TRABAJO FIN DE GRADO

INGENIERÍA EN INFORMÁTICA

Diseño y desarrollo de un sistema multidimensional basado en información geográfica de Andalucía

Autor

Alonso Bueno Herrero (alumno)

Directores

José Samos Jiménez (tutor)

Escuela Técnica Superior de Ingenierías Informática y de Telecomunicación

—

Granada, Julio de 2021

Diseño y desarrollo de un sistema multidimensional basado en información geográfica de Andalucía

Autor
Alonso Bueno Herrero (alumno)

Directores
José Samos Jiménez (tutor)

Diseño y desarrollo de un sistema multidimensional basado en información geográfica de Andalucía

Alonso Bueno Herrero (alumno)

Palabras clave: datos, COVID-19, Andalucía, OLTP, OLAP, Sistema Multidimensional, hechos, dimensiones

Resumen

Este proyecto presenta un sistema multidimensional que se basa en los datos evolutivos desde el inicio «informal» de este proyecto hasta la fecha de su finalización sobre el COVID-19 en Andalucía. El objetivo, aportar, sobre todo a las Administraciones Públicas de nuestra comunidad, una herramienta útil para poder analizar de la mejor forma posible los datos de la pandemia.

Se utilizan diversas tecnologías específicas para este tipo de sistemas, y se incluye, a modo de prueba, un cuadro de mando que se ha elaborado para, básicamente, ilustrar el funcionamiento y uso del sistema.

En Granada, a 17 de Febrero de 2.021.

Alonso Bueno Herrero.

Design and development of a multidimensional system based on geographical information of Andalusia

Alonso, Bueno Herrero

Keywords: data, COVID-19, Andalusía, OLTP, OLAP, multidimensional system, facts, dimensions

Abstract

This project presents a multidimensional system that is based on the «evolutionary data» from the beginning of this project to the date of its completion about the COVID-19 pandemic in Andalusia. The objective is to offer, especially to the Public Administrations of our community, a useful tool in order to analyse the data of the pandemic in the best way.

Some specific technologies for this type of systems are used, and it is also included, as a test, a scorecard which has been made in order to show the operation and use of the system.

Granada, 5th July 2021.

Alonso Bueno Herrero

Yo, **Alonso Bueno Herrero**, alumno de la titulación INGENIERÍA INFORMÁTICA de la **Escuela Técnica Superior de Ingenierías Informática y de Telecomunicación de la Universidad de Granada**, con DNI 76067525Q, autorizo la ubicación de la siguiente copia de mi Trabajo Fin de Grado en la biblioteca del centro para que pueda ser consultada por las personas que lo deseen.

Fdo: Alonso Bueno Herrero

Granada a 6 de Julio de 2021.

D. **José Samos Jiménez (tutor)**, Profesor del Área de Lenguajes y Sistemas Informáticos del Departamento de Lenguajes y Sistemas Informáticos de la Universidad de Granada.

Informa:

Que el presente trabajo, titulado ***Diseño y desarrollo de un sistema multidimensional basado en información geográfica de Andalucía***, ha sido realizado bajo su supervisión por **Alonso Bueno Herrero (alumno)**, y autorizo la defensa de dicho trabajo ante el tribunal que corresponda.

Y para que conste, expiden y firman el presente informe en Granada a 8 de Julio de 2021.

Los directores:

José Samos Jiménez (tutor)

Índice general

Índice de figuras

Índice de cuadros

Capítulo 1

Introducción

1.1. Motivación

La situación que nos ha tocado vivir a raíz de la pandemia de COVID-19 ha provocado que las Administraciones Públicas, ya sean locales, provinciales, autonómicas o el propio Gobierno Central tengan que tomar decisiones muy difíciles, intentando siempre buscar el equilibrio entre las partes afectadas por dichas decisiones (ese malicioso binomio economía *versus* salud). La Informática, como cabría esperar, no puede estar ajena a esta problemática, y debe estar ahí para intentar aportar su granito de arena.

Por otro lado, en cualquier ente, ya sea público (como en este caso) o privado, resulta fundamental poder disponer de herramientas y técnicas adecuadas que permitan, a partir de los datos de que disponen, tomar esas decisiones de la mejor forma posible. Aquí entran los sistemas multidimensionales. Gracias a estos sistemas, se pueden observar las situaciones (los datos, dicho de una forma técnica) desde distintos puntos de vista, pudiendo centrarse en una sola perspectiva, o una combinación de varias, y también con más detalle o menos. Los fundamentos de estos sistemas se explican más adelante.

Con estos dos puntos de partida, presento mi propuesta: un sistema multidimensional con datos actualizados de COVID-19 en Andalucía, con el que aportar mis conocimientos aprendidos en el Grado y, en concreto, en la asignatura «Sistemas Mutidimensionales» (en adelante, **SMD**), a unas hipotéticas administraciones andaluzas que solicitaran este producto de cara al **análisis de los datos** de la pandemia en nuestra comunidad, con especial atención, tal y como advierte el título del proyecto, a la información geográfica que subyace de estos datos, aprovechando también los conocimientos que adquirí en la asignatura «Sistemas de Información Geográfica» (en adelante, **SIG**). Esta *explotación* de la información geográfica también

pretende aportar un valor añadido a nuestro sistema, debido a ese carácter eminentemente geográfico, territorial, que tienen las medidas de contención contra el virus que se están adoptando (cierres perimetrales de municipios, por ejemplo).

1.2. Objetivos del proyecto

La finalidad, a nivel global, de este proyecto se describe en el recuadro 1.1.

Recuadro 1.1 *Objetivo general del proyecto*

Como objetivo general, se pretende proporcionar una herramienta informática para la toma de decisiones en el ámbito de las administraciones andaluzas sobre la situación sanitaria, con especial énfasis en el componente geográfico de los datos.

Los objetivos específicos que se plantean son:

1. Elaborar, de forma integral, un sistema multidimensional aplicando una metodología de desarrollo específica planteada en la literatura y adaptada pertinentemente a nuestro proyecto (en lo que a dimensiones y complejidad se refiere).

2. Experimentar con la integración de la información geográfica con los sistemas multidimensionales[1], intentando explotar al máximo las capacidades que la semántica de estos sistemas puede ofrecer a la interpretación de la información geográfica.

1.3. ¿Cómo se cita la bibliografía?

He optado por seguir el estándar para citar bibliografía en ingeniería que recomienda la Universidad Carlos III de Madrid en `https://uc3m.libguides.com/guias_tematicas/citas_bibliograficas/inicio`: la norma **IEEE v01.29.2021**. Para ello, hago uso de su web didáctica con guía y ejemplos de este estándar, accesible en: `https://uc3m.libguides.com/guias_tematicas/citas_bibliograficas/IEEE`.

[1] Como si de una fusión de las asignaturas se tratase.

1.4. ¿Cómo se estructura esta memoria?

El contenido de este documento se organiza de la siguiente forma:

1. Una introducción, donde se presentan los objetivos del proyecto y se indica la estrategia para citar la bibliografía.
2. Un capítulo dedicado a poner en contexto el proyecto, esto es, a definir y exponer las líneas maestras del área de conocimiento sobre el que versará (Sistemas Multidimensionales) y las tecnologías que podemos aplicar para resolverlo.
3. En el siguiente capítulo, el más extenso, se explica al completo cómo se ha desarrollado el proyecto:
   a) Planificación temporal.
   b) Presupuesto del proyecto.
   c) Repositorio de GitHub para el proyecto.
   d) Explicación del ciclo de vida escogido para guiar el desarrollo del proyecto, así como las diversas metodologías específicas, heurísticas, etc aplicadas en las etapas donde proceda.
   e) Justificación de las tecnologías específicas escogidas, de las descritas previamente.
   f) Etapas del desarrollo: secuencia de apartados donde se explica todo el proceso de desarrollo del proyecto.
   g) Descripción de una pequeña «demo» realizada para comprobar la funcionalidad del sistema implementado.
   h) Visión general del sistema desarrollado, enfocado en las herramientas usadas en cada fase y el flujo de datos.
4. Un capítulo final con las conclusiones extraídas tras este proyecto y una propuesta de ampliación futura del mismo.

Capítulo 2

Contexto del proyecto

El objetivo de este capítulo es poner en contexto este proyecto. Se enmarca dentro de lo que podríamos definir como un «proyecto de datos», pues, a partir de una idea o una posible necesidad que se nos ocurre, buscamos datos para construir una aplicación o sistema que satisfaga esa necesidad.

En primer lugar, se presenta una introducción a los Sistemas Multidimensionales y a algunos conceptos básicos sobre Información Geográfica, en especial, en el territorio de Andalucía.

En segundo lugar, se muestra una breve perspectiva histórica y actual de las herramientas más utilizadas para el desarrollo de Sistemas Multidimensionales. En el Capítulo 3, se indicará qué herramientas se han seleccionado para el desarrollo del proyecto.

2.1. Fundamentos de los Sistemas Multidimensionales

Cabe destacar que el concepto de «Sistema Multidimensional» no es el más extendido para definir este tipo de sistemas. Otros nombres muy relacionados son el de «sistema de almacenamiento de datos» o su acepción anglosajona, «Data Warehouse System», que se deriva del concepto de *Data Warehousing*, que [12] define así:

> *Data Warehousing* es una colección de métodos, técnicas y herramientas usadas para apoyar la toma de decisiones.

La definición anterior queda completada con la siguiente enumeración de necesidades de usuarios finales y organizaciones que dieron lugar a la aparición de estos sistemas:

1. ¡Tenemos montañas y montañas de datos, pero no sabemos cómo acceder a ellos!

2. ¡Necesitamos seleccionar, agrupar y manipular datos de todas las formas posibles! Los procesos de negocio no siempre pueden ser diseñados con antelación, hay veces que es necesario tomar decisiones *ad hoc* ante una decisión en tiempo real. Los usuarios necesitan una herramienta que sea amigable (*user-friendly*) y que les permita obtener de la forma más rápida posible las relaciones que buscan para poder analizarlas.

   Ejemplo 2.1 *Las cifras de fallecimientos en residencias, en cada Distrito Sanitario de la provincia de Sevilla.*

3. ¡Muéstrame solo lo que me importa! O lo que es lo mismo: los usuarios no quieren la información al más mínimo nivel de detalle, solo quieren aquello que les hace falta, el resto puede molestarles o incluso confundirles en su análisis. En el Ejemplo 2.1, los decisores no quieren la información de los casos en residencias a nivel de municipio o de barrio, que podría ser el nivel de detalle de los datos más bajo posible, sino que lo quieren solo por Distritos Sanitarios, que está por encima y es lo que necesitan.

En definitiva, el listado anterior nos sirve para poder elaborar una serie de mínimos o requisitos *a priori* que todo buen sistema de *data warehousing* ha de tener:

1. Accesibilidad para los usuarios.

2. Integración de los datos en base a un modelo de negocio.

3. Flexibilidad para la realización de consultas.

4. Información concisa, presentando solo lo necesario (ni más ni menos) para el análisis.

5. Representación **multidimensional**, aportando todos los puntos de vista posibles sobre los datos a los decisores.

6. Corrección y completitud de los datos integrados (usados).

Aunque no es el único, el principal elemento de este tipo de sistemas es la base de datos multidimensional.

Una **base de datos multidimensional** es una colección de datos que apoya la toma de decisiones y que se caracteriza por:

- estar estructurada en forma de *hechos* y *dimensiones*,
- tener un enfoque subjetivo,
- estar integrada y ser consistente,
- ser capaz de mostrar su evolución a lo largo del tiempo, lo cual implica que **no es volátil**.

En primer lugar, tiene un enfoque subjetivo porque está orientada a la tarea o al módulo específico del cliente que usará ese sistema.

En segundo lugar, un sistema multidimensional utiliza datos de fuentes que pueden ser muy diversas y con formatos distintos, valores nulos, etc. Su obligación es trabajar todos esos datos para conseguir integrarlos, quedándose con la parte de los mismos que interesa según el objetivo empresarial y lograr una consistencia y limpieza plena a nivel global de los datos restantes.

Y, finalmente, la necesidad de que los datos estén siendo frecuentemente actualizados, y de que nunca se borren datos históricos (es decir, menos recientes).

2.1.1. Sistemas OLTP y OLAP

Una vez familiarizados con el concepto de *sistema multidimensional*, conviene saber distinguir entre dos conceptos fundamentales en este:

Sistemas transaccionales (OLTP): encargados del registro de transacciones mediante **almacenamiento** de los datos. Constituyen la «fuente» de información sobre la que trabajar.

Sistemas decisionales (OLAP): orientados a los decisores de las organizaciones, y se sirven de los datos de los sistemas transaccionales para dar soporte a la **toma de decisiones** en la organización.

En la tabla 2.1 se presenta una comparativa básica de estos dos tipos de sistemas.

Pero, ¿cómo se consigue pasar de los datos resultado del registro de transacciones a información útil para los decisores? La clave está en los informes. Un informe es el principal soporte para la toma de decisiones en una organización. Se suelen generar de dos formas:

	OLTP	OLAP
Usuarios	Operadores	Decisores
Función	Operaciones cotidianas	Soporte a la toma de decisiones
Diseño	Orientado a las aplicaciones	Orientado al usuario
Datos	Actuales, actualizados y detallados	Históricos, consolidados y resumidos
Uso	Repetitivo	Ad-hoc
Acceso	Consultas simples y actualizaciones.	Consultas complejas

Cuadro 2.1: Comparación de sistemas OLTP y OLAP. Adaptado de [4].

Informes precocinados: son elaborados como parte del sistema transaccional.

Informes «a medida»: son elaborados e integrados en el sistema a partir de las peticiones de información de un decisor. Las etapas típicas para su desarrollo son:

1. El decisor necesita un nuevo informe, y piensa sobre qué datos versará (se supone que no tiene conocimientos de informática, luego no va a ser una descripción técnica). Se lo comunica al Dpto. de Informática (o equivalente) de la empresa para que obtengan ese informe.
2. El Departamento tendrá que hacer una tarea de **búsqueda** de los datos disponibles en la empresa implicados en esta consulta, les aplica las **transformaciones** (integración, limpieza, etc.) y **genera** ese nuevo informe.
3. Antes de facilitar el informe al decisor, suele haber un equipo con conocimientos sobre el sector de negocio de la empresa que supervisa que el informe realmente es correcto, y se corresponde con lo que se pedía.

Pero hay que decir que este modelo de gestión de la información (basado en la elaboración de informes) no es adecuado, debido esencialmente al tiempo que tiene que esperar el decisor a que le llegue el nuevo informe, pues las tareas que el Departamento de Informática tiene que realizar para generarlo no son en absoluto triviales ni automáticas, en general.

Así las cosas, en la actualidad, y debido a esa necesidad evidente de tener **información en tiempo real** de lo que sucede para intentar *maximizar el beneficio* obtenido con esas medidas tomadas (pensemos en las reuniones de los comités de seguimiento del coronavirus de los gobiernos, por ejemplo), esta filosofía de elaborar informes a medida, no es en absoluto recomendable. ¿Qué hacer, entonces?

Esta enorme necesidad viene a estar hoy en día cubierta en gran medida por los **sistemas multidimensionales**.

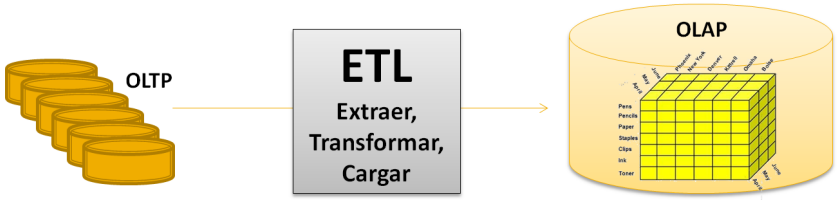


Figura 2.1: Arquitectura «intuitiva» simple para un sistema multidimensional. Fuente: elaboración propia.

2.1.2. La «cocina» de los datos: el componente ETL

El componente ETL sirve de nexo de comunicación entre los datos «fuente» almacenados en las bases de datos OLTP y la información preparada para la toma de decisiones de las bases de datos multidimensionales (componente OLAP).

> **ETL** (Extraer, Transformar, Cargar) es un conjunto de técnicas y procedimientos responsable de que la información esté lista en la base de datos multidimensional para la toma de decisiones.

En la figura 2.1 se muestra la evolución desde los datos originales (OLTP) hasta el sistema OLAP, a partir de lo estudiado en [16].

Las fases del proceso ETL son Extracción, Transformación y Carga. Estas fases deben completarse en este orden ya que, a grandes rasgos, los resultados de una son la entrada de la siguiente, constituyendo así un cauce (*pipeline*) de tareas secuenciales. Veamos con algo más de detalle en qué consiste cada una y qué aporta al desarrollo de nuestro sistema:

Extracción Su función esencial consiste en extraer la información que ha sido actualizada. Existen dos grandes clasificaciones de los métodos posibles para esta tarea:

- ***Diferidos***: se obtienen los cambios respecto a la última vez que se actualizaron. El principal problema que tienen estos métodos es que si ha habido algún cambio entre las dos «capturas» que estamos comparando, no lo hemos podido detectar y se ha perdido.
- ***Inmediatos***: en este caso se captura cada modificación que tiene lugar en la fuente de datos, con lo que ninguna se queda indocumentada y no hay pérdida de información.

Dentro de cada uno de los métodos de extracción también hay subtipos. La principal diferencia entre las dos clasificaciones (diferidos e

inmediatos) es que en los segundos no hay pérdida de información, frente a los primeros, que sí la hay. Aunque hablaremos más adelante de la estrategia seleccionada, se adelanta ya se trata de un método de extracción inmediato usando aplicaciones, pues el software (tanto scripts propios como herramientas de apoyo) nos ayudará en la tarea de obtener todos los datos cada vez que éstos son modificados[1].

Transformación La transformación en el proceso ETL hace referencia a varias tareas:

1. fusionar datos de diversas fuentes,
2. adaptar datos al modelo de datos destino (el modelo multidimensional, en nuestro caso –lo explicaremos más adelante), y
3. corregir problemas con posibles datos erróneos derivados de las fuentes o de la fusión realizada anteriormente, o ambas.

Carga Una vez transformados los datos convenientemente, se cargan, de cara a poder presentarlos a los usuarios (decisores). Existen dos principales filosofías para esta fase:

- Carga inicial: todos los datos de las fuentes se cargan para obtener una primera versión completa del sistema multidimensional.
- Actualizaciones periódicas: se actualizan los datos de la parte OLAP en base al criterio que se considere.

A modo de resumen, se muestra en la figura 2.2 todo el proceso de forma esquemática, poniendo especial énfasis en qué se hace o que se obtiene en cada paso.

2.1.3. El modelo de datos multidimensional

El modelo multidimensional es la clave de los sistemas multidimensionales.

Ejemplo 2.2 *Suponemos un informe real sobre las ventas en una determinada empresa de suministros industriales. ¿Qué tipo de preguntas podemos hacernos sobre la información contenida en dicho informe...?*

- *¿Dónde ... se han realizado las ventas?* $\rightarrow$ *En España, en Italia, en Reino Unido, etc...*

[1] En nuestro caso, tendremos fuentes de datos que se modifican que se modifican a diario, y otras donde no se borra ningún dato, sino que la propia fuente de datos (el *dataset*) se amplía, añadiendo el informe COVID que corresponda a ese día/semana/... en nuevas filas.

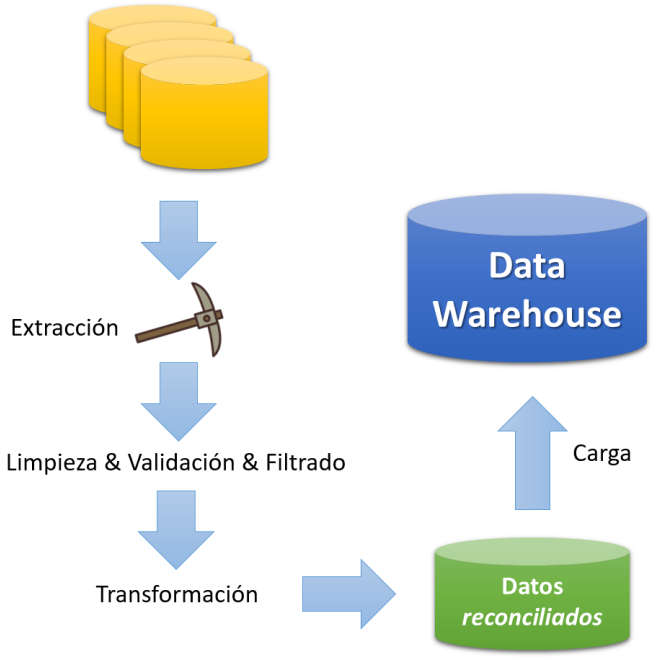


Figura 2.2: Secuencia de tareas del componente ETL. Adaptado y traducido de [12].

- *¿Qué... se ha vendido?* → *Cableado eléctrico, Repuesto pieza de maquinaria, Lote de engranajes, etc...*
- *¿Cuándo...?* → *En Febrero, en Mayo, la semana pasada, etc...*

De esta filosofía parte el modelo de datos **multidimensional**, basado en:

- varias ***dimensiones*** o formas de estudiar dicho informe (hacerle preguntas).
- Un tema central de análisis, los **hechos** (*facts*, en inglés), también llamados *focos de atención*.

Pues bien, estos dos ingredientes son los que definen el modelo de datos multidimensional, es decir, los componenes de una **base de datos multidimensional**. Los sistemas OLAP tienen a esta como principal componente.

Tal y como destaca [12], las bases de datos multidimensionales han generado un gran interés a nivel de investigación y en el ambiente del comercio debido a su preponderancia en multitud de aplicaciones de soporte a la decisión (DSS, *Decision Support Systems)*. La razón principal por la que el modelo multidimensional es usado como un verdadero paradigma en la representación de datos en los *data warehouses* es debido a su relación con su facilidad y sencillez, sobre todo para usuarios poco familiarizados con la informática. Además, esta importancia también puede deberse al intensísimo uso de las hojas de cálculo, tan presentes en la actualidad, en prácticamente todos los ámbitos (empresarial o de cualquier organización), los cuales adoptan el modelo multidimensional como un paradigma de organización de datos.

Aunque más adelante hablaremos de las tecnologías más relevantes para el desarrollo de sistemas multidimensionales y, al hilo de lo que acabamos de comentar respecto a las hojas de cálculo, cabe referir que ha sido Microsoft Excel uno de los primeros software ofimáticos en incluir complementos para el desarrollo del procedimiento ETL (complemento «Microsoft Power Query») y desarrollar el cubo OLAP (complemento «Cubos OLAP» y después «Microsoft Power Pivot»).

2.1.3.1. Representación gráfica de una base de datos multidimensional

La forma más común de representar una base de datos multidimensional es como una figura geométrica sobre unos ejes (como los ejes cartesianos),

pero con n ejes, tantos como dimensiones se hayan diseñado alrededor de ese «foco de atención» (hechos).

Suponiendo que tenemos $n = 3$ dimensiones, la figura se denomina ***cubo OLAP***. En general, para n dimensiones se le denomina **hipercubo**, aunque, debido a la preponderancia del término «cubo» para referirse a cualquiera de ellos, tenga las dimensiones que tenga, será este el término que se usará en esta memoria.

En la figura 2.3 se muestra un cubo OLAP representativo de la situación de la empresa de suministros del Ejemplo 2.2, donde tenemos tres dimensiones:

- *Productos*: responde a la pregunta ¿***Qué***?
- *Meses del año*: responde a la pregunta ¿***Cuándo***?
- *Países*: responde a la pregunta ¿***Dónde***?

y un foco de atención (*hechos*) bastante evidente: las ***ventas***[2]. En este ejemplo, tenemos que cada una de las ***celdas*** que componen el cubo (ver figura 2.3) son una venta realizada, es decir, una *instancia* de los *hechos*.

Un ejemplo de una venta válida se ilustra en la figura 2.4, donde vemos que **se ha vendido el producto «Producto 1» en el mes de «Junio» y en el país «España»**.

2.1.3.2. Operaciones sobre cubos OLAP

Al igual que sobre una base de datos relacional podemos hacer operaciones (agrupaciones, selección de filas, de columnas, seleccionar una tabla completa, etc.), el modelo de datos multidimensional también permite hacer algunas operaciones sobre la base de datos. Estas operaciones son:

1. ***Roll-up***: ascender en el nivel de la jerarquía de una determinada dimensión. Se trata de agrupar datos de los hechos siguiendo el patrón marcado por una jerarquía de la dimensión.
2. ***Drill-down***: descender en el nivel de la jerarquía de la dimensión. Se trata de desagrupar datos.

   Esta operación no se puede realizar directamente si no tenemos los datos originales (antes de ser agrupados mediante *Roll-up*. Es decir, no se puede. Un

[2]Aunque la cuestión *comercial* no nos interesa para nuestra temática, pues nosotros nos centramos en la toma de decisiones en al ámbito de la pandemia de COVID-19 en Andalucía, es cierto que este ejemplo de las *ventas* por país, fecha y tipo de producto vendido es el que primero suele presentarse al introducir los sistemas multidimensionales.

