



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escuela Técnica Superior de Ingeniería Informática

Desarrollo de una herramienta de análisis económico para
la toma de decisiones en cooperativas agroalimentarias de
la Comunidad Valenciana

Trabajo Fin de Grado

Grado en Ingeniería Informática

AUTOR/A: Peláez Bover, Luis

Tutor/a: Martínez Plumed, Fernando

CURSO ACADÉMICO: 2022/2023

Resum

Aquest projecte s'enfoca en el desenvolupament d'una eina d'anàlisi econòmica per a cooperatives agroalimentàries a la Comunitat Valenciana, que permeta a la Conselleria d'Agricultura i a les mateixes cooperatives accedir i analitzar informació comptable per prendre decisions informades. L'objectiu principal del projecte és crear una plataforma intuïtiva i fàcil d'usar, basada en Power BI, que permeta als usuaris consultar i analitzar dades econòmiques de les cooperatives agroalimentàries de la Comunitat Valenciana mitjançant una interfície visual i personalitzable.

Per aconseguir això, es va desenvolupar un magatzem de dades amb la informació dels comptes econòmics del Registre Mercantil de totes les cooperatives agroalimentàries de la regió, i es va crear un esquema relacional en forma d'estrella per poder-hi treballar. L'eina ofereix dues versions, una per a la mateixa Conselleria i una altra per a les cooperatives, amb la diferència que la Conselleria pot consultar dades de totes les cooperatives mentre que cada cooperativa només pot accedir a les dades pròpies. A partir de les dades comptables, es calculen una sèrie de ràtios econòmiques que mostren la situació de la cooperativa, i es presenten en una interfície interactiva en forma de diversos gràfics i visuals. Els usuaris poden aplicar filtres sobre la informació i comparar-se de diverses maneres amb la resta de les cooperatives. A partir de les ràtios, es generen de forma automàtica anàlisis i recomanacions personalitzades per a la situació de cada cooperativa en forma de text per millorar la presa de decisions per part de la seua gerència. El projecte s'ha fet a petició de la Càtedra d'Integració Empresarial de Cooperatives Agroalimentàries (CATIECA) de la UPV.

Paraules clau: Intel·ligència de Negoci, Magatzem de Dades, Anàlisi Econòmica, Cooperatives Agroalimentàries, Power BI, Recomanacions Personalitzades, Presa de Decisions

Resumen

Este proyecto se enfoca en el desarrollo de una herramienta de análisis económico para cooperativas agroalimentarias en la Comunidad Valenciana, que permita a la Conselleria de Agricultura y a las cooperativas mismas, acceder y analizar información contable para tomar decisiones informadas. El objetivo principal del proyecto es crear una plataforma intuitiva y fácil de usar, basada en Power BI, que permita a los usuarios consultar y analizar datos económicos de las cooperativas agroalimentarias de la Comunidad Valenciana a través de una interfaz visual y personalizable.

Para lograr esto, se desarrolló un almacén de datos con la información de las cuentas económicas del Registro Mercantil de todas las cooperativas agroalimentarias de la región, y se creó un esquema relacional en forma de estrella para poder trabajar sobre ellos. La herramienta ofrece dos versiones, una para la propia Conselleria y otra para las cooperativas, con la diferencia de que la Conselleria puede consultar datos de todas las cooperativas mientras que cada cooperativa solo puede acceder a sus propios datos. A partir de los datos contables, se calculan una serie de ratios económicos que muestran la situación de la cooperativa, y se presentan en una interfaz interactiva en forma de diversos gráficos y visuales. Los usuarios pueden aplicar filtros sobre la información y compararse en diversas formas con el resto de las cooperativas. A partir de los ratios, se generan de forma automática análisis y recomendaciones personalizadas para la situación de cada cooperativa en forma de texto para mejorar la toma de decisiones por parte

de la gerencia de las mismas. El proyecto se ha llevado a cabo a petición de la Cátedra de Integración Empresarial de Cooperativas Agroalimentarias (CATIECA) de la UPV.

Palabras clave: Inteligencia de Negocio, Almacén de Datos, Análisis Económico, Cooperativas Agroalimentarias, Power BI, Recomendaciones Personalizadas, Toma de Decisiones

Abstract

This project focuses on the development of an economic analysis tool for agri-food cooperatives in the Comunidad Valenciana, allowing the regional Ministry of Agriculture and the cooperatives themselves to access and analyse accounting information in order to make informed decisions. The main objective of the project is to create an intuitive and easy-to-use platform, based on Power BI, that allows users to consult and analyse the economic data of agri-food cooperatives in the Valencian Community through a visual and customisable interface.

To achieve this, a data warehouse was developed with information from the economic accounts of the Commercial Registry of all the agri-food cooperatives in the region, and a relational schema in the form of a star was created to work with it. The tool offers two versions, one for the Regional Ministry itself and the other for the cooperatives, with the difference that the Regional Ministry can consult the data of all the cooperatives, while each cooperative can only access its own data. From the accounting data, a series of economic indicators are calculated that show the situation of the cooperative and are presented in an interactive interface in the form of various graphs and visualisations. Users can filter the information and compare themselves with other cooperatives in various ways. From the key figures, customised analyses and recommendations for each cooperative's situation are automatically generated in text form to improve decision-making by the cooperative's management. The project was carried out at the request of the Cátedra de Integración Empresarial de Cooperativas Agroalimentarias (CATIECA) of the UPV.

Key words: Business Intelligence, Data Warehousing, Economic Analysis, Agri-Food Cooperatives, Power BI, Customized Recommendations, Decision Making

Índice general

Índice general	V
Índice de figuras	VII
Índice de tablas	VIII
<hr/>	
1 Introducción	1
1.1 Motivación	1
1.2 Objetivos	2
1.3 Impacto esperado	3
1.4 Estructura	3
1.5 Colaboraciones	4
2 Estado del arte	5
2.1 <i>Business Intelligence</i>	5
2.1.1 Fuentes de datos	5
2.1.2 ETL	6
2.1.3 Fase de análisis	7
2.1.4 Fase de visualización	8
2.1.5 Toma de decisiones	8
2.1.6 Problemas en el proceso de BI	9
2.2 Almacenes de datos y alternativas	9
2.2.1 Formatos de orígenes de datos	9
2.2.2 Almacenes de datos	10
2.2.3 Esquemas para almacenes	11
2.2.4 OLAP	15
2.2.5 <i>Data lakes</i>	17
2.2.6 <i>Data Mesh</i>	18
2.2.7 Bases de datos NoSQL	20
2.3 Nuevas tendencias	21
2.3.1 Uso de la nube	21
2.3.2 Inteligencia artificial y <i>Natural Language Processing</i>	22
2.3.3 Procesamiento y análisis de datos en tiempo real, arquitecturas Kappa y Lambda	22
2.4 Soluciones software para BI	23
2.4.1 Servicios	24
2.4.2 Herramientas de programación	26
3 Análisis del problema	31
3.1 Las cooperativas agroalimentarias de la Comunidad Valenciana	31
3.2 Análisis de objetivos, requisitos y asunciones	32
3.3 Análisis de riesgos	34
3.4 Alternativas, presupuesto y elección	35
4 Metodología	37
4.1 Power BI	37
4.2 Herramientas de ETL	38

4.3	Equipo y seguimiento	39
4.4	Plan de trabajo	40
5	Diseño de la solución	43
5.1	Arquitectura de la plataforma	43
5.2	Base de datos	44
6	Desarrollo de la solución	49
6.1	Formato archivo Excel	49
6.2	Creación base de datos	51
6.3	Tablas de medidas	53
6.3.1	Cuartiles	54
6.3.2	Cuartiles acumulados	55
6.3.3	Cuartiles último año	55
6.3.4	Valores último año	55
6.3.5	Último año disponible	56
6.3.6	Primer año disponible	56
6.3.7	Tasas de variación	56
6.4	Textos en lenguaje natural	56
6.4.1	Textos Análisis	57
6.4.2	Textos condicionales	58
6.5	Interfaz visual	59
6.5.1	Filtros y marcadores	59
6.5.2	Menú principal	60
6.5.3	Análisis individual de cooperativas	61
6.5.4	Páginas sector	65
6.5.5	Página territorial	67
6.5.6	Diseño gráfico y ayuda contextual	69
6.5.7	Visuales personalizados	70
6.6	Inicio de sesión cooperativas	71
6.6.1	RLS en Microsoft Power BI	71
6.6.2	Visual personalizado para inicio de sesión	72
6.6.3	Versiones de la herramienta	75
6.7	Publicación	76
7	Resultado	77
8	Conclusiones	79
9	Trabajos futuros	81
	Bibliografía	83
<hr/>		
	Apéndices	
A	ODS	85
B	Código programa Excel	87

Índice de figuras

2.1	Proceso de BI en forma de pirámide	6
2.2	Flujo ETL	7
2.3	Arquitectura almacén de datos	11
2.4	Esquema estrella	12
2.5	Esquema copo de nieve	13
2.6	Esquema constelación	14
2.7	Cubo OLAP	15
2.8	Operaciones OLAP	16
2.9	Esquema <i>data lake</i>	18
2.10	Domino <i>Data Mesh</i>	19
2.11	Arquitectura <i>Data Mesh</i>	20
2.12	Arquitecturas Kappa y Lambda	24
2.13	Interfaz Power BI	25
2.14	Pantalla Power BI Community	26
2.15	Ejemplo Jupyter Notebook	28
2.16	Gráfico evolución herramientas programación para BI	30
2.17	Tabla comparativa herramientas de programación para BI	30
4.1	Diagrama de Gantt	40
4.2	Tabla diagrama de Gantt	40
5.1	Flujo de datos	43
5.2	Modelo relacional base de datos	44
5.3	Tablas de hechos	45
5.4	Tabla DIMCooperativa	46
5.5	Tabla DIMFecha	46
6.1	Primera versión Excel	50
6.2	Excel origen de datos	51
6.3	Pasos tabla DIMFecha	52
6.4	Pasos tablas FACTGeneral y FACTCooperativa	52
6.5	Pasos tabla DIMCooperativa	53
6.6	Tablas de medidas	54
6.7	Ejemplo situación cooperativa	57
6.8	Ejemplos filtros	59
6.9	Menú principal	60
6.10	Ayuda contextual menú principal	61
6.11	Ventana tutorial	61
6.12	Páginas análisis individual cooperativa 1	62
6.13	Páginas análisis individual cooperativa 2	63
6.14	Información contextual pantalla	65
6.15	Aplicación de filtros	66
6.16	Página ratios financieros sector	67
6.17	Página análisis sectorial	67

6.18	Página análisis territorial	68
6.19	Paleta de colores	69
6.20	Ejemplo de <i>tooltips</i>	70
6.21	Text Search Slicer	72
6.22	Tabla codigos	73
6.23	Error <i>show as table</i>	75
6.24	Solución error <i>show as table</i>	75
6.25	Flujo de datos versión cooperativas	76

Índice de tablas

3.1	Requisitos objetivo G-1	32
3.2	Requisitos objetivo G-2	32
3.3	Requisitos objetivo G-3	33
3.4	Requisitos objetivo G-4	33
3.5	Requisitos objetivo G-5	33
3.6	Requisitos objetivo G-6	33
3.7	Requisitos no funcionales	34
3.8	Asunciones de dominio	34
3.9	Riesgos	34

CAPÍTULO 1

Introducción

El análisis de la situación económica de las cooperativas agroalimentarias es un aspecto indispensable para la toma de decisiones informadas y estratégicas en cualquier organismo. Sin embargo, a menudo la falta de acceso a la información necesaria o la dificultad para interpretarla puede ser un obstáculo para las entidades interesadas. En este contexto surge la necesidad de desarrollar una herramienta que simplifique y facilite el entendimiento de estas informaciones económicas.

Este proyecto surge a partir de la colaboración con la Cátedra de Integración Empresarial de Cooperativas Agroalimentarias (CATIECA) de la Universidad Politécnica de Valencia (UPV) y la Conselleria d'Agricultura, Ramaderia i Pesca de la Comunidad Valenciana. El objetivo es desarrollar una herramienta basada en Power BI, intuitiva y fácil de usar, que permita a la Conselleria de Agricultura y a las cooperativas de la Comunidad Valenciana acceder y analizar información de carácter contable de una forma clara y sencilla.

El aspecto más relevante del proyecto es generar un sistema de información para el análisis y la toma de decisiones que aporte la información económica actualizada de las cooperativas agroalimentarias de la Comunidad Valenciana. Este sistema de información se generará a partir de las cuentas económicas del Registro Mercantil de todas las cooperativas agroalimentarias de la región. A partir de este recurso, se aspira a calcular una serie de ratios económicos que ayuden a diagnosticar la situación de cada cooperativa y, así, contribuir a la toma de decisiones informadas. Es fundamental que cada cooperativa solo tenga acceso a sus propios datos, mientras que la Conselleria pueda visualizar la información de todas las cooperativas de la región.

Es relevante destacar la oportunidad que este proyecto representa en términos académicos y a nivel público para la Conselleria, pues les permitirá tener acceso a una base de datos amplia y actualizada, además de una herramienta que facilite la comprensión de la información económica. Este es solo el primer paso de lo que se espera será un proyecto de largo plazo, que se desarrolle y amplíe con el tiempo.

1.1 Motivación

Este TFG se realiza en el seno de CATIECA (Cátedra de Integración Empresarial de Cooperativas Agroalimentarias) de la FADE. La presente memoria describe el trabajo que ha llevado a cabo el alumno durante su estancia de prácticas en la cátedra. Las prácticas se centran en el desarrollo de una herramienta de análisis económico para la toma de decisiones en cooperativas agroalimentarias de la Comunidad Valenciana.

CATIECA tiene contacto constante con la Conselleria d'Agricultura, Ramaderia i Pesca. De esta colaboración surge la idea de crear una herramienta de este tipo. Desde la cátedra se ponen en marcha y buscan a un alumno de prácticas para que se encargue de la parte técnica de la creación de la misma. Desde ese momento nos ponemos a trabajar en el desarrollo.

Personalmente, es un trabajo que me ha motivado mucho ya que al final hemos conseguido un producto tangible y que será lanzado al público para su uso. Este tipo de trabajos me resultan mucho más satisfactorio ya que podemos notar que estamos ayudando a los usuarios (en este caso cooperativas). Es un desarrollo que he realizado prácticamente en solitario de principio a fin, por lo cual siento que he aportado mucho y que podido tomar las decisiones que he querido.

Además, la herramienta ha generado interés externo, lo que siempre aporta a la motivación. El hecho de que sea un proyecto para un organismo público, como es la Conselleria ya nos dice algo. Pero hay que destacar que también han mostrado interés en la herramienta otras personas como algunos responsables de los Registros Mercantiles de las provincias de la Comunitat, así como del Registro Mercantil Central. También lo han hecho los responsables de algunas de las cooperativas más importantes de la Comunitat Valenciana. El proyecto será presentado el próximo 29 de septiembre de 2023 en un *workshop* que organiza CATIECA en la FADE, donde estarán presentes algunas de las personas interesadas mencionadas además de los profesionales del sector. Todo ello, aporta mucho a la motivación necesaria para realizar el proyecto desde cero, ya que pensamos que si se implementa de forma correcta y se extiende su uso, puede ser muy interesante de cara a los usuarios finales. Este desarrollo es solo el primer paso en este proyecto, ya que se pretende continuar con él a largo plazo.

1.2 Objetivos

No nos extenderemos mucho en este apartado ya que la sección 3.2 es específica sobre los objetivos y sus requisitos asociados teniendo en cuenta ya el problema expuesto. Partimos del sector de las cooperativas agroalimentarias de la Comunitat Valenciana, el cual tiene problemas. Muchas cooperativas cierran o continúan funcionando a duras penas, principalmente debido a problemas económicos y financieros. En muchos casos, esto se junta con que los responsables de las mismas no son profesionales de las finanzas o la gestión de empresas, y les resulta difícil identificar sus fallos y ver en qué aspectos deben mejorar. Podemos englobar todos los objetivos en uno: ayudar a las cooperativas de la Comunidad Valenciana a la toma de decisiones informada a través de su información económico-financiera para mejorar su situación. Esto resume el fin último de la herramienta.

Hay algunos objetivos secundarios. Por ejemplo, la creación de una base de datos con la información contable de los últimos años de las cooperativas de la Comunidad Valenciana. Ésta, se usará para nutrir nuestra herramienta, pero está pensada para ser reutilizada en otros proyectos académicos. Es una base de datos amplia del sector que no existe actualmente. También tenemos como objetivo generar una herramienta que sirva para el estudio y el diagnóstico, no solo para las propias cooperativas si no también en el ámbito académico (para la UPV) y en ámbito público (la Conselleria). Aún así, estos son objetivos que surgen del fin inicial que hemos mencionado anteriormente.

La forma de lograr los objetivos también es importante. Queremos crear un herramienta que sea atractiva y, sobre todo, accesible a nivel de usabilidad pensando en cómo son nuestros usuarios finales, ya que tienen un perfil algo específico, sin mucha formación especializada en la materia.

1.3 Impacto esperado

Resulta difícil determinar cuál será el impacto de la herramienta, ya que es algo completamente nuevo en el sector. Creemos que con una base de datos completa y un uso adecuado, puede ser de gran utilidad para las cooperativas. El interés que ha generado hace pensar que tendrá una cierta acogida inicial. No es una herramienta que vaya a tener un impacto inmediato en el sector, ni mucho menos, pero a medio y largo plazo puede ayudar a los responsables de las cooperativas en ciertos aspectos de su estructura económica y de financiación que les ayude a ser más rentables, y por ende, más competitivos.

Aún así, debemos tener en cuenta que esta es una primera versión de la herramienta, y va a funcionar en cierta forma como piloto para comprobar si es algo que suscita interés entre los usuarios finales. La buena noticia es que, de ser así, tenemos una primera versión sólida pero que tiene margen para mejoras: refinando las funcionalidades existentes e implementando nuevas. Confiamos en que sea el inicio de un proyecto que perdure a largo plazo, colaborando estrechamente con las cooperativas y la Consellería.

1.4 Estructura

Pasemos a comentar brevemente la estructura de la presente memoria. Hemos seguido en gran parte las recomendaciones de la ETSINF en cuanto a estructura en este tipo de trabajos. Tras la presente sección de introducción pasaremos a hablar del marco teórico en el estado del arte. En él, hablaremos de todos los temas que tienen relación con el actual desarrollo, además de algunas de las nuevas tendencias en la materia. Nos centraremos en el *Business Intelligence* (BI) y todos los aspectos técnicos que lo rodean. Abordaremos también de manera extensa la materia de las bases de datos.

A continuación, pasaremos a analizar el problema. Expondremos el contexto del desarrollo y hablaremos de los objetivos, los requisitos, las asunciones y los riesgos. Acabaremos viendo las diferentes alternativas que hemos considerado para el desarrollo y nuestra elección final. Seguiremos con la metodología del desarrollo. Hablaremos de la herramienta que vamos a usar para el mismo (Power BI), y de cómo se ha organizado el equipo de trabajo, tanto para el seguimiento como en el plan de trabajo establecido.

Con esto, podremos pasar a ver el diseño y el desarrollo de la herramienta en detalle. En el diseño, mostraremos como hemos armado la arquitectura del despliegue y la base de datos subyacente. En la parte de desarrollo, detallaremos el proceso de creación: primero de la base de datos y de toda la información necesaria para la herramienta; y después de la interfaz gráfica para su uso. Acabaremos esta parte hablando de las distintas versiones y su forma de publicación en la web.

Por último, comprobaremos cual ha sido el resultado final de todo el proyecto, teniendo en cuenta los objetivos, requisitos y riesgos que nos habíamos marcado. Finalizaremos con una conclusión y un apunte sobre cómo se mejorará la herramienta en la sección de trabajos futuros. Está presente también la correspondiente bibliografía.

En cuanto a los anexos tenemos dos. El anexo A es el habitual sobre los ODS común a todos los TFGs de la UPV. El anexo B contiene el código Java de un programa necesario en el desarrollo de la herramienta.

1.5 Colaboraciones

El proyecto se ha llevado a cabo en colaboración con CATIECA como hemos comentado. Por el equipo han ido rotando varios integrantes de manera temporal, en particular otros alumnos que se han encargado de hacer algunas tareas específicas de recolección de datos y diseño gráfico, como iremos apuntando cuando corresponda a lo largo de la memoria. Sin embargo, hay cuatro miembros del equipo que han estado en el proyecto de principio a fin. Elena Meliá Martí y Natalia Lajara de Camilleri, profesoras de la FA-DE y responsables de la cátedra, se han encargado de la coordinación y de la realización de tareas analíticas en materia financiera. Fernando Martínez Plumed, tutor del presente trabajo y que ha sido el representante de la ETSINF en la cátedra para coordinar el proyecto. Por último, Luis Peláez Bover, autor del TFG, que se ha encargado del desarrollo de la herramienta. Agradecer enormemente a todos los miembros del equipo que han participado en el proyecto su colaboración para sacarlo adelante.

CAPÍTULO 2

Estado del arte

2.1 *Business Intelligence*

La inteligencia de negocios, BI (*Business Intelligence*) [Negash, 2004], o inteligencia empresarial, es un campo de la informática que se enfoca en la recopilación, análisis y presentación de datos para ayudar a las empresas a tomar decisiones informadas y estratégicas. El objetivo principal del BI es convertir datos en información útil que pueda ayudar a las empresas a mejorar su eficiencia, aumentar sus ingresos y reducir sus costos. El proceso de BI incluye la recolección de datos de múltiples fuentes, la transformación de estos datos en información útil mediante técnicas de análisis y modelado de datos, y la presentación de esta información de manera clara y comprensible para los usuarios finales a través de herramientas de visualización de datos. Las empresas utilizan el BI para analizar sus operaciones y procesos internos, comprender mejor a sus clientes y el mercado en el que operan, y para identificar nuevas oportunidades de negocio. Se ha convertido en una herramienta indispensable para las empresas modernas que buscan mantenerse competitivas en un entorno empresarial cada vez más complejo y dinámico por diversas razones. El BI ayuda a las empresas a tomar decisiones basadas en datos y hechos en lugar de suposiciones o conjeturas.

En la parte interna de la empresa pueden utilizar el BI para analizar sus procesos internos y mejorar la eficiencia operativa. Al identificar cuellos de botella, ineficiencias y áreas de mejora, las empresas pueden implementar cambios para mejorar su eficiencia y reducir sus costos.

A nivel externo, ayuda a las empresas a identificar nuevas oportunidades de negocio y a detectar patrones y tendencias en el mercado. Esto permite a las empresas anticiparse a los cambios en el mercado y tomar medidas para capitalizarlos. Además, pueden comprender mejor las necesidades y comportamientos de sus clientes. Al analizar los datos de los clientes, las empresas personalizan su oferta de productos y servicios para mejorar la experiencia del cliente en general.

2.1.1. Fuentes de datos

Antes de pasar a exponer el proceso de recogida de datos, es útil que distingamos entre fuentes de datos estructuradas y no estructuradas ya que serán mencionadas varias veces. Las fuentes de datos estructuradas son datos que se pueden organizar y clasificar fácilmente en un formato tabular o en una base de datos relacional. Estos datos suelen ser numéricos o alfanuméricos y tienen un formato predefinido. Algunos ejemplos de fuentes de datos estructuradas son las ventas por producto, los ingresos por región y los registros de clientes.

Por otro lado, las fuentes de datos no estructuradas son datos que no tienen un formato predefinido y no se pueden organizar fácilmente en un formato tabular. Pueden ser archivos de texto, correos electrónicos, imágenes, videos y mensajes de redes sociales. Las fuentes de datos no estructuradas representan un desafío para las empresas ya que requieren herramientas y tecnologías especializadas para su análisis. En la actualidad, están ganando importancia debido al creciente volumen de información no estructurada que se genera diariamente en la web y las redes sociales. Para abordar este desafío, las empresas están adoptando tecnologías como la minería de datos y el procesamiento de lenguaje natural para extraer información valiosa [IBM, 2021].

Es importante destacar que muchas fuentes de datos empresariales contienen tanto datos estructurados como no estructurados. Por ejemplo, los registros de ventas pueden incluir comentarios de los clientes que contienen información no estructurada. Por lo tanto, es esencial contar con herramientas de BI que puedan manejar ambos tipos.

2.1.2. ETL

Pasando ahora al proceso de BI, este es un conjunto de etapas que tienen como objetivo transformar los datos en información útil y significativa para la empresa. El proceso se compone de varias fases que trabajan sobre los datos: recolección, transformación, análisis, visualización y toma de decisiones. La figura 2.1 nos muestra este proceso en un esquema piramidal donde empezamos por la base.

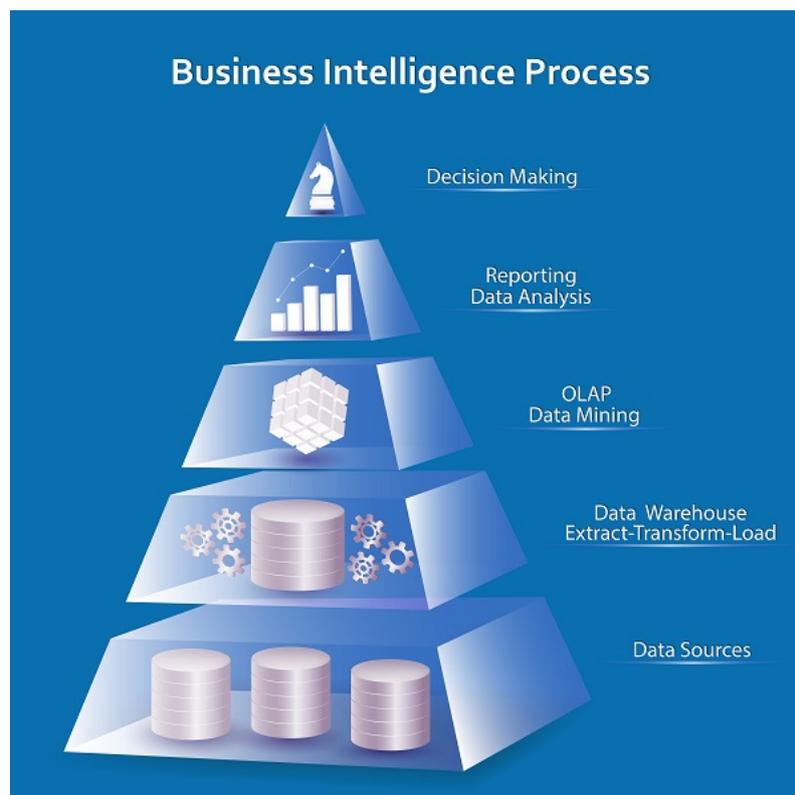


Figura 2.1: Representación del proceso de BI en forma de pirámide [Chatterjee, 2017].

Las dos primeras fases suelen ser llamadas más habitualmente proceso ETL. El proceso ETL es una parte integral del proceso de BI, y es la forma en que se transforman los datos en información útil y significativa. ETL significa Extracción, Transformación y Carga (*Extraction, Transformation, and Loading*), y describe los pasos necesarios para procesar los datos y prepararlos para su uso en informes y análisis.

La primera fase del proceso ETL es la extracción de datos de diferentes fuentes, incluyendo bases de datos internas, sistemas ERP (planificación de recursos empresariales), CRM (gestión de relaciones con los clientes), archivos de texto, redes sociales y otros. La extracción puede implicar el uso de herramientas de integración de datos que pueden extraer datos automáticamente de diferentes fuentes y combinarlos en una sola área de almacenamiento temporal.

Una vez que se han extraído los datos, comienza la fase de transformación, que es el proceso de limpiar, preparar y convertir los datos en un formato adecuado para su análisis. La transformación incluye actividades como la eliminación de datos duplicados, la normalización de datos, la corrección de errores, la eliminación de información irrelevante y la agregación de datos. También puede implicar la creación de nuevos campos y cálculos para hacer que los datos sean más útiles y significativos.

Transformados ya los datos, se cargan en una base de datos o *data warehouse*. La carga puede implicar la inserción de los datos en diferentes tablas o áreas de almacenamiento según su relevancia y uso. La carga también puede incluir la validación de datos y la ejecución de procesos de limpieza de datos adicionales [Martínez et al., 2013] [Doan et al., 2012] [Rizzi and Golfarelli, 2009].

El proceso ETL es esencial para el éxito del proceso de BI, ya que es el medio por el cual se transforman los datos en información útil y significativa. El proceso puede ser automatizado y programado para que se ejecute en intervalos regulares, lo que garantiza que los datos estén actualizados y sean precisos en todo momento. La calidad de los datos también es esencial para el éxito del proceso de ETL. Podemos ver un esquema detallado del flujo de este proceso en la figura 2.2, donde se usa un ejemplo con diversas fuentes de datos y estados intermedios en la transformación para finalmente dar dos resultados distintos en la salida dependiendo de la finalidad del uso de los datos.

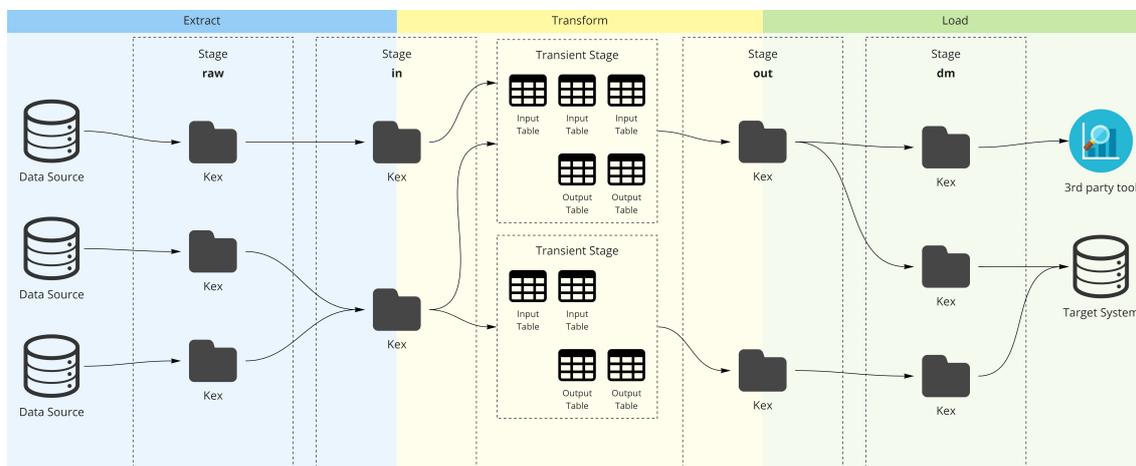


Figura 2.2: Flujo detallado de un ejemplo de proceso ETL [Bizzflow, 2021].

2.1.3. Fase de análisis

Se continúa con la fase de análisis de datos. Esta implica el uso de herramientas y técnicas de análisis para examinar los datos y descubrir patrones, tendencias y relaciones que puedan proporcionar información valiosa para la toma de decisiones empresariales. Podemos distinguir varios tipos de análisis, aunque se suele empezar con una fase de exploración inicial. En ésta, se examina el conjunto de datos para descubrir patrones y tendencias, así como para identificar posibles problemas y oportunidades. Posteriormente se pueden realizar:

- **Análisis descriptivo:** se utiliza para describir los datos y proporcionar información sobre el rendimiento empresarial pasado. Esta etapa puede incluir el uso de *dashboards*, paneles de control e informes para proporcionar una visión general de los datos.
- **Análisis predictivo:** sirve para predecir el comportamiento futuro en función de patrones históricos. Podemos hacer uso de técnicas como el análisis de series temporales para prever la demanda del mercado, el comportamiento del cliente y el rendimiento financiero.
- **Análisis prescriptivo:** usado para recomendar acciones futuras en función de los resultados del análisis predictivo. Se usan herramientas como simulaciones y modelos de toma de decisiones para recomendar estrategias y acciones para mejorar el rendimiento empresarial.

Además, se pueden hacer uso de algunas herramientas adicionales, especialmente para datos no estructurados. Por ejemplo, las herramientas de minería de datos se utilizan para analizar grandes conjuntos de datos y descubrir patrones y relaciones ocultas. También tenemos herramientas de análisis de texto que pueden ser útiles para analizar el sentimiento de los clientes en las redes sociales, comentarios de los clientes en los sitios web y otras fuentes de datos no estructurados [El Morr and Ali-Hassan, 2019].

2.1.4. Fase de visualización

Siguiendo con la fase de visualización de datos, en el proceso de BI se enfoca en presentar los resultados del análisis de datos de una manera fácil de entender para los usuarios finales. El objetivo es hacer que los datos sean accesibles y comprensibles para el personal de la empresa. Las herramientas de visualización de datos permiten a los usuarios crear gráficos, tablas y otros tipos de visualizaciones que muestren los datos de manera clara y concisa. Estas visualizaciones pueden ser interactivas, lo que permite a los usuarios explorar los datos y obtener una comprensión más profunda de los patrones y tendencias.

Debe tenerse en cuenta que la visualización de datos debe ser clara y fácil de entender. Las visualizaciones sobrecargadas o confusas pueden llevar a malentendidos y decisiones erróneas. Por lo tanto, es fundamental seleccionar cuidadosamente las visualizaciones adecuadas para los datos y el público objetivo. Además, las visualizaciones deben actualizarse regularmente para reflejar los cambios en los datos y mantener su relevancia.

2.1.5. Toma de decisiones

Por último, la fase de toma de decisiones es la etapa final del proceso. En esta fase, los usuarios finales utilizan los informes y visualizaciones de datos generados en las fases anteriores para tomar decisiones informadas sobre el negocio. La toma de decisiones en BI es un proceso iterativo que implica la identificación de oportunidades y problemas, la evaluación de alternativas y la selección de la mejor opción. Los informes y visualizaciones generados en las fases anteriores pueden ayudar a los usuarios a identificar áreas de mejora y oportunidades de crecimiento en el negocio. Por ejemplo, los modelos de análisis predictivo pueden predecir la demanda futura de un producto en función de los datos históricos de ventas, lo que puede ayudar a los gerentes de marketing y producción a planificar sus recursos de manera más efectiva.

2.1.6. Problemas en el proceso de BI

A pesar de que el BI puede ser una herramienta muy valiosa para la toma de decisiones informadas en las empresas, existen algunos problemas comunes que pueden surgir durante el proceso. A continuación, se presentan algunos de los problemas más comunes en el proceso.

- **Falta de datos de calidad:** Los datos inexactos, incompletos o inconsistentes pueden afectar la precisión de los informes y análisis, lo que puede resultar en decisiones incorrectas. Es esencial asegurarse de que los datos utilizados sean precisos y confiables.
- **Falta de comprensión por parte de los usuarios finales:** El éxito del proceso de BI depende en gran medida de la comprensión y adopción por parte de los usuarios finales. Si los usuarios finales no comprenden la importancia de los informes y análisis, o no están dispuestos a utilizar las herramientas, el proceso no será efectivo. Es importante educar a los usuarios finales sobre el valor del BI y proporcionar capacitación y soporte adecuados.
- **Limitaciones de la tecnología:** Aunque hay una amplia gama de herramientas de BI disponibles, ninguna es perfecta. Algunas pueden tener limitaciones en términos de funcionalidad, capacidad de procesamiento de datos, integración con otros sistemas, etc. Es importante seleccionar la herramienta adecuada para las necesidades del negocio y estar preparado para enfrentar cualquier limitación que pueda surgir.

