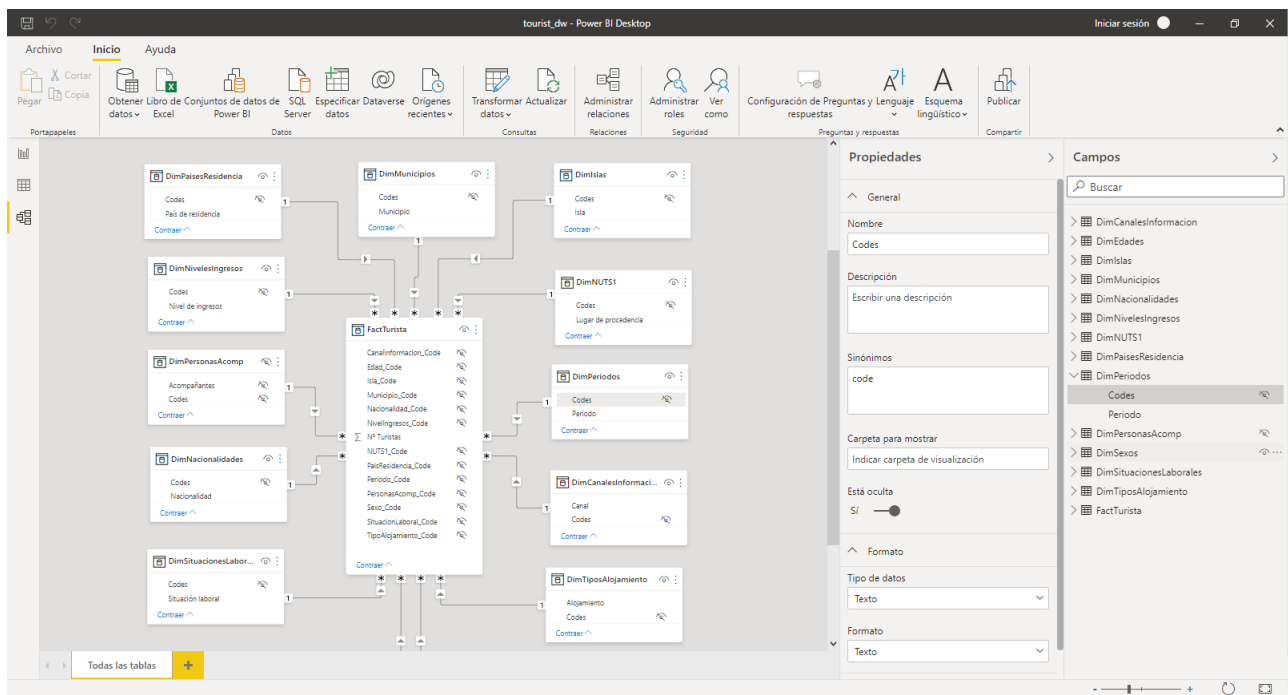


Figura 52: Modelo de datos en Power BI Desktop.

El modelo generado también se debe revisar. Las relaciones entre tablas dimensionales y tabla de hechos deber ser de 1 a muchos, las relaciones entre claves primarias con sus claves foráneas y la visibilidad de algunos campos como las claves primarias y foráneas.

Una de las funciones de las claves primarias y foráneas es la de relacionar las tablas y los datos entre sí. Y este tipo de claves suelen ser códigos alfanuméricos. En nuestras tablas existen otros campos que representan mejor los datos como es el caso de los campos labels en las tablas dimensionales. Este campo label es más legible para los usuarios que las claves alfanuméricas y por esta razón, quitamos la visibilidad de todas las claves de las tablas. Este último ajuste en quitar la visibilidad a las claves se entenderá en la siguiente sección donde se construye el dashboard.