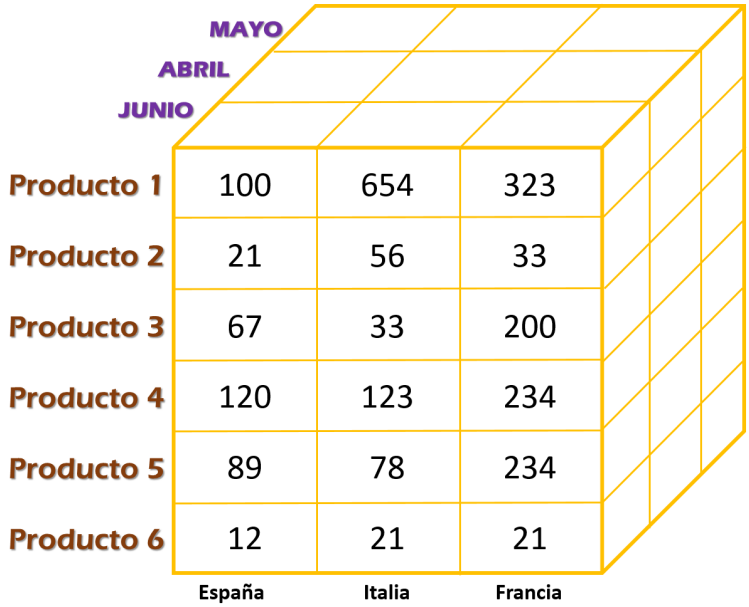


Figura 2.3: Cubo OLAP sobre las ventas de una empresa. Fuente: elaboración propia.

Figura 2.4: Una venta realizada, que se corresponde con un «cubo pequeño» de los que componen el cubo OLAP. Fuente: elaboración propia.

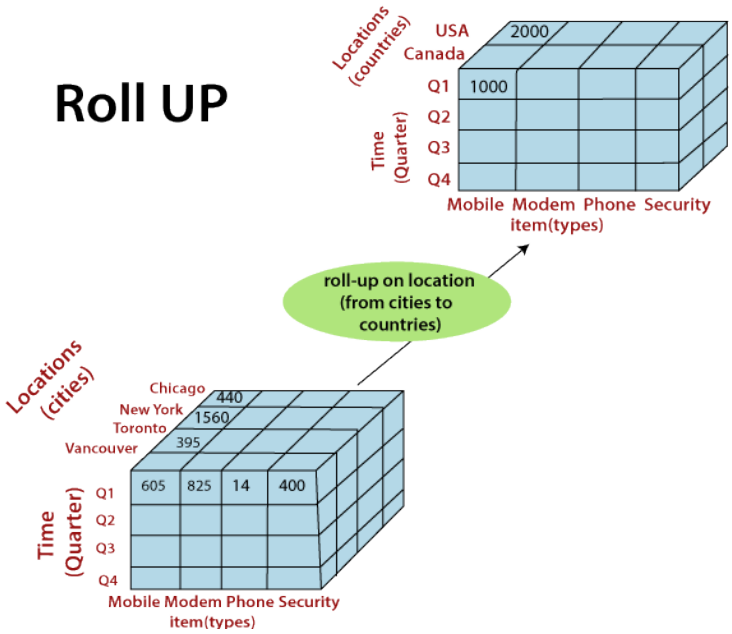


Figura 2.5: Ilustración de la operación *Roll-up*. Fuente: [19]

drill-down es una operación a nivel conceptual. Para implementarla *roll-up* para obtener los datos al nivel de detalle de la operación *drill-down* solicitada.

3. ***Slice&dice***: seleccionar los datos del cubo por una (*slice*) o por varias (*dice*) dimensiones.

Para ejemplificar estas operaciones, en las figuras 2.5, 2.6 y 2.7 se muestran, en el mismo orden en que se han explicado, un ejemplo de aplicación sobre un cubo similar al del Ejemplo 2.2.

2.1.4. Fundamentos del diseño multidimensional

Las fases diseño de una Base de Datos multidimensional son similares a las del diseño de una Base de Datos: diseño conceptual, lógico y físico. La diferencia fundamental radica en los modelos de datos que se usan.

La figura 2.8 muestra notables diferencias entre las etapas que se dan en este diseño en los dos tipos de sistemas: en el caso de sistemas OLAP, se indica explícitamente que el Modelo Conceptual es Multidimensional, de ahí se pasa a un Modelo Lógico OLAP de hasta tres tipos diferentes (M-OLAP, 03-OLAP y R-OLAP), y finalmente una fase de diseño físico con la elección del SGBD.

Observación 1 *En nuestro caso, no haremos explícitamente dos diseños,*

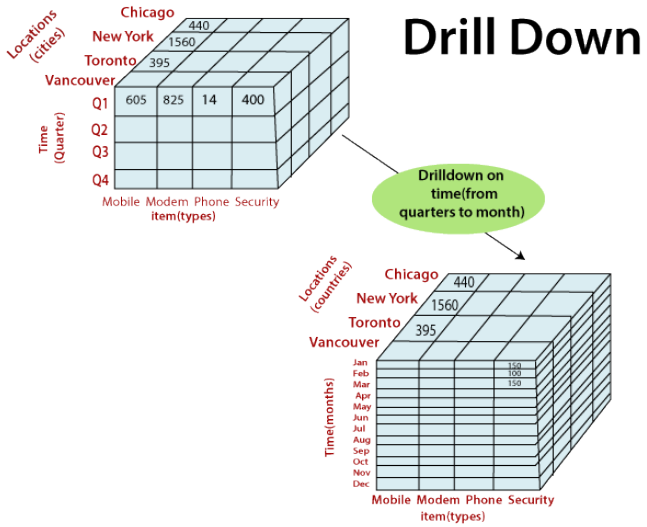
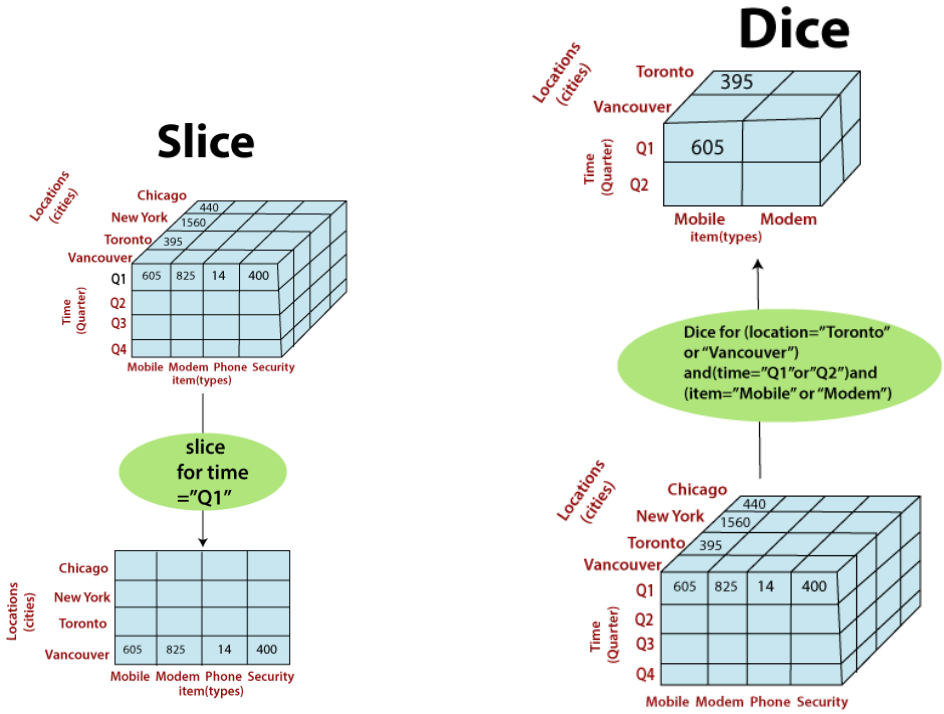


Figura 2.6: Ejemplo de la operación *Drill-down*. Fuente: [19]



(a) Operación *Slice* o *corte por una sola dimensión.*

(b) Operación *Dice* o *corte por varias dimensiones.*

Figura 2.7: Ejemplo de la operación *Roll-up*. Fuente: [19]

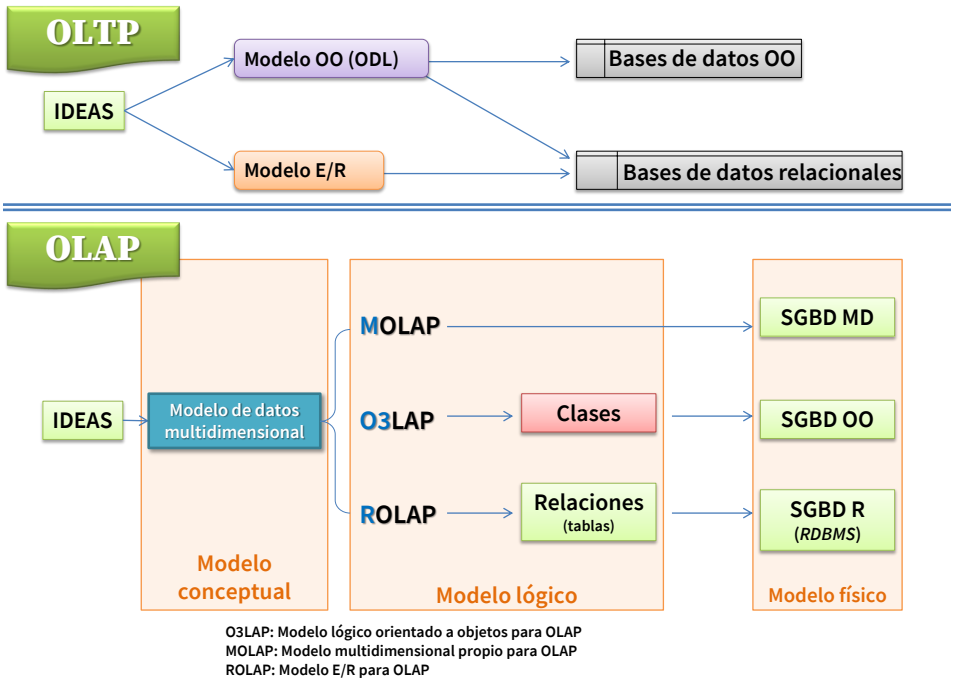


Figura 2.8: Perspectiva general del diseño de las «bases de datos» en sistemas OLTP y OLAP. Fuente: elaboración propia a partir de [16].

uno para el componente OLTP y otro para el OLAP. Para el ciclo de vida seguido para el desarrollo del sistema multidimensional, adaptado a este proyecto, no es necesario tener un esquema explícito que represente a las fuentes de datos (sistema OLTP), sino que directamente entre los datos fuentes transformados y los requisitos de usuario, se afronta el diseño conceptual del sistema OLAP.

Explicamos ya las fases de diseño de los sistemas multidimensionales.

2.1.4.1. Diseño conceptual

En el modelo multidimensional a nivel conceptual, se representan los hechos y las dimensiones, sin entrar en detalles de implementación.

Dentro de cada dimensión hay un conjunto de **niveles**, formando **jerarquías**. A cada valor concreto para cada nivel se le denomina **instancia**. Por ejemplo, para la dimensión *Qué* de una base de datos que gestione ventas de libros, el valor «Crónica de una muerte anunciada» sería una instancia del nivel «Libros».

Otra cosa relevante sobre las dimensiones es la ***cardinalidad*** **de las**

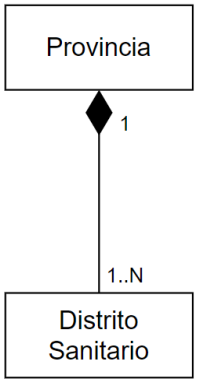


Figura 2.9: Ejemplo de composición UML en nuestro dominio. Fuente: elaboración propia.

relaciones entre los distintos niveles. Se establece que:

Recuadro 2.1 *Cardinalidad entre niveles*

1. Cada instancia de un nivel J se relaciona con solo una instancia del nivel superior $J+1$.
2. Cada instancia del nivel $J+1$ puede relacionarse con N $(N \geq 1)$ instancias del nivel inferior, J.

En definitiva, hablamos de una relación padre (nivel J) e hijo (nivel $J-1$) del tipo «***parte-todo exclusiva***» (una *composición* en UML). La figura 2.9 muestra un ejemplo de esto en nuestro proyecto: los Distritos Sanitarios de Andalucía pertenecen a una provincia, y solo a una, tal que, a la hora de ubicarlas dentro de una jerarquía (o en general, en una dimensión), tendrá que respetarse la composición entre ellas para simbolizar esta relación que comentamos.

Todo esto, además, implica que el nivel superior tiene cardinalidad 1, mientras que los hijos (nivel inferior) tienen cardinalidad $1..N$.

Además, siempre tienen que estar todas las instancias cumpliendo esta relación. Esto conlleva que no puede pasar que una instancia del nivel hijo no tenga ninguna instancia padre asociada, pues esa instancia hijo quedaría *suelta* o *descolgada* dentro de la base de datos. De la misma forma, tampoco debería suceder que haya un libro que sea de dos padres, pues a la hora de hacer un *Roll-up* el sistema no sabría «a qué padre asignarle ese hijo».

Para acabar con el diseño conceptual, es conveniente comprender una serie de propiedades de algunos elementos que hemos referido, y que irán

apareciendo a lo largo del proyecto:

Granularidad es el nivel de detalle para cada dimensión (*Producto* en una dimensión *Qué*, *Tienda* en una dimensión *Dónde* o *Día* en una dimensión *Cuándo*, por ejemplo).

Bases son los niveles de cada dimensión que necesitamos para **identificar unívocamente** cada uno de los hechos (cada venta realizada, por ejemplo), es decir, como si de una llave primaria se tratase[3].

Aditividad puede entenderse como la *viabilidad* de **sumar** los valores de **sumar una determinada medición**. La pregunta que hay que hacerse es si tiene sentido la suma de los valores de esa medición sobre una determinada dimensión.

Ejemplo 2.3 (Analogía sobre la aditividad) *Preguntémonos si tiene sentido sumar el dinero que llevábamos en la cartera cada día de la semana (no es el dinero que hemos gastado, sino el que teníamos en la cartera en el mismo instante concreto el lunes, el martes, etc.) No tiene sentido sumarlo.*

Además, cuando una medición no tiene sentido sumarla por todas las dimensiones del cubo, sino solo por algunas se dice que es una medición *semiaditiva*, frente a las que son puramente *aditivas*, que, evidentemente, lo son por todas las dimensiones.

Dentro del diseño conceptual hay otros conceptos como la clasificación de dimensiones, de hechos, de jerarquías y demás. Para simplificar, si es posible, se recomienda tener jerarquía estrictas y cubrientes (como se explica en el recuadro 2.1.4.1).

[3]De hecho, en términos de tablas relacionales y de usar un modelo lógico ROLAP-estrella (más adelante se explica), la clave primaria de la tabla de hechos (el cubo OLAP) no será otra cosa que un conjunto de llaves externas procedentes de las dimensiones que constituyen las bases de esos hechos, con lo cual todo «cobra sentido».

Recuadro 2.2 *¿Cómo serán nuestras jerarquías?*

Para evitar problemas, en nuestro diseño multidimensional tendremos siempre jerarquías **estrictas** y **cubrientes**:

- *estricta* porque a cada valor de un nivel J le corresponde un único valor1 del nivel superior $J+1$;
- *cubriente*, porque intentaremos que todos los niveles se puedan ajustar de igual forma a la jerarquía que los demás[a].

[a]Un ejemplo de jerarquía no cubriente sería, de entrada, el mapa político de España, organizado en municipios, agrupados en provincias, agrupadas en Comunidades Autónomas; pero ¿y Ceuta y Melilla? No son comunidades autónomas, sino ciudades autónomas, sería necesario que el diseñador lograse una fusión semántica entre ambos conceptos (CCAA y Ciudad Autónoma).

2.1.4.2. Diseño lógico

En esta fase vamos a hacer una descripción explícita de cómo se va a implementar la base de datos multidimensional que hemos definido previamente en el modelo conceptual.

Existen diversos enfoques para diseñar el nivel lógico de una BDMD, destacando:

MOLAP se usa un **SGBD** Multidimensional, conas las operaciones multidimensionales incluidas.

ROLAP representar nuestra Base de datos multidimensional (cubo OLAP) a través de una **Base de Datos Relacional**, y definir a través de una herramienta externa las operaciones necesarias.

OO-DBMS usar **bases de datos orientadas a objetos**. Era una alternativa muy popular en los 90, pero actualmente está en desuso.

HOLAP se refiere a un **modelo** de bases de datos **híbrido**, en tanto que se usarán simultáneamente un modelo MOLAP y ROLAP.

Nosotros nos centraremos en un modelo lógico basado en **ROLAP** internamente, aunque la herramienta seleccionada usará también MOLAP por lo que será HOLAP.

En el caso ROLAP se consideran dos alternativas:

Enfoque ROLAP basado en *estrella* en este caso se establece la siguiente equivalencia:

- por cada dimensión, tendremos una tabla relacional;
- para los hechos, habrá una única tabla que guardará las **llaves externas de las dimensiones** y las **mediciones**.

Enfoque ROLAP basado en *copo de nieve* en este caso la equivalencia es: a) por cada nivel de cada dimensión, una tabla relacional; b) de nuevo, para los hechos una única tabla (igual que el modelo en estrella).

En el modelo ROLAP en estrella se repiten datos en las dimensiones, pero es más sencillo y adecuado para realizar consultas. Es la alternativa de uso más frecuente, también la que usaremos en el proyecto.

La necesidad de simplificar las tablas: concepto de «Llave Generada»

El concepto de llava generada surge ante la necesidad de usar identificadores de cada tupla que fueran únicos pero que ocuparan el **menor espacio posible**.

Por ejemplo, un DNI es una buena opción para una llave primaria, porque en ningún caso hay dos DNIs que sean iguales. Pero necesitamos 8 (o 9) Bytes para cada uno.

En su lugar, podemos numerar secuencialmente las filas y usar ese número como una llave generada. Las filas quedan perfectamente distinguidas unas de otras dentro de la tabla y hemos ahorrado espacio, especialmente cuando se usa como llave externa en los hechos.

En definitiva, nuestro modelo lógico vendrá presidido por esta sencilla estrategia:

Recuadro 2.3 *Estrategia para el diseño lógico*

Para el diseño lógico aplicaremos la heurística basada en el modelo **ROLAP en estrella**, esto es, definir cada dimensión y los hechos en una tabla relacional independiente cada uno, haciendo uso de llaves generadas como claves primarias en las dimensiones y externas en los hechos.

2.1.4.3. Diseño físico

La idea básica en esta etapa final radica en adaptar el modelo lógico al SGBD donde se implementa..

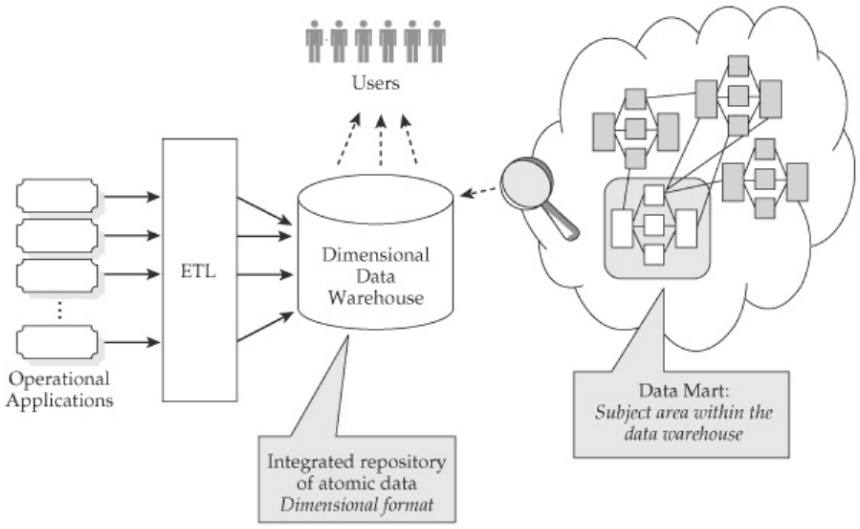


Figura 2.10: Arquitectura en bus de Kimball. Fuente: [11].

2.1.5. Arquitectura de sistemas multidimensionales

Las propuestas arquitectónicas para los sistemas multidimensionales son muy variadas, pero entre todas ellas podemos apreciar un ingrediente común: el enfoque al usuario. Y tiene sentido que esto sea así, pues en definitiva se trata de proporcionarle la información que necesita de la forma más flexible posible y facilitarle la labor de análisis con la herramienta que use.

Quizás el enfoque arquitectónico que mejor represente esta filosofía de orientación al usuario sea la conocida por «Arquitectura de Kimball» o «Arquitectura en bus», que el autor expone en [13] y que aquí se muestra en la figura 2.10.

Lo más significativo de esta arquitectura, es la separación entre tareas de servidor y tareas de cliente, esto es, se queda perfectamente delimitado qué partes corresponden al desarrollo del sistema y qué parte a las aplicaciones de los usuarios finales que solo hacen consultas.

Otro enfoque arquitectónico para los sistemas multidimensionales fue la arquitectura de Inmon, la conocida como «Fábrica corporativa de Información», que guarda unas cuantas similitudes con la arquitectura de Kimball, pero que difiere con respecto a esta en dos aspectos clave:

- En la arquitectura de Inmon la base de datos multidimensional sigue un modelo de datos relacional (ER) al menor nivel de detalle posible,

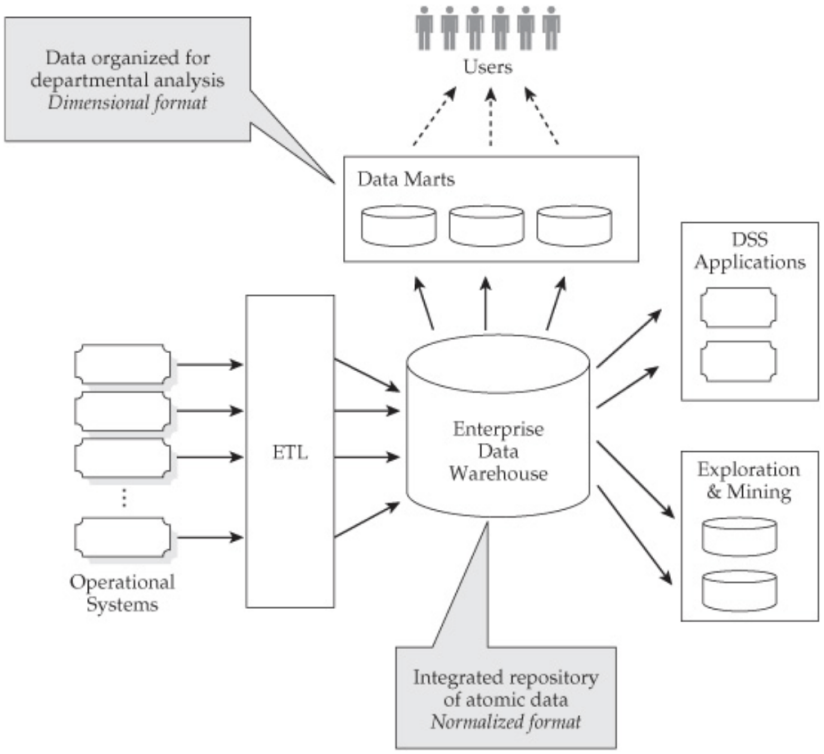


Figura 2.11: Fábrica Corporativa de Información, arquitectura de Inmon. Fuente: [11].

frente a la arquitectura de Kimball, que la diseña según los preceptos que hemos estado explicando hasta ahora, es decir, a través de una serie de cubos OLAP que almacenan la información, aunque también se incide en que sea con el menor nivel de detalle posible.

- En la arquitectura de Kimball, el cliente tiene acceso directo a la base de datos multidimensional (para consultarla, evidentemente), mientras que en la arquitectura de Inmon la misma base de datos sirve para generar pequeños cubos OLAP (llamados *data marts* en inglés) especializados para las áreas de interés que quieran usarla.

En la figura 2.11 se muestra el esquema de la arquitectura de Inmon.

En dicha figura podemos ver cómo, en efecto, la «idea general» se mantiene (zona de datos fuente/operacionales, zona ETL y la base de datos -*data warehouse*-, en esencia).

2.2. Información geográfica en Andalucía

En este apartado se presenta la información geográfica de las distribuciones territoriales de nuestra Comunidad Autónoma relevantes para nuestro proyecto.

Debido a que la temática de los datos es sanitaria, estos van a organizarse, geográficamente, en función del *Mapa de Atención Primaria de Salud de Andalucía*. Se trata de una distribución territorial elaborada por el gobierno de nuestra comunidad que divide el territorio andaluz en los siguientes niveles:

- Andalucía se divide en 8 provincias.
- Cada provincia se compone de un conjunto de Distritos Sanitarios.
- Cada Distrito Sanitario, a su vez, se organiza en Zonas Básicas de Salud (ZBS).
- Finalmente, las Zonas Básicas de Salud se componen de términos municipales.

En [9] se puede encontrar la publicación oficial del Servicio Andaluz de Salud sobre el Mapa de Atención Primaria de Salud de Andalucía.