En conclusión, el *Business Intelligence* es fundamental para la toma de decisiones informadas en las empresas. Es esencial considerar cuidadosamente todos los aspectos del proceso, desde la extracción de datos hasta la visualización y la toma de decisiones. Es importante abordar los desafíos comunes y garantizar una colaboración efectiva entre los equipos de TI y los usuarios finales. Con la selección adecuada de herramientas y técnicas de análisis, el BI puede ser valioso para mejorar la eficiencia empresarial.

2.2 Almacenes de datos y alternativas

2.2.1. Formatos de orígenes de datos

Profundizaremos ahora brevemente en los orígenes de los datos. Ya hemos ilustrado previamente la diferencia entre fuentes estructuradas y no estructuradas. Podemos introducir ahora los formatos más relevantes en los que se suelen encontrar estas fuentes. Estos formatos varían según la naturaleza de los datos y la fuente de donde se obtienen. A continuación, se presentan algunos de los formatos más utilizados:

- **Bases de datos relacionales:** son una fuente de datos estructurados muy común en el contexto del BI. Utilizan el lenguaje de consulta estructurado (SQL) para almacenar y recuperar datos en forma de tablas relacionadas. Los formatos comunes de bases de datos relacionales incluyen MySQL, Oracle y Microsoft SQL Server.
- **Archivos CSV (*Comma-Separated Values*):** Los archivos CSV son un formato simple y ampliamente utilizado para almacenar datos tabulares. Cada registro se representa como una línea de texto y los campos se separan por comas o cualquier otro delimitador. Los archivos CSV son fáciles de crear y leer. Se consideran semiestructurados o no estructurados.

- Archivos Excel: Los archivos de hojas de cálculo de Excel, como XLS y XLSX, son muy populares para almacenar datos estructurados. Estos archivos contienen hojas con celdas organizadas en filas y columnas. Pueden contener múltiples hojas y suelen ser estructurados.
- Archivos JSON (*JavaScript Object Notation*): JSON es un formato ligero y legible por humanos utilizado para el intercambio de datos. Es ampliamente utilizado en aplicaciones web y servicios web. Los archivos JSON representan datos en forma de pares clave-valor y también pueden contener estructuras anidadas. Al igual que CSV, pueden ser semiestructurados o no estructurados.
- Archivos XML (*eXtensible Markup Language*): Las fuentes de datos XML son semi-estructuradas, y contienen información organizada en etiquetas y elementos. XML ofrece flexibilidad en la representación de datos y es ampliamente compatible. Sin embargo, trabajar con fuentes de datos XML puede requerir herramientas y técnicas específicas para su análisis y procesamiento.
- APIs (*Application Programming Interfaces*): Las APIs son interfaces que permiten la comunicación y el intercambio de datos entre aplicaciones. Muchas plataformas y servicios ofrecen APIs que permiten acceder a datos en tiempo real o a través de consultas programáticas.

Sobre estos formatos se aplican procesos de transformación y limpieza que los terminan por modificar en la forma que se desee. Del mismo modo, hay fuentes de datos que no podemos meter dentro de ningún formato en concreto, sobre todo en el apartado de las no estructuradas.

2.2.2. Almacenes de datos

Un almacén de datos, también conocido como *data warehouse*, es un sistema de almacenamiento centralizado y organizado de datos que se utiliza para el análisis y la toma de decisiones en una empresa. Es una infraestructura diseñada para recopilar, integrar y almacenar grandes cantidades de datos procedentes de diversas fuentes, como sistemas transaccionales, bases de datos, archivos, aplicaciones y más. El objetivo principal de un almacén de datos es proporcionar un entorno optimizado para el análisis de datos empresariales. Los datos se extraen, transforman y cargan en el almacén de datos, siguiendo el proceso ETL ya comentado.

Los almacenes de datos están diseñados para ser eficientes en la consulta y el análisis de datos, lo que permite realizar consultas complejas y generar informes en tiempos razonables. Utilizan técnicas de optimización de consultas y estructuras de índices para agilizar el acceso a los datos. Además, suelen implementar modelos de datos dimensionales, como el esquema en estrella o en copo de nieve, que facilitan la estructuración y organización de los datos en dimensiones y hechos. Permite centralizar y consolidar datos de diferentes fuentes en un único repositorio, lo que facilita la integración y la visión global de la información. Proporciona un entorno adecuado para realizar análisis avanzados, como minería de datos, detección de patrones y modelado predictivo. Son especialmente interesantes para almacenar datos históricos a lo largo del tiempo, lo que facilita el seguimiento y análisis de tendencias a largo plazo. Este tipo de estructuras son ampliamente utilizadas por multitud de empresas. En la figura 2.3 podemos ver un esquema completo de un almacén de datos.

Dentro de los almacenes de datos podemos encontrar los *data mart*. La relación entre un almacén de datos y un *data mart* es que un *data mart* se deriva o se construye a partir de

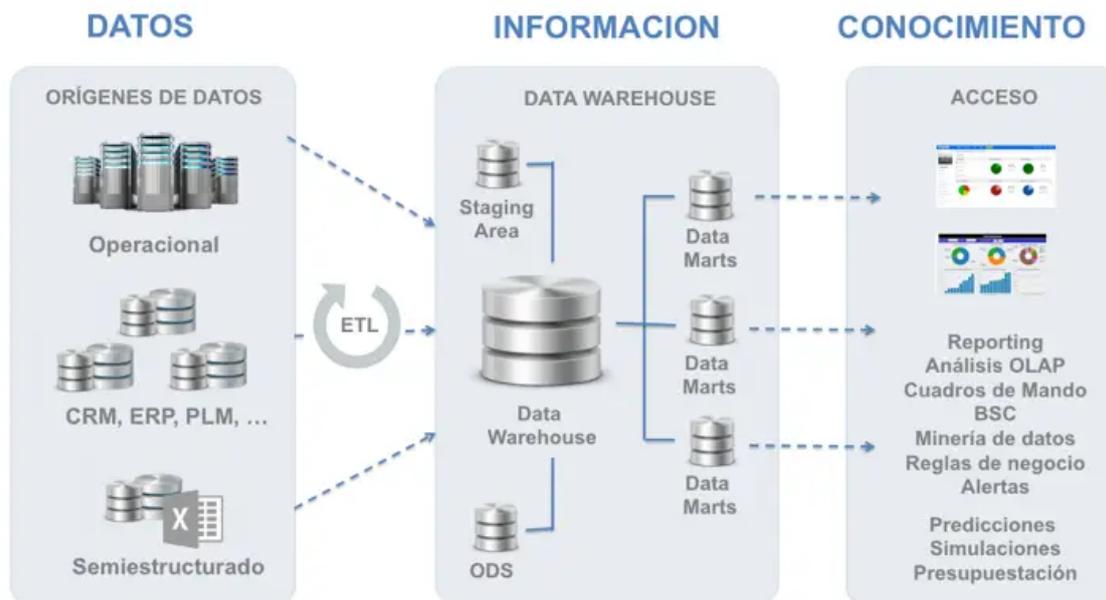


Figura 2.3: Esquema de la arquitectura de un almacén de datos [Calvo, 2016].

un data almacén de datos. En otras palabras, un *data mart* es una subdivisión o segmento del almacén de datos que se crea para satisfacer las necesidades de un grupo específico de usuarios o departamento. Se construye utilizando una porción seleccionada de los datos almacenados en el data almacén de datos y se optimiza para las consultas y análisis relacionados con ese dominio particular [Rizzi and Golfarelli, 2009].

2.2.3. Esquemas para almacenes

Antes de presentar los tres esquemas más habituales que usamos en la organización de los almacenes de datos introduciremos los conceptos de *fact table* (tabla de hechos) y dimensiones.

La *fact table* representa los hechos o eventos del negocio que se desean analizar en el almacén. Contiene las medidas cuantitativas o numéricas del negocio, como ventas, ingresos, cantidades, tiempos, etc. Estas medidas son valores que se pueden agregar o analizar mediante operaciones como sumas, promedios, máximos o mínimos. La *fact table* tiene generalmente una estructura de columnas simples que almacenan los valores de las medidas y claves ajenas que se relacionan con las dimensiones.

Por otro lado, las dimensiones representan los atributos o características descriptivas de los datos en el almacén. Estas tablas contienen información contextual que permite analizar y clasificar los datos de la *fact table*. Cada dimensión tiene generalmente una clave primaria única y se relaciona con la *fact table* mediante una clave ajena. Las dimensiones suelen tener una estructura jerárquica que permite desglosar y agrupar los datos de manera significativa.

La relación entre la *fact table* y las dimensiones se establece mediante las claves ajenas. La *fact table* contiene estas claves ajenas que se utilizan para relacionarse con las dimensiones, lo que permite realizar análisis multidimensionales y responder preguntas sobre cómo las diferentes dimensiones afectan a las medidas o hechos del negocio. La *fact table* actúa como el centro de la relación, y las dimensiones proporcionan el contexto y la granularidad necesarios para realizar consultas y análisis.

Podemos verlo a través del ejemplo de la figura 2.4. Usa el esquema más básico, llamado estrella el cual explicaremos a continuación. La tabla central SALES es la tabla de hechos, y las otras cuatro son dimensiones de esa venta. Cada tupla de la tabla central representa una venta. En la tabla central se almacenan los datos cuantitativos de cada venta mientras que los cualitativos se almacenan a través de las tablas de dimensiones. La dimensión TIME por ejemplo, se relaciona con la tabla de hechos mediante clave ajena y representa el momento en el que se realizó esa venta. Esta dimensión nos va a permitir agregar los datos de la tabla central en base a distintas granularidades para el tiempo: por año, mes, semana... Lo mismo se aplica a las otras tres dimensiones.

Los tres esquemas principales utilizados en un almacén son el esquema estrella (*star schema*), el esquema copo de nieve (*snowflake schema*) y el esquema constelación (*constellation schema*). Estos esquemas definen la forma en que se organizan y relacionan las tablas en el almacén de datos, haciendo uso de las tablas de hechos y las dimensiones.

El esquema estrella es el esquema más simple y comúnmente utilizado en los *data warehouses*. En este esquema, hay una tabla central de hechos que contiene las medidas clave o numéricas del negocio. Las dimensiones se modelan como tablas separadas y se relacionan directamente con la tabla de hechos. Cada dimensión se conecta directamente a la tabla de hechos, creando una estructura en forma de estrella. Un ejemplo de uso de esquemas en estrella es el caso de la empresa de telecomunicaciones AT&T. En su caso, utilizan un esquema en estrella para analizar los datos de facturación y los datos de uso de sus clientes. Este esquema es fácil de entender, facilita el análisis y la generación de informes, y ofrece un rendimiento eficiente para consultas simples y agregadas. Lo hemos usado anteriormente como ejemplo y lo vemos en la figura 2.4.

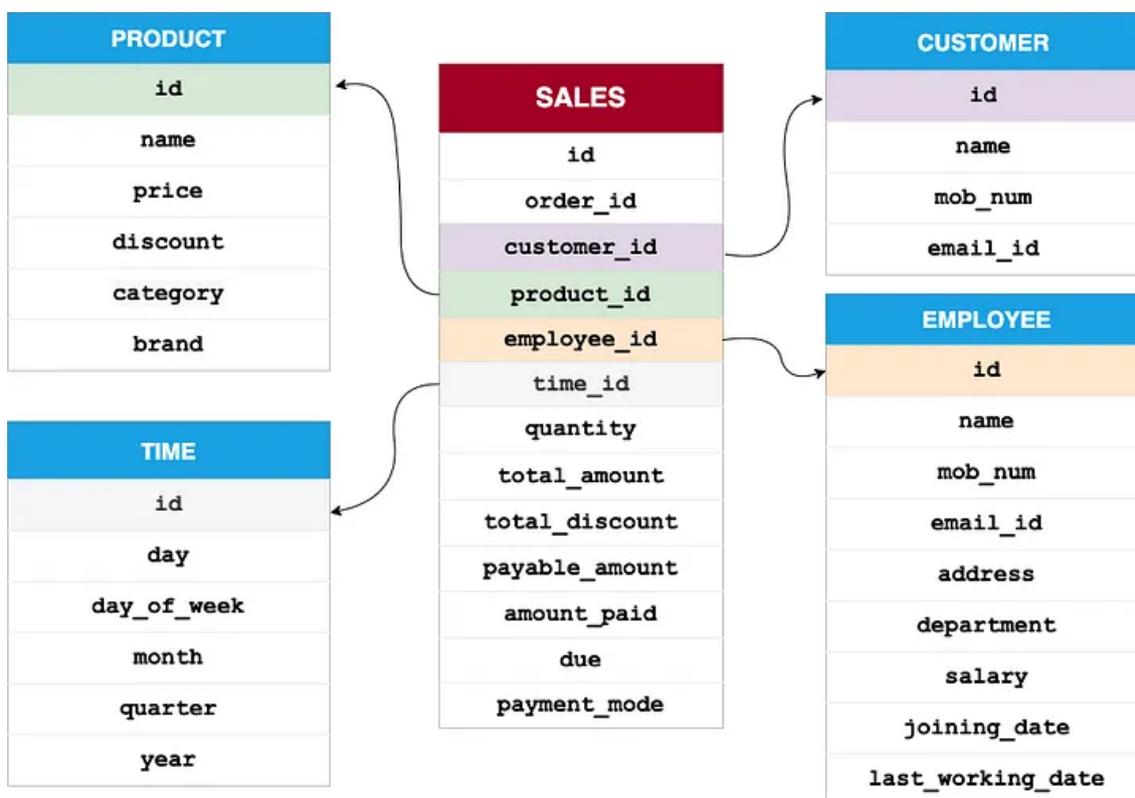


Figura 2.4: Ejemplo de un esquema en estrella [Verma, 2022].

El esquema copo de nieve es una extensión del esquema estrella donde se busca una mayor normalización de las dimensiones. En este esquema, las tablas de dimensiones se descomponen en subdimensiones que se normalizan en tablas separadas. Esto significa

que las tablas de dimensiones tienen relaciones adicionales entre sí, creando una estructura en forma de copo de nieve. Esta descomposición en subdimensiones puede ayudar a reducir la redundancia de datos y mejorar la eficiencia del almacenamiento. Sin embargo, también puede hacer que las consultas sean más complejas y requieran más *joins*. El esquema copo de nieve se utiliza cuando se requiere una mayor normalización y cuando hay una gran cantidad de atributos y relaciones en las dimensiones. Ofrece menos rendimiento en las consultas pero reduce el espacio de almacenamiento que ocupan los datos. Lo podemos ver en la figura 2.5, las dos subdimensiones son *PRODUCT_CATEGORY* y *EMPLOYEE_CONTACT*. En el caso del producto se crea una subdimensión para representar la categoría. Así evitamos repetir todas las características de la categoría para cada producto y solo las tenemos que almacenarlas una vez para cada categoría en una subtabla, ahorrando espacio en memoria. Sin embargo, si realizamos una consulta donde tenemos en cuenta, por ejemplo, el *supplier* de la categoría, la consulta será algo menos eficiente que un esquema en estrella, ya que tendrá que realizar *joins* adicionales con la tabla de la subdimensión.

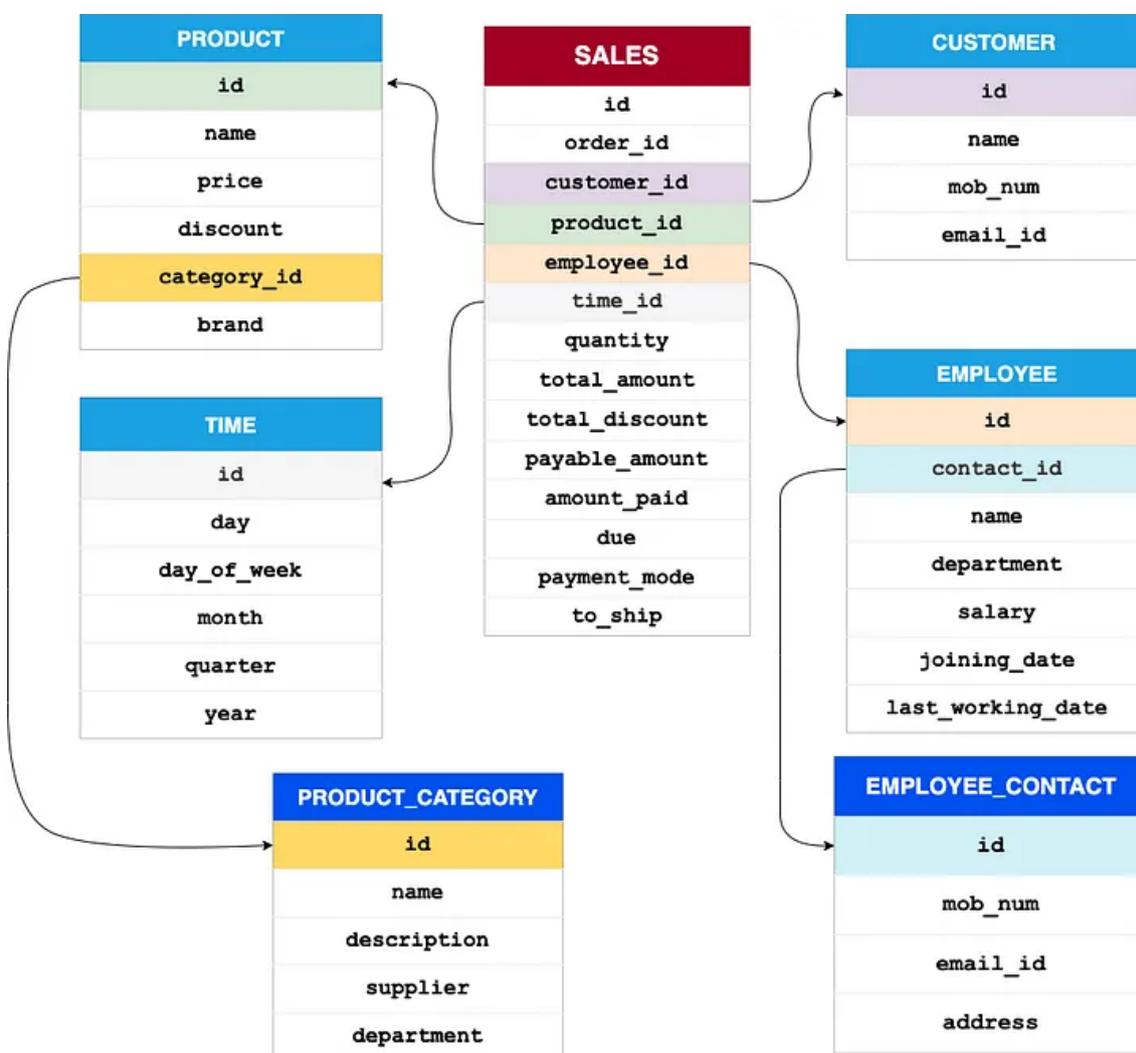


Figura 2.5: Ejemplo de un esquema tipo copo de nieve [Verma, 2022].

El esquema constelación es una combinación de múltiples esquemas estrella interconectados. En este esquema, hay múltiples tablas de hechos que comparten algunas dimensiones comunes. Cada tabla de hechos tiene sus propias dimensiones únicas, y las dimensiones compartidas se modelan como tablas separadas y se relacionan con cada tabla de hechos correspondiente. El esquema constelación se utiliza cuando hay diferentes

procesos de negocio o áreas funcionales que requieren diferentes medidas y dimensiones, lo que permite una mayor flexibilidad y escalabilidad en el diseño del almacén de datos. Una posibilidad es la de la figura 2.6. Se parte del esquema de copo de nieve anterior y se añade una nueva tabla de hechos *SHIPPING*, que tiene dimensiones propias y otras comunes con *SALES* lo que le permite reutilizar dimensiones como la de producto en este caso.

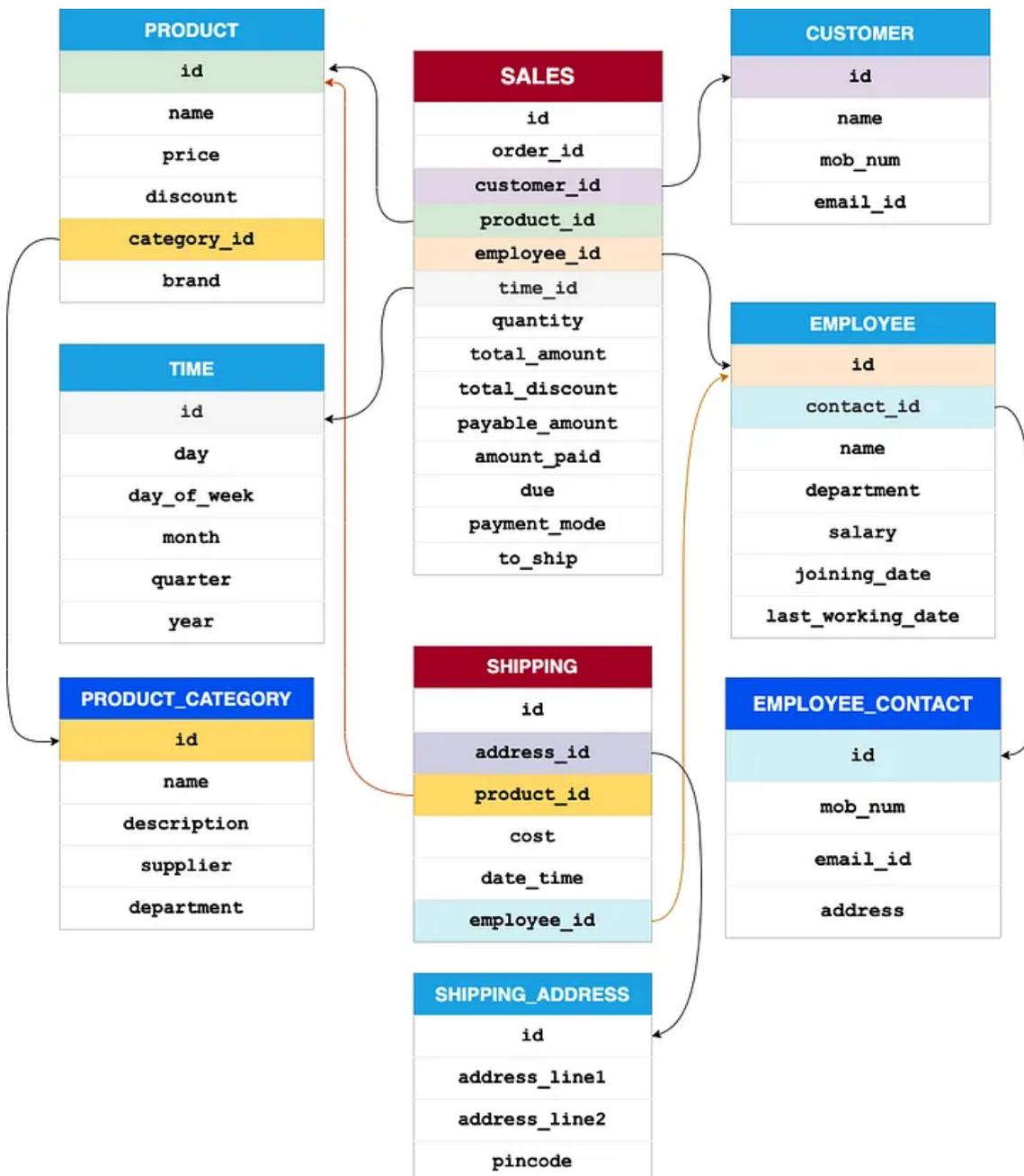


Figura 2.6: Ejemplo de un esquema en constelación [Verma, 2022].

En resumen, el esquema estrella es simple y eficiente para consultas agregadas, el esquema copo de nieve ofrece una mayor normalización para reducir la redundancia de datos y el esquema constelación brinda flexibilidad para diferentes áreas funcionales [Lechtenbörger, 2001].

2.2.4. OLAP

OLAP (*Online Analytical Processing*) es una tecnología y enfoque de análisis de datos que permite a los usuarios realizar análisis multidimensional sobre grandes volúmenes de datos almacenados en un *data warehouse* u otras fuentes de datos. La finalidad principal de OLAP es brindar una forma rápida y flexible de explorar y analizar datos.

La característica clave de OLAP es su capacidad para organizar los datos en una estructura multidimensional conocida como un cubo OLAP. Un cubo OLAP se compone de dimensiones y medidas (lo que antes llamábamos hechos). Estas dimensiones y medidas se organizan en una estructura de cubo, donde cada eje del cubo representa una dimensión y las celdas del cubo contienen los valores de las medidas. Esta estructura multidimensional permite a los usuarios visualizar y analizar los datos desde múltiples perspectivas y niveles de agregación. En el ejemplo de la figura 2.7 podemos identificar que la columna amarilla representa todos los productos para la región de Asia, y la fila verde los productos de tipo TV. La intersección de ambas es el cuadrado de color naranja: los productos de la región de Asia que sean de tipo TV.

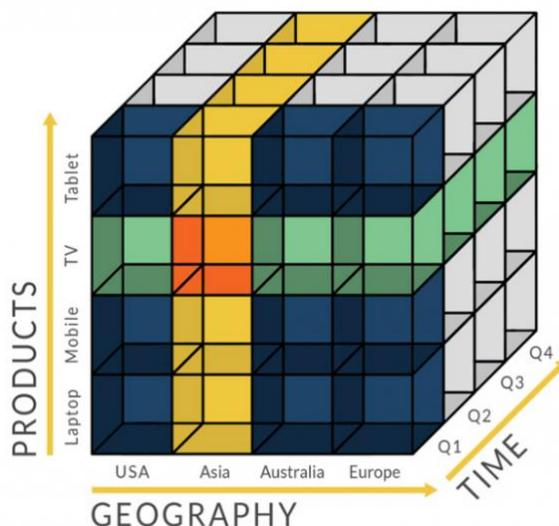


Figura 2.7: Representación gráfica de un cubo OLAP con tres dimensiones: tipo de producto, zona geográfica y período temporal [Holistics, 2020].

La principal ventaja de utilizar OLAP en el análisis de datos es su capacidad para realizar operaciones analíticas complejas y rápidas. Podemos ver una representación visual de ellas en la figura 2.8. Las cinco principales son:

- **Slicing:** Esta operación consiste en seleccionar un subconjunto de los datos del cubo en una dimensión específica. Por ejemplo, se puede seleccionar un período de tiempo específico en la dimensión de tiempo para analizar los datos correspondientes a ese período.
- **Dicing :** Implica seleccionar un subconjunto de datos del cubo en múltiples dimensiones simultáneamente. Por ejemplo, se puede seleccionar un período de tiempo específico, un producto particular y una ubicación geográfica específica para analizar las ventas de ese producto en esa ubicación durante ese período.
- **Drill-down:** Consiste en ir de un nivel de detalle más alto a uno más bajo en una jerarquía dimensional. Por ejemplo, se puede pasar de analizar las ventas a nivel

anual a analizarlas a nivel trimestral o mensual, lo que proporciona una visión más detallada de los datos.

- *Roll-up*: Es lo opuesto al *Drill-down*, ya que implica subir de un nivel de detalle más bajo a uno más alto en una jerarquía dimensional. Por ejemplo, se puede pasar de analizar las ventas a nivel diario a analizarlas a nivel semanal o mensual, lo que proporciona una vista más resumida y agregada de los datos.
- *Pivotar*: La operación de pivotar implica reorganizar los datos del cubo para cambiar las dimensiones y las medidas que se utilizan en el análisis. Permite crear diferentes perspectivas y análisis de los datos sin cambiar la estructura subyacente del cubo.

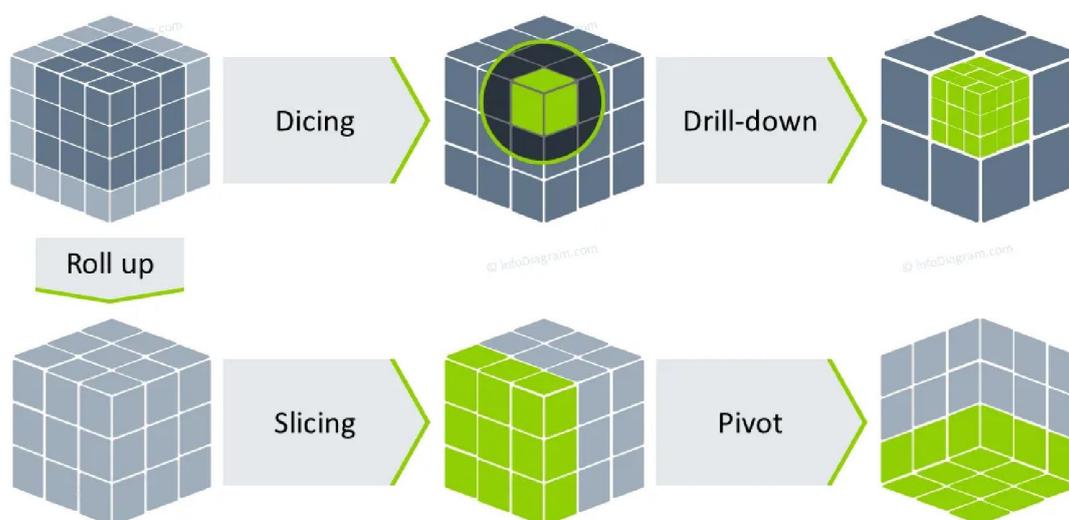


Figura 2.8: Representación gráfica de las operaciones que se realizan sobre un cubo OLAP [InfoDiagram, 2023].

Otro aspecto importante de OLAP es su capacidad para realizar cálculos y comparaciones entre diferentes conjuntos de datos. Los usuarios pueden aplicar funciones de agregación, como suma, promedio, máximo, mínimo, contar, etc., para obtener métricas clave. También pueden realizar cálculos personalizados, como tasas de crecimiento, participación de mercado, variaciones porcentuales, entre otros.

Dentro de OLAP podemos encontrar tres tipos de enfoques diferentes: MOLAP (*Multidimensional OLAP*), ROLAP (*Relational OLAP*) y HOLAP (*Hybrid OLAP*). Estos enfoques se refieren a cómo se almacenan y se acceden a los datos en el contexto del análisis multidimensional.

En MOLAP, los datos se almacenan en una estructura multidimensional optimizada para el análisis rápido. Esto significa que los datos se precálculan y se almacenan en un formato específico, lo que permite un acceso rápido y eficiente a través de consultas multidimensionales. En MOLAP, tanto los datos de resumen como los detalles se almacenan en el cubo OLAP, lo que proporciona un rendimiento óptimo para consultas complejas. Además, MOLAP permite una mayor funcionalidad, como cálculos avanzados y niveles de agregación personalizados. Sin embargo, el almacenamiento de datos en formato multidimensional puede requerir más espacio en disco en comparación con otros enfoques.

En ROLAP, los datos se almacenan en una base de datos relacional tradicional, como un sistema de gestión de bases de datos (DBMS). En lugar de precalcular y almacenar datos en una estructura multidimensional, ROLAP utiliza consultas SQL para acceder

a los datos directamente de la base de datos subyacente. Esto significa que los datos se almacenan en tablas relacionales y se utiliza el poder del motor de la base de datos para realizar operaciones analíticas. ROLAP ofrece flexibilidad en términos de acceso a los datos y permite aprovechar las capacidades de un DBMS existente. Sin embargo, puede haber una degradación del rendimiento al trabajar con grandes volúmenes de datos o al realizar consultas complejas debido a la naturaleza relacional de los datos.

HOLAP es una combinación de los enfoques MOLAP y ROLAP. En HOLAP, los datos se almacenan de manera híbrida, lo que significa que los datos de resumen se almacenan en una estructura multidimensional (similar a MOLAP) y los datos detallados se almacenan en una base de datos relacional (similar a ROLAP). Esto permite combinar la eficiencia del acceso multidimensional para consultas de alto nivel con la flexibilidad y la capacidad de respuesta de las consultas SQL para detalles específicos.

Es interesante que veamos también OLTP y lo diferenciamos de OLAP. OLTP (*Online Transaction Processing*) es un enfoque que se utiliza para el procesamiento eficiente de transacciones en tiempo real. Se centra en el registro y procesamiento de transacciones comerciales diarias, como la inserción, actualización o eliminación de datos en una base de datos.

En un entorno OLTP, las transacciones suelen ser pequeñas en tamaño y se realizan en tiempo real. El objetivo principal de OLTP es garantizar la disponibilidad, integridad y consistencia de los datos transaccionales. Esto implica asegurar que todas las transacciones se realicen de manera segura, sin pérdida de datos y con la capacidad de revertir o deshacer cambios en caso de errores o fallas. Las bases de datos OLTP están diseñadas para admitir altos niveles de concurrencia.

OLAP se utiliza para el análisis y la generación de informes basados en grandes volúmenes de datos históricos, mientras que OLTP se ocupa del procesamiento eficiente de transacciones en tiempo real para respaldar las operaciones comerciales diarias. Ambos enfoques son importantes en el ámbito de la inteligencia empresarial, pero difieren en sus objetivos, volumen y tipo de datos que manejan, así como en el diseño de la base de datos y las consultas que realizan [Rizzi and Golfarelli, 2009].

2.2.5. *Data lakes*

Un *data lake* es un enfoque de almacenamiento de datos que permite almacenar grandes volúmenes de información en su forma original, sin necesidad de una estructuración previa. A diferencia de los enfoques tradicionales de almacenamiento de datos, que implican definir un esquema rígido y aplicar transformaciones antes de almacenar los datos, un *data lake* almacena los datos en su estado bruto, tal como se generan o reciben.

En un *data lake*, los datos se almacenan en su forma cruda y no procesada, sin una estructura predefinida o un modelo de datos riguroso. Esto significa que se pueden almacenar diferentes tipos de datos, como archivos de texto, documentos JSON, registros de eventos, imágenes, archivos de audio, videos, datos de sensores, datos de transacciones, entre otros, sin necesidad de convertirlos o transformarlos previamente. Esto proporciona una gran flexibilidad para almacenar datos de diversas fuentes y formatos en un solo lugar. La arquitectura de un *data lake* generalmente se basa en tecnologías de almacenamiento distribuido, como *Hadoop Distributed File System* (HDFS) o sistemas de almacenamiento en la nube.