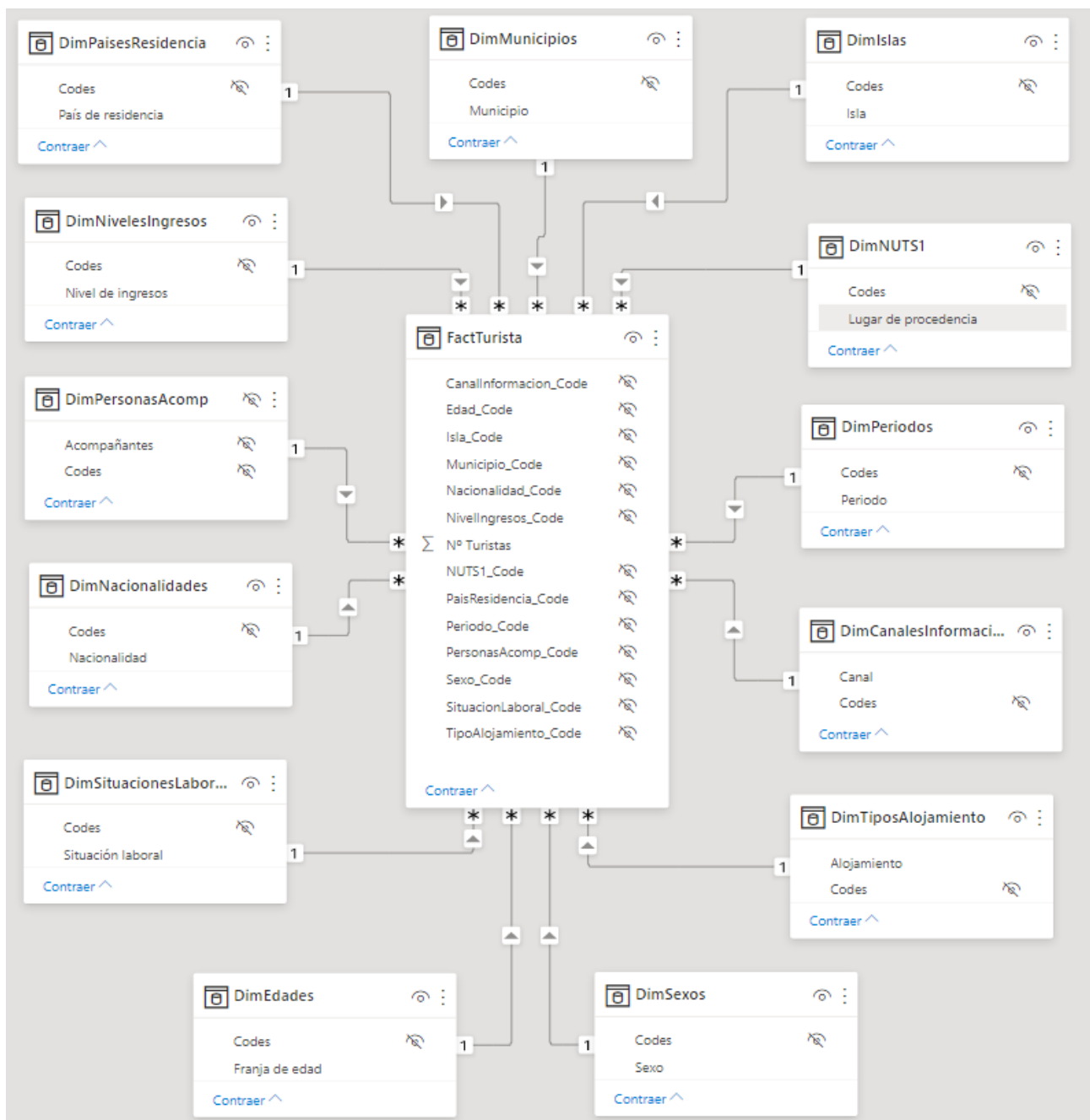


Figura 53: Modelo de datos.

Otras de las correcciones realizadas es la de cambiar el nombre de los campos labels por nombres más representativos de cada tabla.

4.6.3. CONSTRUCCIÓN DEL DASHBOARD

La función de un panel de datos o dashboard es la de mostrar la información de manera clara y relevante para los usuarios a través de visualizaciones, gráficas e indicadores. Y para este papel, he elegido el software de Power BI Desktop.