Como ejemplo, se muestra en la figura 2.12 la distribución del Mapa de Atención Primaria de Salud andaluz para la provincia de Granada.

2.3. Tecnologías para el desarrollo de sistemas multidimensionales

Para desarrollar un sistema multidimensional, ya que éste se compone de partes heterogéneas, no podemos decir que exista una herramienta única y unificada que albergue todas estas tareas, al menos desde el punto de vista profesional, del ingeniero o técnico informático.

A pesar de esto, han surgido en los últimos años algunas empresas que desarrollan cierto software orientado al usuario «no informático», con una formación ofimática, que se conocen como «**herramientas (integradas) de usuario final**». Estas herramientas permiten cargar los datos, procesarlos, limpiarlos, generar incluso los hechos y las dimensiones y un soporte integral para las consultas, todo ello a través de una interfaz gráfica. Por tanto, son adecuadas (en general) para pequeños proyectos. Algunas herramientas de este estilo son *Tableau* y *Power BI*.

Figura 2.12: Mapa de Atención Primaria de Salud en Granada. Fuente: [9].

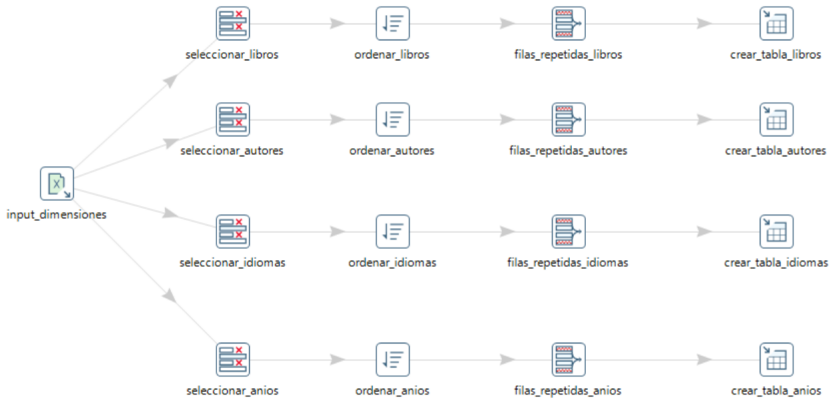


Figura 2.13: Ejemplo de transformaciones en PDI. Fuente: elaboración propia.

En la otra cara de la moneda tenemos las **herramientas profesionales**, que permiten al ingeniero elaborar el sistema teniendo perfecto conocimiento de cómo esa herramienta va integrando las fases del diseño. Se usan en ámbitos profesionales por sus prestaciones (volumen de datos, funcionalidad, disponibilidad de los datos, entre otros). Aunque son más difíciles de usar que las de usuario final.

En dichas herramientas, las transformaciones se van concatenando, formando «cauces de transformaciones», que pueden ramificarse y extenderse tanto como la herramienta permita (ver figura 2.13). Así, se pueden ir ejecutando y depurando las transformaciones que se van haciendo, viendo cómo quedan las tablas tras las transformaciones, etc.

Además, las herramientas profesionales de la fase OLAP, es decir, las que permiten desarrollar los cubos, ofrecen opciones mucho más técnicas y sofisticadas que las herramientas de usuario final, como tareas internas de almacenamiento, definir atributos para los niveles de las jerarquía, procesos de depuración al generar las dimensiones y los cubos y la posibilidad de desplegar el proyecto en un servidor al que se pueden conectar las herramientas de usuario final.

En definitiva, estas herramientas aportan el enfoque profesional que solo pueden realizar profesionales de la Informática por los conocimientos que se requieren.

La tabla 2.2 muestra una comparativa con las características que podemos considerar más interesantes (para nosotros) de estas herramientas y que nos ayudarán a decidir en el futuro.

Tipo herramienta	Herramienta	Gratuita	Transformaciones que soporta	¿Permite crear transformaciones nuevas?	Conexión con fuentes de datos
ETL	*SSIS*	No (versión educativa: Sí)	30+	Sí, en Visual Basic	Ficheros planos, Excel, SaaS (Hadoop y SAP BW)
	PDI	Sí	20+	Sí, en Python/SQL/Js/...	SaaS (Google Analytics/Salesforce) y 40 BD
	Paquetes en R	Sí	Todas las combinaciones posibles	Sí, en R	Ficheros planos, Excel, Bases de Datos
OLAP	*SSAS*	No (versión educativa: Sí)	–	–	–
	Mondrian	Sí	–	–	–

Cuadro 2.2: Comparativa de herramientas útiles. Elaboración propia a partir de lo expuesto en [10].

Capítulo 3

Desarrollo del proyecto

3.1. Planificación temporal del proyecto

En este apartado, se pretende aportar una perspectiva general de la planificación temporal de nuestro proyecto.

El hilo conductor de las tareas realizas ha sido la metodología de trabajo utilizada. Debido a la versatilidad de estos sistemas, tan dependientes de los datos y donde los requisitos de usuario, tal y como los conocemos en el mundo del desarrollo de software, toman un segundo plano, resulta muy sencillo aplicar uno u otro ciclo de vida.

Aunque se explicará con más detalle, podemos adelantar ya que se trata de un ciclo de vida básicamente iterativo, clásico, que subyace de la tan conocida y usada estrategia de diseño de una base de datos: desde unas fases de toma conciencia del mundo a modelar, se atraviesa una etapa de modelización «técnica» o adaptada al software a utilizar para posteriormente lidiar con él y dejar esa base de datos (nuestro sistema multidimensional, en términos «globales») listo para su uso y manipulación.

Para elaborar esta planificación de tareas, se utiliza un **diagrama de Gantt**[1], elaborado usando el software ofimático para Hojas de cálculo *Microsoft Excel*.

En el diagrama (ver figura 3.1) se puede apreciar que se han distinguido unas cinco fases, que se corresponden con:

- A: Contextualización del proyecto y selección del dominio de la información.
- B: Análisis de requisitos

[1]Para conocer cómo elaborar diagramas de Gantt en una hoja de cálculo, he seguido lo explicado en [8].

Nombre de la actividad	Fecha inicio	Duración (días)	Fecha final
A	22/02/2021	21	15/03/2021
B	15/03/2021	4	19/03/2021
C	19/03/2021	72	30/05/2021
D	31/05/2021	28	28/06/2021
E	29/06/2021	9	08/07/2021

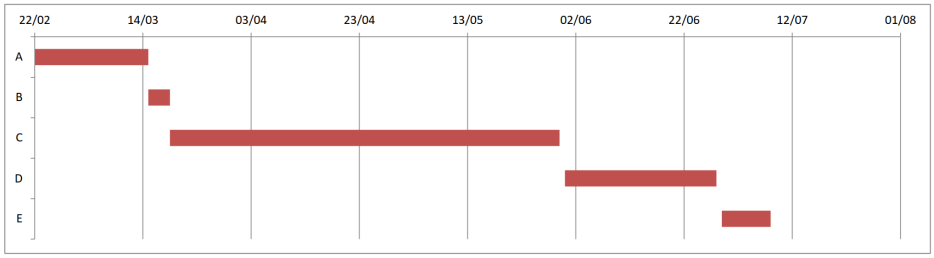


Figura 3.1: Diagrama de Gantt para el proyecto. Fuente: elaboración propia.

- C: Tareas ETL: extracción y transformación (en *R*)
- D:
  - Tareas ETL (ii): creación de la Base de Datos *SQL Server* y carga de datos (*SSIS*).
  - Tareas OLAP: desde *SSAS*, generación de la Base de Datos Multidimensional.

  Temporización en esta fase: aproximadamente la mitad del tiempo cada tarea (*ETL (ii)* y *OLAP*).
- E: Revisión general del proyecto (memoria incluida).

3.2. Presupuesto del proyecto

En este apartado trato de detallar un presupuesto aproximado de un supuesto proyecto real equivalente al que aquí se presenta. Para realizarlo, me baso en un guión de prácticas [3] que se nos proporcionó para una asignatura de mi mención, «Ingeniería de Sistemas de Información» (ISI). En aquel caso, el proyecto también era un Sistema de Información y tanto las referencias necesarias como la forma de elaborarlo se han mantenido, en general. De esta forma, recurro a las indicaciones dadas en el material docente de dicha asignatura así como a diversas referencias legales para elaborar el presupuesto para el presente proyecto.

La suposición que se hace en este caso para elaborar el presupuesto puede ser: «Una pequeña empresa joven de Granada ha sido la adjudicataria de cierto contrato con la Consejería de Salud de la Junta de Andalucía para

desarrollar el sistema multidimensional que nos ocupa, y decide encargar el proyecto a un profesional que trabajará dentro de la empresa temporalmente.»

El presupuesto que se elabora es un presupuesto de desarrollo[2], el cual incluye: gastos de personal y de ejecución.

3.2.1. Gastos de personal

Con este nombre se enmarcan los gastos asociados a costear el «capital humano» necesario para desarrollar el proyecto. En mi caso, se considera que el proyecto, como ya se ha indicado, es realizado por una sola persona, y en base a esto se hará el cálculo.

Detallo las especificaciones generales:

- 1 ingeniero informático (con conocimientos en el dominio del proyecto);
- contrato temporal de 3 meses para el ingeniero[3];
- contribuyentes tipo A[4].

A partir de lo especificado en [3] y en [2], se ha realizado el cálculo manualmente, teniendo en cuenta que:

1. Vamos a fijar el **salario base** (bruto) en 1600 € (respetando el SMI mínimo de 950 €).
2. **Cálculo del coste de cada empleado al mes** (sin contar pagas extras ni vacaciones): 2153 €. Se puede ver en la tabla 3.1.

   Por tanto, el coste del empleado en los **3 meses** se calculará como:

$$2153€ \times 3\ meses = 6459\ € \tag{3.1}$$

3. **Coste total de las vacaciones** sabiendo que el mínimo legal son 30 días de vacaciones por cada 12 meses trabajados (cfr. [2]), se tiene que para 3 meses, el pago proporcional de las vacaciones es de 2,5 días por cada mes de trabajo, con lo cual corresponderá pagar $2,5 \times 3 = 7,5$ días de vacaciones.

[2]Para más información, consultar [3].

[3]El primer mes de mi proyecto estuvo centrado en definir de forma específica el trabajo, ver metodologías, herramientas y demás; en el supuesto real el ingeniero no tendría que hacer estas tareas y por eso se le asignan 3 meses de trabajo, aproximadamente el tiempo que he necesitado desde las primeras etapas del ciclo de vida hasta el final del trabajo.

[4]Los tipos legales de contribuyentes reglados en España se pueden consultar en [6].

Indicador	Porcentaje sobre el salario bruto	Cantidad
Contingencias comunes	23,6 %	377,6 €
Prestación por desempleo	6,7 %	107,2 €
IT/IMS	3,5 %	56 €
Formación profesional	0,6 %	9,6 €
FOGASA	0,2 %	3,2 €
TOTAL		2153 € de coste/empleado

Cuadro 3.1: Coste total de cada empleado al mes, a pagar por la empresa. Elaboración propia a partir de: [3].

Coste total por empleado			
Coste total por empleado	***Nº empleados***	***Coste unitario***	***Coste parcial***
Salario neto por 3 meses	1	6.459,00 €	6.459,00 €
Vacaciones	1	298,63 €	298,63 €
Pagas extraordinarias	1	1.078,80 €	1.078,80 €
TOTAL			**7.836,43 €**

Cuadro 3.2: Coste total de un empleado. Fuente elaboración propia.

Si hacemos una proporción, tenemos que el coste por cada empleado es:

$$\frac{2153,6\ € \times 7,5\ días}{30\ meses} = 298,63\ €\ por\ empleado \tag{3.2}$$

4. **Pagas extraordinarias:** hay que pagar la proporción correspondiente a los 3 meses trabajados al empleado. Dado que por 12 meses se corresponden 2 pagas extraordinarias por empleado, tenemos que pagarle media paga ($\frac{3\ meses \times 2\ pagas}{12\ meses} = \frac{1}{2}$ paga por empleado). En total, el coste de las pagas extraordinarias ascenderá a:

$$2153,6\ € \times \frac{1}{2} \times 1\ empleado = 1078,8\ € \tag{3.3}$$

En total el gasto de pagas extraordinarias ascendería a 1078,8 €.

A partir de los apartados anteriores, en la tabla 3.2 se detallan los gastos totales de personal para el desarrollo del proyecto.

1 – Gastos inventariables					
Descripción	**Unidades**	**Coste unitario**	**Subtotal**	**Amortización**	
Ordenador Portátil "Acer" [1]	1	699,00 €	699,00 €	139,80 € (vida útil=10 años)	
Impresora Wifi/Bluetooth "HP" [2]	1	229,00 €	229,00 €	22,90 € (vida útil=10 años)	
SQL Server[3]	1	11.610,25 €	11.610,25 €	1.935,04 € (vida útil=6 años)	
Power BI Pro[4]	1	25,20 €	25,20 €	4,20 € (vida útil=6 años)	(8,40 €/mes)
TOTAL			**12.563,45 €**	**2.101,94 €**	

1 https://www.elcorteingles.es/electronica/A38773056-portatil-acer-aspire-3-amd-ryzen-5-16gb-1tb-ssd/

2 https://www.elcorteingles.es/electronica/A32365052-impresora-multifuncion-tinta-hp-smart-tank-plus-555-wi-fi-y-bluetooth/

3 https://www.microsoft.com/es-es/sql-server/sql-server-2019-pricing#OneGDCWeb-ContentPlacementWithRichBlock-pp5ed24

4 https://powerbi.microsoft.com/es-es/pricing/ (lo adquirimos por 3 meses, por simplicidad)

2 – Gastos fungibles				
Descripción	**Unidades**	**Coste unitario**	**Subtotal**	**Amortización**
Paquete de folios	3	4,90 €	14,70 €	1,84 € (vida útil=8 años)
Útiles de escritura (estimación)	1	10,00 €	10,00 €	1,25 € (vida útil=8 años)
Cartuchos de tinta negra	2	16,00 €	32,00 €	4,00 € (vida útil=8 años)
TOTAL			**56,70 €**	**7,09 €**

	Sin amortización	*Con amortización*
GASTOS TOTALES DE EJECUCIÓN	**12.620,15 €**	**2.109,03 €**

Cuadro 3.3: Gastos de ejecución del proyecto. Fuente: elaboración propia.

3.2.2. Gastos de ejecución del proyecto

Los gastos de ejecución del proyecto se dividen en:

- Costes de adquisición de ***material inventariable***[5], como equipos informáticos.
- Costes de adquisición de ***material fungible***[6], como el material de oficina.
- Coste de ***alquiler*** de una oficina. En este caso la actividad se desarrolla en la empresa y no es necesario alquilar ninguna oficina o similar.

La tabla 3.3 detalla los gastos de ejecución del proyecto, con las referencias indicadas para los cálculos.

Para el cálculo de la amortización de los gastos inventariables se ha seguido lo explicado en [7], y para tomar la vida útil de los productos, la referencia oficial de la Agencia Tributaria [1].

[5]Se realizará una estimación de los costes que supondrá para la empresa, aunque haya, probablemente, buena parte de este material que, en mayor o menos medida, ya esté presente en la empresa.

[6]Ídibem.

Coste total del proyecto		
	Sin amortización	*Con amortización*
Gastos de personal	7.836,43 €	7.836,43 €
Gastos de ejecución	12.620,15 €	2.109,03 €
TOTAL	**20.456,58 €**	**9.945,46 €**

Cuadro 3.4: Gasto total del proyecto (aunque para los Gastos de personal no se hace amortización, se añade en la celda correspondiente para clarificar de dónde surge el valor total de esa columna). Fuente: elaboración propia.

3.2.3. Coste total del proyecto

En base a los dos tipos de gastos que acabamos de detallar (de personal y de ejecución), podemos hacer el cálculo del coste total del proyecto, detallado en la tabla 3.4.

3.3. Repositorio para el proyecto

Este proyecto tiene asociado un repositorio en mi cuenta personal de GitHub, accesible en

`https://github.com/alonso-bh/TFG`

La estructura del repositorio es la misma que la de la carpeta «local», cuyos directorios más importantes se pueden consultar en la figura A.1.

3.4. Ciclo de vida para el desarrollo del proyecto

Debido a las peculiaridades de los Sistemas Multidimensionales, conviene detenerse a estudiar cuál va a ser la estrategia, metodología o «ciclo de vida» que vamos a tomar como referencia para su desarrollo.

En [12] se presentaron tres modelos de ciclos de vida diferentes:

Ciclo de vida basado en datos Los datos eran la parte fundamental del desarrollo y los requisitos del cliente quedaban relegados a un segundo plano.

Ciclo de vida basado en requisitos Al contrario que el anterior, los requisitos centraban las etapas de diseño (conceptual, lógico, físico).

Ciclo de vida mixto Este enfoque trataba de trabajar los datos por un lado, y los requisitos por otro, y de la correcta combinación y estudio de ambos, surgía el diseño conceptual.

¿Qué modelo de ciclo de vida debemos escoger? El que se recomendó en [16] es el ciclo de vida *mixto*, donde se contruye, como ya se ha comentado, todo el sistema materialmente a partir de los datos y los requisitos. A pesar de lo ideal (al menos teóricamente) de este ciclo de vida mixto, es cierto que en nuestro caso se dan dos circunstancias que nos hacen inclinarnos más por el **enfoque orientado a datos**:

- No tenemos un cliente real que nos haya ofrecido un análisis de requisitos, y he tratado de usar un tipo de análisis de requisitos genérico teniendo en cuenta que los «clientes» hemos sido mi tutor y yo.
- Este proyecto está muy enfocado a explotar al máximo la información que tengamos disponible, en concreto, la que podamos encontrar en los organismo públicos oficiales de Andalucía y España.

Doy así por justificado el uso de este modelo de ciclo de vida. En la figura 3.2 se ilustra el ciclo de vida escogido. Pero nuestro proyecto es más sencillo que el supuesto «proyecto genérico» que aquí se plantea, con lo cual, muestro en la figura 3.3 ese mismo ciclo de vida, adaptado, más intuitivo (a mi modo de parecer) y traducido a nuestra nomenclatura. Lo que podemos apreciar en este nuevo esquema es que la etapa de *Workload refinement* y sus correspondientes resultados no aparece. El motivo es que esta etapa, usando la Metodología de Kimball para el modelo conceptual, ya se realiza de una forma explícita.

Lo que sí creo que es importante que quede claro es lo siguiente:

- Para poder llegar al Diseño Conceptual, la etapa quizás más importante en el diseño de una Base de Datos (Multidimensional) tenemos que hacer dos cosas: limpiar y lograr la coherencia en las fuentes de datos (esto lo haremos en R) y, por otro lado, tener la lista de requisitos del «usuario» (el tutor y yo) lista para poder diseñar un modelo conceptual adecuado a lo que se ha pedido.
- Hasta que no tengamos claro qué tablas (dimensiones y hechos) vamos a tener, no podemos generar las correspondientes tablas en R, ese es el motivo por el cual del Diseño Lógico sale una flecha hacia el ítem «Procedimientos ETL».
- El Diseño Físico no nos debe preocupar en demasía en este proyecto, pues una vez tengamos todos los datos cargados en la base de datos y los cubos generados, será el software SGBD usado el que se encargará de hacer todas las tareas asociadas (optimización de consultas, almacenamiento eficiente, etc.)

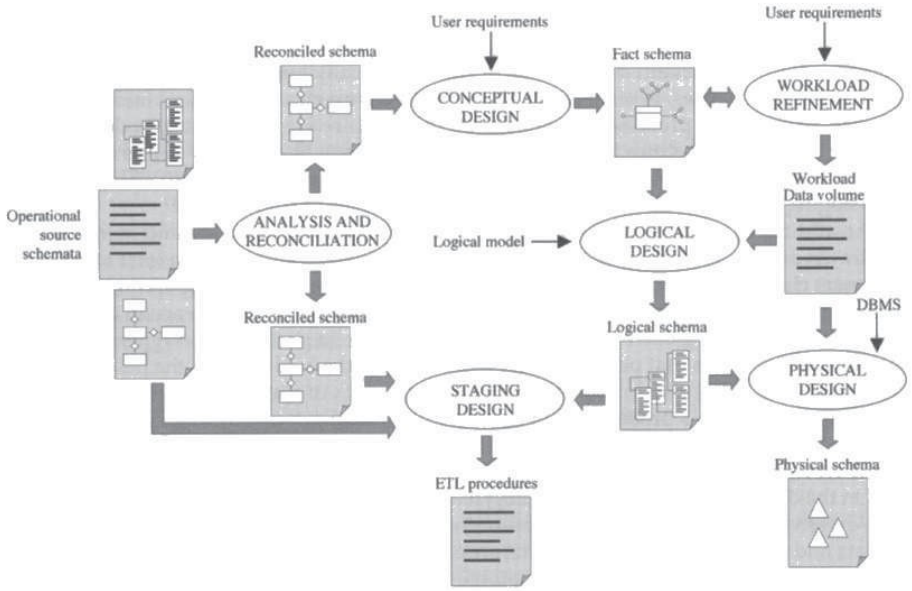


Figura 3.2: Ciclo de vida de un sistema multidimensional con enfoque *orientado a* ***datos***. Fuente: [12].

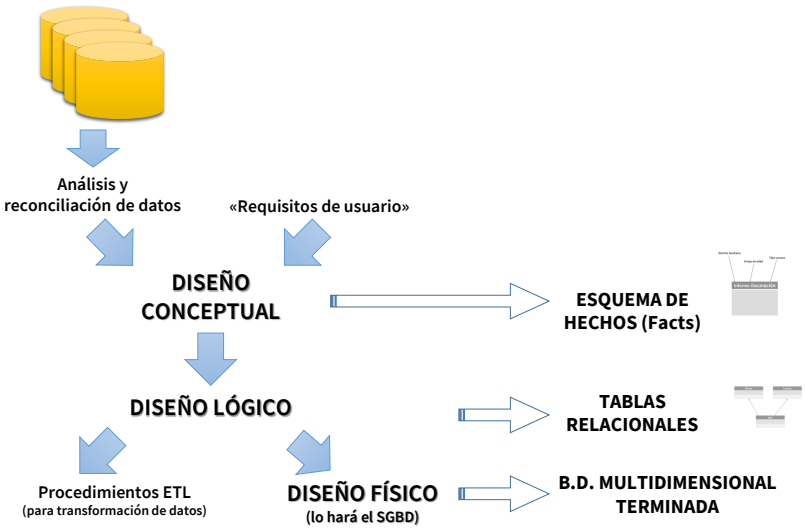


Figura 3.3: Ciclo de vida orientado a datos *adaptado a nuestro proyecto*. Fuente: Elaboración propia a partir de la figura 3.2.

3.5. Estrategia para los diseños conceptual, lógico y físico

En la asignatura **SMD** se utilizó una metodología específica para diseño de Bases de Datos Multidimensionales, que albergaba las siguientes etapas:

Diseño conceptual aplicación de la metodología de Kimball [13], que a su vez se compone de cinco fases, a realizar en el orden propuesto:

Fase 1: Descripción de los procesos de negocio Hay que estudiar cuál es el foco de atención o la cuestión central del cliente. Puede haber un único ***foco de atención*** o varios. Cada foco de atención constituirá un «cubo OLAP».

Fase 2: Establecer la granularidad Consiste en:

1. Determinar y expresar el significado de los focos de atención definidos anteriormente. Por ejemplo: Ventas, Matrículas escolares, Calificaciones, etc.
2. Definir las **bases** del modelo. Las bases de un modelo multidimensional son el nivel de detalle de los puntos de vista que nos hacen falta para identificar cada ítem (registro o fila) de los hechos. Además, hay que definir el atributo concreto para cada uno de ellos. Por ejemplo: una *Venta* puede quedar representada (y «***diferenciada*** ***del resto***) por:
   - La tienda donde se realizó la venta (DÓNDE)
   - El producto comprado (QUÉ)
   - La fecha de la compra (CUÁNDO)

   En este caso, las bases serían los niveles seleccionados de cada *punto de vista*: *Tienda*, *Producto*, *Fecha (dd/mm/aa)*.

Fase 3: Diseñar las dimensiones de análisis Hay varias tareas:

1. Elegir el nombre de las dimensiones, que en general serán «preguntas»: QUÉ, CUÁNDO, DÓNDE, ...
2. Elegir los niveles y las jerarquías.
3. Elegir posibles niveles adicionales, tales como Mes, Código, etc... a partir de datos calculables, y preguntando al decisor.

Recordemos, además, que las dimensiones tienen que ser: Estrictas y Cubrientes.

Fase 4: Identificar las mediciones de los hechos Hay que tener en cuenta que:

- Han de ser tipos numéricos.

- Van a tener un valor concreto (igual o no) para cada registro de la tabla de Hechos, o lo que es lo mismo, para cada celda del cubo OLAP.

También hay que estudiar los fenómenos de:

- La **aditividad**: ¿tiene sentido sumar los valores de una determinada medición?
- La posible inclusión de **mediciones calculadas**: a partir de la medición Mujeres y la de Hombres dentro de un análisis poblacional podemos calcular el total de habitantes.

Fase 5: Estimación del número de instancias Se trata de hacer una estimación de cuántos «registros» se almacenarán, de una forma aproximada, en nuestras tablas de la base de datos correspondientes a los hechos y las dimensiones.