Una de las ventajas clave de un *data lake* es su capacidad para preservar la integridad de los datos y mantener su contexto original. Al almacenar los datos en su forma cruda, se evita la pérdida de información o la introducción de errores causados por transformaciones o agregaciones previas. Dado que los datos se almacenan en su forma bruta,

puede haber una falta de estructura y organización, lo que dificulta su descubrimiento y comprensión. Para abordar este desafío, es necesario implementar mecanismos de catalogación, metadatos y etiquetado de datos, para ayudar a los usuarios a buscar y entender los datos disponibles en el *data lake*. Además, los *data lake* requieren una sólida gobernanza de datos para garantizar la calidad, la seguridad y la privacidad de los datos almacenados [Khine and Wang, 2018].

En los *data lake* se usa un enfoque *pay-as-you-go* evitando al creación de esquemas o relaciones permanentes entre elementos de diferentes orígenes. Se basa en enlaces temporales y metadatos. Si lo comparamos con un almacén de datos tradicional, vemos que en el *data lake* los datos entran de forma cruda en el almacén y después se seleccionamos una parte de ellos para cada necesidad, sobre los que realizamos el proceso de integración, limpieza, etc. En un almacén de datos tradicional los datos se integran previamente dando lugar aun esquema consistente que se introduce en el almacén y los análisis se realizan directamente sobre estos datos ya refinados. Esta diferencia se ve de forma visual en al figura 2.9.

Las empresas pueden crear sus propios *data lakes* o adquirirlos como servicios de otras empresas. Al ser datos en crudo y sin transformar, un mismo *data lake* puede ser ofrecido a varias empresas. Tanto Alphabet (Google) como Amazon o Microsoft ofrecen sus propios *data lakes*. El más relevante es Big Lake, el de Google, siendo el que tiene más cuota de mercado, seguido por el de Amazon.

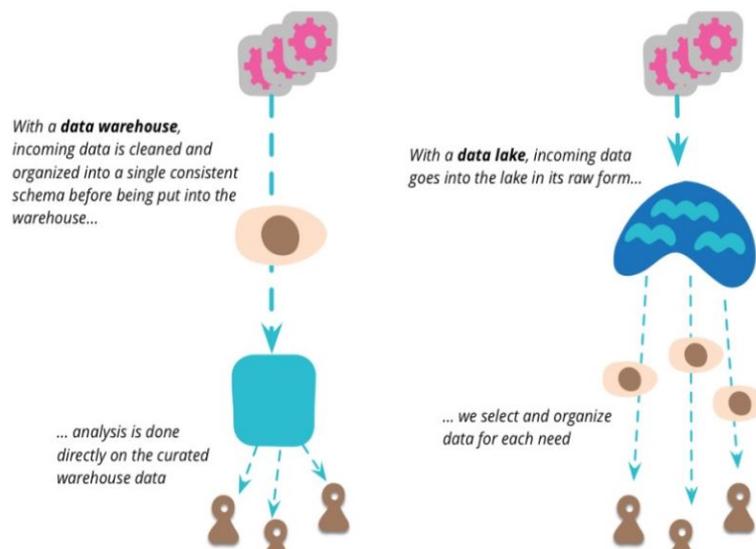


Figura 2.9: Representación gráfica de la diferencia en el uso de los datos en un almacén de datos tradicional (figura de la izquierda) y un *data lake* (figura de la derecha) [Fowler, 2015].

2.2.6. Data Mesh

Data Mesh es uno de los enfoques más revolucionarios en la gestión de datos. Propuesto por Zhamak Dehghani en 2020, *Data Mesh* propone una arquitectura de datos distribuida y descentralizada que distribuye la responsabilidad y propiedad de los datos entre los diferentes equipos y dominios de una organización.

En un *Data Mesh*, los datos se consideran productos y cada equipo o dominio es responsable de gestionar y ser dueño de sus propios productos de datos. Esto implica que los equipos tienen la autoridad y la autonomía para tomar decisiones sobre cómo se capturan, almacenan, procesan y consumen los datos en su dominio específico. Cada dominio se centra en un área de negocio o problema y se organiza en torno a la propiedad y la

gestión de los datos relacionados con ese dominio. Los equipos de dominio son los más adecuados para comprender y gestionar los datos relacionados con su área de negocio. Cada equipo tiene el conocimiento contextual necesario para definir y mantener los estándares de calidad de los datos en su dominio, y es responsable de la documentación, el acceso, la seguridad y la entrega de valor a través de los productos de datos que ofrecen. En la figura 2.10 tenemos varios ejemplos de dominios en una empresa, es el ejemplo gráfico que Dehghani usa en su publicación original.

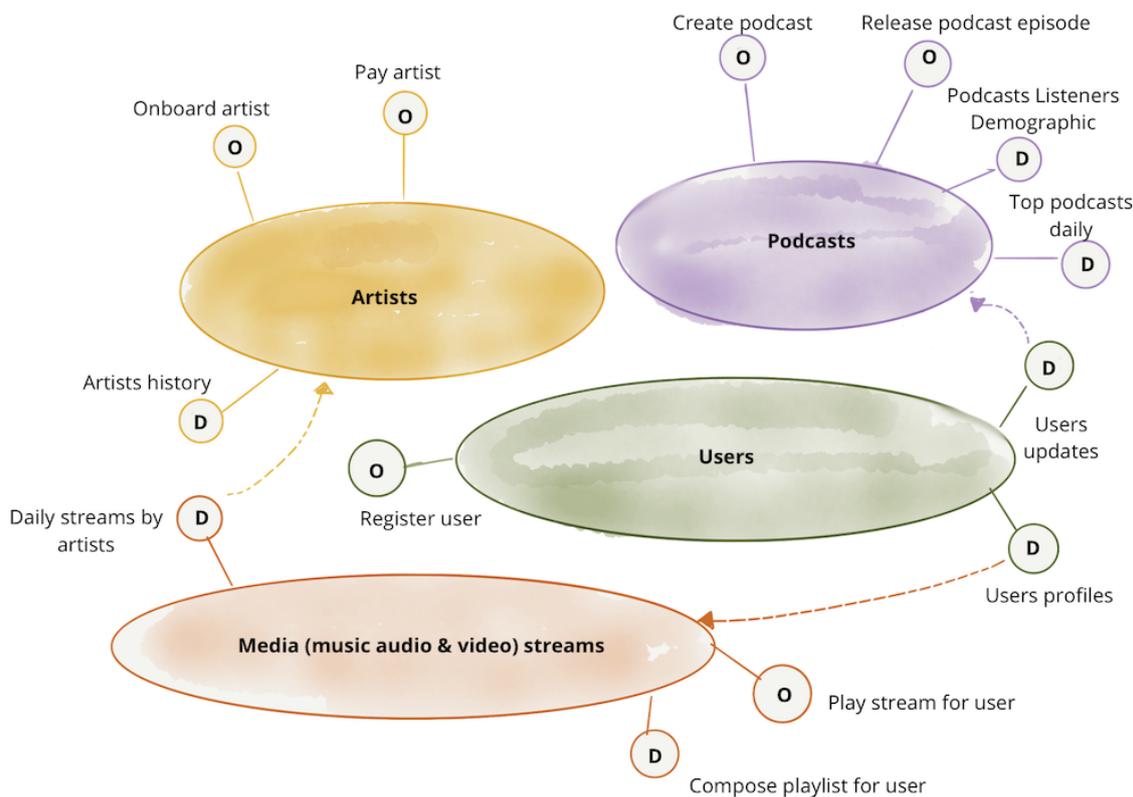


Figura 2.10: Cuatro ejemplos de dominio dentro de una empresa. Las O representan operaciones que se pueden realizar sobre los datos de ese dominio y las D datos analíticos que puede contener [Dehghani, 2020].

Al considerarse los datos como productos en sí mismos, se gestionan de manera similar a como se gestionaría cualquier otro producto en una organización. Los equipos de dominio son los propietarios de estos productos de datos y son responsables de su ciclo de vida completo, desde la captura y procesamiento hasta la entrega y mantenimiento. Cada producto de datos tiene una clara definición de valor y se espera que los equipos de dominio demuestren el impacto y el valor de sus productos de datos en el contexto de su área de negocio. Se establece una plataforma de datos como servicio (*Data Platform as a Service*) para respaldar a los equipos de dominio en la gestión de sus datos. Esta plataforma proporciona herramientas, servicios y recursos técnicos para facilitar la catalogación de datos, el descubrimiento, el acceso seguro, el gobierno de datos y otras capacidades relacionadas.

Aunque cada equipo de dominio es responsable de su propio conjunto de datos, se fomenta la colaboración y la federación de datos entre los equipos. Esto implica establecer acuerdos y estándares claros para compartir datos de manera segura y confiable entre los diferentes dominios. La federación de datos permite obtener una visión más com-

pleta y holística de los datos en toda la organización, fomentando la colaboración y el intercambio de conocimientos entre los equipos [Dehghani, 2020].

En la figura 2.11 vemos como es la arquitectura. En el centro en amarillo tenemos lo que sería un dominio. Arriba en verde tenemos federación y gobierno, que se encarga de interconectar los dominios de forma segura para obtener una visión general. Abajo en azul está la plataforma de datos como servicio que ayuda a los equipos en la gestión de los datos. En la parte derecha en morado tenemos los usuarios de la empresa que finalmente harán uso de esos datos, por ejemplo, consultoría, marketing, decisiones estratégicas, etc. Estos usuarios reales pueden ser departamentos dentro de la empresa o de empresa externas o asociadas que adquieran esos servicios. Es un caso similar al de *data lake*. Un ejemplo interesante es el del banco de inversión JP Morgan, que está implementando un Data Mesh en AWS Cloud [Vellante, 2021].

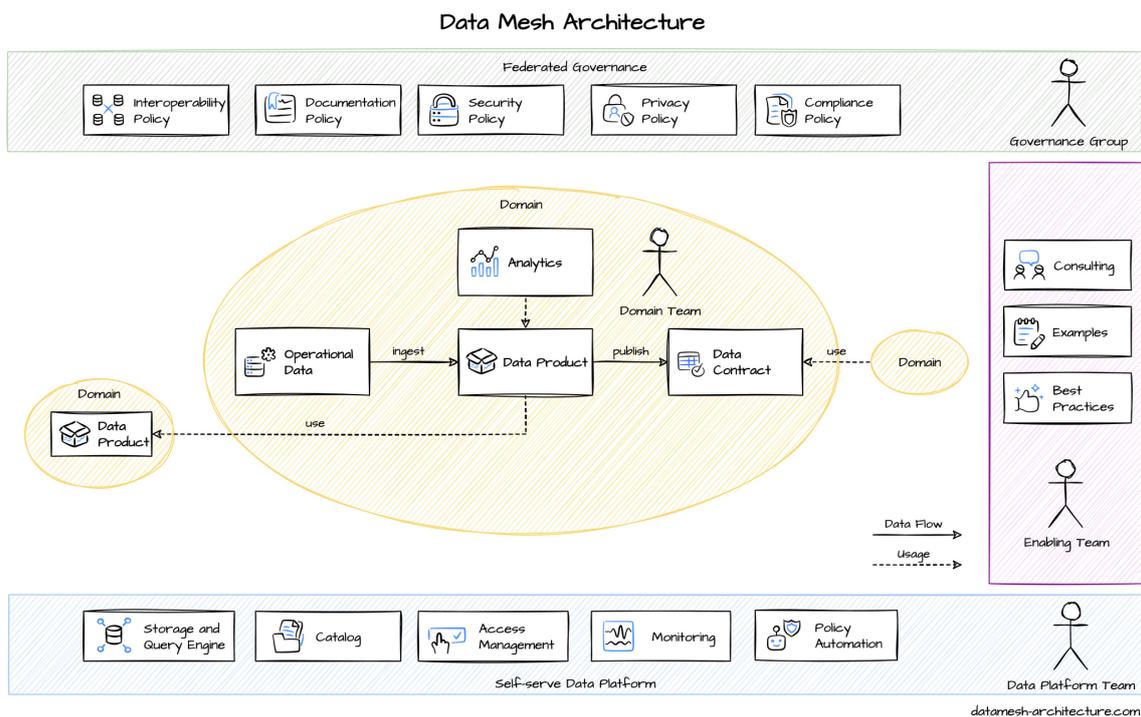


Figura 2.11: Arquitectura de un Data Mesh [Christ et al., 2021].

2.2.7. Bases de datos NoSQL

Las bases de datos NoSQL son sistemas de gestión de bases de datos diseñados para almacenar y recuperar grandes volúmenes de datos de forma eficiente y escalable. Pueden ser una alternativa útil a los almacenes de datos. A diferencia de las bases de datos relacionales tradicionales, las bases de datos NoSQL utilizan modelos de datos flexibles y no estructurados que permiten un almacenamiento y procesamiento más ágil de datos no relacionales. Son altamente escalables, especialmente en entornos distribuidos, ya que están diseñadas para manejar grandes volúmenes de datos y cargas de trabajo distribuidas. Por el contrario, los almacenes de datos tradicionales pueden escalar verticalmente aumentando los recursos de hardware de un servidor, pero no son tan eficientes como las bases de datos NoSQL en términos de escalabilidad horizontal. Además, NoSQL permiten cambios flexibles en el esquema de datos, lo que significa que no es necesario definir una estructura rígida antes de almacenar los datos. Esto permite una mayor adaptabilidad a medida que los requisitos de los datos evolucionan con el tiempo. Están optimizadas para operaciones de escritura y lectura rápidas en grandes volúmenes de datos.

Por ello, ofrecen un alto rendimiento en consultas simples y rápidas, pero pueden tener limitaciones en consultas complejas que requieren operaciones de múltiples tablas. Hay de varios tipos como pueden ser: documentos, clave-valor, columnas o grafos [Li, 2018].

Una de las más usadas es MongoDB, de código abierto. Utiliza un modelo de datos basado en documentos, lo que significa que los datos se almacenan en documentos parecidos a JSON llamados BSON. MongoDB es altamente escalable, lo que le permite manejar grandes volúmenes de datos y soportar crecimiento rápido. Proporciona replicación y tolerancia a fallos para garantizar alta disponibilidad. Además, se integra fácilmente con diferentes lenguajes de programación y es compatible con implementaciones en la nube. Algunas empresas que usan MongoDB son eBay, Electronic Arts o MetLife [upGrad, 2022].

2.3 Nuevas tendencias

Aprovecharemos ahora para comentar algunas tecnologías que están en alza en relación con el BI, y que también podemos considerar estado del arte.

2.3.1. Uso de la nube

El uso de la nube en BI ha revolucionado la forma en que las empresas gestionan y aprovechan sus datos. La nube proporciona un entorno altamente escalable y flexible para almacenar, procesar y analizar grandes volúmenes de datos de manera eficiente. Las plataformas de BI, como Power BI o Tableau, se han adaptado para funcionar en entornos de nube. Estas herramientas permiten a los usuarios conectarse directamente a fuentes de datos en la nube, visualizar y analizar los datos, y compartir informes y paneles de control con otros usuarios de la organización. Esto es especialmente beneficioso para las organizaciones distribuidas geográficamente, facilitando la colaboración entre equipos. Sin embargo, probablemente la mayor ventaja del uso de la nube está en la escalabilidad y en externalizar tareas de mantenimiento y seguridad.

La nube permite escalar verticalmente los recursos de manera rápida y sencilla. Significa que se pueden asignar más recursos, como capacidad de procesamiento y memoria, a medida que crecen las necesidades de la empresa. Esto evita la necesidad de invertir en hardware costoso y garantiza que el sistema pueda manejar cargas de trabajo en crecimiento sin problemas. La nube también permite escalar horizontalmente los recursos distribuyendo la carga de trabajo en múltiples instancias o nodos. La escalabilidad horizontal permite adaptarse rápidamente a los picos de demanda y garantizar un rendimiento constante durante períodos de alta carga.

Por otro lado, en la nube, las tareas de mantenimiento, como la aplicación de actualizaciones y parches de seguridad, son gestionadas por el proveedor de servicios. Esto alivia a las empresas de la responsabilidad de mantener y administrar la infraestructura subyacente, lo que ahorra tiempo y recursos. También ofrecen herramientas de monitoreo y administración que permiten realizar un seguimiento del rendimiento de los sistemas y las aplicaciones de BI. Estos servicios también suelen incluir copias de seguridad automatizadas y replicación de datos en múltiples ubicaciones geográficas.

En materia de seguridad, los proveedores de servicios en la nube implementan rigurosas medidas de seguridad para proteger los datos almacenados y procesados en sus plataformas, por ejemplo, mediante encriptación. También se encargan cumplir con estándares y regulaciones de seguridad establecidos por normativa y que son necesarios

también para la empresa, como el Reglamento General de Protección de Datos (GDPR) en la Unión Europea o las regulaciones específicas de la industria.

La escalabilidad, junto con la externalización de tareas críticas en el proveedor de la nube son aspectos que hacen muy atractivos esta tecnología en el mundo del BI. El uso de Business Intelligence en la nube se considera un factor clave para el éxito en diversos campos. En 2018, cerca del 66 % de las organizaciones con éxito en BI ya utilizan la nube [Elmalah and Nasr, 2019].

2.3.2. Inteligencia artificial y *Natural Language Processing*

El uso de la inteligencia artificial (IA) en el BI ha revolucionado la forma en que las empresas analizan y utilizan sus datos. La IA permite realizar análisis avanzados de datos, como el aprendizaje automático (*machine learning*) y la minería de datos. Por otro lado, puede automatizar tareas repetitivas y tediosas. Por ejemplo, puede realizar la limpieza y transformación de datos de manera automática, lo que ahorra tiempo y reduce errores humanos. La generación de recomendaciones personalizadas también es una aplicación clave de la IA. Tras realizar un análisis de los datos, una IA puede generar recomendaciones o guías que pueden ser muy útiles como base en la fase de toma de decisiones. Con el uso de tecnologías como ChatGPT y similares, este tipo de uso se puede volver algo frecuente [Palmer, 2020].

El Procesamiento del Lenguaje Natural (NLP) es una rama de la inteligencia artificial que se centra en la interacción entre los sistemas informáticos y el lenguaje humano. En el contexto del BI, el NLP desempeña un papel crucial al permitir que las organizaciones comprendan y analicen datos no estructurados. El NLP puede extraer información valiosa de documentos como informes, artículos, correos electrónicos incluso archivos de audio. Mediante técnicas de análisis sintáctico y semántico, el NLP puede identificar entidades clave (como nombres de personas, organizaciones o lugares), relaciones entre entidades y conceptos relevantes en el texto. Junto con esto, puede analizar el tono emocional y las opiniones expresadas en el texto, lo que permite a las organizaciones comprender la percepción de los clientes, empleados o el público en general. Esto puede aplicarse a comentarios en redes sociales, reseñas de productos o encuestas para evaluar el sentimiento general y la satisfacción de los usuarios.

El uso de IA en el mundo del BI es algo que ya está bastante establecido pero que sin duda veremos incrementarse en los próximos años, al ser una de las ramas de la informática que más relevancia está cobrando. En PowerBI por ejemplo, existe una funcionalidad que llaman Q&A, donde el usuario puede hacer preguntas sobre el informe en lenguaje natural y ellos mismos a través de NLP analizan la pregunta y la intentan contestar a través de los datos y generando gráficos, tablas, etc.

2.3.3. Procesamiento y análisis de datos en tiempo real, arquitecturas Kappa y Lambda

A diferencia del enfoque tradicional de BI, que se basa en datos históricos y análisis retrospectivos, el procesamiento en tiempo real se enfoca en datos en constante actualización y análisis en tiempo real para generar información actualizada y relevante. El primer paso para habilitar el análisis en tiempo real es asegurarse de tener fuentes de datos que proporcionen información actualizada de forma continua. Esto puede incluir datos generados por sensores, registros de transacciones, interacciones de usuarios, datos de redes sociales y más. El siguiente paso es la ingestión de datos en tiempo real, que implica la captura y procesamiento de los datos entrantes de forma rápida y eficiente.

Una parte importante del procesamiento y análisis de datos en tiempo real es la capacidad de visualizar y presentar los resultados de manera rápida y efectiva. Las herramientas de visualización de datos en tiempo real, como Tableau o Power BI, permiten crear paneles y gráficos interactivos que actualizan automáticamente los datos a medida que se producen cambios. Todo ello permite a las organizaciones detectar oportunidades, tomar decisiones rápidas y responder a situaciones en tiempo real.

Dos arquitecturas que se están usando comúnmente para el análisis de datos en tiempo real son Kappa y Lambda. Ambas arquitecturas están diseñadas para manejar flujos continuos de datos en tiempo real.

La arquitectura Kappa se basa en un flujo de datos continuo y utiliza una única canalización de procesamiento para manejar tanto los datos en tiempo real como los datos históricos. En esta arquitectura, los datos se ingieren a través de un sistema de mensajería, como Apache Kafka, y luego se almacenan en un sistema de almacenamiento de datos en tiempo real, como Apache Cassandra o Apache HBase. Los datos se procesan utilizando un motor de procesamiento en tiempo real, como Apache Flink o Apache Samza, que permite realizar análisis en tiempo real a medida que los datos llegan. Los resultados del análisis se pueden almacenar en una base de datos o enviar a sistemas de visualización y presentación en tiempo real. La ventaja de la arquitectura Kappa es su simplicidad, ya que utiliza una única canalización de procesamiento para datos en tiempo real y datos históricos. También ofrece baja latencia y permite una escalabilidad horizontal fácil. Sin embargo, una limitación de esta arquitectura es que no puede manejar fácilmente cambios retroactivos en el esquema de datos, ya que los datos en tiempo real se almacenan directamente sin procesamiento previo.

La arquitectura Lambda aborda la necesidad de manejar tanto datos en tiempo real como datos históricos al introducir dos vías de procesamiento paralelo: una vía para datos en tiempo real y otra vía para datos históricos. Los datos en tiempo real se tratan de forma similar a como hemos visto en Kappa. Además, al mismo tiempo, los datos históricos se almacenan en un sistema de almacenamiento a largo plazo, como HDFS. Los datos históricos se procesan utilizando un motor de procesamiento por lotes, como Apache Hadoop, y los resultados se almacenan en un repositorio de datos históricos. La ventaja de la arquitectura Lambda es que permite un procesamiento más flexible y escalable de datos tanto en tiempo real como históricos. Además, permite el procesamiento retrospectivo de datos históricos en caso de cambios en el esquema de datos o en las necesidades de análisis. AWS por ejemplos ofrece la implementación de Lambda a sus usuarios de servidores. Sin embargo, la arquitectura Lambda es más compleja debido a las dos vías de procesamiento paralelo y puede tener una mayor latencia debido al procesamiento por lotes de datos históricos [Feick et al., 2018].

Las tenemos en la figura 2.12. Es interesante fijarse en el orden de magnitud de tiempo que se tarda en realizar una consulta dependiendo de si son datos en tiempo real desde un almacenamiento rápido (ms) o desde un almacenamiento histórico (min/hr).

2.4 Soluciones software para BI

Continuamos con las diferentes opciones que tenemos para llevar a cabo el proceso de BI. Vamos a dividirlos en dos tipos diferentes. Por un lado tenemos las soluciones que podemos considerar servicios o *suites* y por otro lado, herramientas o librerías que podemos usar a través de lenguajes de programación tradicionales. Es importante destacar que las soluciones de BI pueden variar en cuanto a su alcance y complejidad. Algunas son suites completas que abarcan todas las etapas del proceso de BI, desde la extracción de datos hasta la visualización y generación de informes. Otras son herramientas más

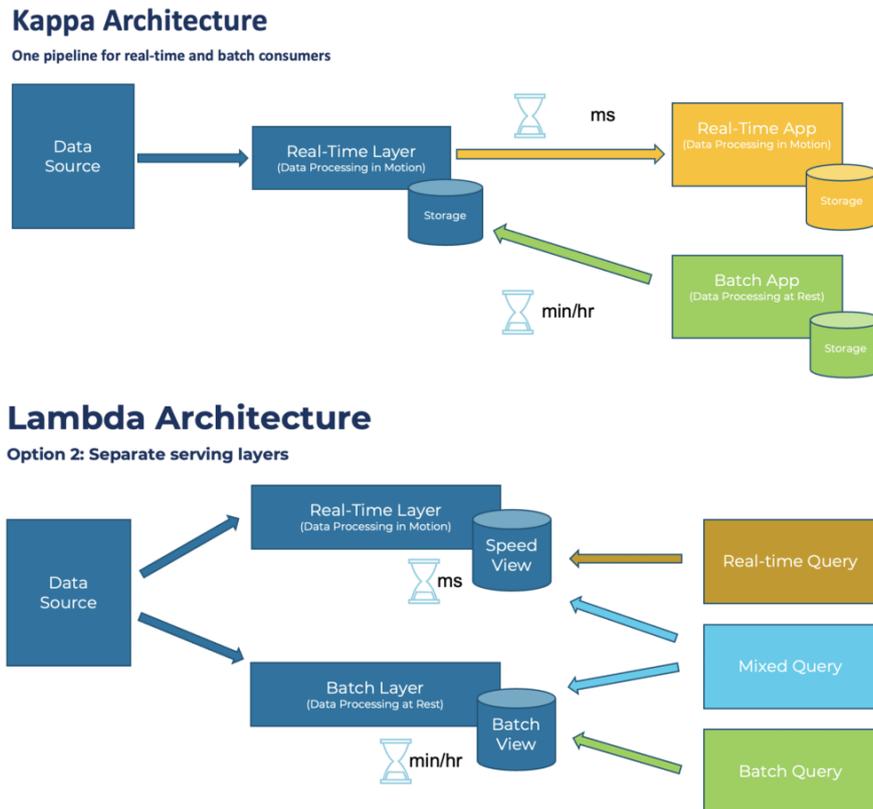


Figura 2.12: Representación de las arquitecturas Kappa (arriba) y Lambda (abajo) [Waechner, 2021].

especializadas que se centran en áreas específicas, como la visualización de datos o el análisis estadístico. La elección de la solución adecuada dependerá de las necesidades y recursos de la empresa.

2.4.1. Servicios

Las *suite* para BI son conjuntos integrados de herramientas y aplicaciones que ofrecen una amplia gama de funcionalidades para el proceso completo de BI. Generalmente incluye varios componentes y módulos que se complementan entre sí para brindar una solución completa. Tienen diversas características que las hacen muy atractivas con respecto al resto de alternativas y que comentaremos a continuación.

Integración completa: Están diseñadas como soluciones integradas que cubren todo el proceso de BI, desde la extracción y transformación de datos hasta el análisis y la visualización. Esto significa que los diferentes componentes de la suite están diseñados para trabajar de manera coherente y se integran sin problemas entre sí, lo que facilita la implementación y el uso de la solución. Si utilizamos herramientas de programación, es necesario desarrollar y mantener todos los componentes individualmente, lo que puede requerir más tiempo y esfuerzo.

Facilidad de uso: Suelen tener interfaces intuitivas y amigables para los usuarios, lo que facilita su uso incluso para aquellos sin conocimientos profundos de programación. Estas interfaces suelen contar con funcionalidades visuales y de arrastrar y soltar, lo que permite a los usuarios realizar tareas como el diseño de informes o la creación de visualizaciones sin tener que escribir código. Esto agiliza el proceso y permite a los usuarios centrarse en el análisis y la interpretación de los datos, en lugar de dedicar tiempo a la

programación. Vemos en la figura 2.13 un ejemplo de interfaz, donde se aprecian claramente las similitudes con los programas del paquete Office de Microsoft.

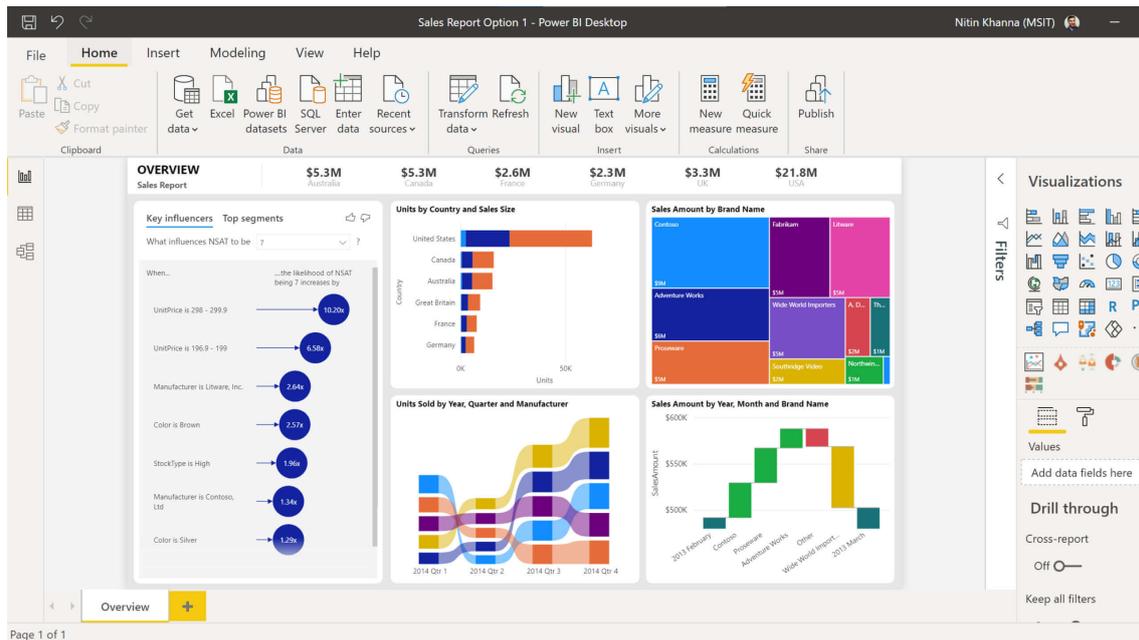


Figura 2.13: Ejemplo de interfaz del programa Power BI Desktop [Microsoft, 2023].

Rapidez en el despliegue: ofrecen opciones preconfiguradas y plantillas que permiten una implementación rápida y sencilla. Esto es útil para las organizaciones que desean obtener resultados rápidamente o que no tienen recursos o experiencia en desarrollo de software. Con otras herramientas de programación, se requiere más tiempo y esfuerzo para diseñar y desarrollar soluciones desde cero.

Mantenimiento y soporte: Las suites de BI aportan soporte técnico y actualizaciones regulares para mantener la solución actualizada y funcionando sin problemas. Esto libera a las organizaciones de la carga de tener que mantener y actualizar constantemente las herramientas y componentes de programación utilizados en un enfoque personalizado. Además, los proveedores de suites de BI suelen proporcionar documentación detallada, recursos de aprendizaje y comunidades de usuarios activas, lo que facilita el aprendizaje y el desarrollo de habilidades. La figura 2.14 muestra una comunidad específica para la herramienta Power BI de Microsoft, donde los usuarios comparten ideas de forma muy activa.

Mencionaremos brevemente algunas de las más populares:

- Microsoft Power BI: Es una suite de desarrollada por Microsoft. Es la herramienta líder en el sector. Ofrece capacidades de extracción, transformación y carga de datos (ETL), análisis de datos, visualización interactiva y generación de informes. Power BI permite conectar a una amplia variedad de fuentes de datos, incluidas bases de datos, servicios en la nube y aplicaciones web. Es conocido por su facilidad de uso, su integración con otros productos de Microsoft y su sólida comunidad de usuarios. Es la que usaremos en el desarrollo del presente proyecto.
- Tableau: Es la otra plataforma líder en visualización de datos y análisis empresarial junto con Power BI. Es apreciado por su enfoque intuitivo y su capacidad para manejar grandes volúmenes de datos.
- QlikView y Qlik Sense: son dos suites de BI desarrolladas por Qlik. Estas plataformas permiten la exploración, visualización y análisis de datos en tiempo real.

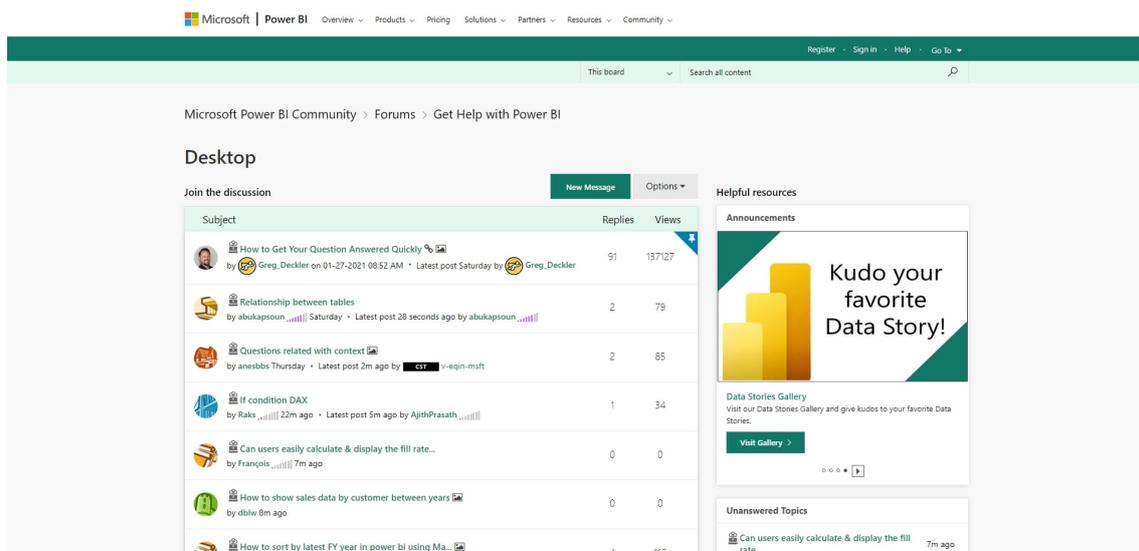


Figura 2.14: Pantalla de inicio de los foros de Power BI Community donde se pueden hacer preguntas y consultar respuestas a otras.

QlikView ofrece funcionalidades de desarrollo más avanzadas y personalizadas, mientras que Qlik Sense se enfoca en la creación de aplicaciones de autoservicio y visualización intuitiva. Ambas suites tienen una reputación sólida en términos de capacidades analíticas y facilidad de uso.

- **SAP BusinessObjects:** Desarrollada por SAP. La suite se integra con otros productos de SAP y permite acceder a una amplia gama de fuentes de datos.
- **MicroStrategy:** Es una suite conocida por su enfoque empresarial y sus capacidades de escalabilidad. Permite a las organizaciones conectarse a diversas fuentes de datos y proporciona opciones de implementación en la nube.

La principal desventaja que presentan este tipo de soluciones frente a las que se pueden considerar más de programación pura es la falta de personalización en desarrollos más complejos y los costes que esto puede generar en ciertas ocasiones. Primeramente porque algunas de las funcionalidades más complejas, sobre todo las que tiene que ver con servidores y grandes volúmenes de datos son de pago, bien a través de licencias o por pagos por capacidad de almacenamiento. Por otro lado, podemos encontrarnos que algunas funcionalidades más concretas simplemente no es posible desarrollarlas dentro del ecosistema que se nos proporciona. La facilidad de uso viene dada a costa de una cierta rigidez.