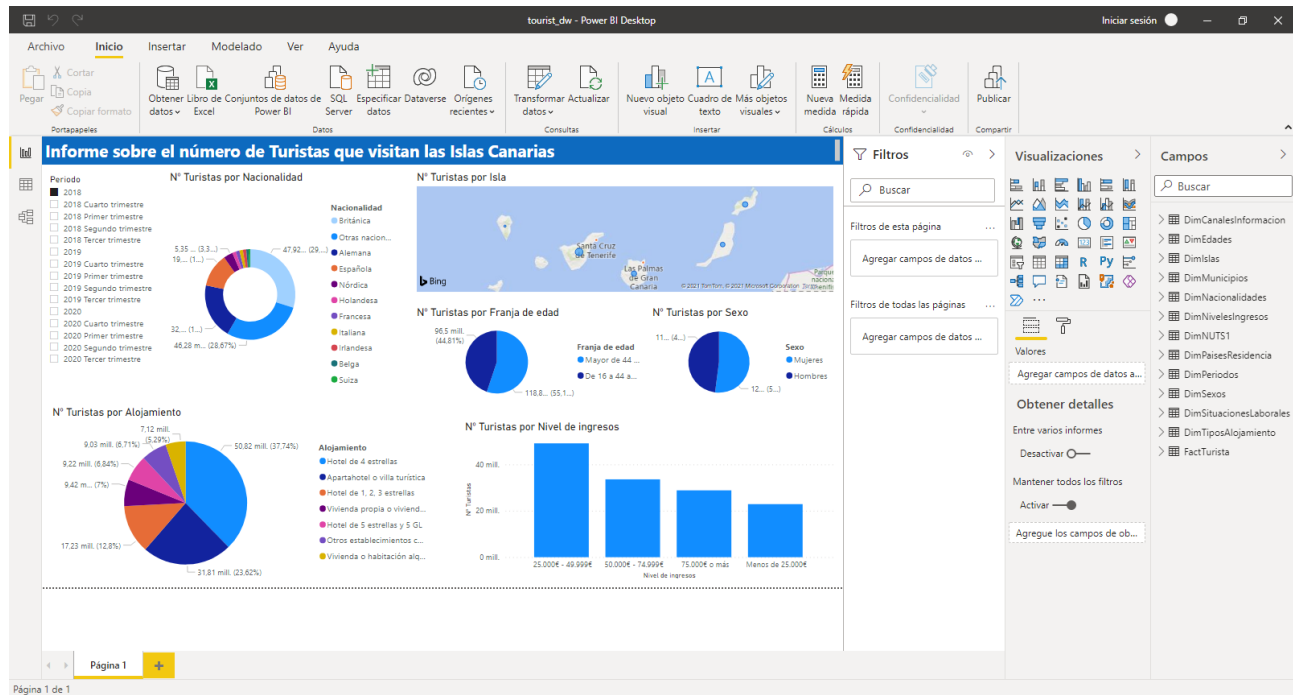


Figura 54: Escritorio de Power BI Desktop.

Power BI Desktop nos ofrece, entre otras muchas opciones, un catálogo amplio de visualizaciones para representar los datos de nuestro Data Warehouse. Las visualizaciones elegidas son las siguientes:

- Un gráfico de anillos que divide el número de turistas que llegan a Canarias por su nacionalidad.