Diseño lógico (ROLAP en estrella) o lo que es lo mismo, aplicar una heurística muy sencilla para transformar los elementos del *Diseño Conceptual* (hechos, dimensiones, atributos, mediciones, etc.) en elementos de la semántica de una Base de Datos relacional.

Diseño físico de él encarga el DBMS (Sistema Gestor de Base de Datos) que utilizaremos.

3.6. Descripción y encuadre de las herramientas usadas dentro del ciclo de vida del proyecto

En el Capítulo 2, sección 2.3, se hizo una descripción comparativa del conjunto de tecnologías que se usan en el desarrollo de nuestro proyecto. Ahora, pasamos a hacer una selección de esas herramientas y se aporta lo que puede definirme como el «motivo» de su elección para la etapa de turno.

En primer lugar, la herramienta para la parte ETL, es decir, obtención, transformación y carga de datos se usarán:

Obtención y transformación de datos se realizará íntegramente programando un conjunto de *scripts* de R, un lenguaje de origen estadístico y con amplio recorrido (en términos de paquetes desarrollados) en la manipulación más que exhaustiva de datos.

Contamos además con la ventaja de que tenemos un IDE propio *RStudio*, que ya he utilizado (recientemente) antes en la asignatura SIG[7] y para numerosas tareas no curriculares personales.

[7] *Sistemas de Información Geográfica*, optativa de Cuarto Curso, cursada en el Cuatrimestre pasado.

Destaquemos también algo clave: debido a la naturaleza de nuestros datos, el hecho de poder programar un *script* que se ejecute con la frecuencia que se necesite y que descargue, almacene y transforme los datos dónde y cómo se quiera también es algo que me hizo decantarme por esta opción frente a otras. Además, y esto ya se puede decir que es algo menos técnico, mi ambición personal por manejar y profundizar en un lenguaje orientado al trabajo estadístico y de datos como R también fue un motivo de peso.

Aunque no se haya indicado ya explícitamente, **también con *R* generaremos todas las tablas relacionales de hechos y dimensiones**, listas para llevarlas a la Base de Datos (pero esta fase de generación de tablas de hechos y dimensiones no llega hasta acabar el diseño lógico, evidentemente).

Carga de datos En esta fase, en nuestro caso, tenemos que limitarnos a cargar los ficheros de cada tabla relacional generada antes en la Base de Datos relacional a través de la herramienta seleccionada. De las ídem que se proporcionaban para esta fase nos quedaremos con SSIS (SQL Server Integration Services), que se integra en el software de Microsft *SQL Server Data Tools*.

OLAP: creación de la base de datos multidimensional la herramienta será *SSAS* (*SQL Server Analysis Services*), que nos ofrece una interfaz, desde mi punto de vista, mucho más amigable y ligera que las herramientas equivalentes de Mondrian, y además, algo que considero clave: tiene habilitada una pequeña herramienta de consultas integrada en el propio software para hacer pruebas sobre el cubo, y así el desarrollador puede verificar si se han generado bien los cubos, **sin necesidad de usar un cliente externo de consultas**; las herramientas de Mondrian, por su parte, generan los hechos y las dimensiones, pero no ofrecen herramientas de pruebas al desarrollador.

Con todo ello, y a modo de colofón, podemos concluir así:

En la fase de **obtención de datos**, estamos programando todas las tareas en *R*, un lenguaje con todos los paquetes de software libre y un IDE propio e interactivo.
En las fases posteriores, usamos herramientas de Microsoft (SSIS/SSAS), con una interfaz muy amigable y una herramienta integrada de consultas simples para hacer todas las pruebas que necesitemos sin un cliente de consultas externo.

3.7. Análisis y reconciliación de fuentes de datos

Esta es quizás la etapa crítica de todo el proceso de desarrollo. Recibe como entrada todo el conjunto de fuentes de datos, con sus esquemas y peculiaridades, y debe devolver un esquema (o conjunto homogéneo de esquemas) reconciliado que represente a todas las fuentes de datos, de cara a poder realizar de forma óptima y coherente, junto con los requisitos del usuario, el diseño conceptual de nuestra Base de Datos multidimensional (y, evidentemente, el resto del proceso que le sigue)[8].

3.7.1. Obtención de datos

En la planificación del proyecto, en base al período estimado para la realización total del trabajo, se especificaba que había una fase previa de estudio de qué datos vamos a utilizar dentro del ámbito de Andalucía. Es por ello que la principal fuente para obtener los datos con los que construir el sistema propuesto es el **Instituto de Estadística y Cartografía de Andalucía** (*IECA*)[9]. En concreto, tomamos el informe sobre COVID-19 que se publica, bajo esta misma página, en el recurso `/salud/static/index.html5`, a través del servicio *BADEA* (Banco de Datos Espaciales de Andalucía). De esta referencia, que fue acordada conjuntamente con el tutor, vamos a tomar los «datos centrales» del proyecto.

La información que se ofrece es muy completa y permite su descarga gratuita en diversos formatos (XLS, CSV, ODS, etc.); en mi caso, y por mi conocimiento personal de este formato opté por *XLS*, que además se trabaja muy bien con el lenguaje de programación *R*.

Adicionalmente, y para la fase de **enriquecimiento de las dimensiones**, se han usado conjuntos de datos menores procedentes del *Instituto Nacional de Estadística* (`https://ine.es/`).

3.7.2. Descripción de las fuentes de datos usadas en el proyecto

En este apartado se realiza una descripción de los conjuntos de datos que se han utilizado, resaltando algunos detalles que considero relevantes en algunos de ellos, como por ejemplo si son datos que se descargan diariamente, cambia su estructura en origen, etc.

Distingo por tanto dos tipos de ficheros:

[8] Podemos repasar el ciclo de vida en la figura 3.3.

[9] Se puede acceder a la web del IECA en `http://www.juntadeandalucia.es/institutodeestadisticaycartografia/`.

- los conjuntos de datos centrales del proyecto, de los que se obtienen las cifras COVID clave;
- los conjuntos de datos y otros recursos auxiliares del proyecto, que han servido básicamente para poder explotar al máximo los datos de la web para enriquecer las dimensiones.

3.7.2.1. Datos «principales» del proyecto

Todos estos datos han sido descargados del ya mencionado portal del *IECA*. Son los siguientes[10]:

1. **Datos de notificación por días naturales** Este conjunto de datos se actualiza de Lunes a Viernes, como el resto, y se va actualizando para añadirle 9 filas asociadas a cada provincia andaluza y otra fila con los datos globales de Andalucía de ese día. Las estadísticas que se muestran son, en cada fila:
   - Confirmados PDIA en ese día
   - Total de confirmados (Confirmados PDIA y confirmados por otra prueba diagnóstica distinta) en ese día
   - Hospitalizados en ese día
   - Curados en ese día
   - Personas que han entrado en la UCI en ese día

   En la figura 3.4 se muestra una captura realizada desde *RStudio* de los datos, ya formateados y limpios.
2. **Datos de notificación a nivel de municipio** Este conjunto de datos, como el resto, se actualiza de Lunes a Viernes (salvo festivos) mediante una estrategia de acumulación de datos, es decir, cada día se van sumando los casos notificados, curados, fallecidos y demás cifras en el último día al del día anterior. Esta es la tónica de este y del resto de conjuntos que nos quedan por describir aquí.

   En cuanto al detalle de los datos que ofrece esta tabla, se proporcionan los siguientes indicadores para cada fila (términos municipales de Andalucía):
   - Confirmados PDIA

[10] Cabe mencionar que los «nombres» que les he asignado a cada conjunto de datos son una versión simplificada de los que aparecían en la fuente original, por ser los de ésta, en mi opinión, demasiado largos. De estos nombres simplificados saldrán los ídem de los diversos ficheros de datos que se utilizan a lo largo de toda la implementación del proyecto, con los correspondientes prefijos donde sea necesario, tal y como explicaré más adelante.

Fecha diagnóstico	Territorio	Confirmados PDIA	Total confirmados	Hospitalizados	UCI	Fallecidos
08/04/2021	Almería	18	18	1	0	0
08/04/2021	Cádiz	16	16	1	0	0
08/04/2021	Córdoba	31	31	1	0	0
08/04/2021	Granada	34	36	0	0	0
08/04/2021	Huelva	68	68	0	0	0
08/04/2021	Jaén	37	37	0	0	0
08/04/2021	Málaga	52	52	1	0	0
08/04/2021	Sevilla	88	88	0	0	0
07/04/2021	Almería	227	227	2	1	0
07/04/2021	Cádiz	254	254	2	0	0

Figura 3.4: Conjunto de datos a nivel de provincia y para los días naturales. Fuente: elaboración propia.

- Confirmados PDIA 14 días
- Tasa PDIA 14 días
- Confirmados PDIA 7 días
- Total Confirmados
- Curados
- Fallecidos

Destacamos además el hecho de que la autoridad de los datos haya añadido la columna `Tasa PDIA 14 días` en esta tabla de datos por municipios, debido a la importancia de este valor para las decisiones que se toman por parte de los gobiernos.

Además, para cada municipio han añadido una columna con su población[11] y el Distrito Sanitario (DS) al que pertenecen, lo cual nos facilitará la tarea posterior al generar la dimensión correspondiente a nivel de código *R*.

Aunque no implica ningún problema o error, también considero importante destacar que este conjunto de datos se encuentra originalmente fragmentado por provincias, lo cual nos obliga a fusionarlos diariamente y después ya trabajar con el conjunto de datos completo día a día.

De nuevo, se muestra en la figura 3.5 un fragmento de este conjunto para un día concreto, donde se muestra también la columna temporal (fecha), como en todos.

[11] La población de cada municipio se incluye originariamente en este conjunto de datos, no lo he añadido yo.

Lugar de residencia	Población	Confirmados PDIA	Confirmados PDIA 14 días	Tasa PDIA 14 días	Confirmados PDIA 7 días	Total Confirmados	Curados	Fallecidos	Fecha
Almería (distrito)	306142	20117	861	281.24203800850586	307	20297	11293	261	25/02/2021
Abla	1248	43	0	0	0	43	26	2	25/02/2021
Abrucena	1183	48	4	338.12341504649197	0	48	36	1	25/02/2021
Alboloduy	609	4	0	0	0	4	3	0	25/02/2021
Alcudia de Monteagud	138	0	0	0	0	0	0	0	25/02/2021
Alhabia	677	14	0	0	0	15	11	0	25/02/2021
Alhama de Almería	3691	265	8	216.74342996477918	2	267	158	6	25/02/2021
Alicún	201	4	0	0	0	4	0	0	25/02/2021
Almería (capital)	201322	13696	612	303.99062198865499	220	13852	7822	159	25/02/2021
Almócita	176	5	0	0	0	5	0	1	25/02/2021

Figura 3.5: Conjunto de datos al nivel geográfico más bajo posible: el municipio. Como el resto de conjunto de datos que nos faltan, la fecha es de día hábil (día de descarga). Fuente: elaboración propia.

	Territorio	Confirmados PDIA	Confirmados PDIA 14 días	Confirmados PDIA 7 días	Total Confirmados	Curados	Fallecidos	Vive en residencia	Fecha
30	Sevilla Sur	589	2	2	611	448	91	Residencias de mayores	04/04/2021
31	Aljarafe	385	1	0	412	298	95	Residencias de mayores	04/04/2021
32	Sevilla Este	80	0	0	83	66	17	Residencias de mayores	04/04/2021
33	Sevilla Norte	289	0	0	294	215	48	Residencias de mayores	04/04/2021
34	Sevilla (distrito)	587	0	0	623	455	124	Residencias de mayores	04/04/2021
35	Almería (distrito)	25	0	0	58	51	5	Otro tipo de institución	04/04/2021
36	Levante-Alto Almanzora	2	0	0	2	2	0	Otro tipo de institución	04/04/2021
37	Poniente de Almería	7	0	0	7	4	3	Otro tipo de institución	04/04/2021
38	Campo de Gibraltar Este	0	0	0	0	0	0	Otro tipo de institución	04/04/2021
39	Campo de Gibraltar Oeste	12	0	0	12	12	0	Otro tipo de institución	04/04/2021

Figura 3.6: Primer conjunto de datos de residencias (I). Fuente: elaboración propia.

3. **Datos de residencias (I)**: este es el primer conjunto de datos que nos ofrece el *IECA* sobre cifras en residencias. En cuanto a sus características, muestra los datos a nivel de *Distrito Sanitario*, e indicando también el tipo de residencia (hay dos tipos) en que se hace la notificación. También tendremos, como siempre, la fecha de notificación del día en que se descargaron los datos.

   Más concretamente, los indicadores COVID que se aportan en esta tabla son:

   - Confirmados PDIA
   - Confirmados PDIA 14 días
   - Confirmados PDIA 7 días
   - Total Confirmados
   - Curados
   - Fallecidos

   En la figura 3.6 se muestra una captura de los datos, donde podemos ver que ya se ha añadido la fecha de notificación, se indica el tipo de residencia y también el Distrito Sanitario, junto a los indicadores comentados.

Vive en residencia	Edad	Confirmados PDIA	Total Confirmados	Curados	Fallecidos	Sexo	Fecha
Residencias de mayores	Menos de 65 años	215	237	206	23	Hombres	05/04/2021
Residencias de mayores	De 65 a 69 años	237	257	214	26	Hombres	05/04/2021
Residencias de mayores	De 70 a 74 años	296	321	242	66	Hombres	05/04/2021
Residencias de mayores	De 75 a 79 años	369	392	281	91	Hombres	05/04/2021
Residencias de mayores	De 80 a 84 años	549	572	394	154	Hombres	05/04/2021
Residencias de mayores	De 85 a 89 años	670	710	455	230	Hombres	05/04/2021
Residencias de mayores	De 90 a 94 años	410	434	288	132	Hombres	05/04/2021
Residencias de mayores	De 95 a 99 años	104	108	56	50	Hombres	05/04/2021
Residencias de mayores	De 100 y más años	10	12	8	4	Hombres	05/04/2021
Otro tipo de institución	Menos de 65 años	633	702	627	25	Hombres	05/04/2021

Figura 3.7: Segundo conjunto de datos de residencias (II). Fuente: elaboración propia.

4. **Datos de residencias (II)**: este conjunto de datos sobre residencias difiere del anterior en los «puntos de vista» de análisis que ofrece, esto es:

   - No tiene información geográfica, como el anterior, que indicaba el Distrito Sanitario asociado a los datos notificados en la tabla.
   - Añade información sobre el sexo (Mujer/Hombre) y el rango de edad.

   A pesar de esto, también cabe destacar que aquí también se ofrece el tipo de residencia.

   Los indicadores que se ofrecen en este caso son:

   - Confirmados PDIA
   - Total Confirmados
   - Curados
   - Fallecidos

   En la figura 3.7 se proporciona una vista previa de este conjunto de datos.

5. **Datos de vacunas**. Este conjunto de datos es quizás uno de los más importantes desde el punto de vista «estratégico». Se muestran los datos por **provincias**, por **rango de edad** de vacunación (es distinto de los rangos de edad de otros conjuntos de datos) y, evidentemente, por fecha de notificación. Los indicadores de vacunación que se proporcionan aquí son:

   - Num personas con 1 dosis,
   - Porcentaje de personas con 1 dosis,
   - Num personas con pauta completa,

Provincia	Num personas con 1 dosis	% personas con 1 dosis	Edad	Num personas con pauta completa	% personas con pauta completa	Fecha
Almería	30077	104.55746367239102	De 80 y más años	29448	102.37085448098449	11/06/2021
Cádiz	57568	107.04351059873559	De 80 y más años	56572	105.19152101152845	11/06/2021
Córdoba	51795	104.31603963586562	De 80 y más años	51022	102.75920406025941	11/06/2021
Granada	52125	103.79537625201617	De 80 y más años	51222	101.99725203608196	11/06/2021
Huelva	26282	106.38332321392431	De 80 y más años	25780	104.35134588140053	11/06/2021
Jaén	43037	101.58381721191522	De 80 y más años	42372	100.01416229995752	11/06/2021
Málaga	76154	100.9986604952189	De 80 y más años	74644	98.996034535350987	11/06/2021

Figura 3.8: Conjunto de datos sobre estadísticas de vacunación por grupos de edad y provincia de inoculación de la dosis. Fuente: elaboración propia.

Sexo	Categoría profesional	Confirmados PDIA	Confirmados PDIA 14 días	Confirmados PDIA 7 días	Total Confirmados	Fallecidos	Curados	Fecha
Hombres	Sanitario en centro sanitario	2843	19	6	3175	11	2924	10/04/2021
Hombres	Sanitario en centro sociosanitario	425	1	0	453	2	420	10/04/2021
Hombres	Sanitario en otro centro	250	4	1	255	2	196	10/04/2021
Hombres	No sanitario en centro sanitario	473	1	0	532	1	491	10/04/2021
Hombres	No sanitario en centro sociosanitario	477	4	1	501	3	440	10/04/2021
Hombres	Atención al público	630	3	0	649	0	631	10/04/2021
Hombres	Fuerzas Armadas y Cuerpos de Seguridad	1344	24	7	1433	0	1091	10/04/2021
Hombres	Personal de ayuda a domicilio	102	3	1	104	0	85	10/04/2021
Hombres	Personal de oficina de farmacia	63	3	1	69	0	58	10/04/2021

Figura 3.9: Conjunto de datos sobre grupos profesionales de riesgo. Fuente: elaboración propia.

- Porcentaje de personas con la pauta completa.

Una «vista previa» de los datos se aporta en la figura 3.8.

6. **Datos de profesionales**. Estos datos ofrecen información sobre los colectivos laborales de mayor riesgo en esta pandemia. El conjunto de datos no tiene información geográfica, pero indica el **sexo** (Mujer/Hombre) y el **colectivo profesional de riesgo** (hay diez). Los indicadores COVID que se proporcionan en este caso son:

   - Confirmados PDIA
   - Confirmados PDIA 14 días
   - Confirmados PDIA 7 días
   - Total Confirmados
   - Curados
   - Fallecidos

   La «vista previa» de este conjunto de datos, en este caso, se aporta en la figura 3.9.

cod_provincia	nombre_provincia
02	Albacete
03	Alicante/Alacant
04	Almería
01	Araba/Álava
33	Asturias
05	Ávila
06	Badajoz
07	Balears, Illes
08	Barcelona
48	Bizkaia
09	Burgos
10	Cáceres

Figura 3.10: Fragmento del conjunto de datos sobre los códigos de las provincias españolas del *Instituto Nacional de Estadística*. Fuente: elaboración propia.

3.7.2.2. Datos adicionales usados en el proyecto

Para este proyecto hemos necesitado tres conjuntos de datos adicionales:

1. Datos sobre los **códigos de las provincias** de España: está disponible en [21]. Se usa para enriquecer las dimensiones, en este caso, con un atributo asociado a la Provincia. Un fragmento se muestra en la figura 3.10.

2. Datos con los códigos de municipios de España: tiene el mismo cometido que el conjunto anterior, solo que ahora se aporta información sobre el municipio, su código oficial del Instituto Nacional de Estadística (INE). Está disponible en [22].

3. Listado de municipios de Andalucía con su Provincia y su Distrito Sanitario. Es de elaboración propia (y manual) a partir de los datos descritos en la sección anterior, y se usará para enriquecer la dimensión que contenga como nivel al municipio.

El motivo del su uso de los dos primeros es para el «enriquecimiento de las dimensiones», dentro del Diseño Conceptual, tal y como se indicó al explicar la Metodología de Kimball. Por su parte, el último es clave para la construcción de una hipotética dimensión geográfica.

3.7.3. Tareas en *R*

Todos los pasos que hemos descrito hasta ahora en la presente sección de «Análisis y reconciliación de fuentes de datos» se materializa en una serie de tareas en el lenguaje de programación *R*.

En este apartado se realiza una breve descripción de esas tareas, indicando qué *scripts* de *R* he creado, por qué, y qué tareas se hacen en cada uno.

3.7.3.1. Descarga de datos

La descarga de los datos se hace automáticamente de Lunes a Viernes (salvo festivos), es decir, los días hábiles, que son en los que el IECA publica datos. Para ello se usa el script `descargar_datasets.R`, ubicado, dentro de la carpeta del proyecto, en `proyecto_tfg`, que no es más que un *proyecto* de *RStudio*[12]. En este script se hacen las siguientes tareas:

1. Lee la ruta de la carpeta raíz del proyecto, que se le pasa por parámetro.

2. Crear dentro del directorio `datos` un directorio nuevo para guardar los datos del día de turno con el formato `dd-mm-yy` [día-mes-año] (por ejemplo, el día 25 de Mayo de Mayo la carpeta se llamará `25-05-21`).

3. Se procede a la descarga vía URL de los ficheros de datos «vírgenes» del IECA. Se le asignan nombres significativos para poder acceder a ellos cómodamente pero también poder identificarlos correctamente sin generar confusiones.

4. Una vez descargados los datos, se llama a una serie de procedimientos que se encargarán de la limpieza y homogeneización de todas las fuentes de datos descargadas. A continuación hablaremos de esto en detalle.

5. Cuando termina la ejecución del script, tenemos una serie de ficheros en el directorio `datos` con los datos globales «limpios».

De esta forma, tendremos modificados (actualizados) diariamente dos tipos de ficheros:

[12]Los proyectos R se limitan a añadir un fichero que «representa» al proyecto y otro con la bitácora sobre las variables del entorno del proyecto, para poder restaurar la sesión en el IDE cada vez que se abre el proyecto.

- Los ficheros que almacenan los datos en formato de tabla plana (y no de tabla dinámica[13]) desde el primer día que se están recogiendo y procesando. Son un total de seis ficheros, que coinciden justamente con los seis conjuntos de datos fuente que explicábamos en la subsección 3.7.2.1. Los nombres de estos ficheros son:
  - `datos_dias_naturales.csv`
  - `municipios.csv`
  - `residencias.csv`
  - `residencias_edad_sexo.csv`
  - `profesionales.csv`
  - `vacunas.csv`
- Una serie de ficheros paralelos a éstos de carácter auxiliar, dentro del directorio `datos/ayer`, que se usan para poder extraer de las tablas descargadas los datos, restando a los del día actual los del anterior, y para eso se almacena los del día anterior.

3.7.3.2. Procesamiento de conjuntos de datos: limpieza y homogeneización

En el mismo directorio que el script anterior tenemos otros que empiezan todos por el prefijo `preparar_`, y que almacenan, por separado, esas funciones que acabamos de comentar para poder procesar los datos fuentes y dejarlos preparados para justo después poder generar todas las tablas de los hechos y las dimensiones.

Cabe referir que en este caso se da la circunstancia de que todos los datos centrales del proyecto son de la misma fuente, el IECA, y ésto nos va a facilitar la homogeneización y también a la comprensión (por parte del programador) de la información.

Más concretamente, los ficheros de los que hablamos son:

1. `preparar_datos_dias_naturales.R`
2. `preparar_datos_municipio.R`
3. `preparar_datos_profesionales.R`
4. `preparar_datos_vacunacion_basicos.R`
5. `preparar_datos_residencias.R`

[13]Para poder elaborar las tablas de las dimensiones y de hechos, el formato de tabla plana es absolutamente imprescindible.

6. `preparar_datos_residencias_edad_sexo.R`

Cada fichero contiene al menos una función para el conjunto de datos correspondiente (de los seis conjuntos «clave» que estamos manejando). Esta función es la que será llamada desde el ya comentado script `descargar_datasets.R`, de tal forma que, como ya se ha comentado, todos los días se descarguen y actualicen los datos y se almacenen «preparados» para poder llegar «cuando se quiera» y lanzar los scripts que generan los hechos y las dimensiones.

Todos estos ficheros siguen una lógica de procesamiento muy parecida, que se puede resumir de la siguiente forma:

1. Lectura de los datos del día (ya se han descargado y almacenado previamente).
2. Tareas de limpieza; por ejemplo, se eliminan filas nulas, se rellenan columnas con datos correctos, se cambian los nombres de columnas problemáticos (tildes y símbolos extraños, entre otros), etc...
3. El paso más importante: preparar los datos para almacenarlos junto con los del resto de días anteriores en los conjuntos indicados en el apartado 3.7.3.1. En este paso distinguimos dos vertientes:
   a) El caso concreto del conjunto de datos asociado a los **días naturales** (`datos_dias_naturales.csv`), que no presenta los datos acumulados (históricos), sino que para cada día -natural- muestra solo los de ese día, y no el valor acumulado, se procesa reemplazando el fichero que tenemos almacenado actualmente en nuestro directorio `datos` por el que se ha descargado.
   b) El resto de conjuntos de datos sí que requieren una transformación. Cada día tenemos la misma estructura de fichero, mismas filas y columnas, y los datos se van **acumulando** (o cambiando, donde proceda) del día actual con respecto a los del día anterior. Esto nos obliga a que, en esta fase, tengamos que cargar el conjunto de datos del día anterior (ya procesado y limpio, claro) e ir restando las columnas del día de hoy a las del día anterior, obteniendo así las cifras del día de hoy «netas». Este procedimiento es la «idea general» para trabajar este segundo tipo de ficheros.