2.4.2. Herramientas de programación

Además de las suites mencionadas anteriormente, existen herramientas y bibliotecas más orientadas a la programación que ofrecen capacidades de BI. Estas herramientas permiten a los profesionales de datos y desarrolladores crear soluciones personalizadas y adaptadas a las necesidades específicas de la empresa.

Python es un lenguaje de programación ampliamente utilizado en el análisis de datos debido a su facilidad de uso y a su amplia gama de bibliotecas especializadas. Una de las bibliotecas más populares y poderosas es pandas, que proporciona estructuras de datos flexibles y eficientes para el manejo y análisis de datos. Con pandas, los usuarios pueden cargar, limpiar, transformar y analizar datos de manera eficiente. Proporciona una amplia

gama de funcionalidades, como la manipulación de filas y columnas, el filtrado de datos, el cálculo de estadísticas descriptivas y la fusión de conjuntos de datos. Además, pandas permite realizar operaciones complejas, como agregaciones, agrupaciones y pivotajes de datos.

Python también cuenta con otras bibliotecas populares para el análisis de datos, como NumPy, que proporciona un soporte eficiente para operaciones numéricas y algebraicas en arreglos multidimensionales. Es particularmente útil para el procesamiento y la manipulación de datos numéricos a gran escala. Matplotlib y Seaborn son bibliotecas de visualización de datos que permiten crear gráficos y visualizaciones interactivas para representar los datos de manera clara y efectiva. Scikit-learn es una biblioteca de aprendizaje automático en Python que proporciona una amplia gama de algoritmos y herramientas para el análisis de datos predictivos.

R es otro lenguaje de programación ampliamente utilizado en el ámbito del análisis de datos y la estadística. R se destaca por su enfoque en el análisis estadístico y gráfico, y cuenta con una amplia colección de paquetes especializados en estas áreas. Por ejemplo, ggplot2 es una biblioteca popular en R que permite crear gráficos con una sintaxis intuitiva y flexible. También existen paquetes como dplyr y tidyr, que facilitan la manipulación y transformación de datos, permitiendo realizar operaciones como filtrado, agrupación, combinación y transformación de datos de manera eficiente. Además, R ofrece paquetes específicos para el análisis estadístico y la generación de informes, lo que lo convierte en una opción poderosa para el análisis de datos en el contexto de BI.

Sin embargo, podemos considerar que las principales alternativas a los paquetes de servicios para BI que hemos comentado anteriormente deben ser soluciones que integren más fases del desarrollo. Las herramientas que ya hemos comentado en este apartado se centran más en el tratamiento de los datos (parte de ETL principalmente) y no tanto en las fases posteriores del proceso de BI. Hay diversas librerías o soluciones que en combinación con las herramientas de los respectivos lenguajes en los que se basan (Python y R en este caso) van más allá.

R Shiny se basa en el lenguaje de programación R y su objetivo principal es facilitar la creación de aplicaciones web interactivas sin requerir conocimientos avanzados de desarrollo web. Puede ser utilizado para desarrollar *dashboards* y paneles de control personalizados que brinden visualizaciones y análisis de datos en tiempo real. La ventaja principal de R Shiny es su capacidad para combinar análisis complejos con una interfaz de usuario interactiva. Los usuarios pueden explorar los datos, ajustar los parámetros de análisis y ver los resultados actualizados en tiempo real. Ofrece una amplia gama de opciones de personalización, lo que permite adaptar la apariencia y el comportamiento de la aplicación según las necesidades específicas de la empresa. Es una buena opción para los desarrolladores que quieran usar R en vez de Python.

Python Dash es una biblioteca de código abierto que permite construir aplicaciones web interactivas y paneles de control en Python. Dash se basa en Flask, otra popular biblioteca de Python para el desarrollo web, y utiliza componentes de React.js para construir interfaces de usuario dinámicas y reactivas. Al igual que R Shiny, ofrece una amplia variedad de componentes gráficos y herramientas de visualización de datos, como gráficos interactivos, tablas dinámicas y filtros interactivos. Además, Dash es compatible con diferentes proveedores de servicios en la nube y plataformas de implementación, lo que brinda flexibilidad en cuanto a dónde y cómo se despliegan las aplicaciones.

Streamlit es una biblioteca de Python que permite crear aplicaciones web interactivas para visualizar y compartir datos en el contexto del BI. Su principal ventaja es su enfoque minimalista y su facilidad de uso. Se considera una alternativa más simple y accesible a Dash.

Por último, podemos considerar Jupyter, que se basa también principalmente en Python. Permite combinar código ejecutable, visualizaciones y texto explicativo en un solo documento. A diferencia de las anteriores alternativas se centra más en la creación de *notebooks* y no tanto en los *dashboards*. Los notebooks de Jupyter son documentos interactivos que contienen código ejecutable, visualizaciones y texto explicativo. Un notebook de Jupyter está organizado en celdas, donde cada celda puede contener código en un lenguaje de programación compatible, como Python, R, Julia, entre otros. Estas celdas pueden ser ejecutadas de forma individual y los resultados se muestran de inmediato debajo de cada celda. Esto permite una exploración y análisis iterativos de los datos, donde los usuarios pueden realizar cambios y observar instantáneamente los resultados. Además del código, las celdas también pueden contener texto enriquecido utilizando el lenguaje de marcado Markdown.

Las anteriores herramientas que hemos comentado (Shiny, Dash y Streamlit) buscan como resultado dar paneles similares a lo que se conseguiría con las suites comerciales, como el ejemplo de Power BI que observábamos en la figura 2.13. Sin embargo, los notebooks tienen una presentación diferente y están pensados para otro tipo de usuario final como vemos en la figura 2.15.

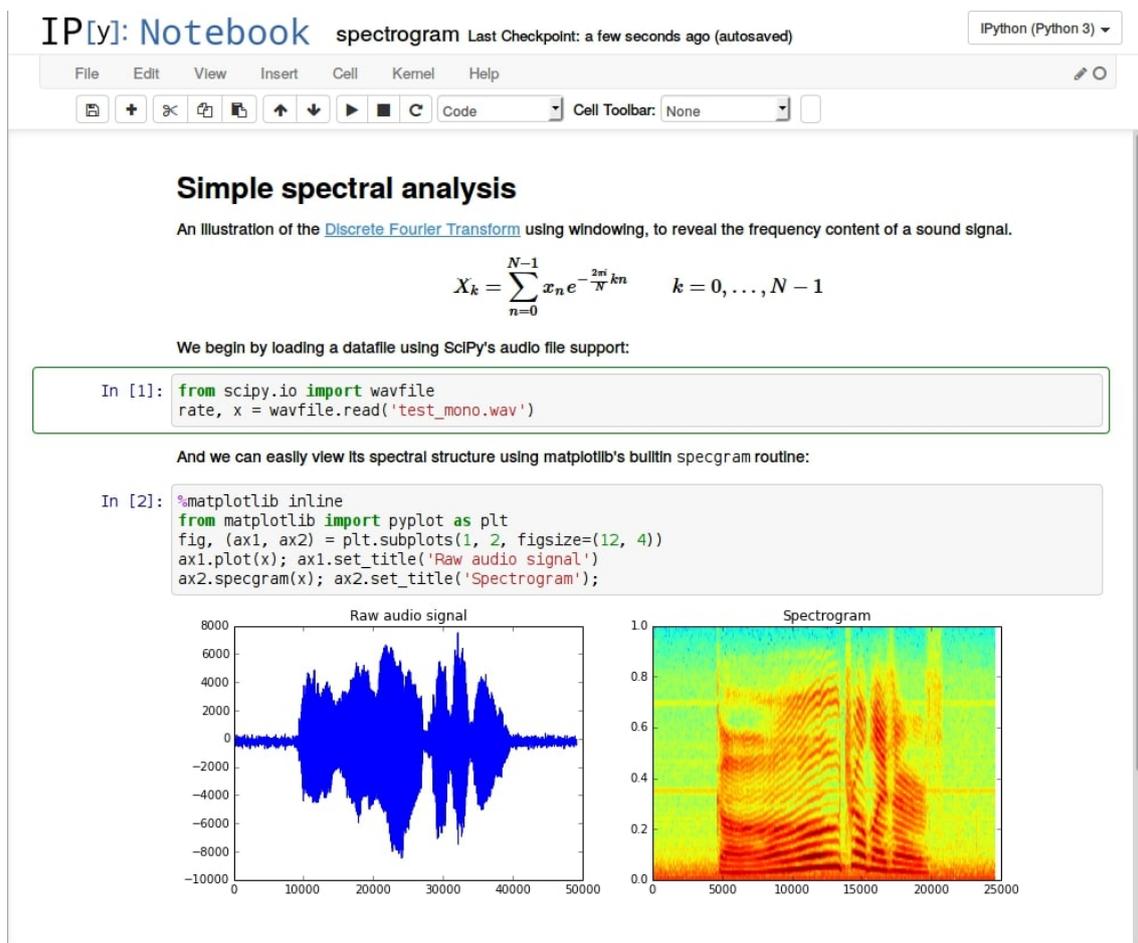


Figura 2.15: Ejemplo de un posible notebook que se puede obtener usando la herramienta Jupyter [IPython, 2023].

Las herramientas de programación en general ofrecen varias ventajas en comparación con las suites:

- **Flexibilidad y personalización:** Las herramientas de programación brindan a los usuarios la libertad de personalizar y adaptar sus soluciones según sus necesidades específicas.
- **Capacidad de integración:** Las herramientas de programación suelen ofrecer una mejor capacidad de integración con diferentes fuentes de datos y sistemas existentes. Los programadores pueden utilizar APIs y bibliotecas para acceder y combinar datos de múltiples fuentes, lo que facilita el análisis de datos desde diversas perspectivas. Además, las herramientas de programación suelen ser más compatibles con lenguajes de consulta estructurados (como SQL) y con tecnologías de almacenamiento y procesamiento de datos, lo que facilita la extracción, transformación y carga (ETL) de datos.
- **Escalabilidad y rendimiento:** Las herramientas de programación pueden ofrecer un mayor rendimiento y escalabilidad en el procesamiento de grandes volúmenes de datos. Al utilizar técnicas de programación eficientes y aprovechar la capacidad de procesamiento paralelo, los programadores pueden realizar análisis y cálculos complejos de manera más rápida y eficiente. Esto es especialmente importante cuando se trata de análisis en tiempo real o de grandes conjuntos de datos.
- **Acceso a algoritmos y técnicas avanzadas:** Los programadores pueden utilizar bibliotecas y paquetes especializados para realizar análisis estadísticos, minería de datos, modelado predictivo y otras técnicas de análisis avanzadas. Esto permite a las empresas aprovechar al máximo sus datos y obtener *insights* más profundos y precisos.
- **Costos y licencias:** En algunos casos, las herramientas de programación pueden ser una opción más económica en comparación con las suites de BI. Las suites suelen tener licencias y costos asociados, especialmente cuando se trata de soluciones comerciales. Por otro lado, muchas herramientas de programación son de código abierto o tienen versiones gratuitas disponibles, lo que reduce los costos de implementación y uso.

A pesar de estas ventajas, también es importante destacar que las herramientas de programación requieren habilidades técnicas y conocimientos de programación. Su implementación y mantenimiento pueden ser más complejos en comparación con las suites de BI, que generalmente tienen interfaces intuitivas y orientadas a usuarios no técnicos. Por lo tanto, la elección entre herramientas de programación y suites dependerá de las necesidades y capacidades específicas de la empresa, así como de los recursos y requisitos del proyecto en particular. Aún con todo, parece que el estándar que se está creando en la mayoría de compañías es decidirse por el uso de las *suites* (especialmente PowerBI, Tableau y Qlik) por encima de estas soluciones más programáticas.

Podemos comparar las diferentes alternativas. En la figura 2.16 se nos sugiere que durante los últimos años las soluciones más populares en GitHub¹ son Dash y Streamlit, las dos basadas en Python. En la figura 2.17 se nos comparan las alternativas en base a ciertos parámetros. [Schmitt, 2020]

¹Se basa en el sistema de *stars* de GitHub, es decir, miran cuantos repositorios destacados hay para cada tipo de herramienta.

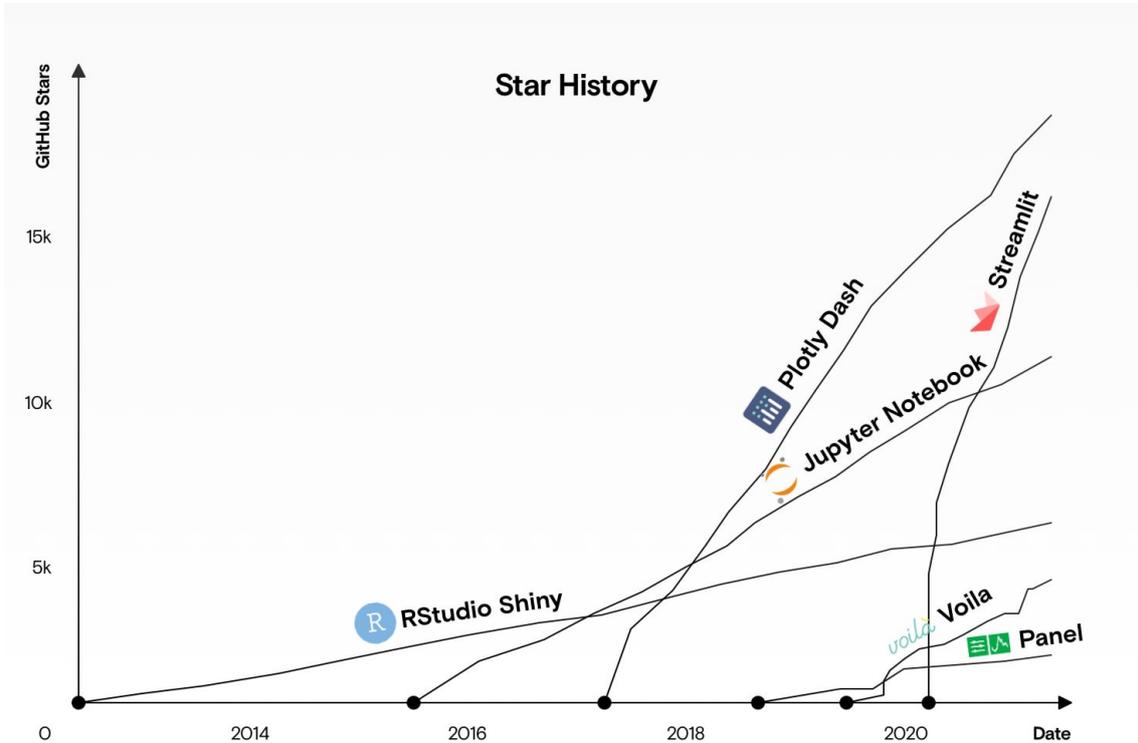


Figura 2.16: Gráfico que muestra la evolución de las principales herramientas de programación para BI a lo largo del tiempo en base al sistema de *stars* de GitHub [Schmitt, 2020].

	Maturity	Popularity	Simplicity	Adaptability	Focus	Language support
Streamlit	C	A	A	C	Dashboard	Python
Dash	B	A	B	B	Dashboard	Python, R, Julia
Panel	C	B	B	B	Dashboard	Python
Shiny	A	B	B	B	Dashboard	R
Voila	C	C	A	C	Dashboard	Python, R, Julia
Jupyter	A	A	B	B	Notebook	Python, R, Julia
Flask	A	A	B	A	Web framework	Python

Figura 2.17: Tabla comparativa de las diferentes alternativas de programación para BI basándose en madurez, popularidad, sencillez, adaptabilidad, en qué se centran y los lenguajes que admiten. Siendo A la mejor puntuación y C la peor [Schmitt, 2020].

CAPÍTULO 3

Análisis del problema

3.1 Las cooperativas agroalimentarias de la Comunidad Valenciana

En la Comunidad Valenciana hay más de 400 cooperativas agroalimentarias. Son un tipo de sociedad mercantil que asocia a titulares de explotaciones agrícolas, ganaderas o forestales y que tiene por objeto la realización de todo tipo de actividades y operaciones encaminadas al mejor aprovechamiento de las explotaciones de sus socios. La principal diferencia que existe con el resto de sociedades mercantiles (anónimas y limitadas principalmente) es en cómo retribuyen a sus socios. En el resto de sociedades mercantiles los socios o accionistas obtienen rendimiento a partir de sus participaciones o acciones en forma de una parte alícuota del beneficio neto de la empresa que se les entrega al final del ejercicio. Sin embargo, en las cooperativas de naturaleza agroalimentaria se suele observar que los beneficios finales que obtiene la empresa suelen ser muy reducidos o incluso negativos. El beneficio de los socios está en que es la propia sociedad cooperativa la que compra el producto a los socios a un precio acordado y después se encarga de distribuirlo a los puntos de venta finales. Por ejemplo, uno de los socios de la sociedad cooperativa puede ser un agricultor que produce naranjas. Este agricultor vende sus naranjas a la sociedad cooperativa, donde él ya obtiene unos beneficios por esa venta y se desentiende de las naranjas. Posteriormente será la cooperativa la que venda esas naranjas a un distribuidor como puede ser un supermercado. Al final del ejercicio, los socios pueden obtener unos pequeños beneficios de la plusvalía de las ventas, pero no es el objetivo principal de la cooperativa. El rendimiento para los socios realmente es el servicio de compra-venta que ofrece la cooperativa. El objetivo de las cooperativas no es otro que proveer a los socios buenos precios por sus productos.

En los últimos años ha sido común la desaparición de algunas de estas sociedades en nuestra comunidad. Uno de los principales problemas es la falta de profesionalización de los trabajadores. Los socios y muchos de los trabajadores no tienen formación financiera o empresarial, lo cual es normal ya que el foco de su producción es el campo. Es por ello que a las cooperativas les puede resultar difícil identificar sus problemas de producción. Al consultar sus cuentas anuales y analizarlas puede ser costoso aislar los problemas económico-financieros a los que muchas se enfrentan si no disponen de una formación más específica. Es necesario democratizar el acceso a este tipo de información en el ámbito de las cooperativas. Son sociedades de vital importancia, especialmente en la Comunidad Valenciana donde el tejido agrícola es muy relevante. Son los productores de gran parte de los productos frescos de alimentación que podemos encontrar en las tiendas, ya sean vegetales o de origen animal.

Desde la Conselleria d'Agricultura, Ramaderia i Pesca de la Generalitat Valenciana se decide abordar el problema, junto con CATIECA (Cátedra de Integración Empresarial de Cooperativas Agroalimentarias) de la facultad de ADE de la UPV se pone en marcha el proyecto. Como comentan desde la cátedra en una entrevista a Las Provincias [Meliá, 2022]: «Como ocurre en general en la mayor parte de sectores empresariales, cuenta con un tejido empresarial compuesto por empresas de muy reducida dimensión. Hoy en día nadie discute que el tamaño es una condición necesaria para competir, en la medida en que los márgenes se van estrechando cada día más, y en este escenario solo pueden sobrevivir empresas con un volumen de actividad suficiente, que les permita tener unos costes unitarios reducidos (economías de escala), y además les posibilite invertir en I+D+i» Para ello se contacta con la ETSINF y el tutor del presente trabajo y se decide desarrollar la herramienta de esta memoria.

Se pretende hacer uso del BI para desarrollar una herramienta cuyo objetivo es ayudar a las cooperativas en la toma de decisiones e identificación de problemas de naturaleza económico-financiera.

3.2 Análisis de objetivos, requisitos y asunciones

Pasaremos ahora a identificar los principales objetivos que existen en el desarrollo de la solución en base al problema que queremos solucionar. Para cada objetivo definiremos una serie de requisitos funcionales (RF).

G-1	Obtener un conocimiento integral y actualizado de la situación de las cooperativas en relación a su funcionamiento y desempeño en el entorno empresarial
RF-1.1	Representar información agregada de las cooperativas
RF-1.2	Permitir granularidad en la agregación de las cooperativas
RF-1.3	Disponer de partes diferenciadas para el análisis individual y agregado

Tabla 3.1: Tabla de requisitos para el objetivo G-1.

G-2	Proporcionar a cada cooperativa información detallada y precisa sobre su situación actual y su progreso a lo largo del tiempo
RF-2.1	Presentar información para el año más reciente que esté disponible
RF-2.2	Ofrecer análisis y recomendaciones en lenguaje natural para el año más reciente
RF-2.3	Incluir en los gráficos datos de los años previos
RF-2.4	Permitir controlar la granularidad de la dimensión del tiempo para los años

Tabla 3.2: Tabla de requisitos para el objetivo G-2.

G-3	Realizar una comparativa entre las cooperativas en términos de sus estructuras organizativas y resultados financieros, con el fin de identificar similitudes y diferencias y así establecer recomendaciones para mejorar la competitividad de las mismas
RF-3.1	Representar conjuntamente información de otras cooperativas en el análisis individual
RF-3.2	Permitir granularidad para seleccionar con qué cooperativas compararse
RF-3.3	Informar sobre su situación en base a como están las cooperativas de su entorno

Tabla 3.3: Tabla de requisitos para el objetivo G-3.

G-4	Evaluar el grado de usabilidad del producto desarrollado, teniendo en cuenta la experiencia del usuario al interactuar con él y determinando posibles mejoras que permitan una mayor eficacia y satisfacción del usuario
RF-4.1	Ofrecer instrucciones iniciales de uso
RF-4.2	Mostrar información sobre la función de los controles de la plataforma
RF-4.3	Mostrar información sobre el significado de la información financiera
RF-4.4	Usar controles intuitivos siguiendo los estándares de diseño
RF-4.4	Mostrar información también en lenguaje natural
RF-4.5	Organizar la información en diferentes apartados temáticos

Tabla 3.4: Tabla de requisitos para el objetivo G-4.

G-5	Diseñar y presentar la información de forma visualmente atractiva y accesible, y mejorar la comprensión del contenido por parte del usuario
RF-5.1	Priorizar el uso de representaciones visuales
RF-5.2	Aplicar un diseño de colores y formas coherente
RF-5.3	Usar visuales de diferentes tipos

Tabla 3.5: Tabla de requisitos para el objetivo G-5.

Definiremos un objetivo más ahora. Es de especial relevancia ya que es el único que se definió durante el desarrollo y no se tenía en cuenta desde el inicio. Surgió más adelante tras reuniones con la Consellería. Es un objetivo que ha condicionado de manera importante el desarrollo como comentaremos más adelante.

G-6	Implementar medidas de seguridad y confidencialidad para proteger la información contable de las cooperativas, asegurando la integridad y privacidad de los datos y garantizando el cumplimiento de los requisitos legales y éticos correspondientes
RF-6.1	Anonimizar los datos de cada cooperativa con respecto al resto de usuarios
RF-6.2	Versión específica para la Consellería y otra para los usuarios de las cooperativas
RF-6.3	Al ingresar, los usuarios solo pueden consultar de manera individual los datos de su cooperativa
RF-6.4	Evitar uso indebido de la información contable

Tabla 3.6: Tabla de requisitos para el objetivo G-6.

Pasaremos ahora a los requisitos no funcionales (RNF). Vamos a definirlos de manera general para toda la herramienta, ya que afectan a la mayoría de objetivos al mismo tiempo.

RNF-1	La herramienta debe estar disponible para el acceso al menos el 99 % del tiempo
RNF-2	Los datos contables deben corresponderse con los que han depositado las cooperativas en el Registro Mercantil de su provincia
RNF-3	Debe ser posible realizar mantenimiento y pequeñas modificaciones por miembros del equipo sin formación técnica específica
RNF-4	Las modificaciones que se realicen sobre los datos subyacentes deben actualizarse automáticamente en la herramienta
RNF-5	Debe ser accesible desde cualquier navegador web
RNF-6	No es necesario la instalación de software adicional para su uso
RNF-7	El tiempo de carga y de respuesta de la herramienta debe ser comparable al de herramientas similares

Tabla 3.7: Tabla con los requisitos no funcionales de la herramienta.

Por último, listaremos las asunciones de dominio (*domain assumptions*) que establecemos para la herramienta. Lo haremos de nuevo de forma general, sin centrarnos en objetivos concretos.

A-1	El usuario tiene un equipo (preferiblemente ordenador) con conexión a internet
A-2	El usuario tiene nociones muy básicas del significado de la terminología contable
A-3	El usuario tiene conocimientos básicos del uso de aplicaciones (navegación, pantallas, controles, etc) sin ser un experto
A-4	Las cooperativas reflejan una situación contable fiel a su situación patrimonial en las cuentas que depositan en el Registro Contable

Tabla 3.8: Tabla con las asunciones de dominio de la herramienta.

3.3 Análisis de riesgos

En este apartado nos centraremos en los riesgos que podemos encontrar el desarrollo y uso final de esta herramienta. Los listaremos primero en forma de tabla como hemos hecho en el apartado anterior para así tenerlos registrados. A continuación, destacaremos los detalles más relevantes de los riesgos identificados.

R-1	Errores contables en los estados de las cooperativas
R-2	Falta de interés en las funcionalidades y beneficios de la herramienta
R-3	Usuarios encuentran la herramienta demasiado difícil de usar
R-4	Imposibilidad de integrar alguna de las funcionalidades deseadas
R-5	Uso indebido de los datos contables de otras cooperativas

Tabla 3.9: Tabla con los riesgos potenciales de la herramienta.

R-1 se relaciona principalmente con posibles fraudes (intencionados o no) o la conocida como «contabilidad creativa». La contabilidad creativa es un eufemismo que hace

referencia a las prácticas contables que pueden seguir la letra de las normas de las prácticas contables estándar, pero desviarse del espíritu de esas normas con una ética contable cuestionable. Tienen como objetivo maquillar la situación patrimonial de la empresa de cara a inversores y otros organismos de una manera legal. Para que la herramienta sea útil los datos deben ser coherentes con la situación real de la empresa. Aunque actualmente no hemos encontrado problemas en los estados contables de las cooperativas, es importante hacer un seguimiento constante para detectar cualquier indicio de falsedad y tomar medidas apropiadas. Sin embargo, solucionar este problema está fuera de nuestras capacidades y del objetivo de este trabajo.

En cuanto a R-2 y R-3, será difícil detectar su posible aparición hasta las últimas fases del desarrollo, cuando los usuarios finales hagan uso de la herramienta. Sin embargo, el surgimiento mismo de la idea junto con el interés que ha despertado tanto en algunos organismos públicos como en algunas de las cooperativas involucradas hace pensar que las funcionalidades sí son de interés. En cuanto a R-4, es posible que surjan nuevas necesidades durante el desarrollo que deban ser integradas en la herramienta. Por esta razón, llevaremos a cabo un seguimiento constante para identificar y abordar cualquier eventualidad que pueda surgir.

En relación a R-5, durante el desarrollo de la herramienta surgieron algunas preocupaciones por la protección de los datos contables de las cooperativas. En principio la herramienta iba a ser completamente abierta, es decir, al entrar la cooperativa podría seleccionar de una lista en la que se encontraban todas las cooperativas de la base de datos su cooperativa y consultar sus análisis. Tras reunirse los responsables del proyecto con la Consellería, ésta les trasladó que después de hablarlo con algunas cooperativas esto podía ser un problema. No existe un impedimento real de protección de datos en sí mismo, ya que la información contable de cualquier empresa es de dominio público y cualquier ciudadano puede acudir al registro mercantil y sacar una copia de las cuentas anuales de la sociedad que desee. Sin embargo, por temas de competencia, las cooperativas no quieren que sus datos contables se encuentren desglosados a dos clics de distancia de cualquier usuario de la herramienta, especialmente competidores directos. Por esta razón, se acordó crear dos versiones de la herramienta, una para uso exclusivo del equipo universitario y la Consellería, con fines informativos y de estudio ya que no son competidores de las cooperativas. La otra, para distribuirla a las cooperativas, con más seguridad en este sentido y los datos anonimizados, implantando medidas para que cada cooperativa solo pueda consultar su información. La representación de información comparativa de otras cooperativas se hace de forma anónima. De todo esto nace el objetivo y los requisitos de la tabla 3.6 y en parte también el riesgo R-4, al estar ya elegida la solución de desarrollo cuando surgió esta problemática.

3.4 Alternativas, presupuesto y elección

La solución que mejor se adecua a los requisitos es crear un *dashboard* que permita a las cooperativas consultar información, gráficos, análisis y otros elementos visuales relacionados con su desempeño a lo largo de los últimos años. Además, se podrían comparar con otras cooperativas y obtener recomendaciones para tomar decisiones informadas.

El desarrollo de esta herramienta se enmarca en la realización de prácticas extracurriculares por parte del alumno autor de la presente memoria. El alumno será el encargado de todo el desarrollo de la herramienta. Se le proporcionan datos ya refinados para ser usados en la plataforma final. Las fases *extraction and transform* del proceso ETL ya realizadas. Además, se le proporcionan textos en forma de análisis y recomendaciones en lenguaje natural (parte de estudio económico-financiero), que se basan en marcadores

establecidos desde la FADE sobre las cooperativas y que no tienen que ver con el ámbito técnico del desarrollo. Será información que se mostrará a las cooperativas en la herramienta. En un principio se establecen entre dos y tres meses a jornada completa para el desarrollo. Posteriormente se podrá extender con algún mes extra a media jornada debido a contratiempos y retrasos en diversos aspectos tanto de la parte técnica del desarrollo como ajenos a la misma. Se debe tener en cuenta para la selección de la solución que es una primera versión de la herramienta (prueba de concepto), con un presupuesto ajustado y unas horas dedicadas limitadas.

Con todo ello, las alternativas que se consideran son las comentadas en la sección 2.4. No volveremos a describirlas ya que ahí ya lo hemos hecho en profundidad. Primeramente, se han descartado las que hemos denominado «herramientas de programación» y se ha optado por las *suites* comerciales más completas. El motivo principal es la complejidad de uso. El desarrollo se va a realizar por una sola persona, por ello, esa integración completa de la que hemos hablado que ofrecen los servicios comerciales es muy atractiva para agilizar el desarrollo, así como la rapidez en el despliegue, el mantenimiento y el soporte técnico. El uso de herramientas de programación requiere de más tiempo de desarrollo para lograr un resultado similar. También son necesarios conocimientos específicos de ciertos lenguajes de programación que el alumno no posee en su totalidad y que le requerirían un poco de tiempo acabar de perfeccionar, quitando tiempo al desarrollo. Se requiere también que los demás miembros del equipo (que no poseen formación técnica) sean capaces de realizar algunas tareas básicas de mantenimiento y pequeñas modificaciones en la herramienta, lo cual resulta muy sencillo con las *suites* comerciales.

Dentro de las *suites*, en el desarrollo se ha decidido usar PowerBI. Las principales alternativas como Tableau o Qlik son opciones muy similares, así que, en este caso la elección viene motivada por algunos detalles, en concreto dos. Primero, porque es la plataforma que tiene un uso más extendido. Es la que más se actualiza y la que tiene una comunidad más grande, lo que resulta útil a la hora de solucionar problemas o buscar soporte. Segundo, hay que tener en cuenta que el ecosistema de la universidad es Microsoft, por lo tanto, la elección de PowerBI resulta muy atractiva pues ya se disponen de licencias e integración con el resto de servicios de Microsoft lo cual hace que el desarrollo y la colaboración en el mismo sean más fáciles si optamos por PowerBI.

CAPÍTULO 4

Metodología

4.1 Power BI

El desarrollo de la herramienta se va a realizar casi en su totalidad con Power BI, una plataforma de inteligencia empresarial (BI) desarrollada por Microsoft. Esta herramienta es ideal para el análisis de datos, la generación de informes interactivos y la toma de decisiones basadas en datos. Power BI ofrece múltiples componentes que trabajan de manera integrada para proporcionar una solución completa e integral de BI:

- **Power BI Desktop** es una aplicación de escritorio que permite a los usuarios crear informes y paneles interactivos. Con Power BI Desktop, los usuarios pueden conectarse a diversas fuentes de datos. Ofrece una interfaz intuitiva y fácil de usar para la creación de visualizaciones personalizadas, consultas y transformaciones de datos, y la definición de relaciones entre los conjuntos de datos.
- **Power BI Service** es la plataforma en la nube de Power BI que permite a los usuarios publicar, compartir y colaborar en los informes y paneles creados en Power BI Desktop. Con Power BI Service, los informes se pueden acceder y ver desde cualquier dispositivo con conexión a Internet, lo que permite a los usuarios obtener acceso instantáneo a los datos y análisis relevantes en cualquier momento y lugar. También permite la colaboración en tiempo real, lo que facilita el trabajo en equipo y la toma de decisiones conjuntas basadas en datos.
- **Power BI Mobile** es una aplicación móvil que permite a los usuarios acceder a los informes y paneles de Power BI desde dispositivos móviles, como teléfonos inteligentes y tabletas. Proporciona una experiencia optimizada para la visualización de informes en pantallas más pequeñas y ofrece la capacidad de interactuar con las visualizaciones, aplicar filtros y recibir notificaciones en tiempo real.
- **Power BI Embedded** es una opción para integrar capacidades de visualización y análisis de Power BI en aplicaciones personalizadas. Con Power BI Embedded, los desarrolladores pueden agregar fácilmente informes y paneles de Power BI a sus propias aplicaciones web o móviles, lo que brinda a los usuarios finales una experiencia de BI sin problemas dentro del contexto de sus aplicaciones existentes.
- **Power BI Gateway** es un componente que permite conectar Power BI con fuentes de datos locales o en la nube de forma segura. El Gateway permite la actualización automática de datos, lo que garantiza que los informes estén siempre actualizados con la información más reciente. Además, facilita la conexión y el acceso a datos almacenados en sistemas locales que no están directamente accesibles desde Power BI Service.

Power BI se ha convertido en una herramienta popular para empresas de todos los tamaños y sectores, ya que permite a los usuarios tomar decisiones más informadas y basadas en datos. La integración con otras soluciones y servicios de Microsoft, como Azure y Office 365, proporciona una infraestructura sólida y escalable para satisfacer las necesidades de análisis de datos y BI de las organizaciones. Además, Power BI ofrece una amplia gama de conectores y compatibilidad con múltiples fuentes de datos, lo que facilita la integración con sistemas existentes y la obtención de datos de diversas fuentes.

4.2 Herramientas de ETL

Uno de los principales intereses del desarrollo es realizar una colección de los datos contables de las cooperativas. Esto presenta un reto, ya que la información se encuentra tanto en los Registros Mercantiles de cada provincia como en el Registro Central. Una buena parte de las cuentas anuales de las cooperativas no está digitalizada, solo está disponible en papel. Por ello en el proceso ETL se ha pasado por varias fases en las que profundizaremos en el apartado de desarrollo de la solución, pero es interesante hablar aquí de algunas de las herramientas. Como se ha mencionado antes, el proceso de extracción y parte del de transformación ha sido realizado por otros miembros del equipo, alumnos que fueron contratados de forma temporal para solucionar algunas de estas tareas. En un principio se hizo una transcripción manual. Tres alumnos pasaban los datos de las cuentas de papel a un Excel que teníamos en común. Posteriormente, tras contactar con el Registro Mercantil Central, fue posible obtener muchas de las cuentas en formato XBRL.

