



UNIVERSIDAD DE VALLADOLID

ESCUELA TÉCNICA SUPERIOR DE INGENIEROS DE TELECOMUNICACIÓN

TRABAJO FIN DE GRADO

GRADO EN INGENIERÍA DE TECNOLOGÍAS ESPECÍFICAS DE
TELECOMUNICACIÓN, MENCIÓN EN TELEMÁTICA

**Diseño e implementación de una base de datos
como soporte para un cuadro de mando para
la Industria 4.0**

AUTOR:

D. Julián Rodríguez Hernández

Tutor:

Dr. D. Ignacio de Miguel Jiménez

Valladolid, 30 de Junio de 2021

TÍTULO: **Diseño e implementación de una base de datos para un cuadro de mando para la Industria 4.0**

AUTOR: **D. Julián Rodríguez Hernández**

TUTOR: **Dr. D. Ignacio de Miguel Jiménez**

DEPARTAMENTO: **Teoría de la Señal y Comunicaciones e Ingeniería Telemática**

TRIBUNAL

PRESIDENTE: **D. Evaristo J. Abril Domingo**

VOCAL: **D. Juan Carlos Aguado Manzano**

SECRETARIO: **D. Ignacio de Miguel Jiménez**

SUPLENTE: **D. Ramón J. Durán Barroso**

SUPLENTE: **Dña. Noemí Merayo Álvarez**

RESUMEN

La irrupción y el auge del Big Data deriva y desemboca en multitud e innumerables colecciones de datos, las cuales deben ser almacenadas en “algún lugar” para posteriormente ser utilizadas y realizar diferentes operaciones o procesos de análisis, y evaluación que se requiera con los mismos. En muchas ocasiones, estos procesos están apoyados en cuadros de mando o *dashboards*, de tal manera que nos proporcionan una apariencia más visual e intuitiva, a la vez que se pueden realizar múltiples escenarios en el acto.

Ahora bien, muchas empresas siguen dependiendo fuertemente de hojas de cálculo Excel para almacenar información, cuando en muchas ocasiones es más eficiente el uso de bases de datos. El objetivo de este proyecto consiste en partir de la información almacenada actualmente por una hipotética empresa en una hoja Excel (potencialmente voluminosa), diseñar e implementar una base de datos para almacenar dicha información, con el objetivo de que sirva como soporte para un cuadro de mando o *dashboard*.

Para ello se parte de unas determinadas especificaciones que marcan las directrices que posee un archivo Excel, el cual contiene esa colección de datos. Una vez analizadas las pautas del archivo Excel se desarrolla el diseño de la base de datos y su implantación funcional. Para acabar desplegando un *script* de migración que hace posible y efectiva la exportación de los datos albergados primeramente en el archivo Excel hacia la base de datos diseñada. Finalmente se pone en funcionamiento todo el conjunto global diseñado.

PALABRAS CLAVE

Datos, archivo Excel, migración, base de datos, *dashboard*

ABSTRACT

The emergence and rise of Big Data derives and leads to a multitude and countless collections of data, which must be stored "somewhere" to be used later and perform different operations or processes of analysis and evaluation that are required with them. In many cases, these processes are supported by dashboards, in such a way that they provide us with a more visual and intuitive appearance, at the same time as multiple scenarios can be carried out on the spot.

However, many companies still rely heavily on Excel spreadsheets to store information, when in many cases it is more efficient to use databases. The objective of this project is to start from the information currently stored by a hypothetical company in an Excel sheet (potentially voluminous), design and implement a database to store this information, with the aim of serving as a support for a dashboard.

To do this, we start from certain specifications that mark the guidelines that an Excel file has, which contains that collection of data. Once the guidelines of the Excel file have been analysed, the design of the database and its functional implementation are developed. Finally, a migration script is deployed to make possible and effective the exportation of the data firstly stored in the Excel file to the designed database. In the end, the entire designed global set is put into operation.

KEYWORDS

Data, Excel file, migration, database, dashboard

AGRADECIMIENTOS

En primer lugar, quiero dar las gracias a mi tutor Nacho, por haberme dado la oportunidad de desarrollar este TFG, por toda su ayuda, asesoramiento y consejos compartidos a lo largo de este camino, sin el cual no habría sido posible llevarlo a cabo.