En el siguiente listado de código muestro el esqueleto general de estos ficheros:

```
1 preparar_datos_profesionales <- function(path_fichero){
2
```

```
3   library(...)     # importar librerias y ficheros necesarios
4
5   datos <- import(fichero.xls) #importar el conjunto de datos
6
7   hoy <- procesar_fichero(datos)  # tareas para procesar el conjunto
        de datos
8
9   if (primera_descarga)  # primer dia que se descargan datos
10  then
11    write(hoy, "datos/ayer/fichero_ayer.csv")
12
13  else   # segundo dia en adelante
14    ayer <- import("datos/ayer/fichero_Ayer.csv")
15    write(hoy, "datos/ayer/fichero_ayer.csv")
16     hoy <- hoy - ayer
17
18     datos_globales <- import("datos/fichero.csv")
19     datos_globales <- datos_globales U hoy  # aniadir datos de hoy al
           conjunto global de referencia
20     write(datos_globales, "datos/fichero.csv")
21  endif
22
23 }
24 # NOTA: para evitar errores de formato, se han escrito las
25 #       palabras SIN acentos ni caracteres 'conflictivos'
```

Existe un séptimo script de *R* que lleva en su nombre también el prefijo `preparar_`, que se llama `preparar_codprovincias.R`; éste contiene a la función que se encarga de obtener uno de los conjuntos de datos adicionales, en concreto, con información de los códigos de provincias.

El resto de scripts de *R* corresponden a fases posteriores del diseño y se explicarán en los lugares de esta memoria donde correspondan más adelante.

3.8. Especificación de requisitos

Esta fase es conceptualmente paralela a la anterior de «Análisis y reconciliación de fuentes de datos», pues una no depende de la otra.

La fase de análisis de requisitos, tal y como se estudió en la asignatura **FIS** (*Fundamentos de la Ingeniería del Software*) [17], aporta al encargado del proyecto (en este caso, yo) una visión más o menos específica (según el análisis) de qué es lo que el usuario quiere para su proyecto.

La forma de realizar el análisis de requisitos, es decir, su estructura, se realiza a partir de lo que se estudió en aquella asignatura. Cabe destacar que la literatura sobre sistemas multidimensionales no se hallan modelos específicos o más recomendables para elaborar este tipo de sistemas, y debido al carácter «generalista» del Análisis de requisitos que se estudió en esta asignatura, considero que nos puede reflejar de forma fidedigna los requerimiento del proyecto en ciernes.

Además, cabe destacar que esta descripción se ha realizado más o menos de forma coordinada entre el tutor y el autor de este trabajo. La descripción es la siguiente:

REQUISITOS FUNCIONALES

- RF1. Estudio gráfico de los datos. El sistema debe permitir generar diferentes tipos de gráficos, tablas, etc (a través de una herramienta externa).

- RF2. Estudio «numérico» de los datos. El sistema deber permitir una visualización flexible de los datos en cuanto a selección del punto de vista, nivel de detalle y variables (mediciones) a mostrar.

  - RF2.1. El sistema permite definir el punto de vista o dimensión (o varios) por las que se visualizan los datos.
  - RF2.2. El sistema permite definir el nivel de detalle en que se visualizan los datos en cada punto de vista (dimensión) seleccionado.
  - RF2.3. El sistema permite definir las variables a mostrar de los datos (hospitalizados, ingresados en UCI, Confirmados por PDIA, etc.)

- RF3. Cuadros de mando. Se debe mostrar al menos un cuadro de mando (a modo de ejemplo) que albergue gráficos interactivos, tablas, etc sobre alguno de los cubos OLAP desarrollados, usando una herramienta de consultas externa.

REQUISITOS DE INFORMACIÓN

RI	1
Descripción	Información sanitaria del coronavirus en base a varios indicadores a nivel de municipio (toda la información posible).
Contenido	RI1.1 Fecha de la notificación. RI1.2 Municipio donde se hace la notificación de casos. RI1.3. Todos los indicadores COVID que puedan obtener directamente de los datos (confirmados, fallecidos, curados, etc.).

RI	2
Descripción	Información sanitaria del coronavirus sobre residencias (toda la información posible)
Contenido	RI2.1. Fecha de notificación. RI2.2. Lugar donde se ha producido, al nivel de detalle más bajo posible, así como franjas de edad, o tipo de residencia donde se produce el contagio, según lo permitan los datos. RI2.3. Todos los indicadores COVID-19 que se puedan obtener directamente de los datos.

RI	3
Descripción	Información sanitaria del coronavirus sobre grupos de profesionales de riesgo (toda la información posible)
Contenido	RI2.1. Fecha de notificación. RI2.2. Lugar donde se ha producido, al nivel de detalle más bajo posible, así como clasificación de grupos profesionales y todo lo que permitan los datos. RI2.3. Todos los indicadores COVID-19 que se puedan obtener directamente de los datos.

RI	4
Descripción	Información sanitaria del coronavirus sobre hospitalizaciones, UCI, etc al nivel de detalle más bajo posible (toda la información posible)
Contenido	RI2.1. Fecha de notificación. RI2.2. Lugar donde se ha producido, al nivel de detalle más bajo posible, y otros puntos de vista que aporten los datos. RI2.3. Todos los indicadores COVID-19 que se puedan obtener directamente de los datos.

RI	5
Descripción	Información sobre la vacunación contra el COVID19 en Andalucía[14]
Contenido	RI2.1. Fecha de notificación. RI2.2. Lugar donde se ha inyectado la dosis, al nivel de detalle más bajo posible y rango de edad. RI2.3. Indicadores sobre número y porcentaje de población con pauta completa e incompleta.

REQUISITOS NO FUNCIONALES

Facilidad de uso

- RNF1. La interfaz de usuario para la realización de informes y consul-

tas será la herramienta Power BI.

- RNF2. Hay dos tipos de usuarios de este sistema: los técnicos, que gestionarán el lado del servidor del sistema (ejecución de scripts de R, volcado/actualización de datos, etc), y los usuarios «finales», que solo manejarán herramientas *cliente* para conectarse con la base de datos multidimensional y hacer gráficos, tablas, etc.
- RNF2. Documentación: se debe proporcionar un manual para el usuario técnico sobre cómo es la carpeta de trabajo que se le entrega, cómo instalar el software necesario para hacer funcionar la aplicación, etc. Para el usuario final, no es necesario, pues dependerá del cliente que use para conectarse al sistema.

Rendimiento

- RNF3. Se deberá hacer una estimación de cuánto espacio (aproximado) puede llegar a ocupar el almacenamiento de los datos[15]

Soporte

- RNF4. El sistema estará disponible en el idioma oficial del Estado, el castellano.

3.9. Diseño conceptual aplicando la metodología de Kimball

Seguimos las indicaciones dadas en el apartado 3.5 para el diseño conceptual, organizado en cinco fases.

3.9.1. Selección de los procesos de negocio a modelar

En base a los requisitos de información que tenemos y a los datos que tenemos disponibles, podemos establecer los siguientes focos de atención, también llamados *hechos* (los siguientes ítems y los dos párrafos que les siguen son una justificación de cómo los hechos seleccionados satisfacen los requisitos y los datos disponibles, **puede obviarse**, pues la definición de cada hecho se dará en el siguiente paso, e ir directamente a la figura 3.11 para verlos).

[15]Esto queda satisfecho, como ya se expuso en el apartado 3.5, en una de las fases del Diseño Conceptual cuando aplicamos la Metodología de Kimball.

1. Datos de días naturales, que satisfacen lo solicitado en el RI4, que pedía datos sobre hospitalizados, UCI y cifras similares. Debido a que el conjunto de datos ofrece también otros indicadores, también los añadiremos.

2. Una serie de 8 conjuntos de datos por provincias y a nivel de municipio, que satisface el RI1, y que permite por tanto dar los datos a niveles superiores mediante la agrupación (Distrito Sanitario, Provincia y toda Andalucía). Incluiremos todos los indicadores que ofrece este conjunto de datos cumpliendo lo especificado por el requisito de «toda la información e indicadores posibles».

3. Datos a nivel de residencias: en este caso encontramos dos conjuntos de datos de residencias, que tienen información y puntos de vista distintos (punto de vista geográfico y por tipo de residencias; y punto de vista de rango de edad, sexo y tipo de residencia, con lo cual los dos nos aportan información nueva, y por tanto los elegimos como dos nuevos focos de atención.

4. Datos sobre profesionales de riesgo: encontramos un conjunto de datos que satisface el RI3, por tanto creamos a partir de él un nuevo foco de atención.

5. Existe un conjunto de datos concreto que ofrece la información solicitada sobre vacunas en el RI5. Lo tomamos como un nuevo foco de atención.

Si intentamos fusionar algunos de los conjuntos de datos, vemos que no es posible en general ya que, bien porque sus puntos de vista son incompatibles (como en el caso de los dos conjuntos de residencias), o bien sí que la tienen pero está a distinto nivel de detalle (por ejemplo, que algunos conjuntos de datos tienen información sobre municipios y otros por distrito sanitario, y si tomamos el conjunto que está a nivel de municipio y obtenemos los datos a nivel de distrito sanitario, estaremos perdiendo la información de municipios, y eso no es correcto).

Además, si nos fijamos, se trata de temáticas que están muy bien diferenciadas entre sí, y conceptualmente consideramos como diseñadores que somos del sistema (estamos en la fase de diseño conceptual) que no es oportuno mezclar temáticas dentro del mismo cubo, y aprovechamos que **podemos tener, desde el punto de vista de la coherencia del sistema, tantos cubos como creamos conveniente**.

Por tanto, tenemos un total de seis focos de atención, que mostramos en la figura 3.11.

Figura 3.11: Focos de atención (hechos) seleccionados. Fuente: elaboración propia.

3.9.2. Establecer la granularidad del proceso de negocio

Se trata de definir el significado de cada foco de atención y con él, las bases de los mismos por separado.

- «**Municipios**»: cada línea de hechos representa la «notificación COVID19 en un municipio en un día (hábil) concreto»

  Esto es así porque si queremos acceder a una determinada instancia/-notificación COVID/fila del conjunto de datos, no podemos decirle solo el nombre del *municipio* (ni tampoco solo la *fecha*), porque nos daría todas las notificaciones COVID de ese municipio, y queremos **solo una** (como una clave primaria), queremos identificar una a una, y eso se consigue dando también la **fecha** de notificación: en ese caso, siempre nos devolverá una única fila, pues **no hay dos filas que tengan el mismo municipio y la misma fecha a la vez, esta es la clave**. Por tanto, **las Bases de** «**Municipios**» son **Fecha y Municipio**.

- «**Días naturales**»: cada línea de hechos representa la «notificación hospitalaria COVID19 en una provincia en un día (natural) concreto»

  Estamos en una situación muy similar a la anterior: si queremos acceder a una única fila/instancia/notificación COVID de este foco de atención no podemos indicarle solo la Provincia, pues nos daría todas las notificaciones que tiene guardadas para esa provincia; tampoco le

podemos indicar solo la fecha, pues entonces lo que estaríamos haciendo es acceder a la notificación de datos de ese día de cada una de las ocho provincias, algo que no queremos. En conclusión, se observa que la **Provincia** y la **Fecha** de notificación **identifican** cada instancia de este foco de atención, por tanto son las **bases** en este caso.

- «**Vacunación**»: cada línea de hechos representa el «informe de vacunación en una provincia en un rango de edad en un día (hábil) concreto»

  Este caso amplía los dos anteriores: ahora, si queremos acceder a cada fila/instancia de este foco, no solo tenemos que indicar la *Provincia* y la *Fecha*, sino también el *Rango de edad* que queremos consultar, de lo contrario nos daría las notificaciones (instancias del foco de atención) asociadas a esa Provincia y esa Fecha, pero para todos los rangos de edad, es decir, que nos daría tantas filas como rangos de edad hay (ocho, inicialmente). En definitiva, las **bases** ahora son: **Provincia, Fecha** y **Rango de edad** de la persona vacunada.

- «**Residencias**»: cada línea de hechos representa el «informe COVID de residencias en un Distrito Sanitario, un tipo de residencia y en un día (hábil) concreto»

  Este caso es similar al anterior, solo que en lugar de tener rango de edad, tenemos el tipo de residencia donde se hace la notificación, y en lugar de tener las Provincias, tenemos Distritos Sanitarios (el nivel de organización sanitaria inmediatamente inferior a la provincia en Andalucía), lo cual no afecta a que sea una base del foco. Las bases por tanto son **Distrito Sanitario**, **Fecha** de la notificación y **Tipo de residencia**.

- «**Residencias Edad-Sexo**»: cada línea de hechos representa el «informe COVID en residencias según tipo de residente (edad, sexo, tipo de residencia) y en un día (hábil) concreto»[16]

  En este caso, y siguiendo el mismo proceso de razonamiento que hasta ahora, las **bases** son: **Fecha**, **Tipo de residencia**, **Rango de edad** y **Sexo** del residente.

- «**Profesionales**»: cada línea de hechos representa el «informe COVID en profesionales según tipo de profesional (categoría profesional y sexo) y en un día (hábil) concreto»[17]

  Ahora las **bases** son: **Sexo**, **Rango de edad** y **Fecha**.

[16] Hemos agrupados varias columnas en el mismo *punto de vista* porque todos ellos definen a la persona y todos los valores de dichas columnas pueden relacionarse entre sí

[17] Igual que el caso anterior de fusión de varias columnas en un mismo *punto de vista*.

La figura 3.12 aporta una primera visión de los puntos de vista (de los que surgirán próximamente las *dimensiones*) posibles sobre cada uno de los focos de atención (hechos). En ellos ya se puede observar que se han añadido algunos atributos dentro de algunos puntos de vista que no estaban explícitamente en los conjuntos de datos, pero que se han añadido porque se pueden deducir del atributo que sí aparece. Esto sucede, por ejemplo, en el punto de vista *Cuándo*, donde solo tenemos la fecha en los conjuntos de datos, pero de la fecha se puede sacar fácilmente el mes, la semana y el año asociado a esa fecha, por eso los hemos añadido.

3.9.3. Diseñar las dimensiones

Para diseñar las dimensiones tomamos cada foco de atención y podemos ir viendo las distintas perspectivas desde las que podemos estudiarlo, en base a las columnas que tenemos. Además, hay que tener en cuenta que las «bases» del foco de atención tienen que ser también puntos de vista a tener en cuenta.

- Para el foco «Municipios», los datos se pueden estudiar desde la perspectiva temporal (la Fecha -día hábil-) y la perspectiva geográfica (el nombre del Municipio). Estos dos puntos de vista responden respectivamente a las preguntas *Cuándo* y *Dónde*, por tanto, podemos definir de momento estas dos dimensiones.

  Pero estas dimensiones por ahora tienen solo un nivel, una altura, que es *Fecha* y *Municipio*. Podemos ampliar las dimensiones de la siguiente forma:

  - Para la dimensión Cuándo, podemos determinar, con un paquete de *R* llamado `lubridate`, varias cosas sobre la fecha: el año, el mes (tanto el nombre como el número) y la semana del año en que está contenida esta fecha.

    La fecha guarda una relación parte-todo exclusiva con el Mes y con la Semana del año (una fecha concreta solo está en una semana y un mes concreto).

    Pero es que además tanto el *Mes* como la *Semana del año* solo pueden están «dentro» del Año, y no por encima ni al mismo nivel.

    Dicho de otra forma, y para concluir, el Año contiene a los meses y a las semanas, y a su vez ambos dos contienen a las fechas.

    La estructura de la dimensión *Cuándo* queda entonces como en la figura 3.13, donde vemos dos jerarquías, bajando por el Mes (la llamamos «Jerarquía Mes») y bajando por la Semana del año (la llamamos «Semana del año»).

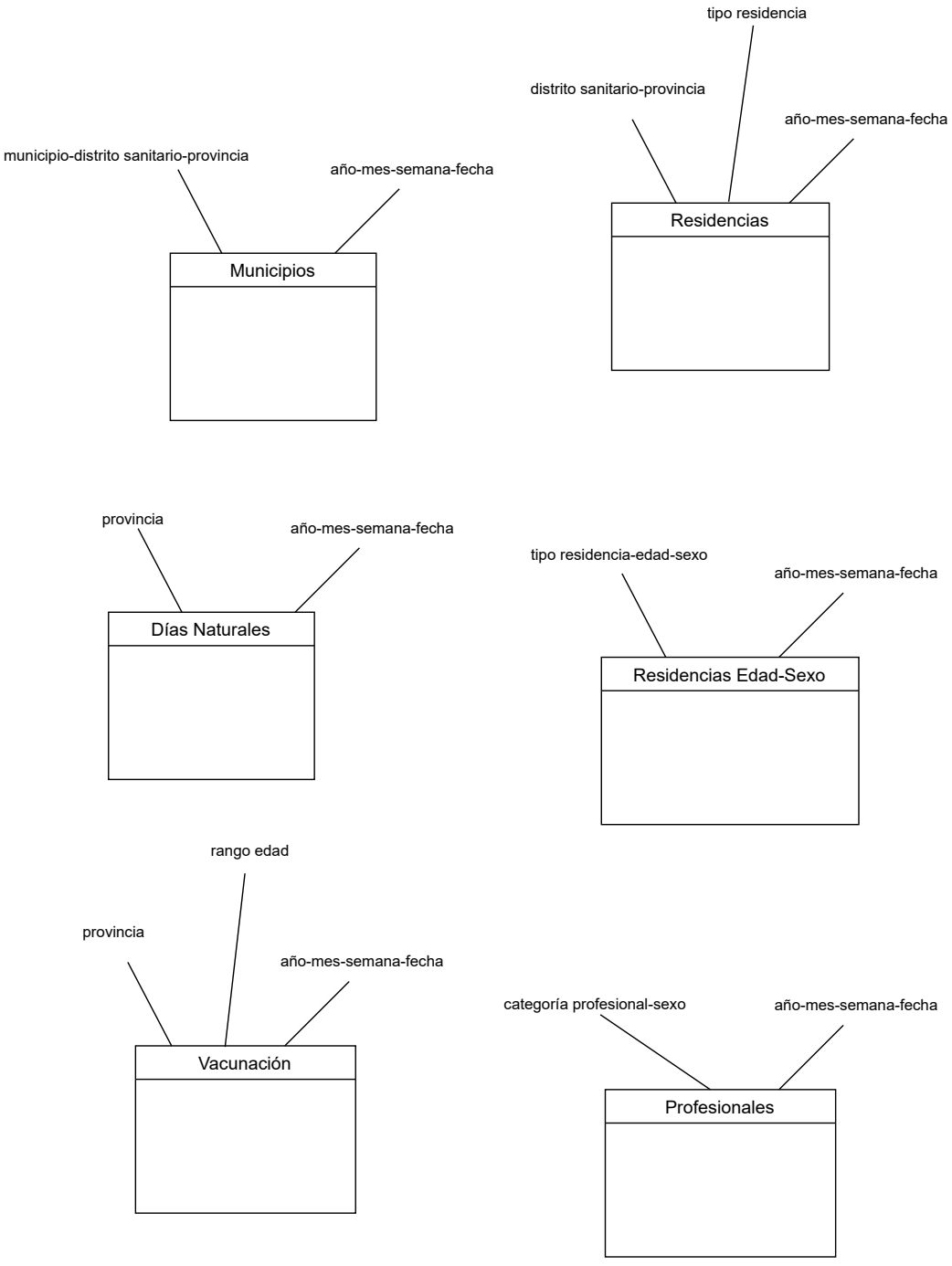


Figura 3.12: Descripción de granularidad de los hechos. Fuente: elaboración propia.

- Para la dimensión *Dónde*, una vez que tenemos los Municipios, podemos obtener también los Distritos Sanitarios por agrupación de datos, y también el nivel de agrupación superior, en que se agrupan los Distritos Sanitarios, que son la Provincia (las ocho provincias andaluzas).

  Tal y como se puede deducir fácilmente, se ve que el Municipio está contenido en el Distrito Sanitario (relación parte-todo exclusiva), a su vez el Distrito Sanitario está contenido en la Provincia y la Provincia está dentro de Andalucía (nivel «Todo» de la jerarquía). La dimensión por tanto tiene una única jerarquía, que podemos llamar «Jerarquía Municipio».

  La situación de la dimensión se muestra en la figura 3.13, donde se aprecia que el nombre de la dimensión no es *Dónde*, sino *Dónde - Municipio*, a continuación se explicará por qué.

- Foco de atención «Días naturales» en este caso tenemos también dos puntos de vista parecidos a los anteriores:

  - el punto de vista de la Provincia (responde a la pregunta *Dónde*) y el de la fecha de notificación (responde al *Cuándo*). Podríamos pensar en reutilizar las dimensiones anteriores pero si nos damos cuenta:

    1. No tenemos información a nivel de municipio, sino a nivel de provincia, por lo que no podemos aprovechar la dimensión anterior, y definimos una dimensión *Dónde - Provincia*, indicando que el nivel de detalle más bajo posible es la Provincia y no podemos obtener más información adicional. La dimensión queda como en la figura 3.13.
    2. En cuanto a la Fecha, de nuevo podemos pensar en la dimensión Cuándo, y aprovechar la que tenemos. El problema es el siguiente: las instancias de la dimensión *Cuándo* del foco de atención anterior eran de fechas de días hábiles (de Lunes a viernes salvo festivos) y ahora tenemos días naturales (todos los días del año, sin excepción), con lo cual para poder almacenar podemos añadir un atributo al nivel Fecha (lo que se denomina un *descriptor de nivel*) que indique si se trata de un día hábil (TRUE/FALSE) y otro atributo que indique que es un día natural (TRUE/FALSE). Aunque realmente el segundo atributo no es necesario, se deja para que queden explícitamente diferenciados cuáles son días naturales y cuáles son solo días hábiles. La dimensión, por tanto, mantiene su estructura original pero añadiendo al nivel Fecha esos dos descriptores, *Es día hábil* y *Es día natural*.

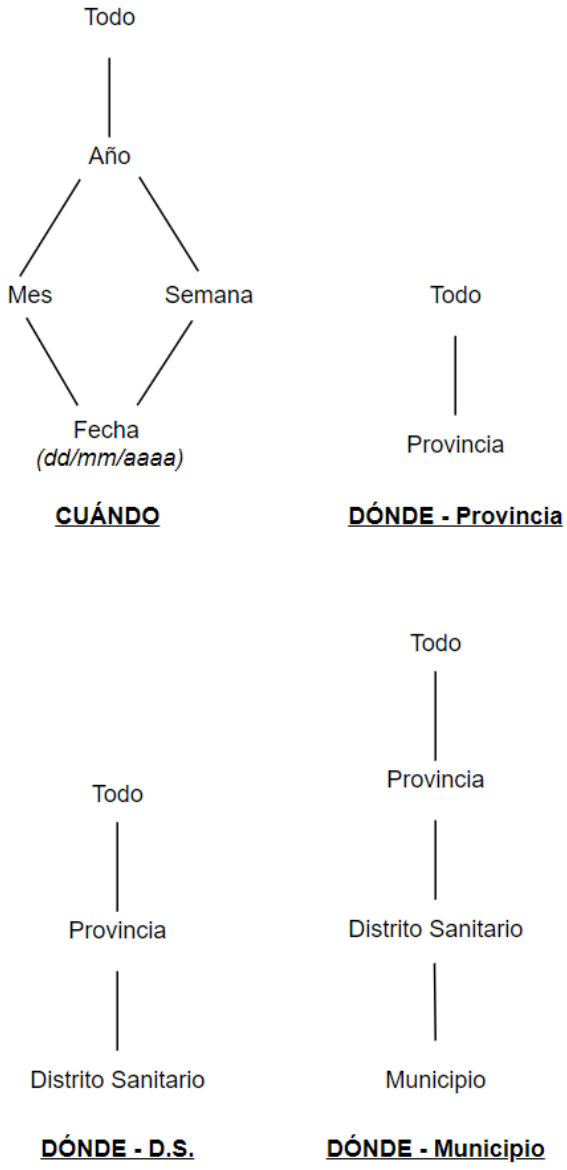


Figura 3.13: Diseño de las dimensiones (1). Fuente: elaboración propia.

- Para el foco de atención «Residencias», tenemos que la Fecha es de nuevo de días hábiles (recordemos que el único foco de atención con fecha de días naturales es el anterior[18]). Por tanto, usamos aquí también la dimensión *Cuándo* tal y como está.

  Los otros dos puntos de vista en este foco son el distrito Sanitario y el tipo de residencia. Siguiendo un razonamiento similar a los anteriores, definimos una nueva dimensión *Dónde - Distristo Sanitario* (porque no podemos aprovechar las otras dimensiones tipo *Dónde* ya definidas, por estar a distinto nivel de detalle) y otra dimensión que se llame *Residencia*. La figura 3.13 muestra la dimensión *Dónde - Distrito Sanitario (D.S.)* y la figura 3.14 hace lo oportuno con la dimensión *Residencia*.

- Foco de atención «Residencias Edad-Sexo»: en este caso de nuevo tenemos el punto de vista de la fecha por día hábil, luego podemos usar aquí también la dimensión *Cuándo*.