Nº Turistas por Nacionalidad

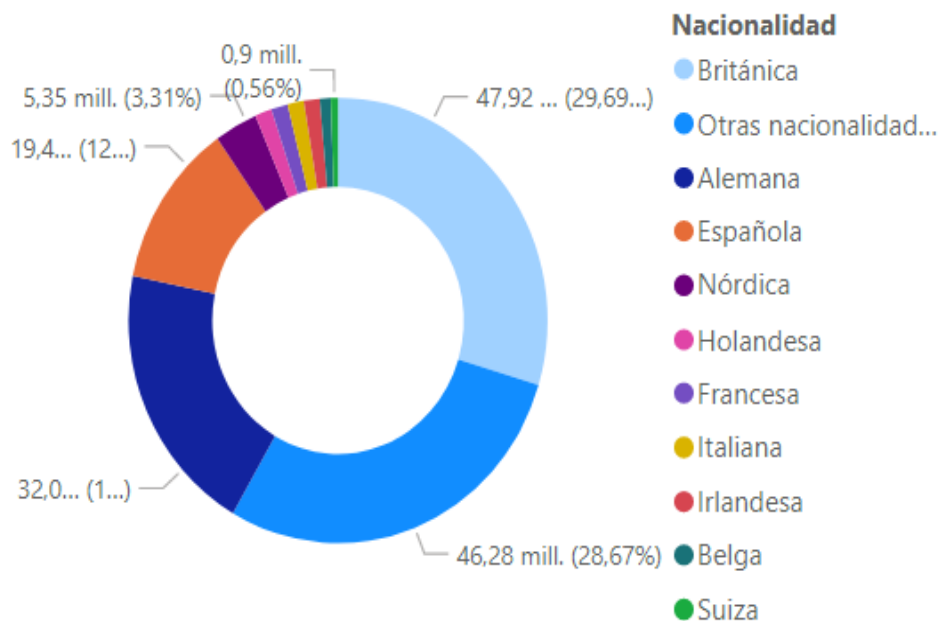


Figura 55: Gráfico de anillos.

- Un gráfico circular que representa el número de turistas por tipo de alojamiento.

Nº Turistas por Alojamiento

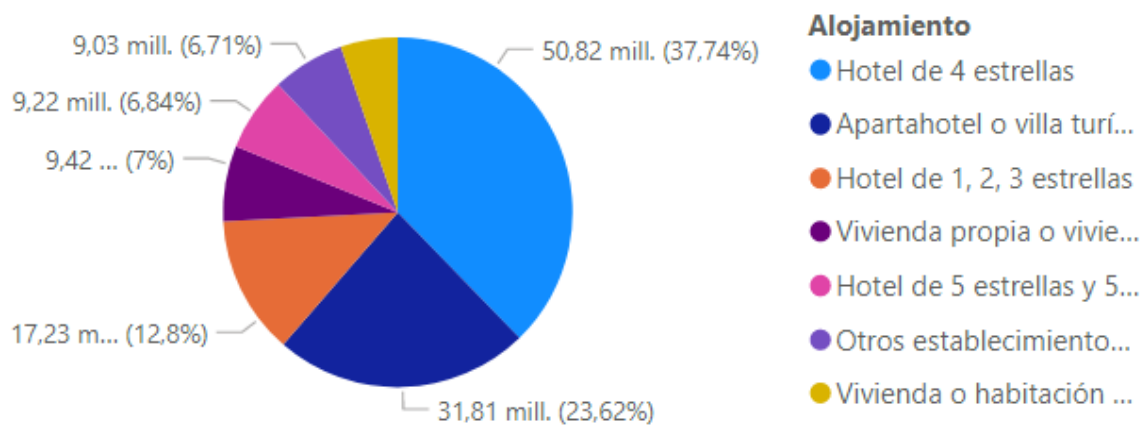
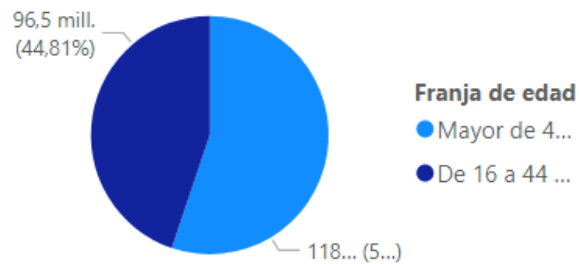


Figura 56: Gráfico circular.

- Otros dos gráficos circulares para mostrar las franjas de edad y el sexo de los turistas.

Nº Turistas por Franja de edad



Nº Turistas por Sexo

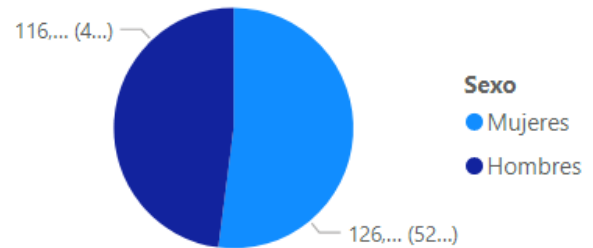


Figura 57: Gráficos circulares.

- Un mapa interactivo para conocer el número de turistas que visitan las cinco islas de las que se recogen datos: Tenerife, Gran Canaria, Fuerteventura, Lanzarote y La Palma.