Me gustaría también mencionar y agradecer a todos los miembros del Grupo de Comunicaciones Ópticas, los cuales me han acogido y proporcionado un ambiente afectuoso y familiar durante mi estancia y desarrollo de las prácticas en él.

Muchas gracias a mi compañero Pablo, quien paralelamente estaba desarrollando el cuadro de mando para el cual ofrecería y serviría mi base de datos creada, con el que me he entendido y compenetrado a la perfección durante su elaboración y por compartir multitud de conversaciones fortalecedoras.

Quiero manifestar y mostrar todo mi agradecimiento a mis padres, Jose e Isabel, y a mi hermana Isabel, por creer siempre en mí y estar siempre a mi lado mostrando su apoyo incondicional.

Finalmente, agradecer a mis compañeros de titulación, especialmente, Francisco, Rodrigo y Sergio, todo el apoyo y ayuda durante la realización de la misma y por todas las vivencias compartidas durante estos años.

ÍNDICE

1	Introducción	1
1.1	Motivación	1
1.2	Objetivos del TFG.....	2
1.2.1	Objetivo general.....	2
1.2.2	Objetivos específicos	2
1.3	Fases y pautas para el desarrollo del TFG	3
1.4	Herramientas utilizadas en la realización del TFG	4
1.5	Estructura de la Memoria del TFG	5
2	Marco teórico	8
2.1	Introducción	8
2.2	Definición y tipos.....	9
2.3	Base de Datos Relacionales	11
2.3.1	Estructura.....	13
2.3.2	Normalización.....	17
2.3.3	Sistemas Gestores de Bases de Datos Relacionales (SGBDR)	21
3	La empresa ficticia ACME. Situación de partida y análisis de requisitos	26
3.1	Introducción	26
3.2	La empresa ficticia ACME	26
3.2.1	Descripción General	26
3.2.2	Descripción Específica. Política de identificación.	28
3.2.3	Funcionamiento de la actividad.	31
3.2.4	Almacenamiento de datos. Archivo Excel.....	35
3.2.5	Información adicional.	39

3.2.6	Implementaciones futuras	40
4	Diseño de la Base de Datos.....	44
4.1	Introducción.....	44
4.2	Despliegue de requisitos	46
4.2.1	Funcionalidades actuales.....	46
4.2.2	Funcionalidades futuras.....	48
4.3	Diseño.....	49
4.3.1	Tablas	50
4.3.2	Relaciones	66
4.3.3	Base de datos (conjunto global)	71
5	Herramienta automatizada de migración. Resultados	74
5.1	Herramienta automatizada para la migración	74
5.1.1	Introducción	74
5.1.2	Descripción.....	74
5.1.3	Ejecución.....	76
5.2	Resultados.....	76
5.2.1	Introducción	76
5.2.1	Situación previa.....	76
5.2.2	Migración	78
5.2.3	Resultados	80
6	Conclusiones y Líneas Futuras.....	84
6.1	Conclusiones.....	84
6.2	Líneas Futuras.....	85
7	Referencias	88
8	Anexo. Guía de ejecución.....	91
8.1	Descarga de WampServer	93

8.2	Ejecución de WampServer.....	95
8.3	Creación de la Base de Datos (phpMyAdmin)	95
8.4	Migración del archivo Excel.....	96