  Luego tenemos una serie de puntos de vista que son el *Tipo de residencia*, la *Edad* y el *Sexo* de los residentes. Si nos fijamos, todos estos campos dan información sobre la persona, tienen esa propiedad, ya que uno nos dice en qué tipo de residencia vive, el otro dice su sexo y el último el rango de edad al que pertenece, por tanto, lo que podemos hacer es definir una dimensión que englobe a todos estos atributos, para evitar tener así tres dimensiones muy pequeñas, aprovechando que guardan esa relación, en témrinos conceptuales.

  Pero para fusionar estos tres campos en una dimensión tenemos que ver su cardinalidad: en la tabla de datos podemos ver que **no existe relación de dependencia entre ellos**, es decir, todos se relacionan con el otro de forma N:N, tal que tendremos una jerarquía con tres niveles a la misma altura (justo encima tendremos simplemente el nivel «Todo», que siempre está en las dimensiones.

  La dimensión se podrá llamar *Quién - Residencias* (pues da información sobre **personas** en residencias). Se muestra en la figura 3.14. En esta figura aparece un nivel más bajo como «combinación» de los niveles superiores. Este nivel existe implícitamente en todas las dimensiones en el Diseño Conceptual (pero no repercute en fases posteriores), para garantizar que todas las jerarquías acaban (abajo) en un nivel común.

- En el foco de atención «Profesionales» tenemos solo dos puntos de vista: la Fecha de notificación y la Categoría Profesional.

  Pues entonces usaremos la dimensión *Cuándo* y una nueva, que llamaremos *Quién - Profesionales*.

[18] Repasar sección 3.7.2.1.

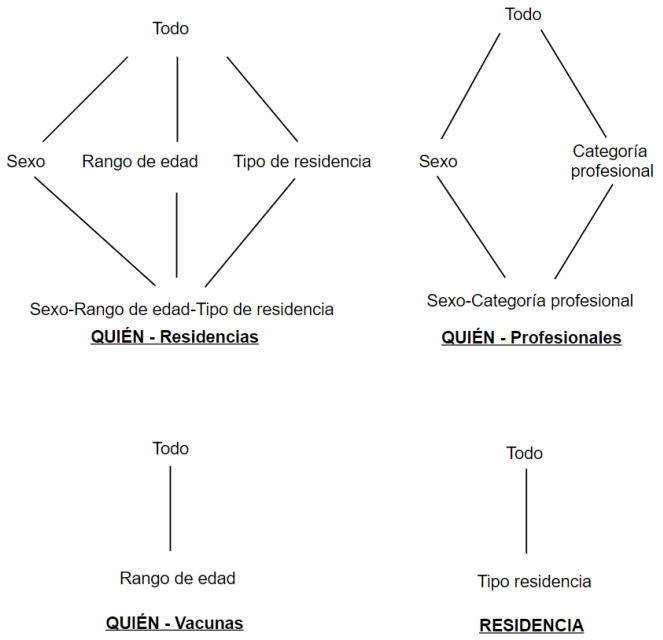


Figura 3.14: Diseño de las dimensiones (2). Fuente: elaboración propia.

La dimensión *Quién - Profesionales* tendrá (en principio) dos niveles (*Sexo* y *Categoría profesional*) que pueden estar a la misma altura porque guardan una relación N:N mutua (no hay relación parte-todo exclusiva entre ellos).

En la altura más baja de la dimensión, como en el caso de la dimensión *Quién - Residencias*, añadiremos el nivel «combinación» de los superiores (es decir, *Sexo-Categoría profesional*). La nueva dimensión se presenta en la figura 3.14.

- Y en el último foco de atención, «Vacunación», tendremos:

  La dimensión *Cuándo*, asociada, como ya es preceptivo, al punto de vista *Fecha*.

  La dimensión *Dónde - Provincia*, referenciando al punto de vista *Provincia* (ya que aquí no tenemos los datos a menor nivel de detalle que este).

  Una nueva dimensión, que podemos llamar *Quién - Vacunas* que englobe así al punto de vista «Rango de Edad», y que contendrá precisamente los rangos de edad de los grupos de vacunación de este conjunto de datos. La dimensión se representa en la figura 3.14.

3.9.4. Definición de las mediciones

En este caso, las mediciones de cada foco de atención coinciden exactamente con todos los indicadores que aparecen en los conjuntos de datos asociados, sin eliminar ninguno, tal y como se solicitaba en los requisitos de información del sistema.

Vamos a clasificar todos los indicadores que tenemos en los focos de atención de cara a estudiar su aditividad:

- *Mediciones aditivas*: son aquellas que expresan datos que tiene sentido sumarlos una vez se hace una agregación (se sube por la jerarquía). Es el caso de las cifras de Confirmados PDIA, Total Confirmados, Fallecidos, Curados, Hospitalizados, UCI, Núm. de personas con la pauta (completa o incompleta) de la vacuna.

- *Mediciones no aditivas*: aquellas que no tiene sentido que las sumemos por ninguna de las dimensiones del cubo (foco de atención): son la Tasa PDIA 14 días y los porcentajes de vacunados con pauta completa/incompleta.

- *Mediciones semiaditivas*: es el caso de la Población (nº de habitantes) de un municipio, que tiene sentido sumarla por la dimensión *Dónde*, y así se sabe la población del Distrito Sanitario de ese municipio e incluso de la provincia conforme se va ascendiendo por la jerarquía. Pero no tiene sentido sumar esta medición por la dimensión *Cuándo*, por ejemplo, pues no tiene sentido sumar el valor de la población de un municipio día a día.

  También el valor «Confirmados PDIA de los 7/14 últimos días» tiene sentido sumarla, pero al ser un valor ya de por sí acumulado de varios días (7 o 14), solo podemos sumarla por la dimensión geográfica, pues así tendríamos el número acumulado de casos a lo largo de un Distrito Sanitario, Provincia, etc.

Sobre la inclusión de **mediciones calculadas**, no consideramos ninguna medida que pueda resultar relevante en base a la información que tenemos. Los indicadores que ofrece el Instituto de Estadística y Cartografía de Andalucía en nuestros datos pueden ser suficientes para este proyecto.

3.9.5. Estimación del número de instancias

El objetivo es realizar una estimación de cuantas instancias (filas) van a ocupar todas las tablas de los hechos y las dimensiones (las fuentes de datos originales no se contemplan aquí, son de fases previas).

El cálculo es el siguiente:

Dimensiones

CUÁNDO: 1 instancia por cada día del año, desde el 26/02/2020. Esta dimensión tendrá una instancia (un registro) nueva cada día.

DÓNDE-Provincia: 1 instancia por cada provincia (8): **8** instancias. Esta dimensión no cambia.

DÓNDE-Distrito Sanitario: 1 instancia por cada Distrito Sanitario andaluz. En Andalucía hay 34 Distritos Sanitarios: **34** instancias. Esta dimensión no cambia.

DÓNDE-Municipio: 1 instancia por cada término municipal de Andalucía: 785 municipios + 8 «municipios sin especificar» = **793** instancias. Esta dimensión no cambia.

QUIÉN: 2 tipos de residencias, 9 rangos de edad y 2 sexos, con relación N:N entre ellos: nº instancias = 2 x 9 x 2 = **36** instancias. Esta dimensión no cambia.

QUIÉN-Vacunas:

→ Hasta el día 16/06/2021 había 8 rangos: **8 instancias**

→ Desde el día 17/06/2021 hay 13 rangos, que incluye a los 8 anteriores: **13 instancias**.

QUIÉN-Profesionales:

→ Hasta el día 28/06/2021: 2 sexos y 10 categorías profesionales, con relación N:N entre ellos: Nº instancias = 2 x 10 = **20** instancias.

→ Desde el día 29/06/2021: 2 sexos y 9 categorías profesionales, con relación N:N entre ellos: Nº instancias = 2 x 9 = **18** instancias.

RESIDENCIA: 2 tipos de residencias, por tanto: **2** instancias.

Focos de atención («hechos»)

MUNICIPIOS: Cada día hábil se añaden: 1 fecha x 793 municipios = **+793 instancias/día**.

DÍAS NATURALES: Cada día se añaden: 1 fecha x 8 provincias = **+8 instancias/día**.

RESIDENCIAS: Cada día hábil se añaden: 1 fecha x 34 distritos sanitarios x 2 tipos de residencias = **+68 intancias/día**.

RESIDENCIAS EDAD-SEXO: Cada día hábil se añaden: 1 fecha x 2 tipos de residencias x 9 rangos de edad x 2 sexos = **+36 instancias/día**.

PROFESIONALES:

→ Hasta el día 28/06/2021: cada día hábil se añadían: 1 fecha x 10 categorías profesionales x 2 sexos = **+20 instancias/día**.

→ Desde el día 29/06/2021: cada día hábil se añaden: 1 fecha x 9 categorías profesionales x 2 sexos = **+18 instancias/día**.

VACUNAS

→ Hasta el día 16/06/2021: cada día hábil se añadían: 1 fecha x 8 provincias x 8 rangos de edad = **+64 instancias/día**.

→ Desde el día 17/06/2021: cada día hábil se añaden: 1 fecha x 8 provincias x 9 rangos de edad = **+72 instancias/día**.

3.10. Diseño lógico ROLAP en estrella

En la fase de diseño lógico vamos a tomar el ya comentado modelo «ROLAP en estrella» que consiste en la siguiente heurística:

Para cada dimensión → 1 tabla relacional

Para cada foco de atención → 1 tabla relacional

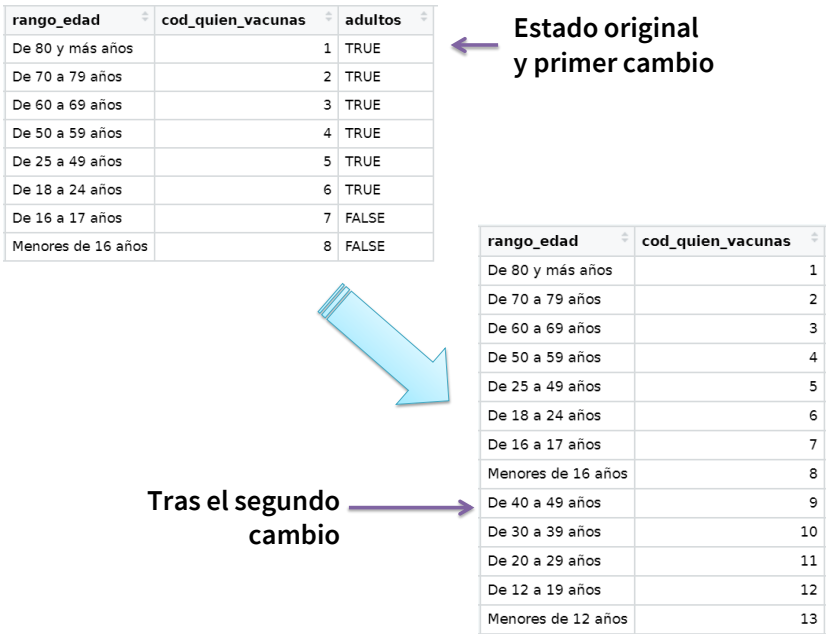
Las claves primarias de las tablas correspondientes a cada dimensión serán una llave generada que empiece en 1 (por convenio seguido en la literatura [16]) y las llaves primarias de las tablas de hechos serán la combinación de **llaves externas** que referencien a las llaves primarias de las tablas de las dimensiones asociadas a ese foco de atención.

Además, para nombrar los nombres de las columnas se seguirá, siguiendo de nuevo lo aconsejado en la literatura, el criterio `camel_case` con el alfabeto inglés, para evitar errores al usar las tablas en algún punto del sistema.

3.10.1. Dimensiones Lentamente Cambiantes

Una **SCD** (*Slowly Changing Dimension*), o **Dimensión Lentamente cambiante** en español, es una dimensión cuyas instancias cambian con el tiempo.

Lo que nos permiten las SCD es poder seguir manteniendo la coherencia en las dimensiones, al máximo posible, a pesar de los cambios que se puedan producir en ellas.



rango_edad	cod_quien_vacunas	adultos
De 80 y más años	1	TRUE
De 70 a 79 años	2	TRUE
De 60 a 69 años	3	TRUE
De 50 a 59 años	4	TRUE
De 25 a 49 años	5	TRUE
De 18 a 24 años	6	TRUE
De 16 a 17 años	7	FALSE
Menores de 16 años	8	FALSE

rango_edad	cod_quien_vacunas
De 80 y más años	1
De 70 a 79 años	2
De 60 a 69 años	3
De 50 a 59 años	4
De 25 a 49 años	5
De 18 a 24 años	6
De 16 a 17 años	7
Menores de 16 años	8
De 40 a 49 años	9
De 30 a 39 años	10
De 20 a 29 años	11
De 12 a 19 años	12
Menores de 12 años	13

Figura 3.15: Cambios en la dimensión *Quién Vacunas*. Fuente: elaboración propia.

En nuestro caso, nos hemos encontrado con que la autoridad de los datos los ha modificado de una determinada forma que nos obliga a hacer uso de esta técnica.

En nuestro caso, han cambiado dos dimensiones:

- En la dimensión *Quién-Vacunas*, se han modificado dos veces los rangos de edad, pasando de tener 8 rangos inicialmente a tener 9 y finalmente, el último cambio alteró algunos de esos 9 últimos reemplazándolos por otros rangos diferentes. La figura 3.16 muestra el estado final tras las tres «versiones» a modo de evolución progresiva.
- En la dimensión *Quién-Profesionales* han eliminado una de las categorías profesionales.

Para afrontar las dimensiones lentamente cambiantes, hemos tenido que hacer lo siguiente:

- Dimensión *Quién-Vacunas*: al producirse el primer cambio, que consistía en «partir en dos» uno de los rangos originales, se modificó el código *R* añadiendo una nueva «versión» con las instrucciones modificadas para que se tomen los dos nuevos rangos y se fusionen (se sumen

Figura 3.16: Forma final de la dimensión *Quién Vacunas* tras aplicar las técnicas SCD. Fuente: elaboración propia.

los valores por filas) para mantener el rango anterior. Se trata de una propuesta sencilla pero que no obliga a modificar la dimensión, y ésta **no se veía alterada en ninguna de sus propiedades o valores de sus niveles**.

Pero con la segunda alteración de los rangos de edad esta sencilla estrategia ya no resultaba válida, y se optó por aplicar una nueva: añadir los nuevos rangos junto con los anteriores; de esta forma, seguimos manteniendo correctamente los datos anteriores y a la vez podemos seguir usando los datos nuevos que descarguemos a partir de ahora, con lo que conseguimos mantener actualizado el foco de atención «Vacunas».

Esto se materializa en una serie de tareas en *R*, que pasaban por modificar el fichero `preparar_datos_vacunacion_basicos.R`, para que se distinguiera una nueva «versión» (la tercera) de los datos y al llamar a este script (la función que contiene) para procesar diariamente los datos de vacunación desde el script principal de descarga (`descargar_datasets.R`), no hubiera errores de procesamiento.

Cabe destacar además que este cambio ha modificado esta dimensión, pues ya no podemos tener el nivel «Adultos», ya que tenemos un rango de edad que abarca mayores y menores, luego no podemos hacer esa distinción. Esto hace que ahora la dimensión tenga un solo nivel, el «Rango de edad» y que quede como en la figura 3.16. Asimismo, la tabla relacional correspondiente a esta dimensión tendrá el mismo aspecto que antes, pero ahora con los rangos nuevos añadidos, y sin la columna `adultos`.

- Dimensión *Quién-Profesionales*: este caso consiste en que la categoría profesional «Atención al público» dejaba de aparecer en los datos des-

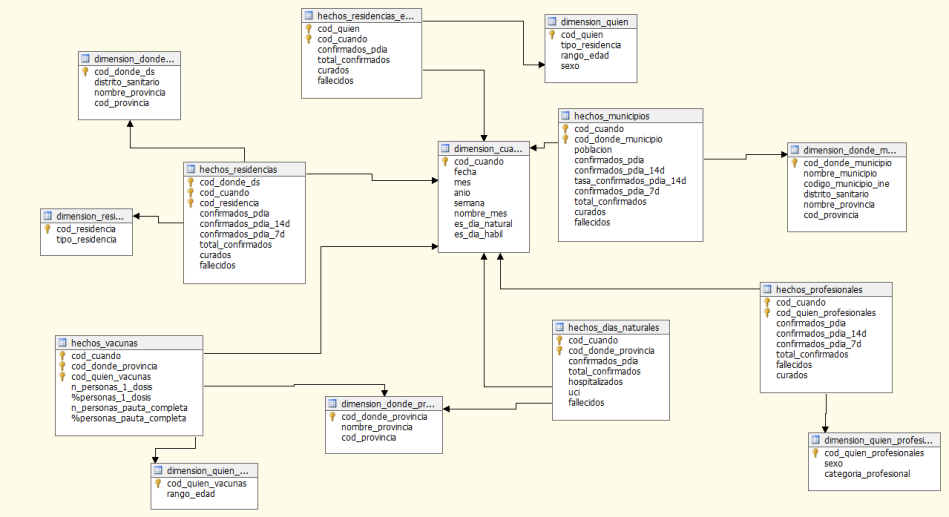


Figura 3.17: Diseño lógico. Fuente: elaboración propia.

cargados. Esto nos obligaba a hacer solamente una modificación del código *R* de cara a manejar correctamente el conjunto de datos (ahora había dos filas menos, la de «Sexo: Hombre, Categoría Prof.: Atención al público» y la de «Sexo: Mujer, Categoría Prof.: Atención al público» y se ha añadido una nueva versión con algunas instrucciones nuevas adaptadas al nuevo conjunto de datos). **La dimensión no se ve modificada** porque los datos antiguos sobre profesionales sí tenían este valor y por tanto no podemos eliminarlo de la dimensión; lo que ocurrirá es que simplemente en la tabla de hechos no tendremos más instancias (filas) asociadas a esta Categoría Profesional.

Pues bien, una vez resuelto el problema de las dimensiones lentamente cambiantes, presentamos ya en la figura 3.17 el diseño lógico para nuestra base de datos relacional, que servirá de base para construir la futura base de datos multidimensional (cubos OLAP), el paso final del desarrollo del sistema.

3.10.2. Tareas en *R* para esta fase

Aparte de las labores ya comentadas para resolver el problema de las *SCD*, lo que falta por comentar es qué scripts de `R` nos han ayudado a generar los hechos y las dimensiones. Dentro de la carpeta de trabajo, en el directorio `proyecto_tfg`, y junto al resto de ficheros de *R*, tenemos:

- `generar_dimensiones.R` es un script con un conjunto de ocho funciones que tras ejecutarse generan el fichero CSV dentro del directorio

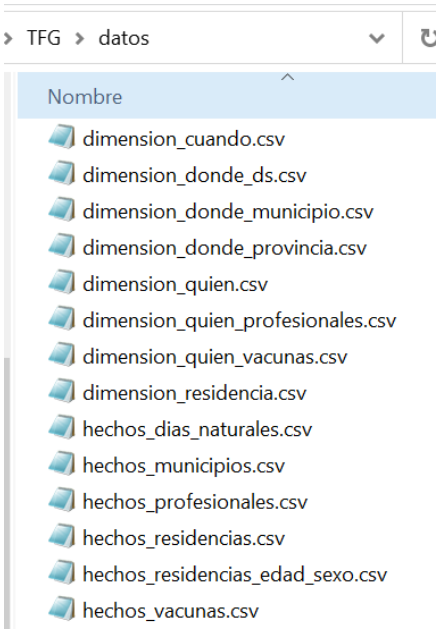


Figura 3.18: Ficheros CSV del directorio `datos/` con todas las tablas del diseño lógico. Fuente: elaboración propia.

`datos/` almacenando la dimensión correspondiente (o la actualizan, si es que ya existe).

- `generar_hechos.R` es un script que contiene seis funciones para generar un fichero CSV dentro del directorio `datos/`, conteniendo cada fichero una de las seis tablas de hechos diseñadas.

La figura 3.18 muestra todos los ficheros con las tablas generadas del diseño lógico dentro del directorio `datos/` de nuestra carpeta de trabajo (`TFG`). Como podemos comprobar en la figura, tenemos exactamente $8+6=14$ *ficheros*, que se corresponden con las 14 tablas generadas.

3.10.3. Tareas en *SQL Server Management Studio*

Una vez tenemos las tablas de los hechos y las dimensiones creadas, tenemos que ir al SGBD de *SQL Server*, *SQL Server Management Studio* (SSMS), que será el que nos permita crear la base de datos en el servidor y poder administrarla.

Un resumen de los pasos que hemos tenido que dar es el siguiente:

1. Abrir la herramienta *SSMS*, conectarse al servidor por defecto «localhost»

y crear la base de datos relacional, que llamamos `covid19and` (se siguen los pasos de la sección 2.1 de [14]).

2. Importar las tablas a partir de los ficheros CSV, usando el asistente de importación de *SSMS*, y siguiendo las indicaciones dadas en el Capítulo 8, sección 8.2.1 de [23]. Esta referencia es a un libro del tutor de este trabajo, y presenta los pasos explícitos para importar un CSV, cosa que no aparece en el guión [14] que estamos siguiendo hasta ahora.

   Estos «paquetes de importación» de los que hablamos son tremendamente útiles. Se ha tomado una nomenclatura para llamar a estos paquetes y así poder entender su cometido de una forma simple:

   - Los paquetes que importen la tabla de una dimensión tendrán como nombre:

     `importar_dimension_<nombre_de_la_dimension>.dtsx`.

   - Análogamente, los que importen una tabla de hechos se llamarán

     `importar_hechos_<nombre_de_los_hechos>.dtsx`.

Tras importar las tablas podemos ir viéndolas mediante consultas tipo `SELECT` de SQL y comprobar que la importación, más allá de que no haya dado errores en el asistente, haya cargado los datos como nosotros queremos.

3.10.4. Tareas en *SQL Server Data Tools* (*SSDT*)

El software *SQL Server Data Tools* es una herramienta de Microsoft que nos permitirá completar: (1) la fase final de ETL: la carga de datos, mediante los paquetes de importación y (2) crear la base de datos multidimensional (cubos OLAP).

3.10.4.1. Creación del proyecto de *SSIS* para la carga de datos en las tablas de la Base de Datos

Seguimos los pasos dados en el guión (sección 3.1 de [14]) para **crear el proyecto de *SSIS*** (que vamos a llamar `covid19and_etl`) e **incluir en él los paquetes de importación** generados desde *SSMS*.

Una vez que tengamos eso listo, podremos ejecutar esos paquetes para ir actualizando los datos de cada tabla, teniendo en cuenta además lo que destacamos en el Recuadro 3.1.

Recuadro 3.1 *Restablecer claves primarias tras ejecutar paquetes de importación*

Cada vez que se ejecuten los paquetes y se carguen las filas en cada tabla de la Base de Datos, es necesario que establezcamos la clave primaria a cada tabla. Para ello, basta con hacer click-derecho sobre el nombre de cada tabla, ir a «Design» y allí fijar la columna `cod_<nombre_tabla>` como clave primaria, en el caso de las dimensiones; pero en el caso de los hechos, recordemos que la clave primaria es el conjunto de claves externas que referencian a las claves primarias de las dimensiones involucradas en ese cubo, por lo que serán éstas las que habrá que marcar como clave primaria en conjunto.

Observación 2 *Recordemos del Capítulo 2 el concepto de transformación como una tarea que se ejecuta* ***encauzada*** *con otras de cara a obtener los datos en un determinado formato.*

Los datos se transformaban para generar las tablas de hechos o dimensiones con el formato que queríamos.

Pero, ¿quién ha creado estas transformaciones? ¿Quién las ha metido dentro de estos paquetes? Pues lo hicimos nosotros, cuando usamos el asistente de importación de tablas en SSMS: el software anotaba las opciones que íbamos seleccionando en el asistente y las volcaba en forma de transformaciones en el paquete resultante (es más, el paquete, si lo abrimos con un editor de texto plano, no es más que un fichero enorme con un montón de parámetros de configuración, que no nos interesa analizar).

3.10.4.2. Creación del proyecto *SSAS* para generar la base de datos multidimensional (cubos OLAP)

Y llegamos al paso final: la creación de la **base de datos multidimensional**. Para ello hay que crear un proyecto de «Analysis Services» en este mismo software, el cual englobará la definición de todas las dimensiones y de todos los cubos, así como sus propiedades, sus componentes y las propiedades de éstos últimos.

De nuevo hay que crear un proyecto de SSAS (*Analysis Services*, el componente OLAP de SQL Server) en SSDT, siguiendo el apartado 3.1 de [15]. El proyecto se puede llamar, siguiendo en la línea de la nomenclatura de elementos previos, `covid19and_olap`.