Nº Turistas por Isla



Figura 58: Mapa interactivo de las Islas Canarias.

- Un gráfico de barras que divide el número de turistas según su nivel de ingresos.

Nº Turistas por Nivel de ingresos



Figura 59: Gráfico de columnas agrupadas.

- Un selector de trimestres y años que muestra los datos asociados a ese periodo de tiempo. Al cambiar de periodo, este selector cambia los datos de todas las gráficas anteriores.

Periodo

- ☒ 2018
- ☐ 2018 Cuarto trimestre
- ☐ 2018 Primer trimestre
- ☐ 2018 Segundo trimestre
- ☐ 2018 Tercer trimestre
- ☐ 2019
- ☐ 2019 Cuarto trimestre
- ☐ 2019 Primer trimestre
- ☐ 2019 Segundo trimestre
- ☐ 2019 Tercer trimestre
- ☐ 2020
- ☐ 2020 Cuarto trimestre
- ☐ 2020 Primer trimestre
- ☐ 2020 Segundo trimestre
- ☐ 2020 Tercer trimestre

Figura 60: Selector de periodos de tiempo.

El conjunto de todas estas visualizaciones dan forma a este dashboard que aprovecha los datos almacenados en el Data Warehouse.

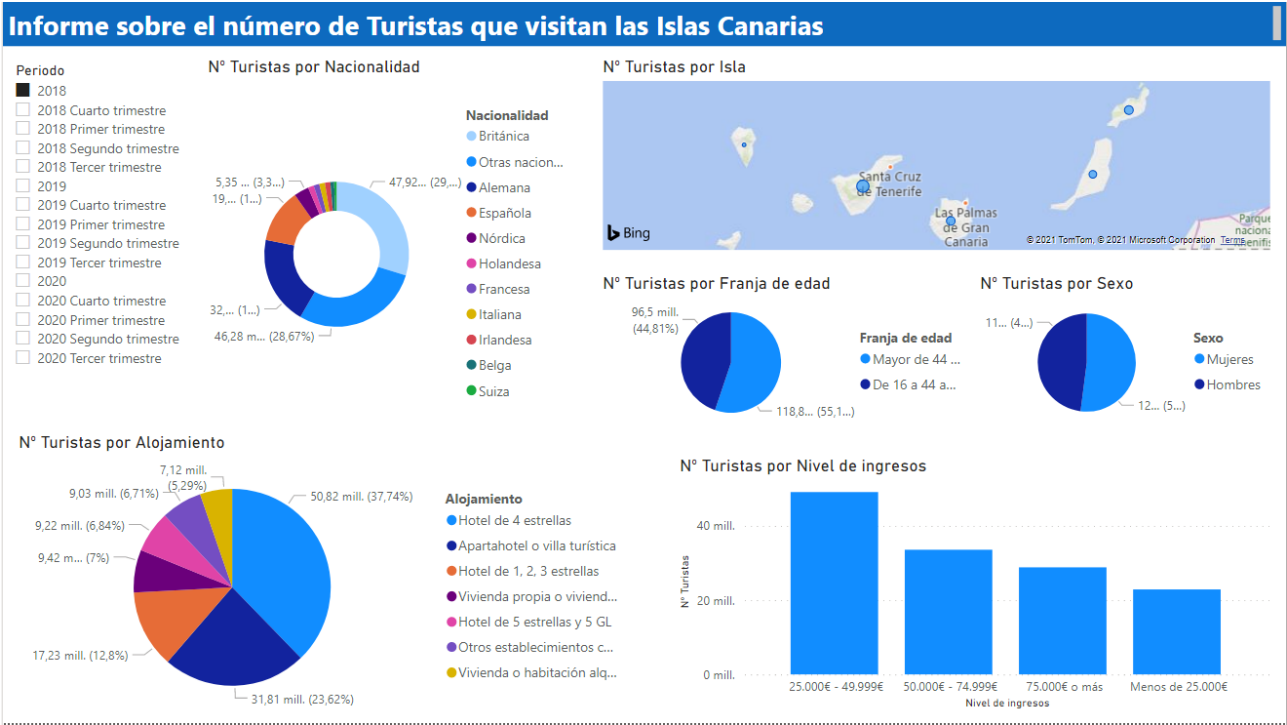


Figura 61: Dashboard con los datos del Data Warehouse.