ÍNDICE DE FIGURAS

Figura 1. Base de datos jerárquica	9
Figura 2. Base de datos de red.....	10
Figura 3. Base de datos orientada a objetos	10
Figura 4. Base de datos multidimensional	11
Figura 5. Base de datos relacional.....	12
Figura 6. Ejemplo de base de datos relacional	13
Figura 7. Niveles de normalización.....	19
Figura 8. Popularidad de los sistemas gestores de bases de datos relacionales más utilizados	22
Figura 9. Fases de diseño y desarrollo de una base de datos	45
Figura 10. Tabla tests	51
Figura 11. Tabla maquinas	54
Figura 12. Tabla operadores.....	55
Figura 13. Tabla configuracion_calidad.....	57
Figura 14. Tabla software.....	58
Figura 15. Tabla proveedores.....	59
Figura 16. Tabla variante_protocolo_1	60
Figura 17. Tabla medidas_consolidadas	61
Figura 18. Tabla medidas_software_no_aceptadas.....	63
Figura 19. Tabla medidas_software_por_elemento_no_aceptadas.....	65
Figura 20. Relaciones de clave simple	68
Figura 21. Relaciones de clave compuesta.....	70
Figura 22. Relaciones de clave compuesta asociados con id_elemento	70
Figura 23. Diagrama Relacional de la Base de datos	72
Figura 24. Primer fragmento de la hoja 1 del archivo Excel.....	77
Figura 25. Segundo fragmento de la hoja 1 del archivo Excel	77

Figura 26. Tercer fragmento de la hoja 1 del archivo Excel	77
Figura 27. Fragmento del script de migración (I).....	78
Figura 28. Fragmento del script de migración (II)	79
Figura 29. Fragmento del script de migración (III)	79
Figura 30. Primer fragmento del aspecto final de la tabla tests.....	80
Figura 31. Segundo fragmento del aspecto final de la tabla tests.....	81
Figura 32. Aspecto final de la tabla maquinas.....	81
Figura 33. Aspecto final de la tabla configuracion_calidad	81
Figura 34. Aspecto final de la tabla operadores	82
Figura 35. Fragmento del aspecto final de la tabla variante_protocolo_1.....	82

ÍNDICE DE TABLAS

Tabla 1. Sinónimos empleados en bases de datos relacionales	14
Tabla 2. Distribución en Excel para Productos Propios	39
Tabla 3. Distribución en Excel para Productos Externos	39
Tabla 4. Registros contenidos en tabla tests	53
Tabla 5. Índices de la tabla tests	53
Tabla 6. Registros contenidos en tabla maquinas	55
Tabla 7. Índices de la tabla maquinas	55
Tabla 8. Registros contenidos en tabla operadores.....	56
Tabla 9. Índices de la tabla operadores.....	56
Tabla 10. Registros contenidos en tabla configuracion_calidad	57
Tabla 11. Índices de la tabla configuracion_calidad	57
Tabla 12. Registros contenidos en tabla software	58
Tabla 13. Índices de la tabla software	58
Tabla 14. Registros contenidos en tabla proveedores.....	59
Tabla 15. Índices de la tabla proveedores.....	59
Tabla 16. Registros contenidos en tabla variante_protocolo_1	60
Tabla 17. Índices de la tabla variante_protocolo_1	60
Tabla 18. Registros contenidos en tabla medidas_consolidadas	62
Tabla 19. Índices de la tabla medidas_consolidadas	62
Tabla 20. Registros contenidos en tabla medidas_software_no_aceptadas.....	63
Tabla 21. Índices de la tabla medidas_software_no_aceptadas.....	64
Tabla 22. Registros contenidos en tabla medidas_software_por_elemento_no_aceptadas	65
Tabla 23. Índices de la tabla medidas_software_por_elemento_no_aceptadas.....	65

1

Introducción

Este primer capítulo introductorio de la presente memoria del Trabajo Fin de Grado refleja las motivaciones vigentes para llevarse a cabo y los objetivos que se pretenden lograr mediante desarrollo y realización del proyecto implicado. Del mismo modo, se presentan las pautas o fases marcadas para su consecución y las herramientas o medios empleados para la obtención del conjunto global. Finalmente, para concluir se describe la estructura y distribución del resto del documento.

1.1 Motivación

En la actualidad, casi la totalidad de las empresas basan sus operaciones, transacciones o registros de los innumerables datos con los que trabajan apoyándose en las bases de datos. Pero esto antes no era así como tal, y tenían que basar ese apoyo en un papel o cuaderno en el que plasmar los registros pertinentes de lo que se quisiera reflejar.