A partir de aquí se suceden algunas tareas necesarias para poder acceder a los datos correctamente y tenerlos preparados para usarlos, como una fase previa a la configuración de las dimensiones y los hechos. Estos pasos han

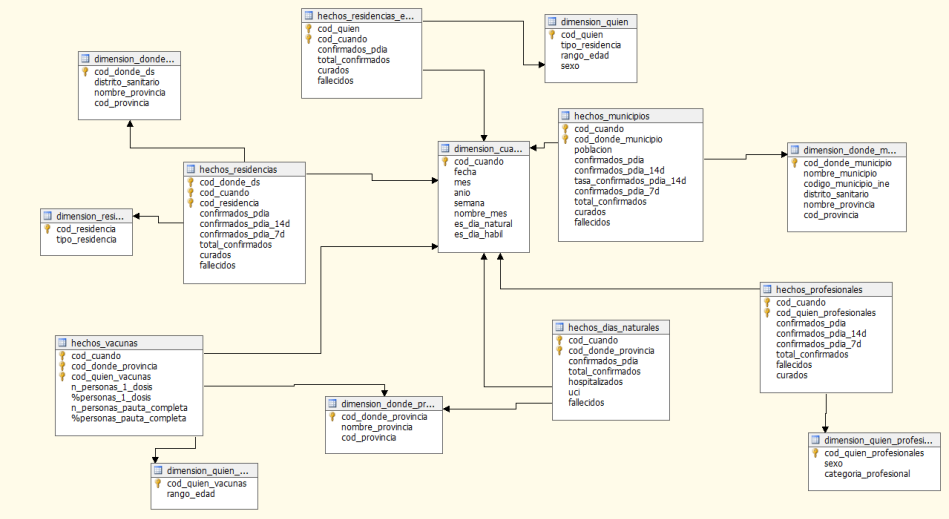


Figura 3.19: Perspectiva del diseño lógico desde la vista creada en el proyecto de SSAS (*Analysis Services*). Fuente: elaboración propia.

sido:

1. Configurar la conexión a los datos, siguiendo la sección 3.3.1 del guión [15].

2. Establecer el usuario de suplantación siguiendo exactamente lo que se indica en la sección 3.3.2 del guión [15].

Una vez realizados estos dos pasos, el usuario que se conecta a la base de datos multidimensional que vamos a crear tendrá permiso «de lectura» sobre la base de datos relacional, que es donde realmente están almacenados los datos, pudiendo así realizar consultas.

Ya podemos empezar a definir las dimensiones y los cubos. En primer lugar, tenemos que crear lo que se conoce como «Data Source View», que es una vista, una representación de la base de datos relacional dentro de nuestro proyecto SSAS, para acceder a ella más cómodamente. Hemos seguido los pasos exactos indicados en la sección 3.4 del guión, y como resultado, en la figura 3.19 tenemos una perspectiva de todas las tablas de la base de datos relacional, con sus atributos, sus claves primarias y las llaves externas referenciando donde corresponden.

Una vez configurada la vista y habiendo comprobado que las tablas están correctamente creadas, vamos a crear las dimensiones.

◁ **DIMENSIONES** ▷ Para crear las dimensiones, seguimos la sección 3.5 del guión [15]. Debido a la especial importancia que tienen, vamos a mostrar a continuación cómo hemos creado una de ellas, la dimensión **Dónde**

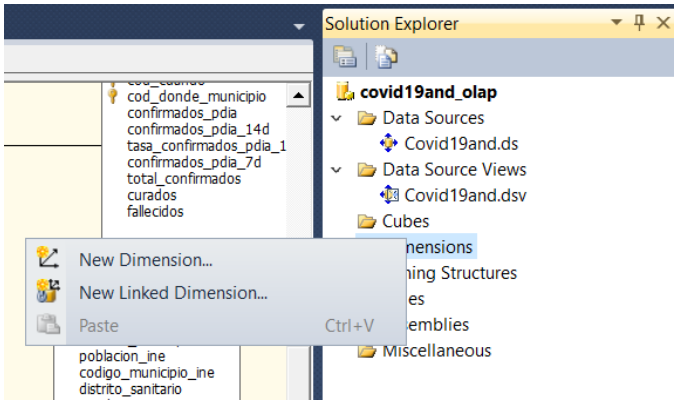


Figura 3.20: Definiendo una dimensión (1). Fuente: elaboración propia.

- Provincia, mientras que del resto daremos los detalles particulares sobre el proceso de creación para evitar sobrecargar este documento con procedimientos e imágenes prácticamente iguales.

1. Abrir el asistente de definición de dimensiones (ver figura 3.20).

2. En la pantalla siguiente, indicamos que **vamos a crear la dimensión a partir de una tabla individualizada para ella**, y no vamos a extraer los datos de otra.

3. Pasamos de pantalla (figura 3.21) y seleccionamos la tabla de la Base de Datos asociada a la dimensión que vamos a definir, que se llamaba (`dimensión_donde_provincia`) y vemos justo debajo que el asistente ha detectado la clave primaria de la tabla. Comprobamos que sea correcta (aunque lo será, porque en la figura 3.19 ya vimos que la herramienta detectó todas las llaves y relaciones correctamente) y pulsamos en `Next`.

4. Este paso es importante: tenemos que seleccionar qué atributos queremos que se usen (en nuestro caso, todos) y tenemos que activar la opción `Enable Browsing` para permitir que se usen los atributos al construir las dimensiones.

   Adaptamos los nombres de los atributos para que sean comprensibles, pues serán los nombres que aparecerán en los informes.

   El resultado se muestra en la figura 3.22.

5. La siguiente pantalla es un resumen del proceso, donde tenemos que definir el nombre de la dimensión: *Dónde - Provincia*, en nuestro caso. Pulsamos en *Finish* y se cierra el asistente y al volver a la ventana principal del prorama vemos que al proyecto se ha añadido un nuevo

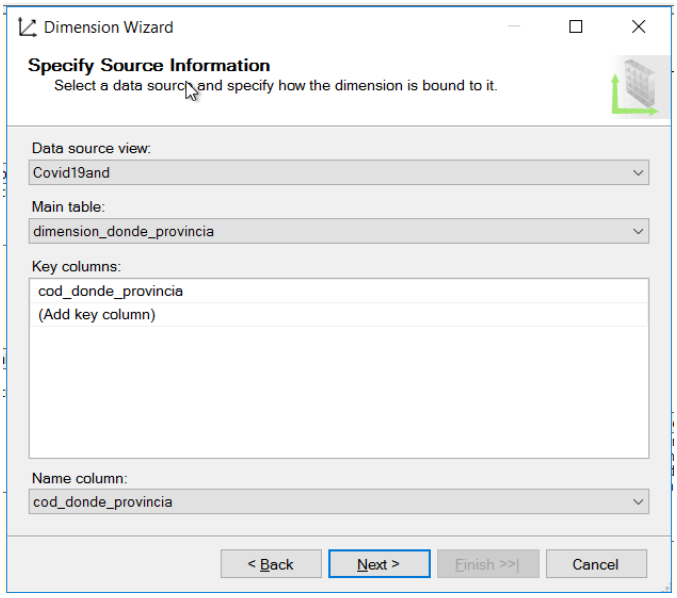


Figura 3.21: Definiendo una dimensión (2). Fuente: elaboración propia.

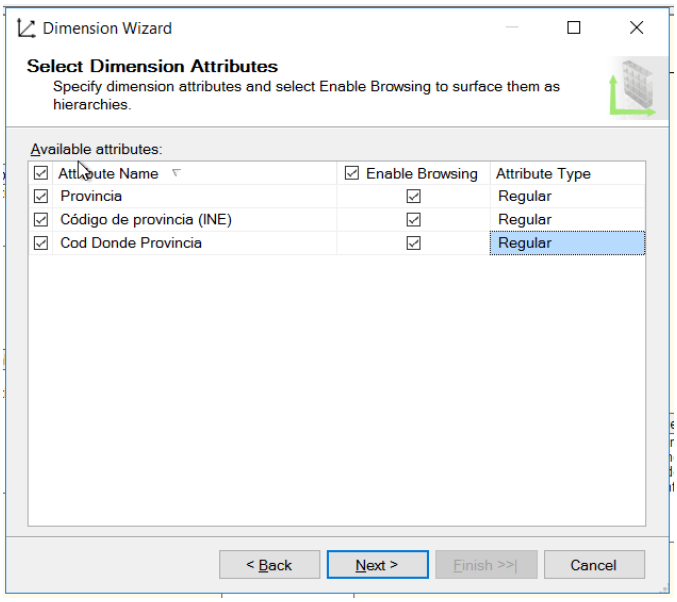


Figura 3.22: Definiendo una dimensión (3). Fuente: elaboración propia.

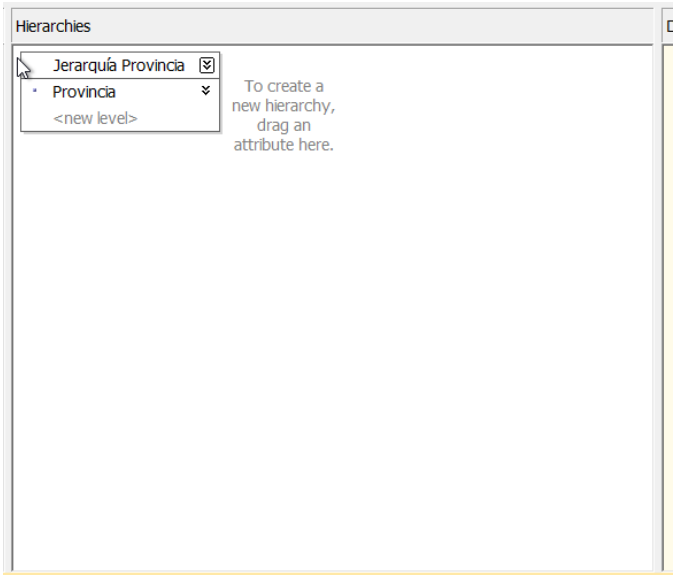


Figura 3.23: Definiendo una jerarquía. Fuente: elaboración propia.

elemento, *Dónde - Provincia.dim*, que es el objeto que representa a la dimensión.

6. Ahora hay que configurar la dimensión, para poder definir los niveles y jerarquías y sus propiedades.

   Para definir jerarquías, teniendo abierta la dimensión en la ventana principal (basta hacer doble-click sobre la dimensión para que se abra), nos vamos a la zona de definición de jerarquías («Hierarchies») y ahí definimos la única jerarquía de esta dimensión «pulsando-arrastrando y soltando» el único nivel que hay de la lista de atributos de la izquierda. Se puede cambiar el nombre a la dimensión. La jerarquía queda como en la figura 3.23.

   A continuación, para completar la definición de la dimensión, tenemos que ir a la pestaña «Attribute Relationships» para establecer las relaciones (1:N, N:N, etc.) entre los niveles de toda la dimensión. Esto lo hemos realizado siguiendo las indicaciones de la sección 3.5.1 del guión [15], adaptándolo. En concreto, el atributo «Código de provincia INE» de nuestra dimensión se debe establecer como un *descriptor de nivel* del nivel *Provincia*.

   Dentro de la pestaña «Atributte Relationships» tendremos la situación mostrada en la figura 3.24.

   Finalmente, una vez definida la dimensión, la procesamos tal y como se indicaba en el guión y comprobamos que se ha generado con los datos estructurados correctamente (ver figura 3.25).

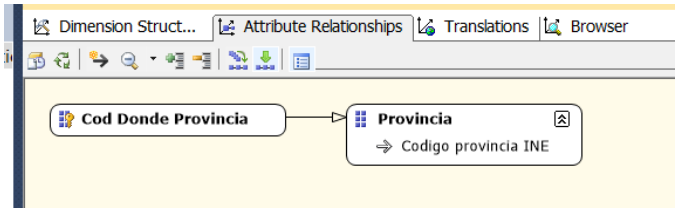


Figura 3.24: Definiendo relaciones entre niveles de la dimensión. Fuente: elaboración propia.

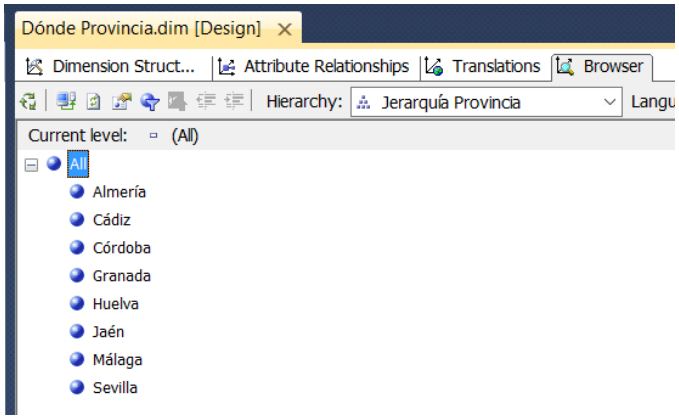


Figura 3.25: Estado de los datos de la dimensión. Fuente: elaboración propia.

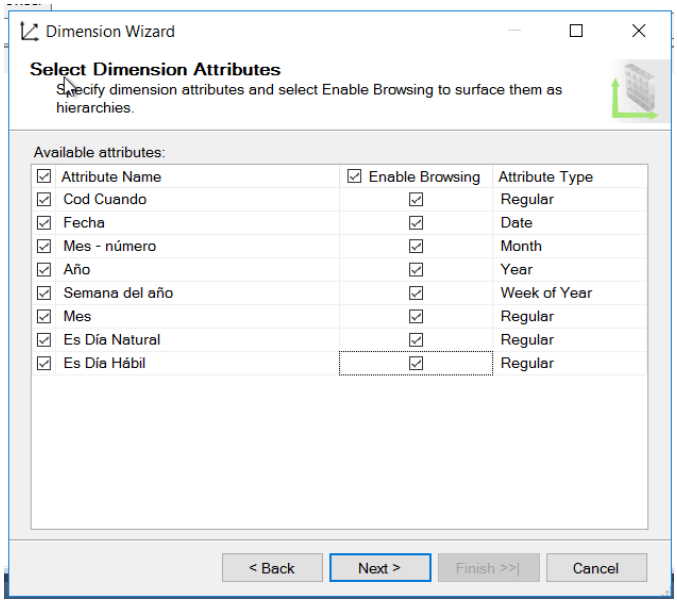


Figura 3.26: Especificación del tipo de atributos (niveles) de la dimensión *Cuándo*. Fuente: elaboración propia.

Siguiendo este procedimiento hemos definido el resto de las dimesiones, aunque con algunas particularidades, que es importante dejar claras:

- En el caso de las dimensiones *Dónde - Municipio* y *Dónde - Distrito Sanitario*, como en *Dónde - Provincia*, que acabamos de definir, la *Provincia* y el *Municipio* tienen su propio código identificativo procedente del INE; por tanto, esos códigos (columnas) se han añadido como descriptores de nivel (provincia o municipio, donde corresponda) como ya hemos hecho para *Provincia* en la dimensión *Dónde - Provincia*.

- El caso de la dimensión *Cuándo* es algo particular porque cuando estamos importando la tabla, y seleccionando y renombrando los atributos, tenemos que fijarnos en la columna «Attribute Type» (recordar en figura 3.22), y vemos que todos los atributos por defecto tienen tipo `Regular`. Tal como indica el guión [15], es conveniente cambiar los tipos de datos para que el software identifique que se trata de atributos temporales (una fecha, un año, un número de mes, etc) para poder darles ese «trato especial» de cara a optimizar consultas hipotéticas en lenguaje MDX (entre otros) que se puedan hacer contra el sistema.

  En defintiva, el tipo de los atributos quedó definido como en la figura 3.26.

  Otro aspecto a tener en cuenta es que, recordando la estructura de la dimensión *Cuándo*, los niveles *Semana del año* y *Mes* están justo en el

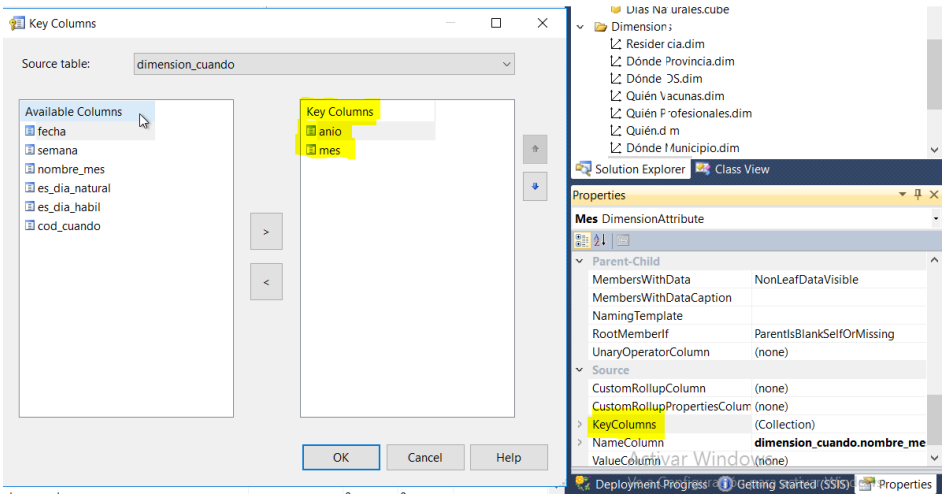


Figura 3.27: Definición de la propiedad «KeyColumn» para el nivel *Mes*. Fuente: elaboración propia.

nivel inferior al *Año*, lo que implica que los meses del año no se pueden repetir para dos años diferentes (y lo mismo con las semanas del año). Pero sabemos que los meses y las semanas del año son todos los años iguales, por tanto, conceptualmente, se están violando las relaciones:

$\rightarrow$ Año $1 : N$ SemanaDelAño y

$\rightarrow$ Año $1 : N$ Mes.

El software es consciente de esta particularidad de la dimensión *Cuándo* y ofrece, para casos como este, una propiedad que permite vicular un atributo (*Mes* y *Semana del año*) a su «atributo clave», de tal forma que se soluciona el problema; esto se hace incluyendo el *Año* dentro del valor de la propiedad «KeyColumns» de *Mes* y de *Semana del año*. Para que esto quede claro, mostramos el valor de dicha propiedad para los dos niveles referidos en las figuras 3.27 y 3.28.

Finalmente, hay que resaltar que se ha definido el atributo *Mes - número* (es decir, los meses del año en números naturales) como un *descriptor* del nivel *Mes*, ya que su razón de ser en origen es la de enriquecer con más información la dimensión en la que está.

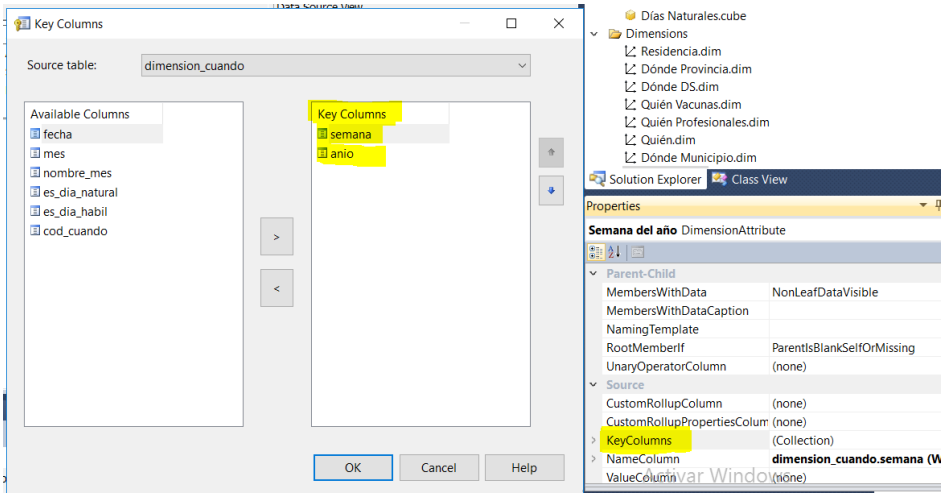


Figura 3.28: Definición de la propiedad «KeyColumn» para el nivel *Semana del año*. Fuente: elaboración propia.

Recuadro 3.2 *Ocultar las claves primarias*

Las claves primarias de las tablas no sirven para nada en los informes. Si al importar una dimensión con el asistente, se dejan marcados todos los campos, incluido el atributo que es clave primaria, tendremos que ocultarlo más tarde. El proceso es sencillo:

1. Una vez tenemos abierta la dimensión en *SSDT*, vamos al cuadro de «Attributes», seleccionamos con click-derecho el atributo que es clave primaria y pulsamos en «Properties».
2. En el recuadro de «Properties» del atributo (abajo a la derecha de la ventana) buscamos la propiedad `AttributeHierarchyVisible` y lo ponemos a `False`.
3. Guardamos el proyecto y procesamos la dimensión desde la pestaña «Browser» → «Process», como es habitual.
4. Haremos esto con cada dimensión. Las claves primarias de los cubos las descarta automáticamente el asistente de importación, por lo que no tenemos que preocuparnos en ese caso.

◁ **CUBOS** ▷ Y llegamos al paso final. Para definir los cubos se han seguido las indicaciones, muy breves, de la sección 3.6 del guión [15].

Una vez que hemos creado los seis cubos, a partir de las seis tablas de hechos, tendremos, **para cada cubo**, una serie de mediciones que el asistente ha importado y las dimensiones vinculadas a ese cubo.

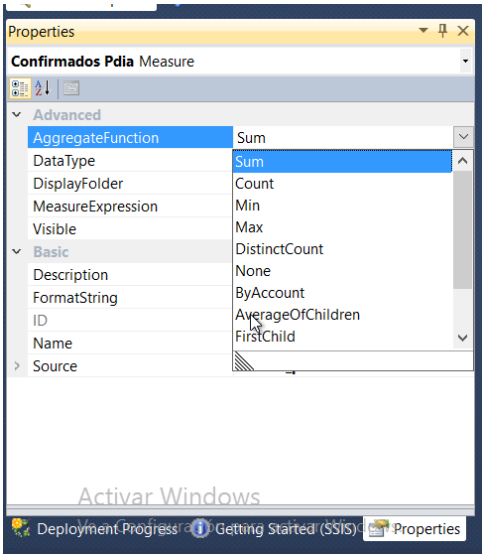


Figura 3.29: Establecer la aditividad de una medición de un cubo. Fuente: elaboración propia.

Lo que nos falta es **establecer la aditividad** o no-aditividad **de las mediciones** y en caso de que sí haya aditividad para una medición, seleccionar la función de agregación (suma, media, mínimo, ...) adecuada. La aditividad de las mediciones ya quedó establecida en el diseño conceptual. Para fijarla, se accede a las propiedades de la medición a configurar y en la propiedad «Aggregate Funcion» fijamos la función de agregación que queramos (`Sum`, `Count`, `Min`, ...); si la medición es *no-aditiva*, fijaremos el valor `None` en este campo. Podemos ver un ejemplo para la medición «Confirmados PDIA» en la figura 3.29.

Una vez tenemos definidos todos los cubos, podemos hacer las pruebas que queramos de forma **independiente** a cada uno desde la pestaña «Browser» para las comprobaciones que queramos.

3.10.4.3. Construir el proyecto y desplegarlo para que esté accesible por las herramientas cliente de consulta

En el panel «Solution Explorer» de SSDT, hacemos click-derecho sobre el nombre de nuestro proyecto y damos a «Build» para (re)constrir el proyecto y después a «Deploy» para desplegarlo. Si todo ha ido bien, tenemos la base de datos multidimensional, con los seis cubos habilitados para recibir consultas desde cualquier herramienta cliente que se conecte de forma válida a ella.

3.11. Elaboración de un cuadro de mando desde un cliente: *Power BI*

Power BI es una herramienta de usuario final que permite conexión a datos almancenados en orígenes muy diversos. Uno de esos orígenes admitidos es una Base de Datos de Analysis Services. Cuando se conecta a ella, detecta automáticamente toda la definición de hechos y dimensiones (y todas sus propiedades), tal que el usuario solo tiene que elaborar las consultas que quiera (gráficos, tablas, consultas MDX, etc).

Aprovechando esta funcionalidad, se elabora aquí un «cuadro de mando» a modo de ejemplo de cara a mostrar cuál podría ser un ejemplo de uso del sistema que hemos diseñado.

Un cuadro de mando es una herramienta del mundo empresarial que contiene varios objetos visuales (gráficos, mapas, tablas, etc) dentro de un mismo «marco de visualización», con la idea de dar una perspectiva rápida de la información que se pretende mostrar.

El cuadro de mando es elaborado siguiendo las indicaciones dadas en [24].

Un cuadro de mando en Power BI puede estar compuesto de varias pantallas. Se pueden añadir «objetos visuales» (gráficos, tablas, botones, etc).

3.11.1. Conexión a la Base de Datos de *Analysis Services*

Para que Power BI, que va a actuar como cliente de consultas sobre nuestro Sistema Multidimensional, pueda conectarse a la Base de Datos Multidimensional, tenemos que seguir unos pasos muy sencillos:

1. Tras abrir *Power BI Desktop*, en la ventana principal, en barra de herramientas de «Inicio» seleccionamos «Obtener datos» → «Analysis Services».

2. Nos saldrá una nueva ventana como la de la figura 3.30. Rellenamos los mismos datos que ahí aparecen y seleccionamos «Conectarse en directo» y pulsamos «Aceptar».

3. Si todo ha ido bien, tiene que mostrar una nueva ventana como la de la figura 3.31, donde se muestren los seis cubos de nuestro sistema. Seleccionamos el que queramos y damos en «Cargar». Ahora ya podremos hacer los informes, gráficos, cuadros de mando, etc que queramos sobre estos datos.