XBRL (*eXtensible Business Reporting Language*) es un estándar de lenguaje de marcado desarrollado específicamente para la presentación y el intercambio de información financiera y de negocios. Proporciona una estructura estandarizada para la creación de informes financieros electrónicos, lo que facilita el análisis y la comparación de datos financieros de diferentes empresas. El formato XBRL se basa en XML, lo que significa que utiliza etiquetas y atributos para representar los datos. Estas etiquetas y atributos están diseñados específicamente para describir los elementos financieros, como ingresos, gastos, activos, pasivos, etc. La principal ventaja del formato XBRL es su capacidad para representar datos financieros de manera estructurada y legible tanto para los humanos como para las máquinas. Esto permite a las empresas y organizaciones generar informes financieros de forma automática y estandarizada, lo que ahorra tiempo y reduce los errores en la presentación de informes. XBRL se utiliza ampliamente en la presentación de informes financieros a organismos reguladores, como la Comisión de Valores y Bolsa (SEC) en Estados Unidos, y a entidades financieras. También es utilizado por empresas para el intercambio de información financiera con socios comerciales, inversionistas y analistas.

Una vez se han obtenido los archivos necesarios, un estudiante de la facultad ha desarrollado una herramienta específica implementada en Python que permite la extracción y transformación de archivos en formato XBRL. Esta implementación permite pasar los archivos directamente a la base de datos existente en Excel que utilizamos, simplificando así el proceso de análisis y gestión de los datos.

La parte final del proceso de transformación y el *load* se realiza directamente en Power BI, usando principalmente Power Query. Es una herramienta de transformación y preparación de datos que forma parte de la *suite* Microsoft Power BI. Permite a los usuarios realizar el proceso ETL desde una amplia variedad de fuentes de datos para su posterior análisis. Una de las principales ventajas de Power Query es su capacidad para conectarse a una amplia gama de fuentes de datos. Power Query ofrece una interfaz basada

en pasos, donde se puede controlar las transformaciones que se realizan en los datos de forma secuencial, distinguiendo cada paso. Además, Power Query tiene la capacidad de detectar cambios en las fuentes de datos subyacentes y actualizar automáticamente los conjuntos de datos, lo que garantiza la actualización y precisión de los datos en tiempo real. Esto es especialmente útil cuando se trabaja con datos en constante evolución, como bases de datos en línea o archivos compartidos, siendo este nuestro caso.

Al hacer uso conjunto de todas las herramientas mencionadas, se lleva a cabo el proceso ETL. Es importante destacar que esta es la fase principal en la que se desarrolla el trabajo colaborativo entre varios miembros del equipo, mientras que las fases posteriores son principalmente llevadas a cabo por el alumno autor del proyecto. La colaboración interdisciplinaria en esta fase es fundamental para garantizar la precisión y consistencia de los datos y facilitar la integración entre diferentes herramientas y técnicas.

4.3 Equipo y seguimiento

En términos de equipo podemos considerar que ha habido cuatro miembros fijos y luego una serie de temporales. Se ha ido contratando a varios alumnos de forma temporal para solucionar problemas particulares, en concreto tres: transcripción de datos manual desde los registros en papel, desarrollo de herramienta para procesar archivos XBRL y diseño gráfico de logos y portada. Por otro lado, los miembros fijos son los dos responsables de la cátedra (CATIECA) de la FADE y el tutor del presente TFG como miembro de la ETSINF. Son los tres profesores de la UPV que han actuado como supervisores y coordinadores del proyecto, asegurándose del funcionamiento y colaboración de los miembros. El último miembro fijo desde el inicio del proyecto es el autor de la memoria. Es el encargado del grueso del desarrollo de la plataforma en Power BI.

En el desarrollo ha habido dos partes diferenciadas. Una primera parte que comprende los tres primeros meses y que ha sido de desarrollo intensivo dejando la herramienta prácticamente finalizada. Luego, una segunda parte, de varios meses donde el desarrollo ha sido mucho más ligero, retocando funcionalidades y arreglando errores. En esta segunda fase el equipo se ha centrado sobre todo en extraer los datos contables de los registros y en temas de relaciones públicas con la Consellería y las cooperativas.

La metodología de desarrollo seguida ha sido altamente flexible, siendo similar a *Scrum* [Schwaber, 1997] en cuanto a su enfoque colaborativo y de iteración continua, aunque sin aplicar de forma estricta dicha metodología.

Durante la primera fase del desarrollo, se han llevado a cabo reuniones semanales entre los cuatro miembros fijos del equipo, incluyendo a los tres profesores coordinadores y al alumno encargado del desarrollo de Power BI. En estas reuniones se presentaban los avances realizados durante la semana y se discutían los siguientes pasos a seguir. Los coordinadores proporcionaban retroalimentación y se intercambiaban ideas y demostraciones.

De esta manera, se trabajaba en iteraciones cortas (sprints de scrum semanales), lo que permitía una mayor flexibilidad y adaptación a los cambios en los requisitos y condiciones del proyecto. Con una metodología de iteración continua y enfoque colaborativo, se aseguraba la implementación adecuada de las funcionalidades y la mejora continua durante todo el proceso de desarrollo. Esta metodología permitió mantener la transparencia, la inspección y la adaptación continua, obteniendo mejores resultados y un mayor control del proyecto.

En la segunda parte del desarrollo no había reuniones constantes de forma periódica, ya que el trabajo a nivel de implementación era mucho más ligero. Se hacían reuniones

esporádicas conforme se creía conveniente para rematar algunos aspectos de las funcionalidades, aplicar algún cambio en la base de datos y sobre todo, solucionar errores que se iban detectando en la fase de test.

4.4 Plan de trabajo

Pasaremos ahora a ver un plan de trabajo aproximado del desarrollo. Lo vemos en el diagrama de Gantt de la figura 4.1. En la figura 4.2 vemos un desglose con la información de cada tarea de forma más detallada. Lo hemos desglosado en tareas que representan los bloques más importantes del desarrollo. No hemos incluido dependencias ya que la única que existe es la de la primera tarea (ETL) con el resto. Evidentemente debemos tener datos iniciales para empezar a trabajar en el resto de partes del proyecto. En el resto ha habido solapes constantes debido a las revisiones que se iban produciendo fruto de las reuniones semanales. El diagrama representa únicamente los tres primeros meses de desarrollo donde si que se trabaja de forma más sistemática. La dedicación de horas era de lunes a viernes 7 horas al día según se estipula en el contrato de prácticas del estudiante. El desarrollo al inicio fue algo lento ya que el alumno no estaba familiarizado con el uso de Power BI y algunas de sus herramientas, por lo tanto, en paralelo había un proceso de aprendizaje que ocupaba algunas de las horas. Por otro lado, también había ciertas horas que se usaban a investigar modelos de *dashboards* y algunas de las tendencias y soluciones más novedosas que se están usando en la creación de informes en Power BI. Algunas de éstas han ido aplicándose en nuestra herramienta para añadir valor a la solución final.



Figura 4.1: Diagrama de Gantt aproximado del desarrollo.

ID	Name	Start Date	End Date	Duration	Color
1	ETL	Jun 01, 2022	Jun 16, 2022	12 days	[Naranja]
2	Gráficos análisis individual	Jun 17, 2022	Jul 04, 2022	12 days	[Amarillo]
3	Introducción de filtros	Jun 27, 2022	Jul 11, 2022	11 days	[Verde]
4	Menú, botones y navegación	Jul 07, 2022	Jul 20, 2022	10 days	[Cian]
5	Página análisis sectorial	Jul 15, 2022	Jul 25, 2022	7 days	[Azul]
6	Página análisis territorial	Jul 18, 2022	Jul 27, 2022	8 days	[Azul oscuro]
7	Cuadros análisis/recomendación	Jul 25, 2022	Aug 11, 2022	14 days	[Violeta]
9	Tutoriales y ayuda contextual	Aug 04, 2022	Aug 15, 2022	8 days	[Magenta]
8	Login usuario y códigos	Aug 12, 2022	Aug 31, 2022	14 days	[Rosa]

Figura 4.2: Tabla con el desglose de las tareas del diagrama de Gantt.

El desarrollo se termina durante el curso académico 2022/2023. Esta fase no se incluye en el diagrama ya que fueron muchas menos horas y no podríamos dividirlos en tareas concretas. Como hemos comentado era una fase más de retoque y solución de *bugs* y no tendría sentido representarlo de esa forma.

CAPÍTULO 5

Diseño de la solución

En esta sección avanzaremos aún más en la fase de diseño, donde profundizaremos en las decisiones que hemos tomado para concebir la infraestructura de la herramienta de manera precisa y detallada.

5.1 Arquitectura de la plataforma

Podemos dividir la herramienta en dos partes diferenciadas. Por un lado, la base de datos creada a partir de los datos contables de las cooperativas. Por otro lado, la interfaz gráfica para su consulta. En la figura 5.1 vemos la representación gráfica de cómo es el flujo de los datos a nivel general en la plataforma.

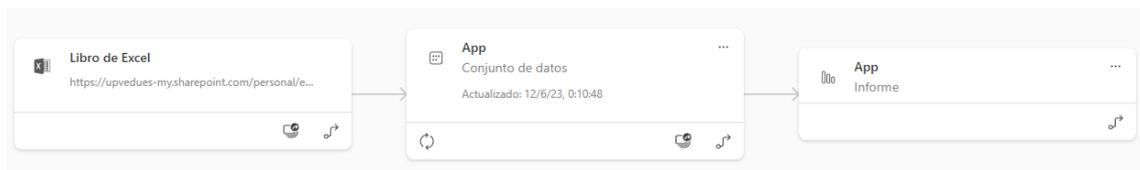


Figura 5.1: Diagrama que muestra el flujo de datos de la plataforma desplegada.

En un primer lugar tenemos un archivo Excel. Éste es el origen de los datos para la herramienta. Ese archivo contiene únicamente los datos que se usan para la herramienta, sin información contable extra que no sea de interés a nivel de representación. Más adelante detallaremos el formato en el que se encuentra la información dentro del archivo, por el momento solo es necesario saber que en ese archivo ya se ha realizado el proceso de extracción y parte de la transformación del proceso ETL. El archivo se encuentra directamente en OneDrive para que sea accesible por todos los miembros del equipo para su consulta y modificación.

A partir del archivo Excel se crea una base de datos relacional. Está representada por el segundo paso de la figura. En este conjunto de datos ya podemos ver la fase de transformación completa. Está almacenada directamente en servidores de Microsoft que guardan las bases de datos asociadas a informes de Power BI. Desde ahí, programamos una actualización diaria de forma cíclica cada veinticuatro horas. De esta forma, una vez al día se actualiza e incluye los posibles cambios que se hayan hecho sobre el archivo de Excel de manera automática. También es posible actualizarla de forma manual desde la plataforma de Service de Power BI si queremos representar algún cambio de los datos subyacentes de forma inmediata.

Por último, el tercer cuadro es la representación de los datos en un *dashboard* de forma visual. Esto será lo único que vea el usuario final. Aquí se realiza la fase de *load*. En este

punto el informe interactivo para la consulta se nutre con los datos. El usuario accede directamente a través de su navegador a este panel.

5.2 Base de datos

A partir del archivo Excel creamos una base de datos que será la que nutra de forma directa la parte visual de la herramienta. El modelo relacional de la misma lo vemos en la figura 5.2 con sus respectivas multiplicidades. El esquema que hemos decidido usar es de tipo constelación aunque muy simplificado. Tenemos dos tablas de hechos y dos dimensiones. Las dos tablas de hechos son exactamente iguales y contienen la misma información por razones de visualización de información agregada y desagregada al mismo tiempo, como comentaremos más adelante, son «FACTGeneral» y «FACTCooperativa». Las dimensiones son «DIMFecha» y «DIMCooperativa».

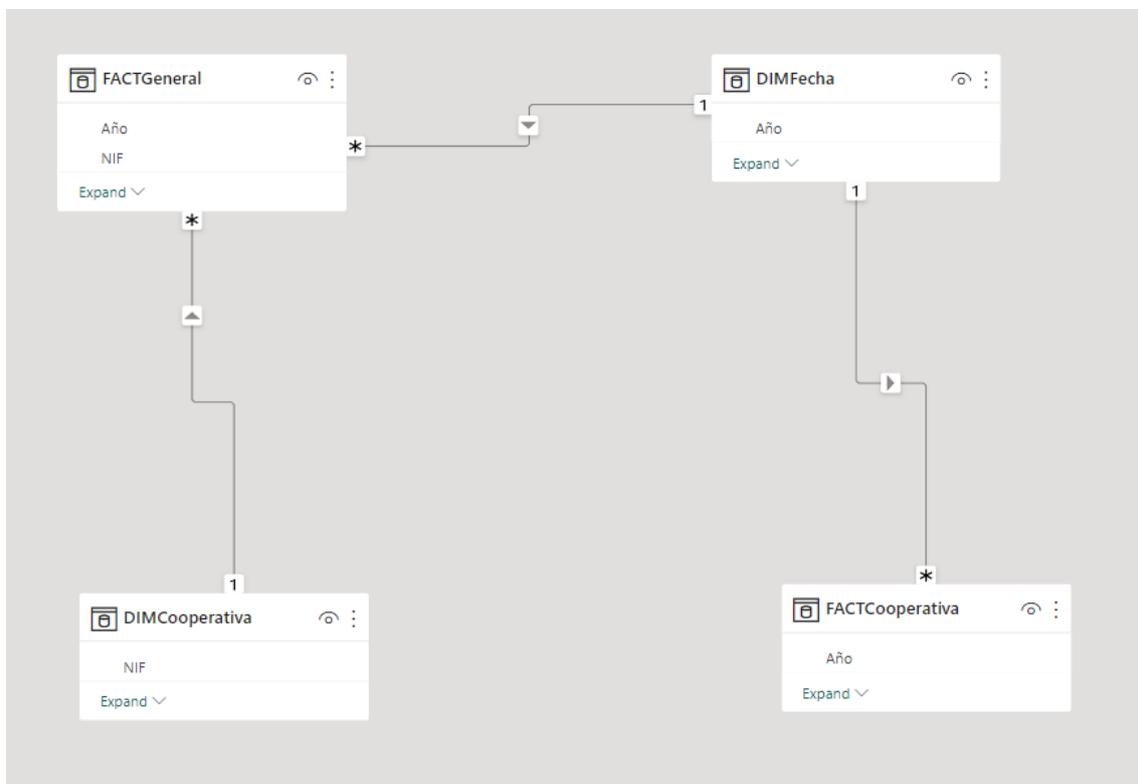


Figura 5.2: Modelo relacional de la base de datos de la plataforma.

Podemos ver las dos tablas de hechos en la figura 5.3. Usamos solamente dos tipos de datos en los atributos de todas las tablas, tanto en las de hechos como en las de dimensión. Los atributos con el símbolo del sumatorio al lado son de tipo numérico decimal, el resto son de tipo texto. La clave primaria de las tablas de hechos es el NIF de la cooperativa y el año. La estructura de cada fila es la siguiente: NIF para identificar a qué cooperativa nos referimos, año para identificar a que año en específico se refiere la información y después una serie de datos cuantitativos. Estos datos son de diverso tipo: cifra de negocio, activo total y diversos ratios financieros, pero todos ellos son marcadores económicos que varían de un año a otro. De modo que, cada tupla de las tablas de hechos representa una serie de datos para una cooperativa en concreto y un año en concreto. Cada cooperativa tiene varias filas en la tabla representando los diferentes años de los que se dispone información. Estas tablas contienen el grueso de la información cuantitativa que será presentada de forma visual y sobre la que se realizarán los cálculos necesarios.

FACTGeneral		NIF	F98314610
Año		Nombre	ABN ACEITES COOP V
∑ AT		Año	2016
∑ INCN		AT	416.059,81
NIF		RDEJ	-2.073,40
Produccion según cnae		INCN	705.813,17
∑ R10PAFDA		R1GPV	0,042
∑ R11PINIA		R2RA	1,696
∑ R12PTA		R3MARGS	0,787
∑ R13END		R4RENTS	0,009
∑ R14ENDEC		R5GFV	0,004
∑ R15ENDLP		R6INCN	705813,170
∑ R16ENDCP		R7PINFA	0,000
∑ R17CALD		R8PINMA	0,004
∑ R18CDP		R9PACA	0,002
∑ R19CMP		R10PAFDA	0,000
∑ R1GPV		R11PINIA	0,000
∑ R20CD		R12PTA	0,002
∑ R21FMV		R13END	0,003
∑ R22LG		R14ENDEC	0,000
∑ R23T		R15ENDLP	0,002
∑ R24DISPO		R16ENDCP	0,000
∑ R2RA		R17CALD	0,057
∑ R3MARGS		R18CDP	0,162
∑ R4RENTS		R19CMP	0,018
∑ R5GFV		R20CD	0,000
∑ R6INCN		R21FMV	0,210
∑ R7PINFA		R22LG	16,805
∑ R8PINMA		R23T	15,519
∑ R9PACA		R24DISPO	15,000

Figura 5.3: Tablas FACTGeneral y FACTCooperativa junto con un ejemplo de una tupla de la tabla.

La razón de tener dos tablas de hechos idénticas es que vamos a necesitar realizar operaciones y representaciones diferentes sobre la misma información de forma simultánea debido a que queremos mostrar la información de forma comparativa. Por un lado, se va a representar la información de la cooperativa propia de cada usuario y por otro lado, la agregación del resto de cooperativas según elija el usuario, para así poder compararse. Todo ello en un mismo gráfico y a la vez. Para ello, Power BI obliga a usar dos tablas diferentes aunque contengan la misma información.

Pasemos ahora a las dimensiones. Primero tenemos «DIMCooperativa» en la figura 5.4. Esta tabla contiene la información cualitativa de las cooperativas y está conectada únicamente con la tabla de hechos «FACTGeneral» a través del NIF de la cooperativa. Su clave primaria es el NIF. Hay una fila por cooperativa presente en la base de datos y cada fila representa los datos cualitativos de cada cooperativa en concreto: provincia en la que se encuentra, tamaño de la cooperativa, tipo de producción, coordenadas, etc.

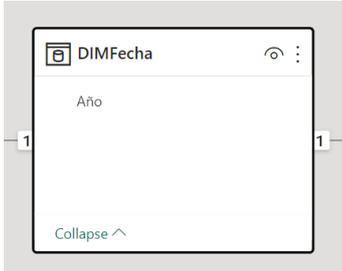
La otra dimensión es «DIMFecha», en la figura 5.5. Solo contiene información del año y es también su clave primaria y su conexión con ambas tablas e hechos. Contiene muy pocas filas y solo una columna. Una fila para cada año para el cual se dispone de infor-



NIF	F98314610
Nombre	ABN ACEITES COOP V
Localidad	NAVARRES
Comarca	La Canal de Navarrés
Provincia	Valencia
CP	46823
Coord X	-0,69194598
Coord Y	39,10319191
CP CNAE	1043
CS CNAE	
Estado	Activa
Tamano	Micro (entre 700.000 y 2.000.000)
Produccion	Aceite

Figura 5.4: Tabla DIMCooperativa y ejemplo de una tupla de la tabla.

mación contable. Para este caso en concreto no es una dimensión que aporte mucha información pero la hemos incluido de esta manera por convención en el desarrollo de este tipo de herramientas, ya que tiene una finalidad incremental. Ahora mismo solo tenemos información sobre el año porque la granularidad que buscamos en la representación de la información con respecto al tiempo es anual. Los datos de las cuentas anuales de una empresa son para cada año. Sin embargo, si en un futuro queremos añadir información trimestral o mensual, por ejemplo, con añadir esa granularidad a la tabla de la dimensión fecha sería suficiente. Es por ello que siempre se añade esta dimensión cuando creamos almacenes de datos de este tipo.



Año
2016
2017
2018
2019
2020
2021

Figura 5.5: Tabla DIMFecha y su contenido, tiene una sola columna que representa el año.

El objetivo de las tablas de dimensiones es la de filtrar. A través de la dimensión fecha, podemos filtrar los datos de las cooperativas para mostrar solo un año en concreto. Con la dimensión cooperativa podemos filtrar con el objetivo de mostrar solo las cooperativas que cumplan unas ciertas características concretas que indicará el usuario, por ejemplo, una comarca determinada o que tengan un tipo de producción específica. Las combinaciones de características son una herramienta muy potente en el filtrado. Por ejemplo: datos del año 2018 a 2020, de cooperativas de pequeño tamaño de la provincia de Castellón que produzcan frutas.

Es por ello que DIMFecha está conectada con ambas tablas de hechos y DIMCooperativa solo con FACTGeneral. La tabla FACTGeneral representa la agregación de todas las cooperativas, es la que se usa a modo de comparación. FACTCooperativa representa la cooperativa del usuario. El filtrado por fecha va a ser necesario tanto para la cooperativa propia del usuario como para la información agregada del resto. Sin embargo, filtrar

por los datos cualitativos de DIMCooperativa solo va a ser necesario para la información agregada en FACTGeneral.

Una posible alternativa habría sido usar un enfoque más cercano al esquema *snowflake*, desplegando la DIMCooperativa en varias subdimensiones. Esto nos ahorraría espacio de almacenamiento, comprimiendo la información de la base de datos. Sin embargo, la base de datos no es tan grande como para que reducir su tamaño sea una gran ventaja, por ello preferimos usar el esquema de constelación ya que nos proporciona más rendimiento en las consultas a cambio de tener algo de información duplicada.

CAPÍTULO 6

Desarrollo de la solución

En este punto se describe en profundidad el desarrollo de cada una de las partes de la plataforma de principio a fin, basándonos en el diseño que ya se ha descrito.

6.1 Formato archivo Excel

Lo primero de todo es organizar la fuente de datos de la que bebe la plataforma, realizando un proceso de transformación para adecuarlo a Power BI. Para ello debemos tener en cuenta que Excel presenta los datos en forma tabular. Para el tipo de datos que manejamos se suelen usar dos organizaciones posibles que se conocen con sus nombres en inglés: *wide* y *narrow*.

El formato *wide* tiene una estructura ancha donde cada columna representa una variable y cada fila es una observación. Es útil para conjuntos de datos pequeños y análisis rápidos. El formato *narrow* tiene una estructura estrecha donde se usan múltiples columnas para representar variables, lo que es útil para grandes conjuntos de datos y modelos de aprendizaje automático. La transformación entre los formatos se puede hacer mediante técnicas de pivotaje y similares.

El archivo original tenía una organización como la que vemos en la figura 6.1. Había una fila por cooperativa. En cada fila estaban tanto los datos cualitativos de la cooperativa como los cuantitativos. Por lo tanto, había columnas para los mismos datos pero una para cada año. Podemos ver en la figura 6.1 que tenemos columnas abreviadas como AT (activo total) seguida de 20, 19, 18, 17 y 16, representando el año. Algo similar al estilo *narrow*. Sobre este archivo era sobre el que iban volcando todos los datos que se extraían tanto en papel como de forma digital del registro mercantil.

Nombre	Localidad	Provincia	Comarca	denada X	denada Y	Código NI	Código NIF	Estado	Activo total					PN20	PN19	PN18	PN17	
									Activo mil EUR 20					propios mil EUR 20				
COOPERATIVA LA COFRENTINA	COFRENTES	Valencia	El Valle de Cofrentes-Ayora	-1061020	3,9E+07	F46412330	A-1237							30,78164				
COOP VALENCIANA DE LABRADORES Y GANADEROS SAN FRANCISCO GREEN FRUITS COOP Y	BALONES	Alicante	El Comtat	-0.803770	3,8E+16	F30302306	A-539	Activo	4.101	3.774	1.219	n.d.	n.d.	1300	660	403	n.d.	n.d.
COOP AGRICOLA GANADERA DE ALMUSAFES COOP Y	OLIVA	Valencia	La Safor	-0.41925	39,30481	F46024460	CV-315	Activo	n.d.	n.d.	n.d.	n.d.	n.d.	n.d.	n.d.	n.d.	n.d.	n.d.
CEREZAS MONTAÑA ALICANTE COOP VALENCIANA DE VALL DE GALLINERA	ALMUSSAFES	Valencia	La Ribera Baixa	-0.36273	39,475	F03164369	CV-55	Activo	n.d.	n.d.	n.d.	n.d.	n.d.	n.d.	n.d.	n.d.	n.d.	n.d.
AGROFIT S DAD COOPERATIVA (EXTINGUIDA)	PICASSENT	Valencia	L'Horta Sud	-0.454715	39,351566	F46671172	CV-207	Extinguida	n.d.	n.d.	n.d.	n.d.	n.d.	n.d.	n.d.	n.d.	n.d.	n.d.
FEMFEINA S.COOP.	PICASSENT	Valencia	L'Horta Sud	-0.572	39,27538	F40659340	CV-251	Activo	70	n.d.	n.d.	n.d.	n.d.	9	n.d.	n.d.	n.d.	n.d.
AGRICOLA SAN JOSE OBRERO DE ALBALAT COOP VALENCIANA	ALFARFAR	Valencia	L'Horta Sud	-0.46049	39,53253	F46024014	CV-288	Activo	n.d.	n.d.	n.d.	n.d.	n.d.	n.d.	n.d.	n.d.	n.d.	n.d.
COCERVA COOP. Y. (EXTINGUIDA)	ALBALAT DE	Valencia	La Ribera Baixa	-0.46049	39,53253	F46020072	CV-402	Extinguida	n.d.	n.d.	n.d.	n.d.	n.d.	n.d.	n.d.	n.d.	n.d.	n.d.
COOP AGRICOLA VIRGEN DEL ROSARIO COOP Y DE ALFAHUIR	NAQUERA	Valencia	El Camp de Túria	-0.154730	38,96663	F46024329	CV-341	Activo	n.d.	n.d.	n.d.	n.d.	n.d.	n.d.	n.d.	n.d.	n.d.	n.d.
NAVAFLORES COOP VAL (EN LIQUIDACION)	ALFAHUIR	Valencia	La Safor	-0.110446	39,899206	F36622230	CV-1422	Disuelto	n.d.	n.d.	n.d.	n.d.	n.d.	n.d.	n.d.	n.d.	n.d.	n.d.
SERVICIOS INT DE VALENCIA COOP VALENCIANA	NAVARRÉS	Valencia	La Canal de Navarrés	-0.61549	38,82111	F37222569	CV-1849	localizable según maestro finca	n.d.	n.d.	1.100	1.436	1.714	n.d.	n.d.	-13	50	n.d.
EL FLEIX S.C.V.	VALENCIA	Valencia	Valencia	-0.37670805	39,48802	F36435664	CV-1263	Baja de oficio	n.d.	n.d.	n.d.	n.d.	n.d.	n.d.	n.d.	n.d.	n.d.	n.d.
PROYECTOS AGRICOLAS COOP Y	BUÑOL	Valencia	La Hoya de Buñol	-0.52395	39,0187	F03347215	CV-1147	Activo	n.d.	n.d.	n.d.	n.d.	n.d.	n.d.	n.d.	n.d.	n.d.	n.d.
SAT 364 CY NOVACITRUS	CASTALLA	Alicante	L'Alcoià	-0.44469	39,3717	F37222079	CV-200	Activo	n.d.	n.d.	4.757	4.725	4.411	n.d.	n.d.	2.812	2.549	n.d.
SAT N 4530 GAVIGA	OLIVA	Valencia	La Safor	-0.125775	38,92246	F46201810	CV-200	Activo	n.d.	n.d.	n.d.	n.d.	n.d.	n.d.	n.d.	n.d.	n.d.	n.d.
INTERPLANT SAT 276 CY	ALDAIA	Valencia	L'Horta Oest	-0.46274	39,46226	F36375859	CV-200	Activo	n.d.	n.d.	n.d.	n.d.	n.d.	n.d.	n.d.	n.d.	n.d.	n.d.
COOP PARA EL DESARROLLO Y PROGRESO AGRICOLA SAN RAMON DE AGOST COOP Y	PICASSENT	Valencia	L'Horta Sud	-0.439306	39,317217	F03016490	A-1550	Activo	n.d.	n.d.	n.d.	392,678	353,83445	n.d.	n.d.	n.d.	n.d.	184,62942
	AGOST	Alicante	L'Alacantí	-0.64197	38,43802													

Figura 6.1: Captura de la primera versión del Excel de datos financieros.

Sin embargo, para analizarlos en Power BI es mucho más conveniente usar una aproximación de tipo *wide*. El problema reside en que para transformar este archivo al formato necesario es necesario realizar pivotajes complejos y otras operaciones avanzadas. Por ejemplo, necesitamos seleccionar las columnas en grupos de 4 o 5 (cada una de las cuales representa un dato para un año) y convertirlas en filas. Para un archivo con pocas filas no es un problema, y se puede realizar de forma manual, pero este archivo tiene más de mil filas, por lo que no es una opción. Estas opciones avanzadas no están disponibles de forma directa en Power Query que usa Power BI.

Para solucionarlo, se desarrolla un programa simple en Java al que se le introduce este archivo y produce un archivo de salida en el formato que deseamos. Para ello hemos usado de la librería Apache POI disponible para Java que nos permite trabajar con archivos Excel y otros formatos tabulares. El código del programa se puede consultar en el anexo B. Así obtenemos como salida un archivo como el de la figura 6.2. Este será el archivo que usaremos como base y que se representaba en la figura 5.1 como el primer recuadro. Este será el nuevo archivo sobre el que se irán introduciendo los nuevos datos de los que se vaya disponiendo y que actualizará automáticamente la base de datos de la plataforma. Aquí tenemos que cada cooperativa tiene una fila para cada año del que tiene datos contables. En todas sus filas tiene la misma información cualitativa, ya que esa no cambia con los años: nombre, localidad, coordenadas, etc. Lo que cambia en cada fila son sus datos contables cuantitativos, que son los que hay reflejados en las cuentas anuales del año que está representando esa fila. Desde la columna «AT» hasta la última columna (más allá de «R2RA» que es última que vemos) son los datos cuantitativos.

Nombre	Ano	CP	Localidad	Provincia	Comarca	Coord X	Coord Y	NIF	CP CNAE	CS CNAE	Estado	AT	RDEJ	INCN	R1GPV	R2RA
ANECOOP	2016	46010	VALENCIA	Valencia	València	-0,36636	39,47703	F460992224631	0150	Activa	114726,404	1737,02	574140	0,017827	5,004428	
ANECOOP	2017	46010	VALENCIA	Valencia	València	-0,36636	39,47703	F460992224631	0150	Activa	119184,101	1658,375	599018,1	0,019249	5,02599	
ANECOOP	2018	46010	VALENCIA	Valencia	València	-0,36636	39,47703	F460992224631	0150	Activa	125370,56	1392,516	650003,4	0,018192	5,184657	
ANECOOP	2019	46010	VALENCIA	Valencia	València	-0,36636	39,47703	F460992224631	0150	Activa	132659,6	1631,388	658276	0,019019	0,004962	
ANECOOP	2020	46010	VALENCIA	Valencia	València	-0,36636	39,47703	F460992224631	0150	Activa	142325,74	1406,419	726571,6	0,017613	0,005105	
SURINVER	2016	03190	PILAR DE L.Alicante		El Baix Seg	-0,81371	37,87972	F543415650161	4631	Activa	42834,31363	1258,99	69932,91	0,105721	1,632638	
SURINVER	2017	03190	PILAR DE L.Alicante		El Baix Seg	-0,81371	37,87972	F543415650161	4631	Activa	55681,32832	15,65981	72446,28	0,117475	1,301088	
SURINVER	2018	03190	PILAR DE L.Alicante		El Baix Seg	-0,81371	37,87972	F543415650161	4631	Activa	58817,32607		70702,74	0,119612	1,202073	
SURINVER	2019	03190	PILAR DE L.Alicante		El Baix Seg	-0,81371	37,87972	F543415650161	4631	Activa		8,93291				
SURINVER	2020	03190	PILAR DE L.Alicante		El Baix Seg	-0,81371	37,87972	F543415650161	4631	Activa						
SAN ALFO	2016	12549	BETXI	Castellón	La Plana Br	-0,21015	39,93151	F126545964631	4110	Activa	25555,27766	2885,34	55787,32	0,159156	2,183006	
SAN ALFO	2017	12549	BETXI	Castellón	La Plana Br	-0,21015	39,93151	F126545964631	4110	Activa	26811,79461	480,3443	58206,93	0,198622	2,170945	
SAN ALFO	2018	12549	BETXI	Castellón	La Plana Br	-0,21015	39,93151	F126545964631	4110	Activa	27292,76767		69561,02	0,18821	2,548698	
SAN ALFO	2019	12549	BETXI	Castellón	La Plana Br	-0,21015	39,93151	F126545964631	4110	Activa		2297,941				
SAN ALFO	2020	12549	BETXI	Castellón	La Plana Br	-0,21015	39,93151	F126545964631	4110	Activa						
SAT N 935€	2016	03550	SANT JOAN	Alicante	L'Alacantí	-0,40888	38,41652	F038426714631		Activa	90247,974	81,916	59648,18	0,153842	0,660936	
SAT N 935€	2017	03550	SANT JOAN	Alicante	L'Alacantí	-0,40888	38,41652	F038426714631		Activa	57655,409	63,242	76599,02	0,120764	1,328566	
SAT N 935€	2018	03550	SANT JOAN	Alicante	L'Alacantí	-0,40888	38,41652	F038426714631		Activa	65114,167	173,183	77198,01	0,133486	1,185579	
SAT N 935€	2019	03550	SANT JOAN	Alicante	L'Alacantí	-0,40888	38,41652	F038426714631		Activa	61634,833	60,327	71368,93	0,14528	1,157932	

Figura 6.2: Captura del Excel del origen de los datos.

6.2 Creación base de datos

Para crear la base de datos relacional a partir del archivo Excel que hemos descrito usamos el Power Query Editor que viene incorporado en Power BI. Como se ha comentado, es un editor que funciona con el sistema de pasos, donde cada modificación que se realiza sobre una tabla se registra como un paso. Esto nos permite modificar de forma separada cada uno de los pasos, eliminarlos o cambiar su orden. Nuestro objetivo aquí es pasar de la tabla tabular que representa el Excel al modelo relacional que hemos mostrado en la parte de diseño. Al introducir los datos por primera vez en Power BI, el programa les da una interpretación inicial y crea una tabla. A partir de ahí es cuando se puede modificar. Las cuatro tablas de la base de datos son copias de esta tabla original pero con una serie de pasos diferentes aplicados sobre cada una.

Comencemos con la tabla más sencilla: DIMFecha. Podemos ver los pasos en la figura 6.3. Los primeros seis pasos, hasta *Changed Type1* veremos que son iguales en las cuatro

tablas. Estos son pasos iniciales que usamos para elegir la primera fila como encabezado de cada columna, definir el tipo de datos que debe interpretar en cada una y renombrar algunas de ellas. Luego, lo único que hacemos es quedarnos solo con la columna que representa el año y eliminar los duplicados, así tenemos una sola fila para cada año representando la dimensión del tiempo.