Posteriormente, se fue introduciendo el mismo método, pero en formato electrónico, y en muchos casos utilizando la aplicación informática Microsoft Excel. Si anteriormente se plasmaban los datos de manera física, es decir, a puño y letra en una hoja de un cuaderno o individual e independiente, con la ayuda de este otro software el registro poseía el mismo fin, pero cambiando el modo o método, ya que bastaba con encender el equipo en el que estuviese instalada dicha aplicación y anotar en sus hojas de cálculo los datos que se precisara. Sin embargo, Excel es una aplicación dirigida realmente a analizar datos y no a administrarlos, pues no es realmente un Sistema de Gestión de Bases de Datos.

El último de los pasos para la mejora y sofisticación del hecho acaecido en las anteriores líneas es, como ya anunciaba al principio, el registro y almacenamiento de todos

ellos en bases de datos, desde y con las cuales apoyarse e interactuar para consultar o modificar cualquiera de los datos de referencia involucrados en la operación que se disponga a llevar a cabo.

Todavía es común que muchas empresas almacenen sus datos en hojas de cálculo en lugar de en auténticas bases de datos, y ese es el escenario de partida que toma este TFG. Se asume una empresa ficticia que tiene sus datos almacenados actualmente en hojas de cálculo Excel y que desea migrarlos hacia una base de datos. El objetivo del TFG es por tanto doble. Por un lado, se trata de diseñar y crear una base de datos que dé cabida a dicha colección de datos. Por otro, se trata de diseñar e implementar un procedimiento para migrar esos datos desde las hojas de cálculo hacia la base de datos, teniendo en cuenta además que dichos datos puedan ser objeto de cualquiera operación que se desee desempeñar a través de un cuadro de mando. Puesto que el TFG asume una empresa ficticia, parte del trabajo realizado en el mismo ha consistido en la generación de los datos de partida en Excel, y con respecto a la elaboración de un cuadro de mando, ese ha sido el objetivo de un Trabajo Fin de Máster relacionado [1].

1.2 Objetivos del TFG

1.2.1 Objetivo general

El objetivo global de todo el conjunto de desarrollo del Trabajo Fin de Grado es diseñar e implementar una base de datos que sirva como soporte para un cuadro de mando o *dashboard* y, a su vez, migrar datos recogidos y almacenados en un archivo Excel a la base de datos diseñada previamente.

1.2.2 Objetivos específicos

Para llegar a lograr lo descrito en el párrafo anterior, lo mejor será identificar y señalar una serie de objetivos específicos que deberán ser alcanzados a lo largo de las distintas fases del proyecto. Estos objetivos específicos son:

- Diseñar y crear datos relacionados consecuentemente en un archivo Excel a partir de unas especificaciones dadas. (Puesto que el TFG se va a desarrollar suponiendo un escenario hipotético, es necesario crear de forma

sintética los datos de partida, y ese es, por tanto, uno de los primeros objetivos del proyecto).

- Diseñar, crear e implementar una base de datos.
- Desarrollar una herramienta automatizada (*script*) que permita la exportación y migración de datos desde un archivo Excel hacia una base de datos.
- Implementar y poner en funcionamiento en un entorno global todos los objetivos citados anteriormente.

1.3 Fases y pautas para el desarrollo del TFG

Una vez reconocidos y marcados los objetivos que se pretenden desarrollar, el planteamiento llevado a cabo para hacerlos posibles ha sido designar y seguir una serie de fases evolutivas y, en consecuencia, mediante las cuales poder llegar a conseguirlo con éxito:

1. Propuesta de un escenario de partida de una empresa que dispone de datos almacenados en Excel y desea migrarlos hacia una base de datos.
2. Creación de un archivo Excel con datos ficticios pero consecuentes con los especificado en el escenario de partida.
3. Reconocimiento y comparación de diferentes herramientas y tecnologías como alternativas para el diseño e implementación de la base de datos.
4. Familiarización con las herramientas y tecnologías elegidas.
 - a. WampServer
 - b. MySQL
 - c. phpMyAdmin
5. Fase de diseño, desarrollo y despliegue de la base de datos.