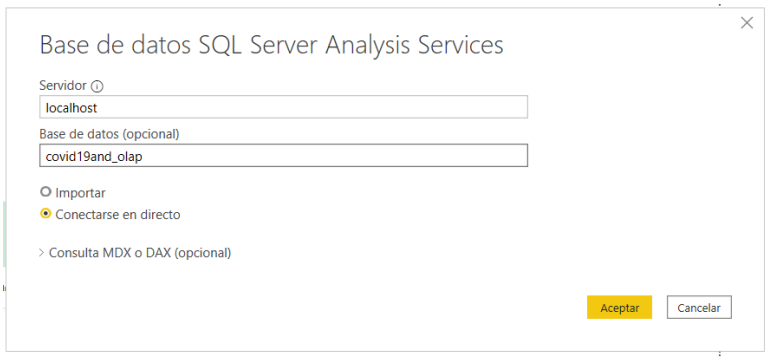


Figura 3.30: Configurando la conexión a Power BI. Fuente: elaboración propia.

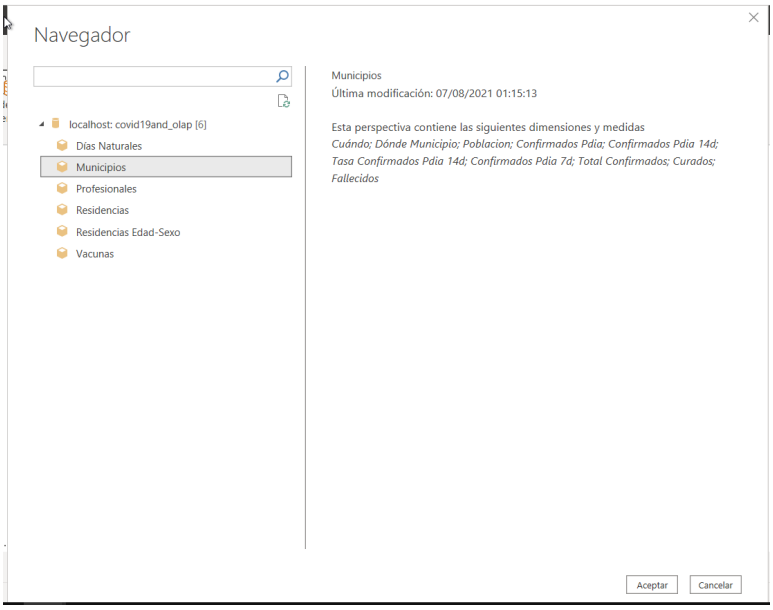


Figura 3.31: Visualizando los cubos disponibles. Fuente: elaboración propia.

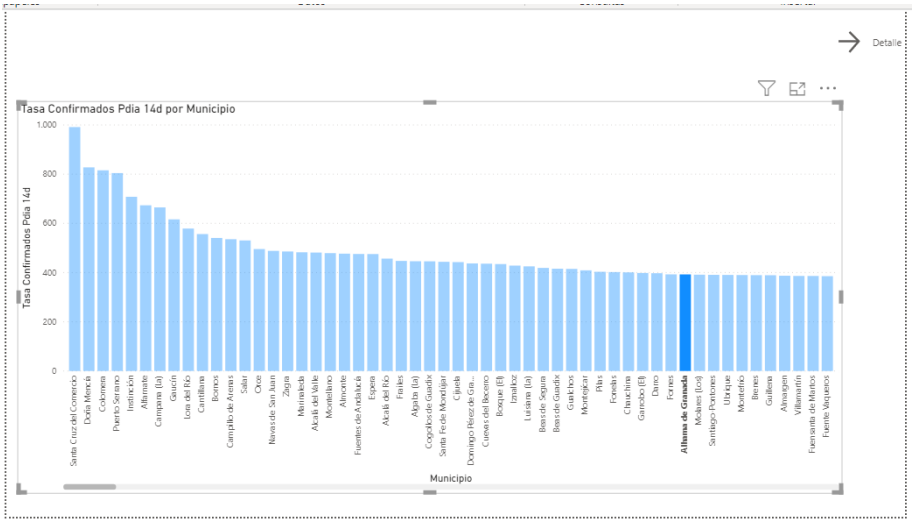


Figura 3.32: Visualizando cuadro de mando (1). Fuente: elaboración propia.

3.11.2. Descripción del cuadro de mando elaborado

Nuestro cuando de mando de ejemplo se compone de tres pantallas secuenciales, por las que nos podemos desplazar cómodamente mediante botones (← y →) colocados a tal efecto. Se ha realizado sobre el cubo «Municipios» y su contenido es:

Pantalla 1 Gráfico de barras con la Tasa de contagios PDIA de 14 días para cada municipio de Andalucía.

Pantalla 2 Hay tres «tarjetas»[19] junto con una lista de los municipios andaluces, de tal forma que cuando se selecciona un municipio (basta con hacer 1 click sobre él) se actualizarán los indicadores de estas tarjetas.

Pantalla 3 muestra la evolución en el período que hemos registrado datos (desde el inicio de este proyecto) de los fallecidos y los confirmados por PDIA.

Las figuras 3.32, 3.33 y 3.34 muestran las pantallas descritas.

Con una cuenta profesional o educativa previamente adquirida, sería posible publicar este cuadro de mando de las diversas formas que el producto ofrece. Para esto vamos a la barra «Inicio» → «Publicar».

[19]Cada tarjeta en Power BI muestra un valor numérico (por ejemplo, el número de fallecidos o de curados)

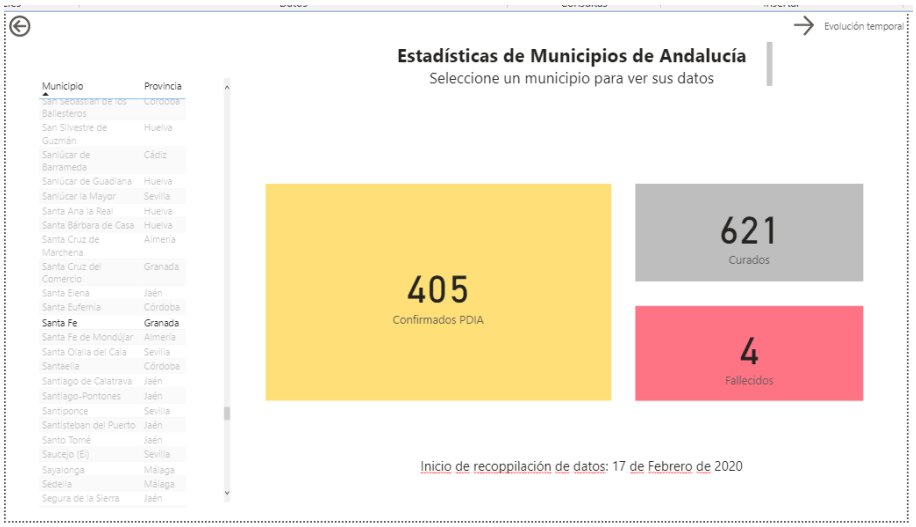


Figura 3.33: Visualizando cuadro de mando (2). Fuente: elaboración propia.

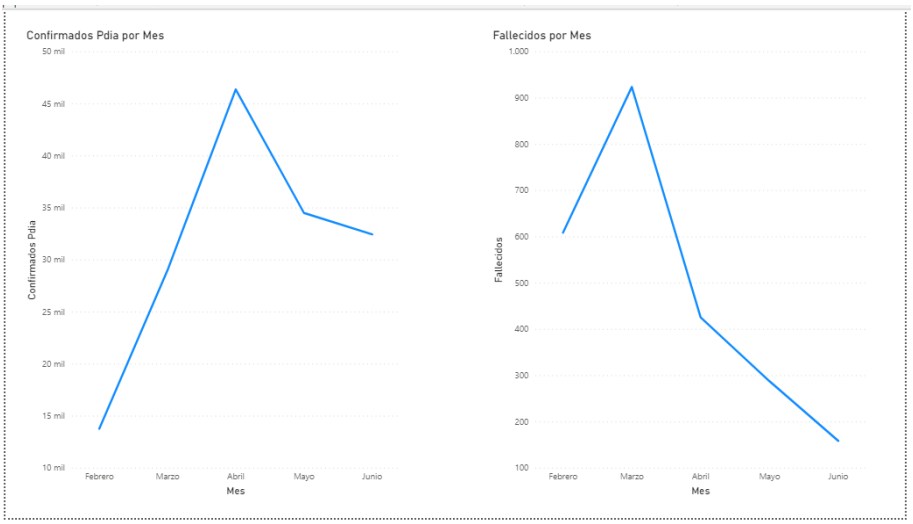


Figura 3.34: Visualizando cuadro de mando (3). Fuente: elaboración propia.

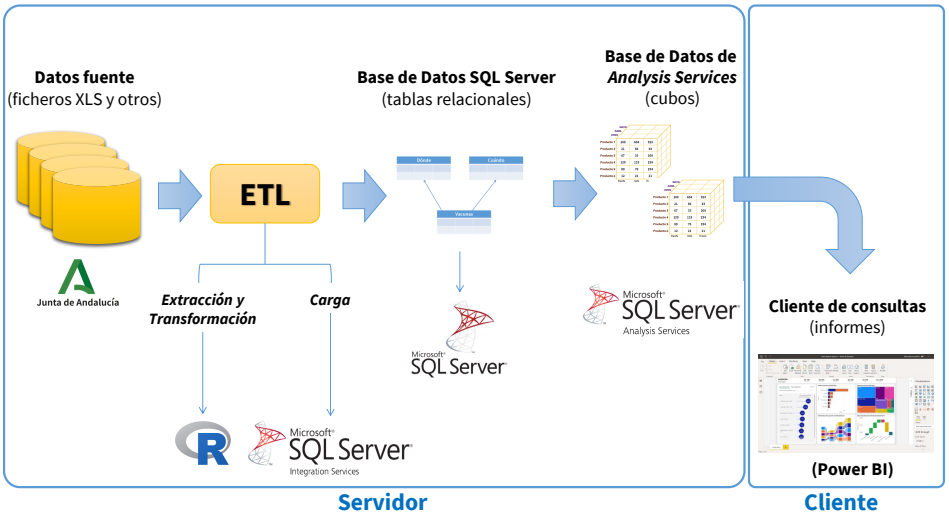


Figura 3.35: Perspectiva general (arquitectura) del sistema. Fuente: elaboración propia.

3.12. Perspectiva general del sistema

Para finalizar, se muestra en la figura 3.35 una visión de la arquitectura «intuitiva» de nuestro sistema.

En ella se destaca la separación entre:

- El lado del **servidor**, con las fuentes de datos, los procedimientos ETL (en *R* y *SSIS*), la Base de Datos *SQL Server* y la Base de Datos Multidimensional (en *SSAS*).
- El lado del **cliente**, con la herramienta de consulta escogida en cada caso (Microsoft Excel, Power BI, ...), accediendo a la Base de Datos Multidimensional.

Capítulo 4

Conclusiones y trabajo futuro

4.1. Conclusiones tras el desarrollo del proyecto

Podemos concluir afirmando que con este proyecto se ha consolidado la adquisición de los conocimientos y las competencias teórico-prácticas impartidos en el Grado.

Esto queda demostrado con la aplicación de conocimientos de asignaturas de programación (a la hora de llevar a cabo las tareas de procesamiento de datos en *R*) o de Ingeniería del Software (al estudiar, comprender y aplicar un ciclo de vida de software y diversas metodologías).

En segundo lugar, también se ve reflejada la influencia de la mención escogida, «Sistemas de Información», por la propia naturaleza de este proyecto, por el trabajo con las Bases de Datos, el manejo del software asociado, etc.

4.2. Trabajo futuro

El sistema que se ha desarrollado ha sido capaz de recoger buena parte de la información esencial del COVID-19 en Andalucía. Pero esto se ha hecho mediante la utilización de datos sobre los que no teníamos ningún tipo de control, y estábamos expuestos a cualquier cambio que pudieran sufrir, con lo que ello implica (modificar código de procesamiento de los datos, actualizar la Base de Datos, etc.).

Por tanto, este sistema se podría ampliar con más información, más controlada, y así poder explotar mejor todos esos datos, aportando mucha

más estabilidad y riqueza al sistema.

Apéndice A

Manual de usuario

En este anexo se describirá cómo debe usarse este software de cara a poder explotar su funcionalidad al máximo y de la forma adecuada.

A.1. Sobre el directorio de trabajo del proyecto

En primer lugar, hay que indicar que todo el trabajo se organiza bajo un directorio llamado `TFG`, donde se encuentra todo organizado en directorios para que el sistema trabaje con rutas relativas dentro de él. La estructura del directorio de trabajo se muestra esquemáticamente en la figura A.1.

En la imagen aparece:

- el directorio `datos`, con todos los los ficheros, descargados o generados por/para este proyecto;
- el directorio `media` incluye las fotografías, esquemas, etc. de elaboración propia usados en la memoria del proyecto;
- el directorio `proyecto_tfg` es un proyecto del IDE *RStudio* con todo el código *R*;
- el directorio `recursos` contiene diversos documentos (en PDF, generalmente) que se han usado;
- el directorio `ssdt` alberga los dos proyectos de *SQL Server*, es decir, `covid19and_etl` y `covid19and_olap`.

En el directorio principal también se incluye el archivo de *Power BI* (`Cuadro de mando - Municipios.pbix`) con el cuadro de mando elaborado para la «demo».

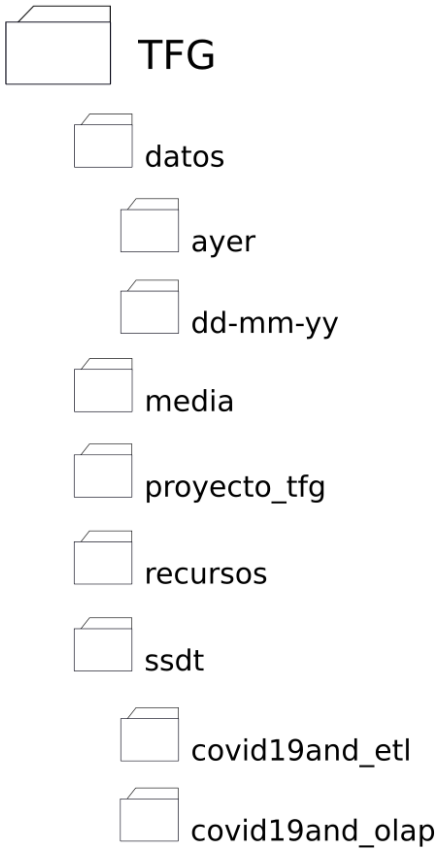


Figura A.1: Estructura de directorios fundamental del directorio del proyecto. Fuente: elaboración propia.

A.2. Requerimientos

Para poder ejecutar los scripts de R necesitamos:

1. Instalar el lenguaje de *R* desde `https://cran.rstudio.com/`, seleccionando el Sistema Operativo que queramos.
2. Opcionalmente, recomiendo instalar el IDE *RStudio* (`https://www.rstudio.com/products/rstudio/download/#download`), un IDE personalizado para R pero que da cobertura al desarrollo con otro lenguaje. Es con este software con el que se han desarrollado todas las tareas en R de procesamiento de los datos, y ofrece un entorno de ejecución muy sencillo y rápido.

Una vez que tenemos el entorno listo, tenemos que acceder a una consola de R e instalar, mediante el comando `install.packages(...)` cada uno de los siguientes paquetes[1], todos ellos necesarios y no incluidos por defecto en la instalación del lenguaje:

1. *RCurl*
2. *rio*
3. *readr*
4. *stringr*
5. *tidyr*
6. *dplyr*
7. *lubridate*

Una vez hecho esto, podremos ejecutar cualquier script de R del proyecto.

A.3. Actualizando los datos

El script *descargar_datasets.R* es el que se ejecuta todos los días hábiles (de Lunes a Viernes sin festivos) para descargar y procesar los datos. Para ejecutarlo hay que hacerlo de la siguiente forma:

```
$> Rscript <path_proyecto>/proyecto_tfg/descargar_datasets.R <path_proyecto>
```

[1] Me han ayudado mucho las «hojas rápidas» (*cheatsheets*) oficiales de algunos de estos paquetes, disponibles en [26].

donde `path_proyecto` es la ruta al directorio «raíz» del proyecto, `TFG`, en el Sistema de Archivos. Pueden usarse barras inclinadas no invertidas (`/`), pues *R* lo detecta correctamente, aunque estemos en un Sistema Operativo Windows.

Para poder actualizar, cuando corresponda, los ficheros CSV con las tablas de hechos y dimensiones (dentro del directorio `datos/`, los que empiezan por `dimension_` o `hechos_`), basta con ejecutar los scripts:

`generar_dimensiones.R` y `generar_hechos.R`

en este orden, porque debido a que los hechos tienen claves externas referenciando a las de las dimensiones, el orden contrario generaría errores en los datos.

Una vez actualizados los CSV asociados a las tablas, actualizamos los datos en *SQL Server*. Para ello:

1. Acceder a los paquetes de importación de las tablas, disponibles en la ruta `ssdt/covid19and_etl/covid19and_etl` de la carpeta del proyecto. Pero es necesaria una modificación de estos paquetes antes de ejecutarlos: debido a que la fuente de datos es un fichero CSV, el paquete de importación tiene configurado el *path* de donde obtener los datos que se usó para desarrollar este proyecto. Por tanto, la primera vez que se usen estos paquetes, hay que:

   a) Abrir cada paquete de importación con un editor de texto plano (esto es posible porque los paquetes de importación son ficheros XML).

   b) Buscar el campo `DTS:ConnectionString` y cambiar su valor (está entrecomillado) por la ruta del CSV correspondiente.

   c) Guardamos el ficheroy y cerrar.

   Ahora ya se pueden ejecutar los paquetes desde el proyecto *SSIS* `covid19and_etl`, como se describía en la sección 3.10.4.1.

2. Restablecer las claves primarias de las tablas desde *SQL Server Management Studio*, tal y como se explicaba en el Recuadro 3.1.

La herramienta para poder procesar los paquetes de importación, *SQL Server Data Tools*, se puede descargar en su versión para estudiante de forma gratuita en

`https://docs.microsoft.com/es-es/sql/ssdt/download-sql-server-data-tools-ssdt?view=sql-server-ver15`

También, para poder acceder a la Base de Datos de *SQL Server* hace falta *SQL Server Management Studio*, que se puede obtener gratuitamente en

`https://docs.microsoft.com/es-es/sql/ssms/download-sql-server-management-studio-ssms?view=sql-server-ver15`

Previamente, en un caso real, se habría comprado el software de *SQL Server* e instalado siguiendo las indicaciones que se proporcionen junto al producto.

Observación 3 *En el caso del desarrollo de este proyecto se ha usado una Máquina Virtual proporcionada por el tutor, que contenía todas las herramientas ya instaladas y configuradas.*

A.4. Despliegue de la Base de Datos Multidimensional

Para desplegar la base de datos se seguirán las instrucciones ya explicadas en la sección 3.10.4.3.

Bibliografía

[1] AEAT. «Tabla de coeficientes de amortización lineal». Agencia Estatal de la Administración Tributaria. https://www.agenciatributaria.es/AEAT.internet/Inicio/_Segmentos_/Empresas_y_profesionales/Empresas/Impuesto_sobre_Sociedades/Periodos_impositivos_a_partir_de_1_1_2015/Base_imponible/Amortizacion/Tabla_de_coeficientes_de_amortizacion_lineal_.shtml.

[2] Boletín Oficial del Estado. «Real Decreto Legislativo 2/2015, de 23 de octubre, por el que se aprueba el texto refundido de la Ley del Estatuto de los Trabajadores». Agencia Estatal del Boletín Oficial del Estado. https://www.boe.es/buscar/act.php?id=BOE-A-2015-11430.

[3] F. Berzal. *Guión de prácticas de Ingeniería de Sistemas de Información - Presupuesto del proyecto* [Documento de PDF]. Disponible en: http://elvex.ugr.es/decsai/information-systems/lab/project/presupuesto.2020.pdf

[4] F. Berzal. *Transparencias de la asignatura «Ingeniería de Sistemas de Información»* [Documento de PDF]. Disponible en: http://elvex.ugr.es/decsai/information-systems/slides/33%20Data%20Access%20-%20Data%20Warehousing.pdf.

[5] IECA. «Informe COVID-19 en Andalucía». Instituto de Estadística y Cartografía de Andalucía. https://www.juntadeandalucia.es/institutodeestadisticaycartografia/salud/COVID19.html.

[6] AEAT. *Contribuyentes por el IRPF*. Agencia Estatal de la Administración Tributaria. https://www.agenciatributaria.es/AEAT.internet/Inicio/Ayuda/Manuales__Folletos_y_Videos/Manuales_de_ayuda_a_la_presentacion/Ejercicio_2016/_Ayuda_Modelo_100/3__Cuestiones_generales/3_1__Ideas_previas/3_1_1__Contribuyentes_por_el_IRPF/3_1_1__Contribuyentes_por_el_IRPF.html

[7] ING. «Qué es y cómo calcular una amortización». En Naranja. https://www.ennaranja.com/economia-facil/

que-es-y-como-calcular-una-amortizacion/ (acceso: Marzo de 2021).

[8] ExcelTotal. «Diagrama de Gantt en Excel». Excel Total. https://exceltotal.com/diagrama-de-gantt-en-excel/.

[9] SAS, «Mapa de Atención Primaria de Salud de Andalucía», Servicio Andaluz de Salud - Dirección General de Asistencia Sanitaria - Subdirección de Coordinación de Salud, España, Guía, 2003. [En línea]. Disponible en: https://www.sspa.juntadeandalucia.es/servicioandaluzdesalud/sites/default/files/sincfiles/wsas-media-pdf_publicacion/2020/libSASmapaAP.pdf.

[10] SitchData. «SQL Server Integration Services (SSIS) vs. Pentaho Data Integration (Kettle) vs. Stitch». Stitch Data. https://www.stitchdata.com/vs/ssis/pentaho/.

[11] C. Adamson, *Star Schema: The Complete Reference*, 1ª edición. EEUU: The McGraw-Hill Companies, 2010.

[12] M. Golfarelli y S. Rizzi, *Data warehouse design: modern principles and methodologies*, 1ª edición. Italia: The McGraw-Hill companies, 2009.

[13] R. Kimball y M. Ross, *The Data Warehouse Toolkit (3rd Edition)*, 3ª edición. EEUU: John Wiley y Sons, Inc, 2013.

[14] J. Samos. (2019/20). *Guion de la Práctica 4 de SMD: SSIS (SQL Server Integration Services)* [Documento de PDF]. Disponible en: /recursos/smd04etlmssis.pdf.

[15] J. Samos. (2019/20). *Guion de la Práctica 6 de SMD: SSAS (SQL Server Analysis Services)* [Documento de PDF]. Disponible en: /recursos/smd06olapmssas.pdf.

[16] J. Samos. (2019/20). *Transparencias de la asignatura SMD* [Presentación de PDF]. Disponible en /recursos/TransparenciasSMD_1920.

[17] Profesores de la asignatura. (2018/19). *Transparencias del Capítulo 2.2 de «Fundamentos de Ingenería del Software»* [Presentación de PDF]. Disponible en recursos/FIS_Capitulo2.2.pdf.

[18] E. Garví, N. Padilla, J. Samos. (2016/17). *Transparencias de la asignatura SMD* [Presentación de PDF]. Disponible en: recursos/TransparenciasSMD_antiguas.

[19] JavaTPoint. *Operaciones OLAP.* JavaTPoint. https://www.javatpoint.com/olap-operations.

[20] R.D.Peng, *Expresiones regulares en R*, [En línea]. Disponible en: https://bookdown.org/rdpeng/rprogdatascience/regular-expressions.html#grepl

[21] INE, 9 feb. 2021, «Relación de provincias con sus códigos» Instituto Nacional de Estadística, doi: https://www.ine.es/daco/daco42/codmun/cod_provincia.htm.

[22] INE, 1 ene. 2020, «Relación de municipios y sus códigos por provincias» Instituto Nacional de Estadística, doi: https://www.ine.es/daco/daco42/codmun/codmun20/20codmun.xlsx.

[23] J. Samos, *Sistemas Multidimensionales. Prácticas con Power BI y Power Query, Tableau y Tableau Prep, SSIS, SSAS, PDI, Mondrian y R*, [En línea]. Disponible en: https://lsi2.ugr.es/jsamos/sm2019/.

[24] J. Samos, *Sistemas Informáticos de Soporte a la colaboración y decisión*, [En línea]. Disponible en: https://lsi2.ugr.es/jsamos/siscd/.

[25] Fotografías usadas para elaborar la figura 3.35: https://es.wikipedia.org/wiki/Junta_de_Andaluc%C3%ADa#/media/Archivo:Logotipo_de_la_Junta_de_Andaluc%C3%ADa_2020.svg, https://es.wikipedia.org/wiki/Junta_de_Andaluc%C3%ADa#/media/Archivo:Logotipo_de_la_Junta_de_Andaluc%C3%ADa_2020.svg, https://codingsight.medium.com/what-is-sql-server-analysis-services-40740ed613fb y https://es.wikipedia.org/wiki/R_(lenguaje_de_programaci%C3%B3n)#/media/Archivo:R_logo.svg.

[26] RStudio. *Cheatsheets de RStudio*. RStudio. https://www.rstudio.com/resources/cheatsheets/.