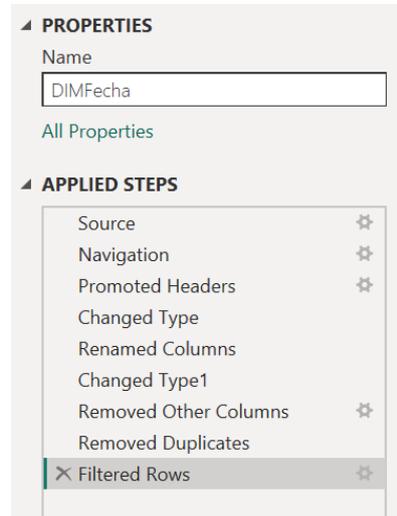


Figura 6.3: Representación de los pasos seguidos para crear la tabla DIMFecha.

Las dos tablas de hechos FACTGeneral y FACTCooperativa son idénticas, solo cambia la conexión que tienen con las tablas de dimensiones. Tenemos los pasos de ambas en la figura 6.4.

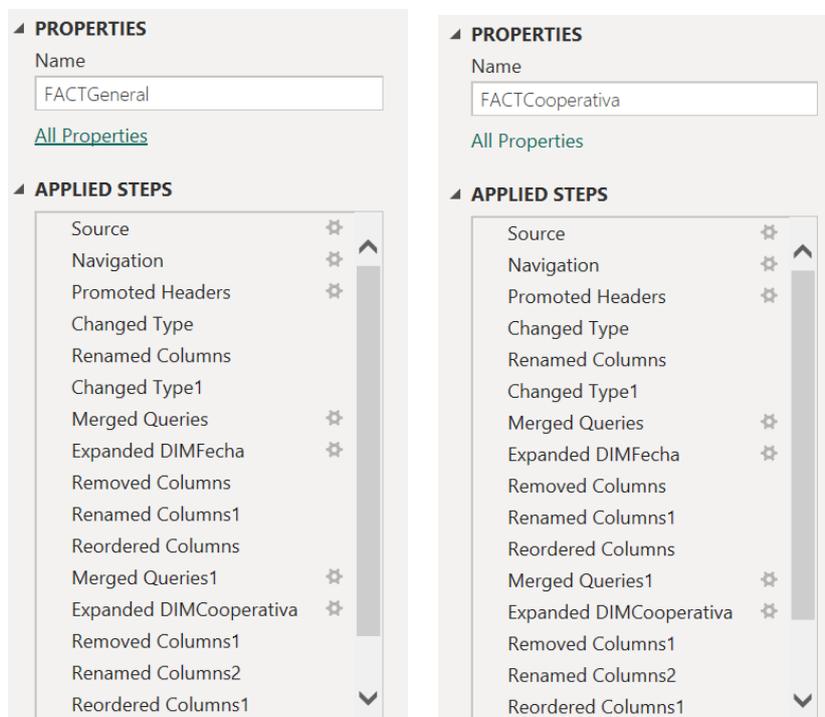


Figura 6.4: Representación de los pasos seguidos para crear las dos tablas de hechos.

Expanded DIMFecha y *Expanded DIMCooperativa* son para crear las conexiones con las tablas de las dimensiones. Después eliminaremos la conexión entre FACTCooperativa y DIMCooperativa ya que no es necesaria. El resto de pasos son para hacer limpieza.

Eliminamos las columnas que ya están representadas por las dimensiones, nos quedamos solo con el NIF, el año y los datos cuantitativos. Además, arreglamos ciertos detalles de formato, nombres y orden.

Por último, tenemos la tabla de dimensión DIMCooperativa. Los pasos están en la figura 6.5.

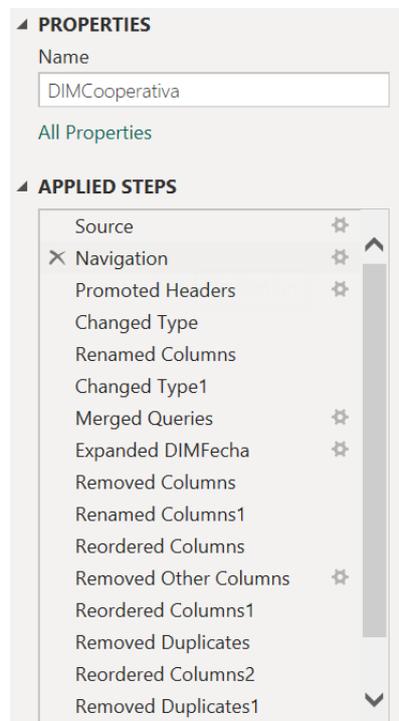


Figura 6.5: Representación de los pasos seguidos para crear la tabla DIMCooperativa

Partimos de una de las tablas de hechos y lo que hacemos es eliminar las columnas que representan los datos cuantitativos y quedarnos solo con los cualitativos que representan las características de la cooperativa. La dinámica seguida es similar a las anteriores, realizando procesos de limpieza y cambio de formato. Importante eliminar los duplicados, ya que si no tendríamos información repetida para cada año al haber partido de una tabla de hechos.

En este punto ya tenemos la base de datos relacional completa, funcionando y cargada en Power BI. Podemos dar por concluido el proceso de ETL.

6.3 Tablas de medidas

El objetivo de la herramienta no es simplemente mostrar los datos contables en crudo. Queremos aportar información extra sobre esos datos. Por ello, además de los datos puros que tenemos ya en la base de datos vamos a necesitar datos derivados que se crean a través de cálculos sobre los datos originales. Después, queremos presentar análisis en lenguaje natural sobre la situación de la cooperativa. La mejor manera de almacenar esta información en Power BI es a través de tablas de medidas.

Las medidas son cálculos que se aplican a los datos de un conjunto para realizar análisis y mostrar resultados. Son fórmulas que se crean utilizando el lenguaje de fórmulas de Power BI (DAX) y se utilizan para realizar cálculos agregados, como sumas, promedios, recuentos, máximos, mínimos, etc. Una ventaja que ofrecen las medidas es que se calculan de forma dinámica, lo que reduce el tiempo de carga inicial de la herramienta. Se

van calculando a medida que van siendo necesarias según la información que se quiera mostrar y la interacción que realice el usuario. Los conjuntos de estas medidas se almacenan en la llamadas tablas de medidas. Estas tablas se almacenan en la base de datos de la plataforma en Power BI y se representan como tablas vacías en el modelo relacional, y que no están conectadas con ninguna otra. Algunas de ellas, como las primeras que comentaremos, contienen muchas medidas, más de cien, las cuales deben ser introducidas manualmente. Otras contienen solo una o dos.

Hemos hecho uso de ellas durante todo el desarrollo para almacenar muchas medidas, podemos verlas en la figura 6.6. A lo largo del resto de la memoria veremos donde se usan. Ahora pasaremos a comentar lo más relevante sobre ellas.



Figura 6.6: Tablas de medidas creadas para la herramienta.

6.3.1. Cuartiles

Se almacenan cada uno de los cuartiles (Q1, Q2, Q3 y Q4) de cada uno de los datos cuantitativos de la tabla de hechos. Es decir, los cuartiles de cada columna que contiene datos numéricos de la tabla de hechos. Tenemos una medida para cada cuartil de cada dato, por lo tanto, cuatro medidas por columna de la tabla de hechos. Como hemos comentado, las medidas se calculan de forma dinámica, por lo que no necesitamos calcular una medida diferente para cada año ya que al mostrar la información en un gráfico por ejemplo, ya desglosa por año y aplica la fórmula para calcular el cuartil únicamente a un año en concreto. Aquí podemos ver la fórmula DAX para calcular los cuartiles para uno de los datos:

```
Q1R1 = PERCENTILE.INC(FACTGeneral[R1GPV];1)
```

```
Q2R1 = PERCENTILE.INC(FACTGeneral[R1GPV];0,75)
```

```
Q3R1 = PERCENTILE.INC(FACTGeneral[R1GPV];0,5)
```

```
Q4R1 = PERCENTILE.INC(FACTGeneral[R1GPV];0,25)
```

Para el resto sería igual, únicamente cambiando la referencia al dato de la tabla que queremos usar, el cual se encuentra entre paréntesis en esta fórmula. Usamos la función PERCENTILE.INC indicándole el valor de los cuartiles.

6.3.2. Cuartiles acumulados

Es similar a la anterior, pero en vez de calcular el valor de los cuartiles estamos calculando la amplitud del cuartil, por ejemplo, la diferencia que hay entre Q2 y Q3. La finalidad de esta medida va a ser usarla en gráficos de barras apiladas, para que así podamos mostrar en una misma barra cada uno de los cuartiles como un tramo diferente. Más adelante los mostraremos. Hacemos uso de la misma fórmula pero restando el percentil previo.

```
Q1R1' = PERCENTILE.INC(FACTGeneral [R1GPV];1) -
        PERCENTILE.INC(FACTGeneral [R1GPV];0,75)
```

```
Q2R1' = PERCENTILE.INC(FACTGeneral [R1GPV];0,75) -
        PERCENTILE.INC(FACTGeneral [R1GPV];0,5)
```

```
Q3R1' = PERCENTILE.INC(FACTGeneral [R1GPV];0,5) -
        PERCENTILE.INC(FACTGeneral [R1GPV];0,25)
```

```
Q4R1' = PERCENTILE.INC(FACTGeneral [R1GPV];0,25)
```

6.3.3. Cuartiles último año

Se calcula en cuál de los cuatro cuartiles se encuentra el valor para el año más reciente de un determinado dato cuantitativo. La salida se devuelve en formato texto: Q1, Q2, Q3 o Q4. Calculamos el valor numérico del cuartil y lo comparamos con el valor más reciente de ese dato de forma sucesiva usando *ifs*. Lo usaremos cuando generemos los textos en lenguaje natural.

```
QR13UA = IF(CALCULATE (PERCENTILE.INC(FACTGeneral [R13END];0,25);
    FILTER(ALL(FACTGeneral [Ano]); FACTGeneral [Ano] =
        UltimoAnoDisponible[UADR13]))>= 'ValoresUltimoAno' [R13UA]; "Q4";
IF(CALCULATE (PERCENTILE.INC(FACTGeneral [R13END];0,5);
    FILTER(ALL(FACTGeneral [Ano]); FACTGeneral [Ano] =
        UltimoAnoDisponible[UADR13])) >= 'ValoresUltimoAno' [R13UA]; "Q3";
IF(CALCULATE (PERCENTILE.INC(FACTGeneral [R13END];0,75);
    FILTER(ALL(FACTGeneral [Ano]); FACTGeneral [Ano] =
        UltimoAnoDisponible[UADR13])) >= 'ValoresUltimoAno' [R13UA]; "Q2";"Q1"))
```

6.3.4. Valores último año

Almacenamos el valor del último año de ciertos parámetros. Es importante tenerlo almacenado ya que nos referimos al último año del cual tenemos datos para esa cooperativa en concreto y ese parámetro. Puede haber cooperativas que tengan datos para años más recientes que otras, por eso es importante que calculemos de forma dinámica esta información. Aquí podemos ver un ejemplo de cada tipo de dato que calculamos en esta parte.

```
R2UA = LASTNONBLANKVALUE(FACTCooperativa [Ano];
    CALCULATE(SELECTEDVALUE(FACTCooperativa [R2RA])))
```

```
NumeroCooperativasUA = CALCULATE ( DISTINCTCOUNT(FACTGeneral [NIF]);
    FACTGeneral [Ano] = MAX(DIMFecha [Ano]))
```

```
MedianaR22 = CALCULATE (MEDIAN(FACTGeneral[R22LG]); FILTER(FACTGeneral;
    FACTGeneral[Ano] = UltimoAnoDisponible[UADR22]))
```

```
FacturacionMediaCooperativasUA = CALCULATE ( AVERAGE(FACTGeneral[INCN]);
    FACTGeneral[Ano] = MAX(DIMFecha[Ano]))
```

6.3.5. Último año disponible

Calculamos lo mismo que en el caso anterior, pero en vez del valor queremos almacenar el año al que corresponde. Lo necesitamos para mostrar la información de forma correcta y sobre todo para la generación de textos.

```
UADR1 = LASTNONBLANK(FACTCooperativa[Ano];
    CALCULATE(SELECTEDVALUE(FACTCooperativa[R1GPV])))
```

6.3.6. Primer año disponible

Exactamente lo mismo que en último, pero calculamos el primero disponible en lugar del último. Necesitamos saber cuál es el primer año disponible y el último para poder calcular tasas de variación.

```
PADR19 = FIRSTNONBLANK(FACTCooperativa[Ano];
    CALCULATE(SELECTEDVALUE(FACTCooperativa[R19CMP])))
```

6.3.7. Tasas de variación

Calculamos la tasa de variación de varios de los datos. Hacemos el cálculo habitual: final menos inicial entre inicial.

```
VarFacturacionMediaCooperativas =
((CALCULATE ( AVERAGE (FACTGeneral[INCN]); FACTGeneral[Ano] = MAX(DIMFecha[Ano]))
- CALCULATE ( AVERAGE (FACTGeneral[INCN]); FACTGeneral[Ano] =
    MIN(DIMFecha[Ano])))
/ CALCULATE ( AVERAGE (FACTGeneral[INCN]); FACTGeneral[Ano] =
    MIN(DIMFecha[Ano])))
```

6.4 Textos en lenguaje natural

Una de las funcionalidades más interesantes que incorporamos en la herramienta es la generación de textos informativos para el usuario. El objetivo es facilitar al usuario la comprensión de la información que se le está mostrando en los diferentes gráficos. Como hemos comentado, los usuarios en su mayoría no tienen formación financiera extensa, por lo que esta parte les puede resultar de gran interés.

Vamos a generar dos tipos de textos. Uno de carácter descriptivo, desglosando de forma verbal la situación de la cooperativa. Los llamamos análisis. El otro tipo, que llamamos recomendaciones, aporta mejoras al usuario que puede aplicar en su cooperativa para mejorar su situación. Más adelante veremos cómo se muestran al usuario estos textos, ahora hablaremos de cómo se generan. Estos textos se almacenan como medidas, en dos de las tablas de medidas que podemos ver en la figura 6.6.

Son textos condicionales donde todos los posibles textos están almacenados como medidas. Basándonos en combinaciones de horquillas en las que se encuentran los valores de ciertos datos de la cooperativa, se muestra un texto u otro al usuario. Estas combinaciones han sido determinadas desde la facultad de ADE por las profesoras responsables de la cátedra basándose en los estudios realizados. En la figura 6.7 podemos ver un ejemplo de una de las numerosas situaciones consideradas.

Situación 1

Liquidez General (R22) $1,25 \leq R22 \leq 1,75$

Ratio Tesorería (R23) $0,9 \leq R23 \leq 1,1$

Ratio de Disponibilidad (R24) $0,2 \leq R24 \leq 0,3$

Desde una perspectiva general, la COOPERATIVA se encuentra en situación de equilibrio económico financiero. Todos los ratios de liquidez: Liquidez General (R22) (QX), Ratio de Tesorería (R23) (QX) y ratio de disponibilidad (R24) (QX) están dentro del umbral de equilibrio. Como conclusión, la cooperativa tiene suficiente liquidez, como para hacer frente a la totalidad de sus deudas a corto plazo.

Figura 6.7: Ejemplo de una de las posibles situaciones en las que se puede encontrar una cooperativa. Vemos las horquillas de datos que corresponden a esta situación y el texto que se debe mostrar. Los subrayados del texto representan variables, las cuales tenemos almacenadas en las tablas de medidas que ya hemos comentado.

6.4.1. Textos Análisis

En esta tabla almacenemos todos los textos posibles del tipo análisis que pueden ser mostrados por la herramienta. Tenemos una medida para cada texto. Aquí podemos ver un par de ejemplos:

```

AnálisisLiquidezS23 = "Desde una perspectiva general en
    "&UltimoAnoDisponible[UADR22]&", la "&ValoresUltimoAno[NombreUA]&" puede
    tener problemas para hacer frente a sus pagos a corto plazo, ya que tanto el
    Ratio de Liquidez General "&FORMAT(ValoresUltimoAno[R22UA]; "#0.000")&"
    ("&CuartilesUltimoAno[QR22UA]&") como el de tesorería
    "&FORMAT(ValoresUltimoAno[R23UA]; "#0.000")&"
    ("&CuartilesUltimoAno[QR23UA]&") o el de Ratio de Disponibilidad más
    inmediata (con efectivo) "&FORMAT(ValoresUltimoAno[R24UA]; "#0.000")&"
    ("&CuartilesUltimoAno[QR24UA]&"), se encuentran por debajo de lo deseado.

Sería aconsejable para la cooperativa tomar medidas para mejorar la liquidez y
    evitar riesgos de impagos."
  
```

```

AnálisisEPES9 = "La "&ValoresUltimoAno[NombreUA]&" en el año
    "&UltimoAnoDisponible[UADR13]&" sigue una estrategia de financiación
    arriesgada, ya que su volumen de deuda ("&FORMAT(ValoresUltimoAno[R13UA];
    "#0.000")&") es elevado y la mayoría de los recursos empleados son ajenos.
  
```

Desde el punto de vista de la exigibilidad, la calidad de la deuda es buena puesto que la mayoría de los recursos son exigibles a largo plazo

```

("&FORMAT(1 - ValoresUltimoAno[R16UA]; "#0.000")&").
  
```

Además, cabe destacar que un "&FORMAT(ValoresUltimoAno[R14UA]; "#0.000")&" de sus activos totales, se financian con deuda con entidades de crédito.

Con respecto al grupo de comparación elegido, la cooperativa se sitúa en cuanto a endeudamiento ("&FORMAT(ValoresUltimoAno[R13UA]; "#0.000")&") en el cuartil "&CuartilesUltimoAno[QR13UA]&".

El coste medio del pasivo de la cooperativa es de "&FORMAT(ValoresUltimoAno[R19UA]; "#0.000")&", ha experimentado una variación del "&FORMAT(TasasVariacion[VarR19Cooperativa]; "#0.000")&" entre "&PrimerAnoDisponible[PADR19]&" y "&UltimoAnoDisponible[UADR19]&".

Como vemos, los textos no son estáticos ya que requieren mostrar valores numéricos que se calculan de forma dinámica usando las medidas que hemos comentado en la sección anterior. Las introducimos en el texto dándoles el formato correspondiente usando la función FORMAT. Éstas se incluyen en medio de los textos entre comillas y entre símbolos &.

6.4.2. Textos condicionales

En esta tabla tenemos medidas almacenamos las condiciones lógicas para mostrar los textos. Es decir, comparamos los valores de la cooperativa que tiene seleccionada el usuario con las horquillas que hemos establecido. Sabiendo en qué situación se encuentra la cooperativa, se decide mostrar un texto u otro de los almacenados en las tablas que hemos comentado anteriormente. A continuación, podemos ver un ejemplo de cómo funciona la lógica de selección. Usamos un *switch* e introducimos todas las combinaciones de horquillas que tenemos para esta sección en concreto. La comprobación es dinámica, cuando el usuario vaya a este apartado se le mostrará el texto correspondiente a su situación en concreto. Para la sección de este ejemplo podemos mostrar doce textos diferentes.

```

AnálisisEPE =
SWITCH(
  TRUE();
  0,4 > [R13UA] && 0,5 > [R16UA] && [R14UA] > 0 ;
    TextosAnálisis[AnálisisEPES1];
  0,4 > [R13UA] && 0,5 > [R16UA] && [R14UA] == 0 ;
    TextosAnálisis[AnálisisEPES2];
  0,4 > [R13UA] && 0,5 < [R16UA] && [R14UA] > 0 ;
    TextosAnálisis[AnálisisEPES3];
  0,4 > [R13UA] && 0,5 < [R16UA] && [R14UA] == 0 ;
    TextosAnálisis[AnálisisEPES4];
  0,4 < [R13UA] && 0,6 > [R13UA] && 0,5 > [R16UA] && [R14UA] > 0 ;
    TextosAnálisis[AnálisisEPES5];
  0,4 < [R13UA] && 0,6 > [R13UA] && 0,5 > [R16UA] && [R14UA] == 0 ;
    TextosAnálisis[AnálisisEPES6];
  0,4 < [R13UA] && 0,6 > [R13UA] && 0,5 < [R16UA] && [R14UA] > 0 ;
    TextosAnálisis[AnálisisEPES7];
  0,4 < [R13UA] && 0,6 > [R13UA] && 0,5 < [R16UA] && [R14UA] == 0 ;
    TextosAnálisis[AnálisisEPES8];
  0,6 < [R13UA] && 0,5 > [R16UA] && [R14UA] > 0 ;
    TextosAnálisis[AnálisisEPES9];
  0,6 < [R13UA] && 0,5 > [R16UA] && [R14UA] == 0 ;
    TextosAnálisis[AnálisisEPES10];
  0,6 < [R13UA] && 0,5 < [R16UA] && [R14UA] > 0 ;
    TextosAnálisis[AnálisisEPES11];

```

```

0,6 < [R13UA] && 0,5 < [R16UA] && [R14UA] == 0 ;
TextosAnalisis[AnalisisEPES12];

"NO hay información a mostrar."
)

```

Habiendo visto ya cómo son todos los datos subyacentes de la herramienta podemos pasar al desarrollo del *front-end*.

6.5 Interfaz visual

En esta sección nos centraremos en cómo se ha desarrollado la interfaz de uso de la herramienta. Nos enfocaremos en la parte visual, las diferentes secciones que tiene, cuáles son los controles y algunas funcionalidades más. Antes de comenzar a ver la interfaz en sí debemos comentar dos opciones que nos ofrece Power BI y que son muy relevantes en nuestro producto final: los filtros y los marcadores o *bookmarks*.

6.5.1. Filtros y marcadores

Los filtros o *slicers* son una de las funcionalidades clave que ofrece Power BI a la hora de crear *dashboards*. Permiten mostrar datos concretos que cumplan unas ciertas características, ofreciendo granularidad en el análisis. Los filtros pueden combinarse entre sí tanto como se quiera para mostrar información muy específica. Hay varias formas de aplicar filtros en Power BI. Una forma es mediante controles visuales como lo que vemos en la figura 6.8. En este caso, el usuario está seleccionando ver datos de entre los años 2016 y 2018, de cooperativas de la provincia de Castellón y con un tamaño micro.

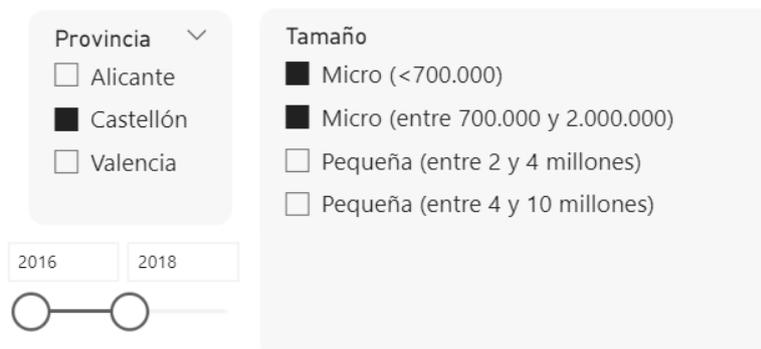


Figura 6.8: Ejemplo de algunos de los filtros que se pueden aplicar en la herramienta.

Otra manera en la que los usuarios pueden aplicar filtros es clickando directamente sobre los gráficos. Por ejemplo, en un gráfico de barras por años, si el usuario clicka en el año 2021, aplicará un filtro para mostrar solo datos del año 2021 en todos los demás elementos visuales que está viendo en ese momento. Los filtros también pueden ser aplicados por el desarrollador de manera predeterminada a alguna página en concreto de la herramienta. Además, pueden mantenerse entre páginas. Podemos sincronizar los filtros para que afecten solo a ciertos visuales o a varias páginas a la vez. En general son de gran utilidad y sus combinaciones casi infinitas, lo que permite ver la información subyacente desde muchos ángulos diferentes.

Los marcadores permiten guardar configuraciones del panel. Los elementos visuales pueden estar ocultos o mostrarse, así como ciertos filtros pueden estar aplicados o no. Los marcadores nos permiten guardar la configuración actual para más adelante poder volver a este estado con un solo click. Una forma muy útil de usarla es asociarlas a botones. Esto nos va a permitir ofrecer al usuario cambiar entre configuraciones predeterminadas por los desarrolladores para mostrar cierta información de una forma concreta. Más adelante veremos cómo usamos este tipo de enfoque en los marcadores. Entendiendo estas funcionalidades, podemos pasar a comentar cada una de las partes que tiene la interfaz visual.

6.5.2. Menú principal

Esta es la página inicial que se encuentra el usuario al abrir la plataforma. Vemos una captura de la misma en la figura 6.9. En la parte central tenemos una lista de todas las cooperativas de la base de datos. Usamos un visual de tipo *slicer* y le introducimos tanto el nombre como el NIF de las cooperativas. El usuario elige la cooperativa sobre la cual quiere consultar la información. Esta lista actúa como un filtro. Cuando el usuario elige una, se aplica un filtro sobre la tabla FACTCooperativa dejando solo visible la información individual de esa cooperativa. Este filtro se aplica a todo el resto de páginas de la herramienta. Es por esta razón que duplicábamos la tabla de hechos. Al aplicar este filtro solo podemos consultar los datos de esa cooperativa. Necesitamos la copia, en este caso FACTGeneral, a la que no se le aplica este filtro, para poder mostrar la información comparativa. La lista es extensa y está ordenada alfabéticamente. De todas formas, debido a la longitud incluimos un buscador donde se puede buscar tanto por nombre como por NIF.

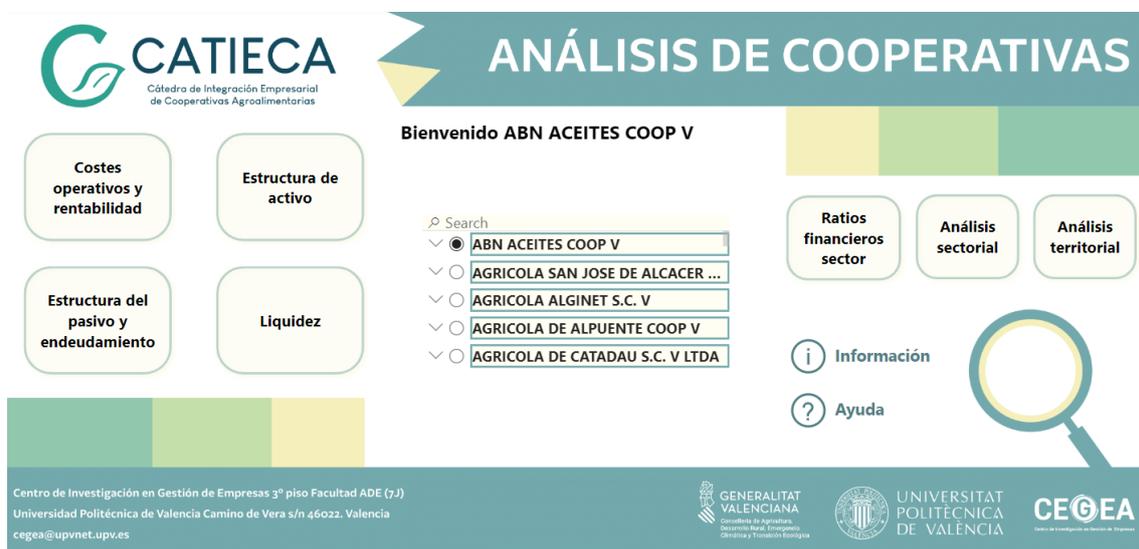


Figura 6.9: Pantalla del menú principal.

En la parte izquierda tenemos cuatro botones que sirven para abrir cada una de las correspondientes secciones. A continuación, veremos cada una de ellas en detalle. Estas cuatro representan las secciones donde el usuario verá la información concreta de su cooperativa. En la parte derecha tenemos tres botones que también nos dirigen a otras tres páginas con datos. La diferencia con los de la izquierda es que estos son análisis para todas las cooperativas, son análisis sectoriales. Es decir, estos tres no cambian según la cooperativa que seleccionamos en la lista central, mientras que los cuatro anteriores sí.

En la zona inferior derecha tenemos dos botones: información y ayuda. El botón de información nos mostrará, de forma contextual sobre esta misma vista, información sobre a dónde nos lleva cada uno de los botones anteriores. Lo tenemos en la figura 6.10. El botón de ayuda abre una ventana con un pequeño tutorial sobre el uso general de la herramienta. Vemos cómo es en la figura 6.11. El resto de elementos son puramente visuales y decorativos, como los títulos, formas y demás.



Figura 6.10: Ayuda contextual que se muestra en el menú principal al pulsar el botón de información.

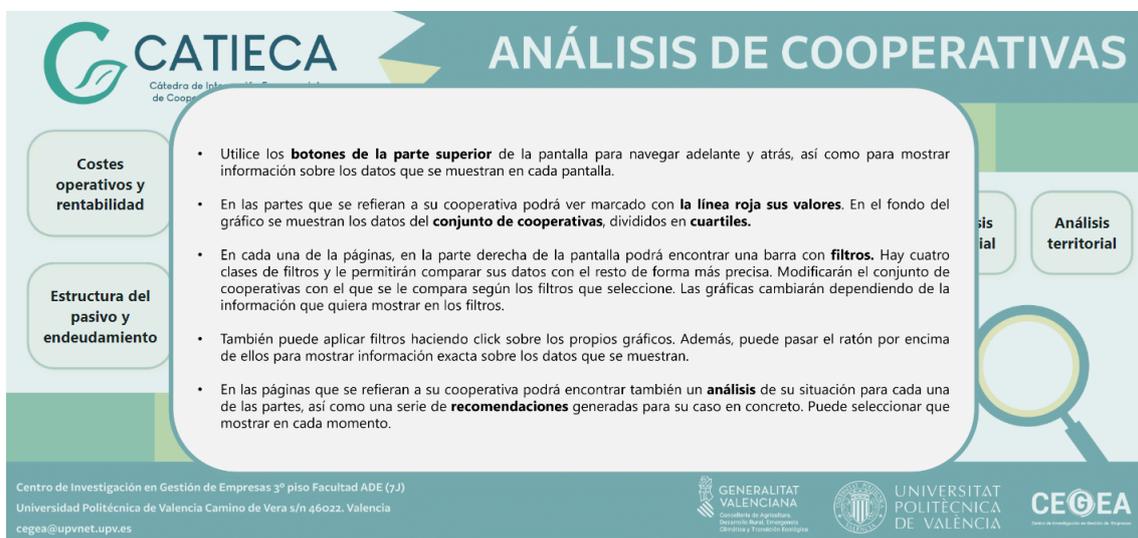


Figura 6.11: Ventana emergente con un pequeño tutorial que se muestra al pulsar el botón de ayuda.

6.5.3. Análisis individual de cooperativas

Se han desarrollado cuatro páginas a las que podemos acceder a través de los botones de la parte izquierda del menú principal. Tenemos cuatro secciones, cada una de ellas identificada por su título tanto en el botón como en el texto superior de cada una de las páginas. Tenemos dos en la figura 6.12 y las otras dos en la figura 6.13.

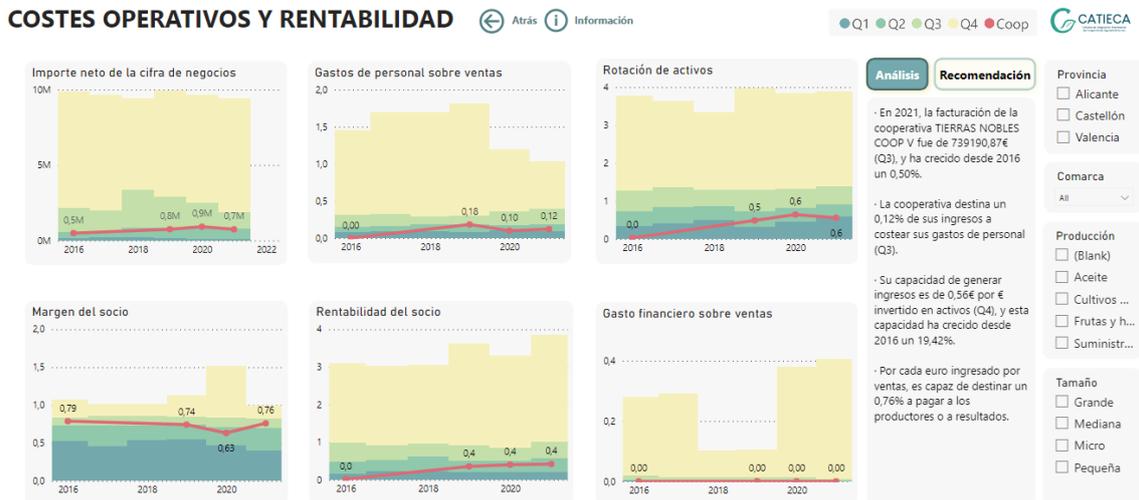


Figura 6.12: Primeras dos páginas del análisis individual de la cooperativa.

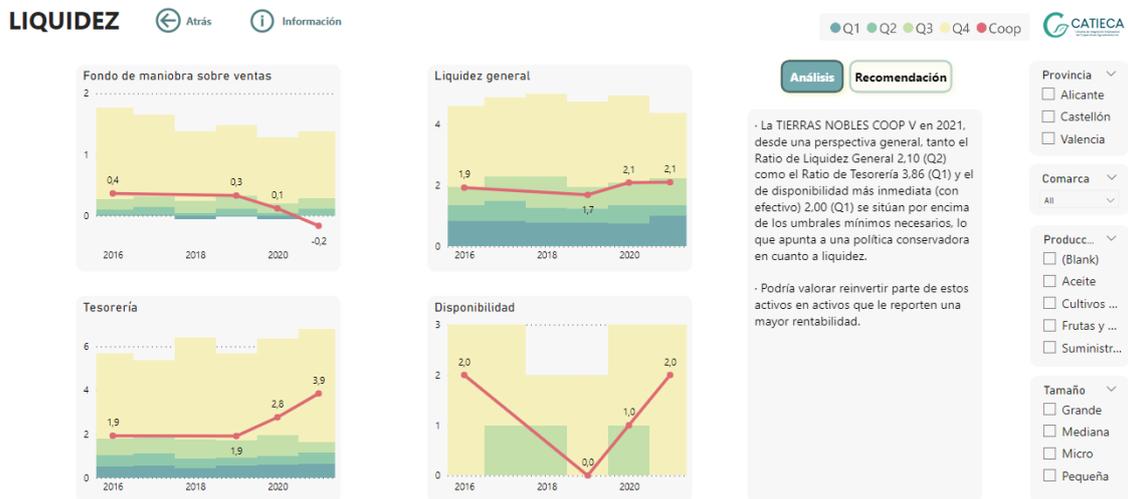


Figura 6.13: Últimas dos páginas del análisis individual de la cooperativa.