CAPÍTULO 5

5.1. POSIBLES LÍNEAS FUTURAS DE TRABAJO

Este proyecto puede tener continuidad, dado que se trata de un sistema de Business Intelligence básico y en el que la implementación del Data Warehouse es el asunto principal de este Trabajo de Fin de Grado. A continuación propongo algunas líneas de trabajo y aspectos a mejorar del proyecto.

- **Automatizar la recogida de datos abiertos a través de peticiones HTTP.** La descarga de los archivos del portal de datos abiertos del Gobierno de Canarias se podría automatizar con SQL Server Integration Services y que estos datos se integraran también en sus Datamarts correspondientes.
- **Profundizar en las transformaciones de los procesos ETL.** Se podría realizar transformaciones de los datos más elaboradas para corregir caracteres no reconocidos o transformar los tipos de datos a otros más idóneos.
- **Ampliar el data warehouse con más tablas dimensiones del turista.** El perfil del turista se puede ampliar con más tablas dimensionales de las que se muestran en este proyecto. Se podrían aumentar con los datos ya disponibles en el portal de datos abiertos del Gobierno de Canarias.
- **Añadir otras tablas de hechos con temas relacionados.** Al estudio sobre el perfil del turista se podrían unir otros asuntos relacionados como como los alojamientos, los centros turísticos, el alquiler de coches, etc.

5.2. CONCLUSIONES

Con el desarrollo, la investigación y la redacción de este Trabajo de Fin de Grado he conseguido entender varios aspectos sobre el desarrollo de Data Warehouse que detallo en las siguientes líneas.

- **La diferencia entre los conceptos teóricos y las tecnologías a utilizar.** En todo lo relacionado con la tecnología de la información y las comunicaciones es muy común usar tecnicismos que no todos entendemos. Hay un vocabulario técnico que, en muchas ocasiones, se confunde su significado incluso cuando se usa entre profesionales. Palabras como big data, blockchain o ransomware son cada vez

más frecuente en la prensa común. Cuando comencé con este proyecto, mi concepto de Business Intelligence era equivocado. Ahora entiendo que es un concepto que se refiere a un sistema basado en un tipo de almacenamiento particular conocido por Data Warehouse y que sirve para aprovechar la información almacenada, siendo modelada y representada gráficamente. Otro punto interesante es que un sistema de Business Intelligence se puede desarrollar en varias tecnologías. Se puede tener un Data Warehouse con bases de datos gestionadas con SQL Server y representar los datos en otros software distintos de Power BI: Tableau, Qlik; o usar otros software de almacenamiento de datos como los servicios de Microsoft Azure y Big Query, entre otras plataformas.

- **La utilidad de un Data Warehouse.** Cuando hablamos de almacenamiento de datos, es muy posible que pensemos en bases de datos. Existen varias formas de clasificar los tipos de bases de datos: si son relacionales o no, por su finalidad o por el servidor de bases de datos utilizado. El término de Data Warehouse se descrito a lo largo de este documento como un tipo específico de almacenamiento formado, como mínimo por una base de datos con tablas relacionales y que es el protagonista de un sistema de Business Intelligence.
- **La manera de elaborar un Data Warehouse.** Su implementación requiere de un enfoque particular como propone el modelo de Ralph Kimball, descrito en este documento, y el modelo de William H. Inmon.
- **La ventaja de la visualización de modelos de datos.** Los datos almacenados no sirven de mucho por sí solos. La forma de aprovecharlos es trabajarlos en modelos de datos, crear visualizaciones e informes basados en ellos. De esta manera, entenderemos mejor el contexto de los datos y descubriremos comportamientos y patrones con información objetiva.
- **Las posibilidades de analizar datos y establecer conclusiones.** Posibilidad de realizar estudios y llegar a conclusiones (tasa de parados por municipios y por edad, localización de hospitales e incidencia por Covid, tasa de parados y políticas de empleo y formación.)
- Con el análisis de los datos a través de un sistema de Business Intelligence, es posible realizar estudios basados en datos, plantear hipótesis y llegar a conclusiones respaldadas con datos. Supongamos un estudio de datos abiertos