6. Análisis y planteamiento para la migración de los datos desde el archivo Excel a la base de datos creada.
7. Creación y desarrollo del *script* automatizado para la migración de los datos desde el archivo Excel a la base de datos creada.
8. Fase de despliegue y evaluación del conjunto creado.
9. Fase de revisión, corrección y consolidación del conjunto global.
10. Obtención y evaluación de resultados.
11. Planteamiento y extracción de conclusiones.
12. Elaboración del documento de la memoria del Trabajo Fin de Grado.

1.4 Herramientas utilizadas en la realización del TFG

Para la realización de todo el trabajo completo se ha requerido del uso de determinadas herramientas, aplicaciones software y lenguajes de programación además de multitud de horas de trabajo a papel y bolígrafo para realizar los primeros bocetos enfocados al diseño de la base de datos.

Como recogía y las anunciaba en el apartado anterior 1.3, tres son las herramientas fundamentalmente sobre las que se cimienta este TFG. En primer lugar, la instalación de la aplicación software WampServer [2], la cual está compuesta por una pila o conjunto de soluciones software que presentan diferentes prestaciones. El acrónimo Wamp, que forma parte del nombre, tiene el siguiente significado:

- W, de soporte únicamente para sistema operativo Windows. Es el sistema operativo base empleado para la realización de este TFG.
- A, de servidor Apache. Ofrece disponer de un servidor web local en nuestro equipo.
- M, con relación a MySQL. Incluye sistema de gestión de bases de datos relacionales MySQL.

- P, perteneciente al lenguaje de programación PHP. Mediante el cual poder crear o emplear multitud de herramientas operacionales al ejecutarse junto con Apache y comunicarse con MySQL.

MySQL ha sido el sistema de gestión de bases de datos relacionales elegido y mediante el cual se ha ido dando forma y confeccionando la base de datos.

Poniendo en ejecución la aplicación anterior se puede hacer uso de otra de las herramientas implicadas, que es phpMyAdmin. Se trata de un software gratuito, el cual también viene instalado internamente al hacer lo propio con WampServer. Este permite realizar un manejo de la administración y gestión de la base de datos mediante una API establecida a la que se tiene acceso a través del navegador. Admite un amplio abanico de operaciones con bases de datos, tablas, relaciones, índices, etc., que se pueden realizar mediante la interfaz de usuario, ejecutándose bajo la comunicación de MySQL. De la misma manera, también posee un cuadro de diálogo donde poder ejecutar directamente declaraciones SQL.

Otra de las herramientas de las que se ha necesitado su uso, es de un editor de texto, en el cual desarrollar todo el código descrito en lenguaje PHP, para conseguir realizar la migración automatizada a la nueva base de datos de los datos recogidos hasta el momento únicamente en el archivo Excel.

1.5 Estructura de la Memoria del TFG

El desarrollo del presente documento está distribuido de la siguiente manera:

1. Empezando por el capítulo en el que nos encontramos, Capítulo 1, se ha introducido una pequeña presentación, no exhaustiva, del TFG. Qué es lo que se pretende conseguir y cómo se ha llevado a cabo.
2. El Capítulo 2 proporciona el marco teórico asociado al TFG.
3. Una vez identificados los temas a abordar es hora de ponerse a desarrollarlos. Por ello, el Capítulo 3 presenta el escenario hipotético que se plantea resolver, y así realiza una descripción detallada del funcionamiento que está realizando

la empresa ficticia ACME para obtener los requisitos necesarios con los que poder diseñar y desarrollar la base de datos óptima.

4. Todo el diseño y desarrollo de la base de datos viene enmarcado y explicado en el Capítulo 4.
5. En poder de la base de datos, el Capítulo 5 denota cómo se ha realizado la herramienta para la migración de los datos del archivo Excel a la base de datos recientemente diseñada. También se describen y recogen los resultados obtenidos al poner en ejecución y sincronizar la herramienta de migración y la base de datos.
6. Desarrollada la totalidad de los fundamentos que basan el TFG, es hora de extraer y compartir las conclusiones pertinentes al respecto y aportar algunas posibles visiones de futuro. Esto es lo que refleja lo escrito en el Capítulo 6.
7. Finalmente, para concluir el documento se ha introducido un Anexo. Guía de ejecución complementario al final, en el que se detalla un ejemplo de ejecución en el que se puede llevar a cabo todo lo desarrollado a lo largo de la presente memoria de este Trabajo Fin de Grado.