Los títulos son bastante autoexplicativos, en cada sección encontraremos desglosados datos económico-financieros de ese tipo. Son las siguientes:

- **Costes operativos y rentabilidad:** visión detallada de los gastos operativos de la empresa y del rendimiento que está obteniendo: retorno sobre activos, retorno sobre inversiones, o retorno sobre capital invertido. Todo esto conduce a una comprensión integral de la eficiencia operativa de la empresa.
- **Estructura de activo:** composición de los activos de la empresa a corto plazo o a largo plazo, tangibles, o intangibles. Estos pueden incluir desde activos físicos como maquinaria y equipos hasta activos intangibles como patentes, marcas registradas o incluso el reconocimiento de la marca. La estructura del activo proporciona una idea de en qué ha invertido la empresa y qué recursos tiene a su disposición.
- **Estructura del pasivo y endeudamiento:** información sobre las obligaciones financieras de la empresa, es decir, sus pasivos. Abarca la deuda a corto y largo plazo, las obligaciones de arrendamiento, las provisiones para responsabilidades futuras y más. Este apartado puede proporcionar perspectivas sobre las estrategias de financiamiento de la empresa y su nivel de endeudamiento, lo que puede ser un indicativo de su estabilidad financiera.
- **Liquidez:** capacidad de la empresa para cubrir sus necesidades financieras a corto plazo. Incluye indicadores como el ratio de liquidez corriente, que compara los activos corrientes (que se pueden liquidar rápidamente) con los pasivos corrientes (que deben pagarse en el corto plazo), o el ratio de liquidez inmediata, que mide la capacidad para pagar deudas inmediatas solo con activos líquidos que no requieran venta o disposición.

Todas tienen el mismo formato. Arriba, el título que indica en cual nos encontramos. Justo a su lado, dos botones. Atrás nos regresa al menú principal. Información, muestra información contextual sobre los elementos visuales. Vemos como sería una de ellas en la figura 6.14. En este caso es especialmente interesante ya que también se explica qué representa cada uno de los gráficos y qué significa. Estos dos botones los vamos a encontrar al lado del título en todas las páginas, en todos los casos tienen las funciones que acabamos de comentar. Arriba del todo a la derecha tenemos la leyenda de los gráficos y el logo de la cátedra.

En la parte central tenemos una serie de gráficos, cada uno de ellos representando un dato o un ratio correspondiente a la sección. Los gráficos son muy interactivos. Usamos una combinación de gráfico de barras apiladas con gráfico de líneas y ponemos en el eje X el tiempo en años. Cambian de forma inmediata al aplicar filtros clickando sobre ellos o usando los filtros del lateral. Tiene un pequeño menú de opciones en la parte superior que se muestra al acercar el ratón, esto nos permite consultar aquí también la explicación sobre el gráfico o ponerlo en pantalla completa. Al pasar el ratón por encima de las barras o de la línea también se muestran los valores numéricos concretos. Todos los gráficos tienen el mismo formato en su representación. Los datos se dividen por años. La línea roja representa el valor de ese dato para la cooperativa que el usuario ha seleccionado en el menú principal. Las barras apiladas del fondo son los cuartiles de todo el resto de cooperativas para ese dato. Por este motivo tenemos medidas representando la amplitud de los cuartiles.

En la parte derecha, pegado al margen, tenemos cuatro menús que representan filtros que podemos aplicar. Todos ellos son visuales de tipo *slicer*. Podemos filtrar por provincia, comarca, tipos de producción o tamaño. Todos ellos pueden combinarse entre sí o

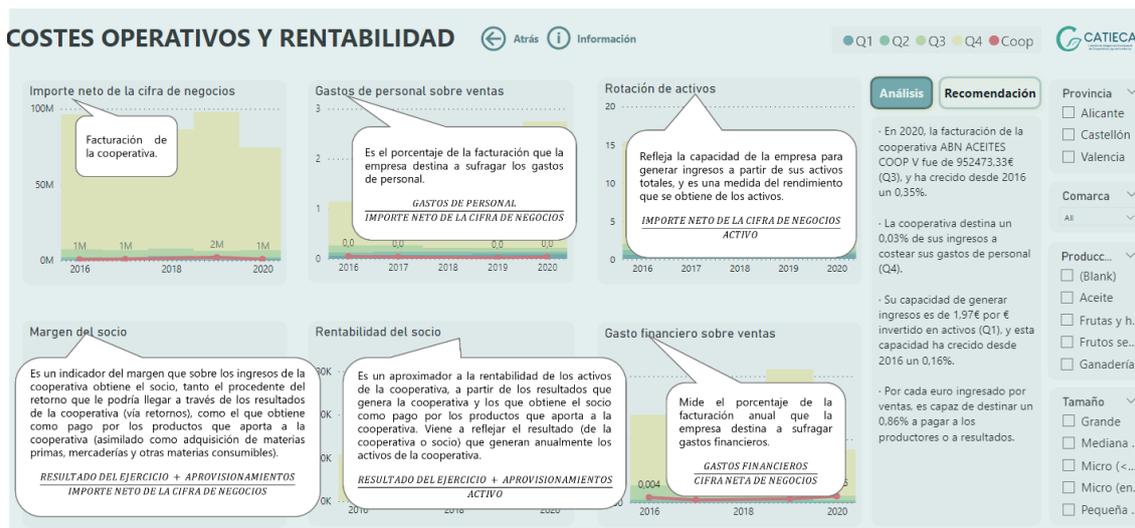


Figura 6.14: Ejemplo con la vista que se muestra al pulsar el botón de información con ayuda contextual sobre los diferentes gráficos.

aplicar selección múltiple sobre varias categorías del mismo filtro. Estos filtros nos cambiarán las barras del fondo de los gráficos según los vayamos aplicando. Nos sirven para elegir con qué cooperativas nos queremos comparar. En la figura 6.15 vemos la comparativa de los mismos gráficos sin aplicar ningún filtro y aplicando ciertos filtros. Hemos aplicado filtro para seleccionar cooperativas pequeñas de la provincia de Castellón. Se puede observar cómo cambian todos los cuartiles. Estos filtros de la parte lateral se aplican sobre la tabla DIMCooperativa, que al estar conectada con FACTGeneral (representada por las barras), selecciona solo las filas que cumplan esos requisitos representados por los datos cualitativos de la tabla de dimensión. Cada uno de los cuatro filtros viene representado por una columna de la tabla de dimensión.

Por último, entre los filtros y los gráficos tenemos un cuadro de texto con dos botones. Este cuadro de texto es el que usamos para mostrar los dos tipos de análisis en lenguaje natural que hemos comentado anteriormente. Cada una de las cuatro secciones tiene textos personalizados que se muestran según sea la situación de la cooperativa. El cuadro de texto tiene asociado la fórmula lógica para mostrar el texto como las que veíamos en el apartado de textos condicionales. Los botones que tiene encima se usan para cambiar entre mostrar el texto de tipo análisis o el de tipo recomendación. Este tipo de comportamiento no está disponible de forma nativa en Power BI pero haciendo uso de los marcadores que se han comentado al principio del capítulo se puede lograr un efecto similar. En la página realmente hay dos cuadros de texto y cuatro botones, pero la mitad de ellos están siempre ocultos. Lo que hacemos es asociar marcadores a cada uno de los botones. Al pulsar el botón se cambia entre una configuración y otra. En cada configuración dejamos solo visible el cuadro de texto de ese tipo y cambiamos la forma visual y la interacción de las botones para representar cuál está pulsado y cuál se puede pulsar.

6.5.4. Páginas sector

Continuamos con dos páginas a las que accedemos con los dos primeros botones de la parte derecha del menú principal: «Ratios financieros sector» y «Análisis sectorial». Contienen análisis generales del sector de las cooperativas de la Comunidad Valenciana, no están afectadas por la cooperativa que elija el usuario en el menú principal.

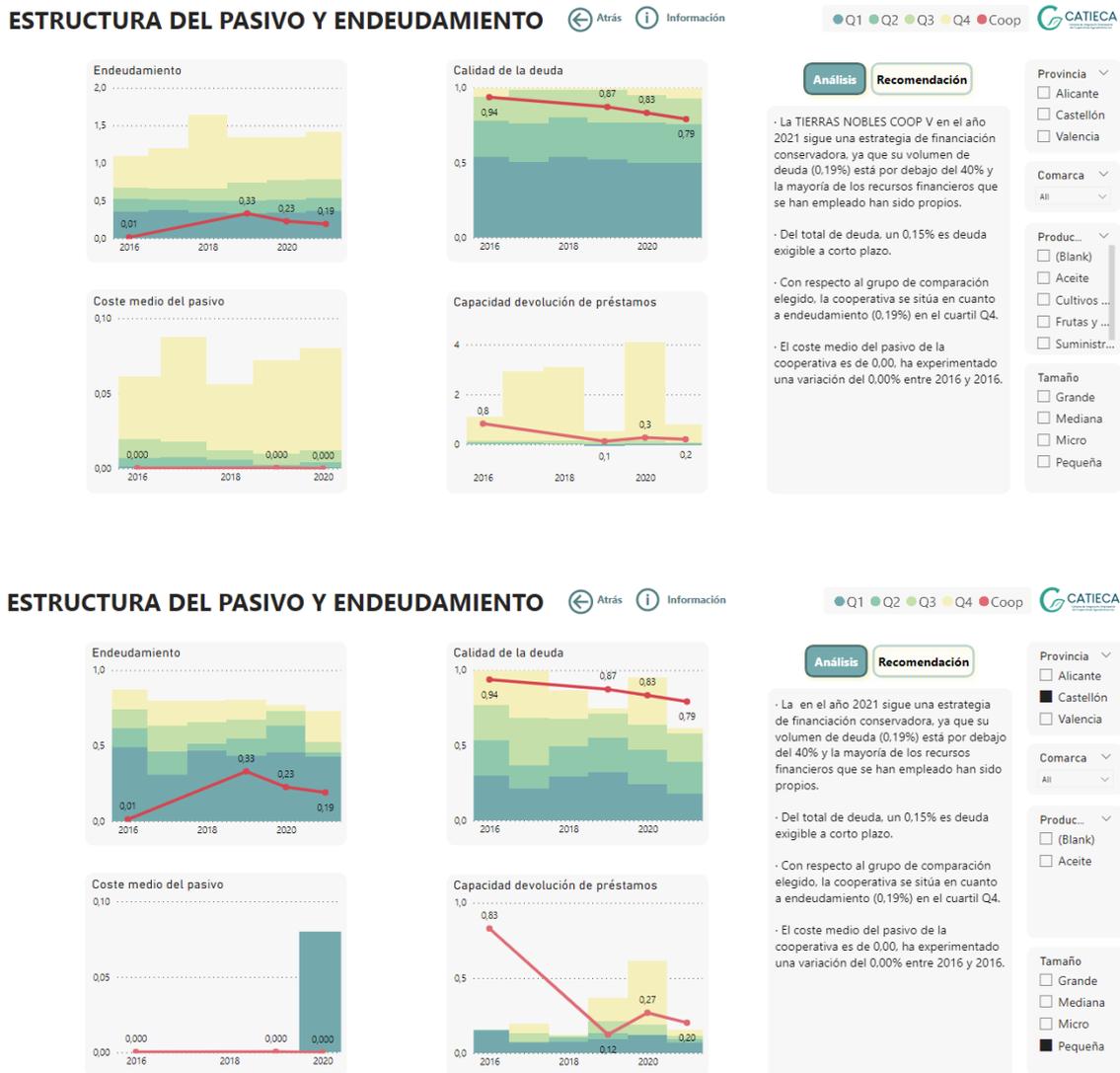


Figura 6.15: Diferencia entre los mismos gráficos al aplicar filtros.

La figura 6.16 contiene «Ratios financieros sector». Esta página es muy similar a las cuatro que acabamos de comentar. Utiliza el mismo tipo de gráfico pero en este caso la línea se refiere a la mediana del sector en vez de a la cooperativa. Es una selección de los gráficos más importante que podemos encontrar en las cuatro páginas anteriores juntas. La intención es dar una imagen general del estado del sector y de su evolución de forma rápida en una sola pantalla usando los KPIs (*Key Performance Indicators*) más importantes. De nuevo todos ellos son interactivos y cambian con los filtros de la barra lateral derecha.

La otra página es «Análisis sectorial», la vemos en la figura 6.17. Es complementaria a la página que acabamos de comentar, ya que su objetivo es también dar una imagen rápida del sector pero desde otra perspectiva. En vez de usar los gráficos de barras apiladas y línea usamos otros dos tipos. Arriba tres gráficos de tarta para mostrar las distribuciones de ciertas características de las cooperativas. Abajo tres gráficos de barras para mostrar la evolución numérica de ciertos datos. De nuevo, todos ellos son sensibles a los filtros, permitiéndonos ver la imagen de algún subgrupo de cooperativas en concreto.



Figura 6.16: Página ratios financieros sector.



Figura 6.17: Página análisis sectorial.

6.5.5. Página territorial

La última página que nos queda por comentar es la de «Análisis territorial», la tenemos en la figura 6.18. Nuevamente, presenta una imagen general del sector, pero en este caso desde una perspectiva del territorio. En la zona central tenemos dos cuadros que nos muestran el número de cooperativas que hay en cada provincia, en cada comarca y en total. El de la izquierda son cuatro visuales de tipo *card* juntos con la dimensión provincia y el número usando una medida. El de la derecha es una *multi-row card* con la dimensión comarca y la medida para el número de cooperativas nuevamente. Abajo tenemos un visual de tipo *matrix* que representa en formato tabla el número de cooperativas y la facturación media agregados por provincia. Representamos tanto el valor absoluto para el último año disponible como la tasa de variación.

El visual más interesante de esta página lo tenemos en el lado izquierdo. Es un mapa donde cada burbuja representa la ubicación de cada cooperativa. Hace uso de las columnas coordenada X y coordenada Y que tenemos en la tabla de dimensión para situar su latitud y longitud. El tamaño de la burbuja viene determinado por el dato que se tenga seleccionado de los tres que vemos en la barra que se encuentra justo debajo del mapa.



Figura 6.18: Página con el análisis territorial.

Podemos representar el tamaño según el activo total, la cifra de negocio o el resultado de cada una de las cooperativas. Por defecto, el visual del mapa solo permite asignar un dato para representar el tamaño. Sin embargo, podemos simular este comportamiento usando lo que se conoce como *field parameters*. Éstos, permiten a los usuarios cambiar dinámicamente las medidas o dimensiones que se analizan en un informe. Lo primero, debemos activar la opción *field parameters* en la configuración de Power BI Desktop. Después, creamos una tabla de medidas con una medida para cada uno de los datos que vamos a querer usar. Nosotros la hemos llamado «Datos Mapa» y es la única que no habíamos comentado de la figura 6.6. Ahora, podemos crear una nueva tabla del tipo tabla de parámetros, éstas no aparecen en el esquema relacional de la base de datos. La creamos por código:

```
Mapa = {
    ("Activo total"; NAMEOF('DatosMapa'[AT])); 0);
    ("Cifra de negocio"; NAMEOF('DatosMapa'[INCN])); 1);
    ("Resultado ejercicio"; NAMEOF('DatosMapa'[RDEJ])); 2)
}
```

Ya podemos introducir esta tabla en un *slicer* y obligar a que siempre tenga que tener una de las tres opciones pulsada usando la opción *single select*, evitando así la agregación de los campos. De esta forma simulamos un menú de selección.

Una vez más, en el margen derecho tenemos los filtros que afectarán a los visuales de la página. Esta sección es un buen ejemplo para ver que gran parte del atractivo reside en usar combinaciones de estos filtros para consultar información muy específica en un par de clicks. Tanto en esta página como en las anteriores puede no interesarnos conocer la facturación media de todas las cooperativas de la Comunidad Valenciana pero puede que sí nos interese conocer la facturación media de las cooperativas de gran tamaño de la Marina Alta que produzcan vino, por ejemplo.

Con esto, hemos acabado de comentar todas las páginas que componen la herramienta. Nos centraremos ahora en algunos puntos relevantes del desarrollo de la interfaz visual.

6.5.6. Diseño gráfico y ayuda contextual

Debemos tener en cuenta que, mayormente, los usuarios de la herramienta no tienen conocimientos específicos ni de informática ni de finanzas, por lo tanto, la sencillez de uso y la claridad de los datos es relevante.

En cuanto al diseño gráfico, optamos por algo sencillo. Con una paleta de menos de 4 colores se diseña toda la herramienta. En este sentido, lo único que se ha realizado de manera externa por otra persona son las franjas de los márgenes superior e inferior del menú principal. Son puramente decorativas y las usamos como fondo para esa página. Optamos por mantener la continuidad usando la misma paleta de colores en todos los elementos visuales de la herramienta. Además del gris que se usa como fondo en algunos de ellos para diferenciarlo del blanco del fondo general, los colores que usamos son los que vemos en la figura 6.19. Se ha intentado seguir los estándares de diseño habituales de las interfaces gráficas, tratando que sea lo más intuitiva posible. Por ejemplo, cambiar ligeramente los botones de forma y color cuando se hace *hover* sobre ellos.

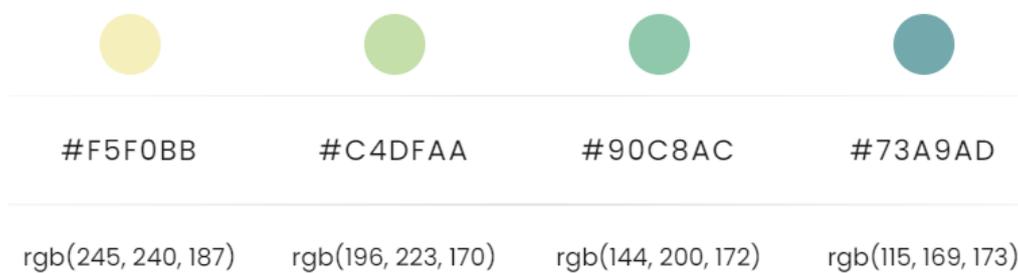


Figura 6.19: Paleta de colores usada en los elementos visuales de la herramienta.

Siguiendo con la facilidad de uso, hemos querido incorporar ayuda contextual para ayudar tanto en el uso de la herramienta como en la comprensión de la información. En todas las páginas disponemos de un botón de información que al pulsarlo nos abre una vista como la que veíamos ya en la figura 6.14. En cada caso, se muestra la información correspondiente a la página en la que se abre. Como vemos, proporciona cierta ayuda sobre el uso de los controles, pero sobre todo explica qué significa el dato que está representado en cada uno de los gráficos u otros elementos visuales. De esta forma, tratamos que el usuario entienda en profundidad lo que está viendo. En el menú principal, además del botón de información tenemos un botón de ayuda, para abrir la vista que habíamos visto en la figura 6.11. Aquí, al contrario de las anteriores, se nos muestra un tutorial más general sobre cómo hacer uso de la herramienta.

Este tipo de ventanas no están predefinidas en Power BI. Sin embargo, nuevamente podemos hacer uso de ciertos trucos para lograr un efecto similar, en este caso usando los marcadores. Lo primero que debemos hacer es crear la vista que se abre. Esta vista es una imagen en formato .png, con un fondo translúcido sobre el que ponemos los bocadillos o ventanas que queremos sobreimpresionar. Podemos usar cualquier editor de imágenes para crearla, pero es importante crearlas sobre una plantilla de una captura de la página sobre la que se van a mostrar para poder alinear todos los bocadillos perfectamente con el visual al que corresponden. Después, creamos dos marcadores: uno con la imagen visible sobrepuesta, otro con la imagen oculta. Asignamos la activación del marcador con la imagen visible al botón de información y el marcador con la imagen oculta al pulsado de la propia imagen que se ha abierto. Así pues, para salir de esta vista o quitar la ayuda, el usuario simplemente debe hacer click en cualquier sitio de la interfaz, ya que la imagen la cubre entera. Es importante usar un fondo con cierta transparencia para lograr el efecto correcto. De esta forma, simulamos la apertura de ventanas emergentes.

Del mismo modo, también usamos lo que Power BI llama *tooltips*. Son pequeños bocadillos de información que se muestra al pasar el ratón por ciertas partes de un elemento visual. Vemos algunos ejemplos en la figura 6.20. Lo usamos para mostrar la misma información que en la anterior vista, el usuario puede optar por ambas opciones. Decidimos incluir la opción de la ventana emergente además de esta última porque considerábamos que era más elegante e intuitiva.

Con la unión de un diseño gráfico intuitivo y limpio; y la implementación de ayuda contextual, podemos considerar que la herramienta es accesible tanto en su uso como en su comprensión.

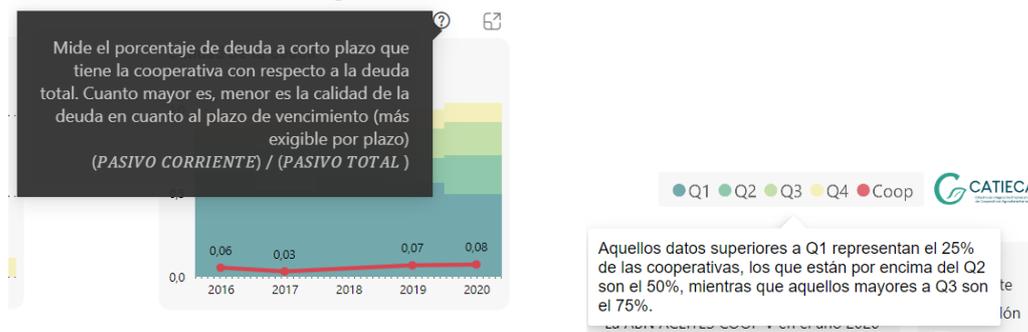


Figura 6.20: Ejemplo de dos usos de los *tooltips*.

6.5.7. Visuales personalizados

Power BI ofrece una amplia gama de gráficos y otros elementos visuales para la creación de paneles. Sin embargo, algunas funcionalidades más específicas es posible que queden fuera del alcance de los visuales nativos. Por ello, Microsoft permite utilizar visuales personalizados. Hay dos posibilidades en este sentido. Por un lado, hay un pequeño *market* de visuales al que podemos acceder desde Power BI Desktop. La mayoría son gratuitos y se incorporan solo con un par de clicks a tu barra de herramientas. Aún así, la oferta es limitada y muchos de ellos son casi inservibles. En nuestra herramienta usamos uno de este tipo. Se trata de «Text Enhancer by MAQ Software». Es un cuadro de texto como el nativo pero que ofrece más opciones de formato y sobre todo admite más tipos de entrada. Lo hemos necesitado para poder mostrar correctamente los textos personalizados que hemos comentado anteriormente, los cuales usan una entrada lógica por código para poder incorporar las condiciones según las que se debe mostrar uno u otro.

La otra manera es subir uno mismo los visuales de forma local para incorporarlo a tu barra de herramientas y usarlo en tus paneles. Puede ser un visual creado desde cero o puede ser uno descargado de la web y que no se encuentra en el *market* oficial. Hay varias opciones para crearlos como por ejemplo, usar visuales creados en R o en Python. Sin embargo, la más común es usar JavaScript. Para desarrollar un visual para Power BI se debe componer una estructura de ficheros específica la cual se compila por consola y genera un archivo en un formato específico (.pbviz) el cual se exporta a Power BI Desktop para poder usarse. Es necesario usar node.js para estos visuales. Estos visuales se deben compilar desde PowerShell. Primero, debemos instalar las herramientas de desarrollador usando el comando:

```
npm i -g powerbi-visuals-tools@latest
```

Una vez instalado, nos dirigimos en PowerShell a la ruta donde se encuentre el archivo `pbiviz.json` de nuestro proyecto. Ahí podemos ejecutar el comando:

```
pbiviz package
```

Con esto, en la carpeta `/dist/` del proyecto se nos habrá generado el archivo `.pbiviz` con el nombre e ID de nuestro visual, archivo que usaremos para importarlo a PowerBI Desktop. Es necesario que conozcamos este proceso ya que lo vamos a usar en la siguiente sección.

6.6 Inicio de sesión cooperativas

Durante el desarrollo, cuando se había realizado ya una parte importante del mismo surgió una funcionalidad crítica que debía implementarse. En un principio, la herramienta iba a ser como hemos venido comentado. En el menú principal, el usuario elegiría de una lista de todas las cooperativas la suya y se le generaría el informe correspondiente. Sin embargo, tras una reunión con la Consellería, nos plantean que esta opción es problemática y que se necesita más seguridad de la información. No porque sea información privada, ya que las cuentas anuales de las empresas son de dominio público, solo hay que solicitarlas en el correspondiente registro mercantil. Si no por un tema de competitividad entre cooperativas. Desde la Consellería nos manifiestan que los gerentes de una cooperativa pueden estar incómodos si cualquiera de sus competidores puede consultar con tan solo un par de clicks la información de su cooperativa desglosada en detalle. Por ello, surge la necesidad de crear algún tipo de mecanismo de inicio de sesión o similar que prevenga el uso indebido de la herramienta para que los usuarios pueden consultar los datos de su cooperativa exclusivamente. Señalar que, la información comparativa que hay en la herramienta no supone un problema, ya que se encuentra agregada y no es posible desglosarla individualmente.

6.6.1. RLS en Microsoft Power BI

Nos encontramos en una situación en la que necesitamos implantar RLS en nuestra plataforma. RLS son las siglas en inglés de *Row Level Security*. Es una función en bases de datos que permite controlar el acceso de usuarios a filas específicas en una tabla. Se definen políticas de seguridad a nivel de fila para determinar qué datos pueden ver o modificar los usuarios según sus credenciales y roles. RLS asegura que los usuarios solo accedan a los datos que les corresponden y mejora la privacidad de la información. Es útil para entornos con múltiples usuarios que necesitan diferentes niveles de acceso.

En nuestro caso, queremos filtrar filas de la tabla «FACTCooperativa». Necesitamos que al usuario le sean visible solamente las filas correspondientes a su cooperativa, ocultando todas las demás. El usuario podrá ver una fila para cada año del que se disponga datos contables para su cooperativa en la base de datos.

Hay que tener en cuenta que Power BI está pensado para usarse dentro de una organización, sin usuarios externos. Desde Power BI Desktop podemos crear roles y asignarles RLS. Se pueden crear *scripts* para la generación de roles, lo cual podríamos usar para crear el gran numero que necesitamos, uno por cooperativa. De este modo, cada rol corresponde a un usuario que solo puede ver la información que le corresponde.

Con ello, podemos usar una licencia de tipo Pro, que permite, entre otras cosas, proteger los reportes o *dashboard* que creamos. Antes de acceder al menú principal, el usuario debe introducir su cuenta Microsoft para así comprobar si le hemos dado autorización y

comprobaría a qué rol está asignado. Sin embargo, para esto, el usuario no solo necesitaría una cuenta de Microsoft y de Power BI (ambas gratuitas) si no que necesita también una licencia Pro si quiere acceder a este informe. No es suficiente con que el creador del informe tenga una licencia Pro, cualquier persona que quiera verlo lo necesita también. Esto, evidentemente, no es una solución aceptable, no podemos suponer que el usuario tenga cuenta en Power BI, mucho menos que tenga una licencia Pro.

Otra opción sería combinar un dominio propio con la otra licencia de pago que tiene Power BI, llamada Premium, la cual ofrece aún más opciones que la Pro. Con Power BI Premium, podemos proteger de igual manera que hacíamos con Pro, obligando a que los usuarios ingresen su cuenta antes de acceder al informe, pero en este caso, si son usuarios de dentro de la organización, pueden ver el informe tras introducir sus datos aunque no tengan licencia Premium o Pro, con la gratuita es suficiente. Intentamos proceder por esta vía con un dominio propio de la cátedra que se nos dio desde la universidad con una cuenta Premium. Sin embargo, para poder asignar licencias de Power BI, aunque sean gratuitas, a perfiles creados dentro del dominio de una organización, debe pagarse una cantidad. Todo esto se realiza desde Azure Active Directory (Azure AD). Las tarifas que se proporcionan son por carga de servidores, y aunque escojamos la más barata para tener las cuentas necesarias para las cooperativas, estamos hablando de una cantidad desorbitada. Varios miles de dólares al mes, fuera de nuestras posibilidades.

Todo esto nos fuerza a usar una alternativa con más inventiva. No es una opción nativa de Power BI pero podemos hacerla funcionar de la forma que necesitamos.

6.6.2. Visual personalizado para inicio de sesión

En el *market* de visuales oficial podemos encontrar uno llamado «Text search slicer» desarrollado por DataBrothers. Podemos verlo en la figura 6.21. Se trata de un visual que actúa como un filtro, similar al que veíamos en el menú principal de la herramienta donde el usuario elegía su cooperativa de la lista. Éste, en cambio, solo tiene la barra de búsqueda, por lo tanto, la lista con todas las cooperativas no está. El filtro se asocia a un campo, el usuario escribe en la barra y al pulsar enter o hacer click en la lupa se aplica un filtro de texto sobre el campo asociado. Por ejemplo, si lo asociamos al NIF, el usuario introduce el NIF de su cooperativa y se le mostrará solo la información correspondiente a ese NIF. En vez de usar el NIF, que es público, la idea es usar códigos personalizados que proporcionaremos a cada cooperativa para que pueda consultar su información de manera anónima.

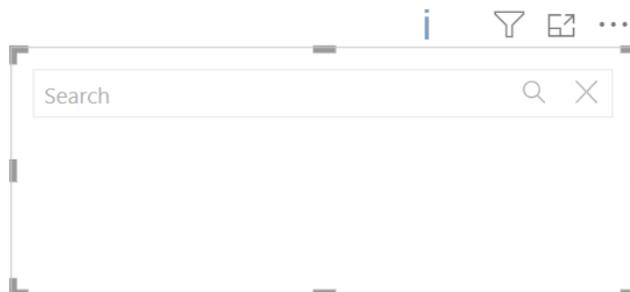


Figura 6.21: Visual Text Search Slicer en un informe.

Creamos una nueva tabla de dimensión en nuestra base de datos que llamamos «Códigos» y la asociamos con FACTCooperativa. Vemos la tabla y la conexión en la figu-

ra 6.22. Es una tabla creada desde un origen de datos nuevo, por lo que ignoramos las multiplicidades que se crean en la conexión, es una convención de Power BI. En esta tabla tenemos dos columnas: el NIF de las cooperativas y el código asociado a ese NIF. La clave primaria es el código (también puede ser el NIF, ambos son únicos) y se conecta a la tabla de hechos mediante el NIF. Asignamos la columna «codigo» como campo al filtro nuevo que estamos usando, simulamos así el funcionamiento de una contraseña.

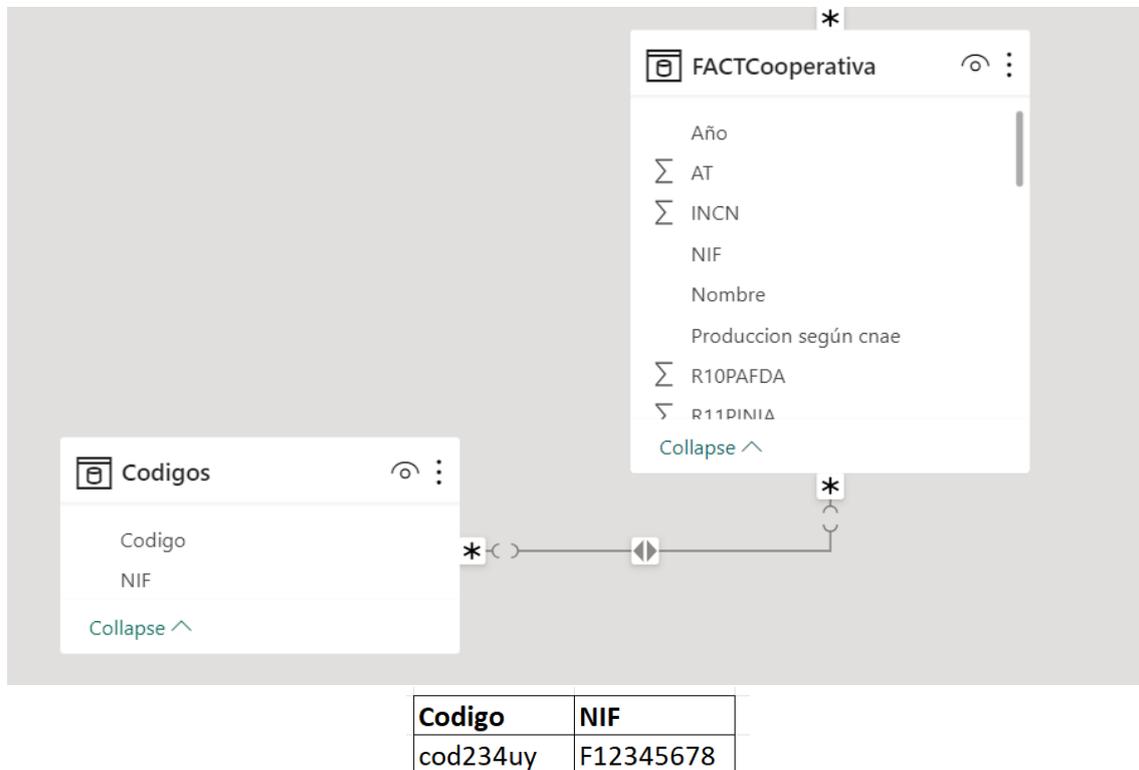


Figura 6.22: La tabla de dimensión Codigos en el modelo relacional y una tupla de ejemplo.

Los códigos se extraen desde un archivo Excel separado que se añade a las fuentes de datos. Para crear los códigos con el formato que deseamos hacemos uso de la consola de VBA (Visual Basic for Applications) disponible en Excel y usamos el siguiente *script* para crear una nueva función Excel que nos genere los códigos [MrExcel, 2013]:

```
Function RandNumLetExclude(numChars As Integer, Optional Exclude As String)
'Generates a string, numChars in length, of random alphanumeric (0-9, A-Z, a-z)
'taht optionally excludes characters in the string argument Exclude
'Can be used on a worksheet --> =RandNumLetExclude(numChars,Exclude)
Dim picknum As Integer
Dim Lets As String
Dim Nums As String
Dim FinalResult As String
Application.Volatile
Randomize

Do
    picknum = Int((2 * Rnd) + 1)
    Select Case picknum
        Case 1: Nums = CStr(Int(10 * Rnd)) 'generates 1-digit string from 0-9
        Case 2: Lets = Chr(Int((26 * Rnd) + 97)) 'generates a-z
    End Select
    If Nums = "" Then
        If InStr(Exclude, Lets) > 0 Then
```

```

        Lets = ""
        GoTo Nxt
    End If
ElseIf InStr(Exclude, Nums) > 0 Then
    Nums = ""
    GoTo Nxt
End If
FinalResult = FinalResult & Nums & Lets
Nxt:
    Nums = ""
    Lets = ""
Loop While Len(FinalResult) < numChars
RandNumLetExclude = FinalResult
End Function

```

Ahora podemos generar los códigos invocando a la función que hemos creado usando `=RandNumLetExclude(8;"0,o,l")`. De esta forma se nos genera un código de ocho caracteres con número y letras en minúscula. Excluimos el cero y la «o» y «l» minúsculas para evitar confusiones. Generamos tantos códigos como cooperativas tenemos. Luego simplemente los seleccionamos todos y comprobamos en el Excel que no haya duplicados, aunque la probabilidad de que eso ocurra es extremadamente baja.