donde se analicen la tasa de parados por municipios, sus franjas de edad y sus perfiles profesionales ¿Podrían ser los datos concluyentes de este estudio un motivo para orientar mejor las políticas de empleo y configurarlas para ciertos municipios? ¿Podríamos relacionar la incidencia de Covid-19 con la geolocalización de los centros sanitarios? ¿Nos imaginamos a las Administraciones Públicas usando análisis de datos para medir el éxito de sus decisiones? ¿Y los partidos políticos basando sus propuestas en este tipo de estudios?

El uso de un Data Warehouse y, a su vez, de un sistema de Business Intelligence para hacer estudios basado en datos abre un campo de oportunidades en varios ámbitos. En el sector público, los datos públicos permitirán fiscalizar la actividad pública. Una muestra de ello es el ingeniero informático Jaime Gómez Obregón ([Twitter: @JaimeObregon](https://twitter.com/JaimeObregon)) que ha publicado la web <https://contratosdecantabria.es> basada en los contratos públicos del Gobierno de Cantabria. En el sector privado, las empresas generan datos que están empezando a utilizarse para tomar decisiones hacia una mejor competitividad. Y eso significa una mayor demanda de especialistas en administración de bases de datos, programadores, analistas y científicos de datos. Y las iniciativas de Datos Abiertos pueden generar herramientas útiles para la sociedad, nuevas oportunidades de negocio y nuevos puestos de empleo.

BIBLIOGRAFÍA Y REFERENCIAS

1. Stephen Baker (2008). *The Numerati*. Editorial Planeta Mexicana.
2. Juan Manuel López Zafra y Ricardo A. Queralt Sánchez de las Matas (2019). *Alquimia: como los datos se están transformando en oro*. Editorial Planeta.
3. PowerData Solutions SL (20 de septiembre de 2021). *Big Data: ¿En qué consiste? Su importancia, desafíos y gobernabilidad*. <https://www.powerdata.es/big-data>
4. PowerData Solutions SL (20 de septiembre de 2021). *Data lake: definición, conceptos clave y mejores prácticas*. <https://www.powerdata.es/data-lake>
5. Wikipedia (20 de septiembre de 2021). *Inteligencia empresarial*. https://es.wikipedia.org/wiki/Inteligencia_empresarial
6. Open Knowledge Foundation (17 de septiembre de 2021). *El manual de Open Data*. <https://opendatahandbook.org/>
7. Microsoft (17 de septiembre de 2021). *Documentación oficial de Microsoft*. <https://docs.microsoft.com/es-es/>
8. Educba (17 de septiembre de 2021). *Data Warehouse Tutorial*. <https://www.educba.com/data-science/data-science-tutorials/data-warehouse-tutorial/>
9. Artículo de Kimball Group (17 de septiembre de 2021) *Data Warehouse fácil*. <https://datawarehouse.es/>
10. Exforsys Inc (17 de septiembre de 2021). *Data Warehousing*. <http://www.exforsys.com/tutorials/data-warehousing.html>
11. Linkedin Learning (diciembre de 2020). *Implementing a Data Warehouse SQL Server 2019*. <https://www.linkedin.com/learning/implementing-a-data-warehouse-sql-server-2019>
12. Blog BI-Geek (2 de mayo de 2016). *Arquitectura BI: Comparativa entre Inmon y Kimball*. <https://blog.bi-geek.com/arquitectura-comparativa-inmon-y-kimball/>
13. Euclides Silva Peñafiel, Verónica Marcela Zapata Yáñez, Morales Guamán Klever Patricio y Toaquiza Padilla Luis Marcelo (10 de septiembre de 2019). *Análisis de metodologías para desarrollar Data Warehouse aplicado a la toma de decisiones*

[Archivo PDF].

<https://cienciadigital.org/revistacienciadigital2/index.php/CienciaDigital/article/view/922>

14. Wikipedia (17 de septiembre de 2021). *Modelado dimensional*.

https://es.wikipedia.org/wiki/Modelado_dimensional