El problema que nos surge ahora es que el buscador del filtro aplica una búsqueda de tipo contiene, es decir no busca coincidencias exactas. Por ejemplo, si ponemos la letra «s» y buscamos, nos quedaremos con todas las cooperativas que tengan una ese en su código, cosa que no queremos. Tenemos que cambiar la búsqueda a una coincidencia exacta. Por suerte, el código fuente del visual está disponible en el siguiente repositorio de GitHub:

<https://github.com/databrothers-code-storage/text-search-slicer>

Lo descargamos y tras investigar, concluimos que debemos modificar lo siguiente. En el archivo `textSearchFilterService.ts` que se encuentra en `/src/services/` en la línea 20 cambiamos el operador `Contains` por `Is`. De esta forma aplicamos un *exact match* en vez de un *contains*. Además, debemos cambiar la propiedad `GUID` en el archivo `pbviz.json`. El motivo es que si no, al introducirlo en Power BI, el programa lo reconocerá como el visual original sin modificar y no veremos los cambios, por ello cambiamos el `GUID` por cualquier otra cosa. Ahora simplemente compilamos y lo exportamos a Power BI desktop como hemos comentado en la sección anterior. Es importante activar en la configuración el uso de *custom visuals* para poder usarlo.

Una posible falla de seguridad que existe es que los códigos pueden ser visibles usando la opción *show as table*. No se puede desactivar ni modificar de ninguna forma. Se accede a ella pulsando click derecho sobre cualquier elemento visual. Nos muestra los campos asociados a ese visual en forma de tabla como podemos ver en el ejemplo de la figura 6.23.

No nos dice a qué cooperativa están asociados, por tanto, aunque se usasen no se podría saber exactamente qué datos se están viendo. Aún así, no es algo deseable, pero hemos podido comprobar que esta lista tiene una limitación, ya que solo muestra los mil primeros elementos de esa columna ordenados de forma alfabética. Por lo tanto, lo solucionamos fácilmente añadiendo simplemente mil entradas con texto en la columna códigos que no sean códigos reales, por encima de los mismos. Estas nuevas entradas no las asignamos a ningún NIF, dejamos en blanco ese atributo en este caso. De esta forma, si se accede a esta opción solo se verá esta información, la cual no representa nada, vemos como queda en la figura 6.24.

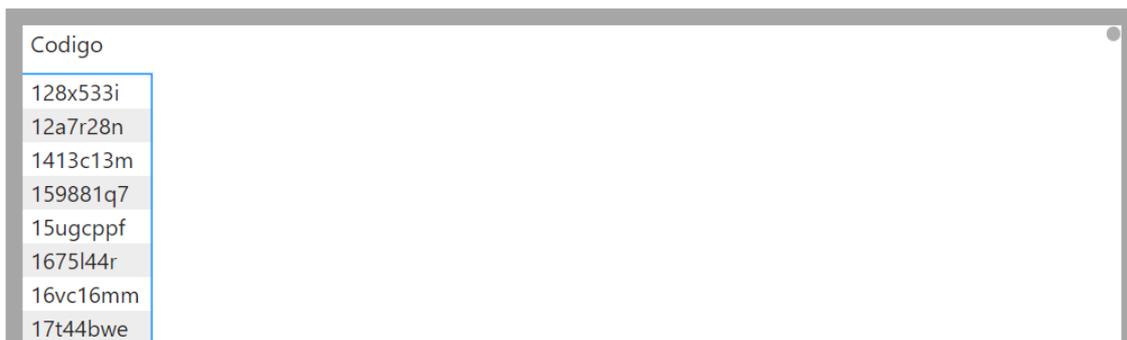


Figura 6.23: Pantalla con ejemplo de uso de la opción *show as table* donde los códigos quedan a la vista.

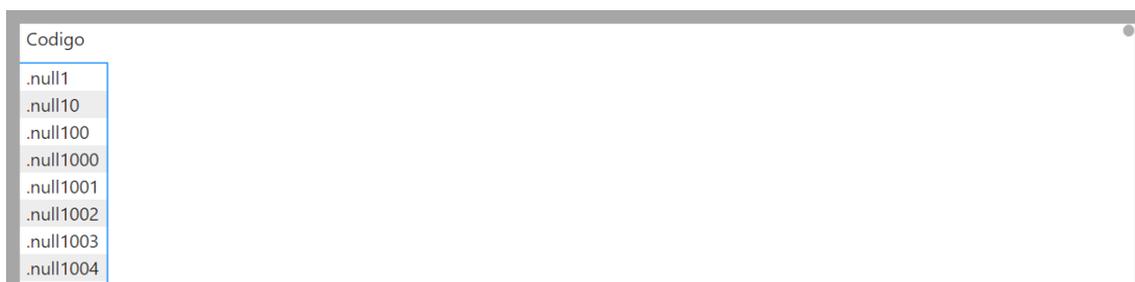


Figura 6.24: Pantalla con ejemplo de uso de la opción *show as table* donde los códigos quedan ocultos.

La opción mostrar como tabla es muy poco usada, por lo que pensamos que nuestros usuarios no harán uso de la misma. De este modo conseguimos tener algo muy similar a un inicio de sesión, usando únicamente contraseña, pero a efectos de seguridad muy similar. Es una buena forma de solventar RLS con las funcionalidades gratuitas de Power BI y se ha considerado como adecuada por los responsables del proyecto.

6.6.3. Versiones de la herramienta

Ahora por tanto vamos a tener dos versiones de la herramienta como hemos comentado anteriormente en la memoria. Una será de uso exclusivo por la Consellería y los profesionales de la universidad, es la que hemos venido comentando en todo el desarrollo. Esta versión tiene una lista completa con los nombres de todas las cooperativas en el menú principal. Tiene una finalidad más de estudio y realización de informes, no de competitividad, por lo que no hay problema con la protección de los datos.

La otra versión, será la que pasaremos a las cooperativas. Esta versión, en el menú principal tiene el buscador donde deben insertar el código que les proporcionaremos. Además, el visual del mapa que estaba en la página de análisis territorial lo modificamos. Presentaba un problema, ya que mostraba la ubicación exacta de las cooperativas, cosa que sabemos que no es deseable ya que se podrían desglosar los datos de manera individual. Por ello, ahora representamos sobre el mapa la información agregada por provincias. Perdemos detalle pero ganamos esa protección de datos necesaria. Esta versión tiene dos fuentes de datos en vez de solo una como la otra versión, ya que los códigos se encuentran en otro archivo, por lo tanto, el flujo de datos queda como el que vemos en la figura 6.25.



Figura 6.25: Flujo de datos de la nueva versión de la herramienta, contiene dos fuentes de datos.

La versión ideal de la herramienta es la primera, con toda la información pública, ya que resulta más informativa. Sin embargo, las preocupaciones en la competitividad nos han llevado a crear esta segunda versión más segura para que la usen las cooperativas.

6.7 Publicación

Ambas versiones se publican en la web y pueden ser accedidas mediante enlace directo. La versión con toda la información de manera visible será de distribución interna para la universidad y Consellería. La versión para uso de las cooperativas la publicaremos en la web de la cátedra para que esté siempre accesible desde ahí. El enlace que nos proporciona Power BI es muy largo, poco atractivo y difícil de recordar. Por ello, hemos comprado un dominio para que las cooperativas accedan a través del siguiente enlace:

Con todo esto, podemos dar por concluido el desarrollo de la herramienta, ya estaría lista y funcionando. Evidentemente es susceptible de cambios, teniendo en cuenta el *feedback* que proporcionen los usuarios, *bugs* o nueva información que se quiera incorporar. Durante toda esta última fase se han ido realizando pruebas de uso para detectar errores. Especialmente, errores a nivel de información, por ejemplo, si los textos que se generan se corresponden con la situación real de la cooperativa. Estas pruebas, al ser de carácter económico-financiero se han realizado principalmente por las profesoras responsable de la cátedra, que son las que tienen los conocimientos necesarios para analizar estos casos.

CAPÍTULO 7

Resultado

Podemos comprobar el resultado final de la herramienta en el enlace que encontramos más abajo. Se trata de la versión abierta, la cual consideramos más completa. Animamos a los lectores a que accedan y hagan pequeñas pruebas. Este enlace dejará de ser accesible al cabo de unas semanas de la publicación de la presente memoria. No podemos dejar toda la información contable abierta para cualquier lector futuro de esta memoria por las razones que ya hemos comentado. Todo el desarrollo ya se ha llevado a cabo en la versión del enlace. Sin embargo, hay dos detalles pendientes de implementar que están a la espera debido a que deben terminar el estudio que se está realizando en la FADE.

Por un lado, en cuanto a los textos, hemos mostrado ejemplos de los de análisis (descriptivos) y faltan los de las recomendaciones (prescriptivos). Los encargados de la redacción de los textos en la FADE todavía están ultimándolos en estos momentos. En cuanto a la implementación es como los de análisis. Ya hemos dejado configurados los dos tipos y el botón para realizar el cambio, aunque por ahora las recomendaciones salen vacías. Solo queda pegar los textos cuando nos los pasen en una tabla de medidas mostrarlos en el cuadro de texto.

La otra parte que queda es completar la base de datos con todos los datos. La herramienta y el formato de la base ya están configuradas para aceptar los datos nuevos de forma automática. Actualmente, en la base de datos no están todos los datos que hemos podido extraer, ya que las profesoras de la FADE están asegurándose que sean correctas cada una de las cuentas anuales antes de incluirlas. A nivel funcionamiento esto no cambia nada, pero cuanto más rica sea la base, visualizaciones más interesantes tendremos. Por ello, en la versión del enlace hemos incluido los datos completos que teníamos en la base de datos y algunos que solo teníamos de forma parcial o que hemos derivado, para conseguir un volumen más cercano al que tendrá la plataforma cuando introduzcamos los datos que faltan. Podemos acceder a través del siguiente enlace:

<https://app.powerbi.com/view?r=eyJrIjoizTVjNmYwYWEtYTNkNS00M2JkLWJmZmUtOGI1M2E5YjA0NDg3IiwidCI6ImJlNDY1NWRmLWFjNzMtNDAxZi1hN2F1LTE5OGMzYjcyZDBjNiIsImMiOjh9&pageName=ReportSectiona5b64bc0eec639350a80>

La herramienta se presentará el próximo 29 de septiembre de 2023 en un *workshop* que realizará CATIECA en la FADE. En la misma estarán presentes diversos representantes del sector, cooperativas y otros organismos.

Es interesante con el producto acabado que lo comparemos con los objetivos y requisitos que establecimos en la sección 3.2. Al repasar los objetivos y requisitos funcionales, cabe recalcar que una buena parte hemos logrado implementarlos a lo largo del proceso de desarrollo. Si nos centraremos en los objetivos G-4 (tabla 3.4) y G-5 (tabla 3.5), aunque se han validado muy favorablemente con las responsables de la cátedra, estos representan un desafío en este momento: necesitamos lanzar nuestra herramienta y re-

cibir retroalimentación de los usuarios para comprender completamente nuestro nivel de éxito en estos aspectos. Más concretamente, el objetivo G-4 se centra en la facilidad de uso y en cómo se relaciona con la propia experiencia y conocimientos del usuario. En cuanto a G-5, se refiere al diseño visual, y aunque creemos que hemos desarrollado un diseño atractivo, esperamos validar esto con las opiniones de los usuarios. Estamos comprometidos con el seguimiento activo y la evaluación de estos dos objetivos durante las primeras fases de lanzamiento del producto. Al mismo tiempo, consideramos que el resto de los objetivos ya se han cumplido con satisfacción.

En referencia a los requisitos no funcionales, podemos afirmar que los hemos implementado de manera adecuada. Vale la pena señalar que para lograr varios de estos requisitos, dependemos de elementos externos, como se mencionó anteriormente en dicha sección. No obstante, en lo que respecta a las áreas que están bajo nuestro control, estamos seguros de haberlos cumplido eficazmente.

En la tabla 3.9 presentamos los riesgos potenciales de este proyecto. Es un desafío garantizar que hemos abordado todos de forma exitosa en esta etapa anticipada, especialmente debido a que muchos dependen del uso de los usuarios finales, aun así, nos hemos esmerado para establecer todas las medidas preventivas posibles. En particular, el riesgo R-4 «Imposibilidad de integrar alguna de las funcionalidades deseadas», relativo al inicio de sesión y la protección de datos ha supuesto un auténtico reto. Aunque la solución que hemos implementado no es perfecta, es la mejor que podemos proporcionar con las tecnologías que manejamos actualmente. De todas formas, estamos convencidos de que esta primera versión de la herramienta es robusta y funcional, dejando espacio para mejoras y desarrollos en el futuro.

CAPÍTULO 8

Conclusiones

Este proyecto ha permitido el desarrollo exitoso de una herramienta intuitiva e interactiva de análisis económico-financiero, basada en Power BI, que presta servicios a las cooperativas agroalimentarias de la Comunidad Valenciana. Para esta iniciativa, se creó un almacén de datos con la información procedente del Registro Mercantil de todas las cooperativas agroalimentarias de la región y se estructuró un eficiente esquema relacional en forma de constelación. La herramienta se ha configurado en dos versiones, una para la Conselleria de Agricultura y la UPV, que brinda acceso a los datos de todas las cooperativas, y otra para las cooperativas individuales, que les permite consultar solo sus propios datos. A partir de estos datos, el sistema calcula ratios económicos, los presenta visualmente a través de gráficos interactivos permitiendo aplicar filtros y hacer comparaciones entre cooperativas. Además, a partir de estos ratios, genera análisis y recomendaciones automatizadas que apoyan la toma de decisiones en cada cooperativa. Finalmente, el proyecto, solicitado por la Cátedra de Integración Empresarial de Cooperativas Agroalimentarias (CATIECA) de la UPV, ha corroborado su compromiso con la mejora de la gestión y la eficiencia operativa de las cooperativas agroalimentarias en la Comunidad Valenciana.

En cuanto a la experiencia personal, hay varios puntos que podemos resaltar tras trabajar en este desarrollo. Personalmente, considero que he aprendido mucho sobre *Business Intelligence*, bases de datos y, sobre todo, Power BI. Cuando comencé las prácticas y tuvimos la primera reunión del equipo, se me propuso desarrollar la herramienta en Power BI. Nunca antes lo había usado, pero me dijeron que era accesible y acepté. Tuve que aprender desde cero a usarlo y considero que ahora mismo tengo un dominio muy profundo de Power BI. Lo he exprimido al máximo, usando las opciones más específicas que ofrece de programación y creación de bases de datos. Para implementar algunas de las funcionalidades he tenido que llevar Power BI al límite y usarlo de ciertas formas para las que no está pensado en un inicio. Pienso que es difícil hacer algo más avanzado a nivel de interfaz con Power BI que lo realizado. Con herramientas sencillas podemos llevar a cabo desarrollos complejos.

La mayor dificultad en este sentido ha sido el apartado del inicio de sesión, que nos ha mostrado los problemas que pueden causar requisitos que aparecen en medio del desarrollo. Aún así, hemos visto que podemos usar Power BI y otras herramientas para lograr los objetivos aunque suponga un esfuerzo extra y usar mucho la inventiva. Se pueden lograr muy buenos resultados, simplemente hay que investigar y probar diferentes cosas. Mencionar la importancia de la comunidad. La de Power BI es pequeña pero muy activa, y en varias ocasiones he dado solución a problemas que han surgido preguntando en la página oficial de la comunidad.

Resaltar también la oportunidad única que representa la creación de bases de datos desde cero, a pesar del desafío que puede suponer cuando los datos provienen de fuentes no estructuradas, no fácilmente accesibles o en diferentes formatos. Este ha sido uno de los retos más significativos en términos de tiempo en el desarrollo, dado que la recopilación de datos ha requerido un período de tiempo considerable debido a circunstancias externas. No obstante, estamos en un proceso continuo de mejora y aprendizaje.

La relación con los estudios cursados la podemos detectar en varios aspectos. Por un lado, todo lo que ha sido la creación del pequeño programa para el formato del Excel se ha hecho en Java al ser el lenguaje conocido por usarse durante la carrera. Además, todo el apartado de interfaz gráfica se ha realizado siguiendo las convenciones y estándares de diseño que se trabajan en varios proyectos de algunas asignaturas. En mi caso particular, parte del desarrollo coincidió con una estancia Erasmus, por lo tanto, no cursé asignaturas como Diseño y Gestión de Bases de Datos donde me consta que se usa Power BI en las prácticas. Sin embargo, durante mi estancia cursé principalmente asignaturas correspondientes al máster de ciencia de datos de mi universidad destino (Politécnico de Milán). Esto me dio un conocimiento profundo de todo lo relacionado con almacenes de datos, estructuras de esquemas, proceso ETL, integración de datos, etc. Los cuales han sido de gran ayuda en el proyecto.

Por otra parte, el contenido de la herramienta tiene relación directa con ADE, la otra titulación del doble grado. El estar en contacto constante con los datos y ser capaz de entenderlos junto con los análisis que se hacían ha sido importante. Todo ello, gracias a los conocimientos específicos que aporta el grado en ADE.

Ahora solo queda poner la herramienta en manos de los usuarios y conocer su opinión. A priori estamos contentos con el resultado conseguido y creemos que satisface el objetivo global que nos habíamos marcado: ayudar a las cooperativas de la Comunidad Valenciana a la toma de decisiones informada a través de su información económico-financiera para mejorar su situación. Como se ha comentado, la intención era crear un proyecto y una primera versión de la herramienta que sirviese como base. Es un proyecto en el que se continuará trabajando en un futuro.

CAPÍTULO 9

Trabajos futuros

Este proyecto, desde el momento en el que surgió, ha sido pensado como un proyecto a largo plazo. Como mínimo, la herramienta se irá actualizando cada año con la información contable nueva que vayan publicando las cooperativas. Aún así, es evidente que hay cabida para varias mejoras y nuevas implementaciones, y que es probable que se realicen por las nuevas personas que se involucren en el proyecto.

Lo primero sería crear una versión con un inicio de sesión en condiciones y aplicando RLS de forma nativa. Esto en Power BI no es posible. Por ello, sería interesante usar como plantilla la herramienta que hemos desarrollado y copiarla usando otra tecnología que ofrezca más personalización. Como hemos comentado en el estado del arte, Dash o Streamlit son opciones muy interesantes en este sentido. Si el nuevo desarrollador tiene conocimientos profundos de desarrollo web también lo puede realizar todo desde cero. Esto queda a elección personal, pero ese sería un primer paso, migrar todo a un ecosistema más acorde. Ello requeriría tener disponible siempre a alguien para el desarrollo y la solución de errores que surjan. En cambio, ahora, al estar en Power BI hay muchas cosas que se pueden cambiar por usuarios sin experiencia.

Una vez se tenga esta nueva versión, se podría construir desde ahí añadiendo nuevas funcionalidades. En este sentido habría que centrarse en el *feedback* que den los usuarios e investigar qué cosas sobran y cuales les gustaría tener, ahora mismo es difícil determinar esto si no tenemos información externa.

Bibliografía

- [Bizzflow, 2021] Bizzflow (2021). ETL Process Structure. <https://wiki.bizzflow.net/docs/getting-started/etl-process-structure/>.
- [Calvo, 2016] Calvo, D. (2016). Data Warehouse. <https://www.diegocalvo.es/data-warehouse/>.
- [Chatterjee, 2017] Chatterjee, A. (2017). Business Intelligence 101. <https://www.fingent.com/blog/business-intelligence-101/>.
- [Christ et al., 2021] Christ, J., Visengeriyeva, L., and Harrer, S. (2021). Data Mesh Architecture. <https://datamesh-architecture.com>.
- [Dehghani, 2020] Dehghani, Z. (2020). Data Mesh Principles and Logical Architecture. <https://martinfowler.com/articles/data-mesh-principles.html>.
- [Doan et al., 2012] Doan, A., Halevy, A., and Ives, Z. (2012). *Principles of Data Integration*. Elsevier. Google-Books-ID: s2YCKGrO10YC.
- [El Morr and Ali-Hassan, 2019] El Morr, C. and Ali-Hassan, H. (2019). Descriptive, Predictive, and Prescriptive Analytics. In El Morr, C. and Ali-Hassan, H., editors, *Analytics in Healthcare: A Practical Introduction*, SpringerBriefs in Health Care Management and Economics, pages 31–55. Springer International Publishing, Cham.
- [Elmalah and Nasr, 2019] Elmalah, K. and Nasr, M. (2019). Cloud Business Intelligence. *International Journal of Advanced Networking Applications*, 10:4120–4124.
- [Feick et al., 2018] Feick, M., Kleer, N., and Kohn, M. (2018). Fundamentals of Real-Time Data Processing Architectures Lambda and Kappa. *{SKILL 2018 - Studierendenkonferenz Informatik*.
- [Fowler, 2015] Fowler, M. (2015). DataLake. <https://martinfowler.com/bliki/DataLake.html>.
- [Holistics, 2020] Holistics (2020). The Rise & Fall of The OLAP Cube. <https://www.foundthisweek.com/items/rise-fall-olap-cube>.
- [IBM, 2021] IBM (2021). Structured vs. Unstructured Data: What’s the Difference? | IBM. <https://www.ibm.com/cloud/blog/structured-vs-unstructured-data>.
- [InfoDiagram, 2023] InfoDiagram (2023). Operations on OLAP cube illustration. <https://www.infodiagram.com>.
- [IPython, 2023] IPython (2023). The Jupyter Notebook — IPython. <https://ipython.org/notebook.html>.

- [Khine and Wang, 2018] Khine, P. and Wang, Z. (2018). Data lake: a new ideology in big data era. *ITM Web of Conferences*, 17:03025.
- [Lechtenbörger, 2001] Lechtenbörger, J. (2001). *Data Warehouse Schema Design*. IOS Press. Google-Books-ID: BBOM25evBHKC.
- [Li, 2018] Li, Z. (2018). NoSQL Databases. *Geographic Information Science & Technology Body of Knowledge*.
- [Martínez et al., 2013] Martínez, A. B., Lista, E. A. G., and Flórez, L. C. G. (2013). Técnicas de modelado de procesos de ETL: una revisión de alternativas y su aplicación en un proyecto de desarrollo de una solución de BI. *Scientia Et Technica*, 18(1).
- [Meliá, 2022] Meliá, E. (2022). «Debemos tender a procesos que posibiliten una dirección y toma de decisiones única». <https://www.lasprovincias.es/extras/coopera/debemos-tender-procesos-20221111234552-ntvo.html>.
- [Microsoft, 2023] Microsoft (2023). Visualización de datos | Microsoft Power BI. <https://powerbi.microsoft.com/es-es/>.
- [MrExcel, 2013] MrExcel (2013). Generate random alphanumeric code. <https://www.mrexcel.com/board/threads/generate-random-alphanumeric-code.678160/>.
- [Negash, 2004] Negash, S. (2004). Business intelligence. *Communications of the association for information systems*, 13(1):15.
- [Palmer, 2020] Palmer, D. (2020). AI-powered business intelligence: The future of analytics. <https://www.ibm.com/blog/ai-powered-business-intelligence-the-future-of-analytics/>.
- [Rizzi and Golfarelli, 2009] Rizzi, S. and Golfarelli, M. (2009). *Data Warehouse Design: Modern Principles and Methodologies*. McGraw Hill.
- [Schmitt, 2020] Schmitt, M. (2020). Data dashboarding tools | Streamlit v.s. Dash v.s. Shiny vs. Voila vs. Flask vs. Jupyter. <https://www.datarevenue.com/en-blog/data-dashboarding-streamlit-vs-dash-vs-shiny-vs-voila>.
- [Schwaber, 1997] Schwaber, K. (1997). Scrum development process. In *Business Object Design and Implementation: OOPSLA'95 Workshop Proceedings 16 October 1995, Austin, Texas*, pages 117–134. Springer.
- [upGrad, 2022] upGrad (2022). MongoDB Real World Use Cases: Advantages & Top Companies [2023]. <https://www.upgrad.com/blog/mongodb-real-world-use-cases/>.
- [Vellante, 2021] Vellante, D. (2021). Breaking Analysis: How JP Morgan is Implementing a Data Mesh on the AWS Cloud. <https://wikibon.com/breaking-analysis-how-jp-morgan-is-implementing-a-data-mesh-on-the-aws-cloud/>. Section: Big Data.
- [Verma, 2022] Verma, A. R. (2022). Data Warehousing Schemas. <https://medium.com/analytics-vidhya/data-warehousing-schemas-961391b3f3de>.
- [Waehner, 2021] Waehner, K. (2021). Kappa Architecture is Mainstream Replacing Lambda. <https://www.kai-waehner.de/blog/2021/09/23/real-time-kappa-architecture-mainstream-replacing-batch-lambda/>.

APÉNDICE A

ODS

Grado de relación del trabajo con los Objetivos de Desarrollo Sostenible (ODS).

Objetivos de Desarrollo Sostenible	Alto	Medio	Bajo	No procede
ODS 1. Fin de la pobreza.			X	
ODS 2. Hambre cero.			X	
ODS 3. Salud y bienestar.			X	
ODS 4. Educación de calidad.				X
ODS 5. Igualdad de género.				X
ODS 6. Agua limpia y saneamiento.				X
ODS 7. Energía asequible y no contaminante.				X
ODS 8. Trabajo decente y crecimiento económico.	X			
ODS 9. Industria, innovación e infraestructuras.	X			
ODS 10. Reducción de las desigualdades.		X		
ODS 11. Ciudades y comunidades sostenibles.		X		
ODS 12. Producción y consumo responsables.		X		
ODS 13. Acción por el clima.				X
ODS 14. Vida submarina.				X
ODS 15. Vida de ecosistemas terrestres.				X
ODS 16. Paz, justicia e instituciones sólidas.				X
ODS 17. Alianzas para lograr objetivos.				X

Reflexión sobre la relación del TFG/TFM con los ODS y con el/los ODS más relacionados.

Empecemos por los que tienen menos relación. Todos los que hemos marcado como «No procede» consideramos que no tienen relación alguna y que no merece la pena comentarlos. «Fin de la pobreza», «Hambre cero» y «Salud y bienestar» los hemos colocado en «Bajo». Al ser un trabajo que tiene relación con el desarrollo económico, y sobre todo en la producción de alimentos, podemos considerar que tocan tangencialmente estos objetivos. El primero porque ayudamos en el desarrollo de la economía y el segundo y el tercero por ayudar de cierta manera a aumentar la producción de alimentos, los cuales podemos considerar muy sanos y de proximidad. Aunque de nuevo, no son los más relevantes.

En el nivel medio hemos colocado tres objetivos. Primeramente, «Reducción de las desigualdades» en este caso, de carácter económico. Las cooperativas con empresas que sufren mucho a nivel económico, especialmente por los ajustados márgenes de beneficios que sacan de sus ventas. Con nuestra herramienta esperamos ayudar a las cooperativas valencianas a ser más competitivas, tanto con el resto de productos nacionales como con los que llegan del extranjero. «Ciudades y comunidades sostenibles» y «Producción y consumo responsables» van muy de la mano. Esto tiene que ver especialmente con el consumo de productos locales y de proximidad. Si conseguimos que las cooperativas aumenten su producción y presenten una mejor forma financiera, los ciudadanos podrán disfrutar de más alimentos de este tipo y a mejor precio. Así mismo, si queremos considerar la Comunidad Valenciana como una comunidad responsable debemos atender a nuestros agricultores ya que son parte esencial de la misma.

En el nivel más alto hemos colocado dos de los objetivos. «Trabajo decente y crecimiento económico» está claramente relacionado con nuestra herramienta y consideramos que es el ODS que más la representa. El de las cooperativas es un sector muy afectado por los problemas económicos y la competitividad exterior. En este sentido nuestra herramienta está completamente dirigida al crecimiento económico de estas sociedades. El objetivo final de la herramienta es que las cooperativas puedan conocer su situación económico-financiera y puedan analizarla para ver cómo pueden mejorar. Todo ello conlleva un crecimiento económico en el sector. Del mismo modo ayuda a los socios de las cooperativas, los agricultores que tan a menudo tienen un trabajo muy duro y poco remunerado, de ahí el trabajo decente. Los agricultores merecen una mejora de sus condiciones laborales aunque sea solo en la remuneración económica, que sería en lo que ayuda nuestra herramienta.

El otro que consideramos alto es «Industria, innovación e infraestructuras». Lo incluimos sobre todo por la innovación. La recolección de todos los datos contables de un sector y la creación de una herramienta de este tipo, de acceso gratuito y público para todos los afectados, es algo novedoso. Pensamos que un proyecto innovador viendo también el interés que ha generado.

Estos son todos los ODS que hemos considerado relevantes en el proyecto. En general creemos que es un proyecto comprometido con estos objetivos y que representa muy bien lo que se quiere conseguir con los mismos.

APÉNDICE B

Código programa Excel

```
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt
 * to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Main.java to edit this
 * template
 */
package main;

// Import statements
import java.io.File;
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.IOException;
import java.util.Map;
import java.util.Set;
import java.util.TreeMap;
import org.apache.poi.ss.usermodel.Cell;
import org.apache.poi.ss.usermodel.Row;
import org.apache.poi.xssf.usermodel.XSSFSheet;
import org.apache.poi.xssf.usermodel.XSSFWorkbook;
import org.apache.poi.hssf.usermodel.HSSFSheet;
import org.apache.poi.hssf.usermodel.HSSFWorkbook;
import org.apache.poi.ss.usermodel.Cell;
import org.apache.poi.ss.usermodel.FormulaEvaluator;
import org.apache.poi.ss.usermodel.Row;
/**
 *
 * @author Luis
 */
public class ExcelGenerator {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) throws Exception {

        //CREAMOS EL NUEVO EXCEL
        // Blank workbook
        XSSFWorkbook salida = new XSSFWorkbook();

        // Creating a blank Excel sheet
        XSSFSheet hoja
            = salida.createSheet("Datos");
```

```

//PREPARAMOS EL EXCEL BD PARA LEERLO
// Reading file from local directory
FileInputStream file = new FileInputStream(
    new File("BD.xlsx"));

// Create Workbook instance holding reference to
// .xlsx file
XSSFWorkbook workbook = new XSSFWorkbook(file);

// Get first/desired sheet from the workbook
XSSFSheet sheet = workbook.getSheetAt(1);

// Creating an empty TreeMap of string and Object[] []
// type
Map<Integer, Object[]> data
    = new TreeMap<Integer, Object[]>();

// Writing data to Object[]
// using put() method
//Fila con los headers.
data.put(1,
    new Object[] {"Nombre", "Año", "CP", "Localidad",
        "Provincia", "Comarca", "Coord X", "Coord Y", "NIF", "CP
        CNAE", "CS CNAE", "Estado",
            "AT", "RDEJ", "INCN", "R1GPV", "R2RA", "R3MARGS",
            "R4RENTS", "R5GFV", "R6INCN", "R7PINFA",
            "R8PINMA",
            "R9PACA", "R10PAFDA", "R11PINIA", "R12PTA",
            "R13END", "R14ENDEC", "R15ENDLP", "R16ENDCP",
            "R17CD",
            "R18CDP", "R19CMP", "R20CD", "R21FMV", "R22LG",
            "R23T", "R24DISPO"});

int k = 2;
//BUCLE QUE RECORRE CADA EMPRESA
for(int i = 3; i <= 473 ; i++) {
    //BUCLE PARA CADA AÑO
    for(int j = 0; j < 5; j++) {
        Row row = sheet.getRow(i);
        data.put(k,
            new Object[] {row.getCell(3).toString(), 2016+j,
                row.getCell(5).toString(), row.getCell(6).toString(),
                row.getCell(7).toString(),
                    row.getCell(621).getStringCellValue(),
                    row.getCell(8).toString(),
                    row.getCell(9).toString(),
                    row.getCell(10).toString()});
    }
}

```

```

        row.getCell(11).toString(),
        row.getCell(12).toString(),
        row.getCell(17).toString(),
        row.getCell(22-j).toString(),
        row.getCell(177-j).toString(),
        row.getCell(342-j).toString(),
        row.getCell(483-j).getNumericCellValue(),
        row.getCell(488-j).getNumericCellValue(),
        row.getCell(493-j).getNumericCellValue(),
        row.getCell(498-j).getNumericCellValue(),
        row.getCell(503-j).getNumericCellValue(),
        row.getCell(508-j).getNumericCellValue(),
        row.getCell(513-j).getNumericCellValue(),
        row.getCell(518-j).getNumericCellValue(),
        row.getCell(523-j).getNumericCellValue(),
        row.getCell(528-j).getNumericCellValue(),
        row.getCell(533-j).getNumericCellValue(),
        row.getCell(538-j).getNumericCellValue(),
        row.getCell(543-j).getNumericCellValue(),
        row.getCell(548-j).getNumericCellValue(),
        row.getCell(553-j).getNumericCellValue(),
        row.getCell(558-j).getNumericCellValue(),
        row.getCell(563-j).getNumericCellValue(),
        row.getCell(568-j).getNumericCellValue(),
        row.getCell(573-j).getNumericCellValue(),
        row.getCell(578-j).getNumericCellValue(),
        row.getCell(583-j).getNumericCellValue(),
        row.getCell(588-j).getNumericCellValue(),
        row.getCell(593-j).getNumericCellValue(),
        row.getCell(498-j).getNumericCellValue()});
    k++;
}
}

```

```

//file.close();

// Iterating over data and writing it to sheet
Set<Integer> keyset = data.keySet();

int rownum = 0;

for (Integer key : keyset) {

    // Creating a new row in the sheet
    Row row = hoja.createRow(rownum++);

    Object[] objArr = data.get(key);

    int cellnum = 0;

    for (Object obj : objArr) {

        // This line creates a cell in the next
        // column of that row

```

```
        Cell cell = row.createCell(cellnum++);

        if (obj instanceof String)
            cell.setCellValue((String)obj);

        else if (obj instanceof Double)
            cell.setCellValue((Double)obj);

        else if (obj instanceof Integer)
            cell.setCellValue((Integer)obj);
    }
}

// Try block to check for exceptions
try {

    // Writing the workbook
    FileOutputStream out = new FileOutputStream(
        new File("salida.xlsx"));
    salida.write(out);

    // Closing file output connections
    out.close();

    // Console message for successful execution of
    // program
    System.out.println(
        "salida written successfully on disk.");
}

// Catch block to handle exceptions
catch (Exception e) {

    // Display exceptions along with line number
    // using printStackTrace() method
    e.printStackTrace();
}
}
```