2

Marco teórico

2.1 Introducción

A nadie sorprende ya en los días que vivimos el término base de datos, puesto que llevan siendo utilizadas desde hace mucho tiempo. En concreto, la primera base de datos creada data en la década de los 60 [3]. Si bien es cierto, que cada día con abundante frecuencia desempeñan activamente un papel importante en cualquier acción cotidiana que realizamos a través de cualquier dispositivo electrónico, fundamentalmente. Pero también, un simple ejemplo de ellas es la realización de una lista de la compra, donde plasmamos en un papel los productos y la cantidad de estos necesarios, y de esta manera estamos confeccionando nuestra pequeña base de datos. Otro ejemplo de ellas es nuestra agenda telefónica, en la cual anotamos conjuntamente nombres y números de teléfono con un significado implícito.

Otra de las certezas que existe, es que aproximadamente la totalidad de operaciones que se desarrollan en la red están sujetas a una o incluso varias bases de datos. Desde el comercio electrónico, la banca on-line, hasta la reserva de una habitación para unas merecidas vacaciones o para realizar la inspección técnica de tu vehículo necesitan de esa conexión con las mismas.

Actualmente, en cualquier actividad desarrollada se genera multitud de información la cual, con cada vez más frecuencia, es recomendable almacenarla en algún lugar para poder tener acceso a ella en un momento dado. Cada vez la cantidad de datos producida es más elevada continuamente, hasta el punto de que ha sido acuñado el anglicismo *Big Data* para referirnos a ese indeterminado y desmesurado volumen de información.

Es por ello por lo que, dependiendo de la actividad desarrollada, cantidad y volumen que adquiera esa información así será desarrollado de una u otra manera ese lugar donde depositarla.

2.2 Definición y tipos

El significado que le damos al término *base de datos* puede ser muy amplio y va desde lugar donde se almacena información, hasta conjunto de datos pertenecientes a un mismo contexto y almacenados sistemáticamente para su posterior uso; pasando por estructuras especializadas que permiten a sistemas computarizados guardar, manejar y recuperar datos con gran rapidez; conjunto de datos almacenados en memoria externa que están organizados mediante una estructura de datos, o lugares diseñados para almacenar grandes cantidades de datos de forma sistematizada con el fin de que podamos acceder a ellos de forma rápida y eficiente. No nos podemos quedar con una definición concreta puesto que todas ellas son verídicas, por ello que lo mejor sería realizar una abstracción de ideas de cada una y crear una fusión híbrida a raíz de todas ellas.

Dependiendo de la información que se vaya a almacenar en la misma y de la función que se desee desempeñar con ellas, existen distintos tipos de modelado de las bases de datos [4]. Algunos de los más destacados son:

- **Jerárquicas:** la estructura que tienen es semejante a la estructura de un árbol. De un nodo raíz o padre pueden derivar otros nodos (hijos) con el objetivo de preservar los datos organizándose en un orden particular. Podemos hacernos una idea si recordamos como distribuye Microsoft Windows las carpetas y archivos. La **Figura 1** representa una idea visual de la distribución de este tipo de bases de datos.

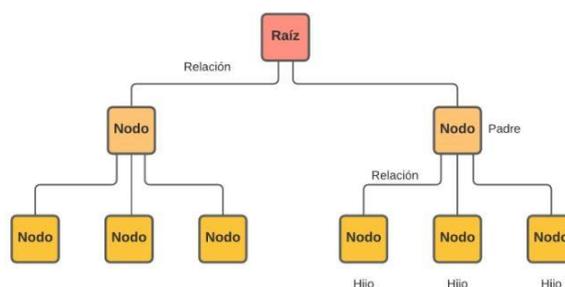


Figura 1. Base de datos jerárquica

- **De red:** son muy similares a las anteriores. La diferencia radica en que un nodo hijo puede tener más de un nodo padre, en lugar de únicamente uno como ocurría en las jerárquicas. Se puede representar este otro tipo es como un conjunto de nodos conectados unos y otros mediante enlaces (**Figura 2**).

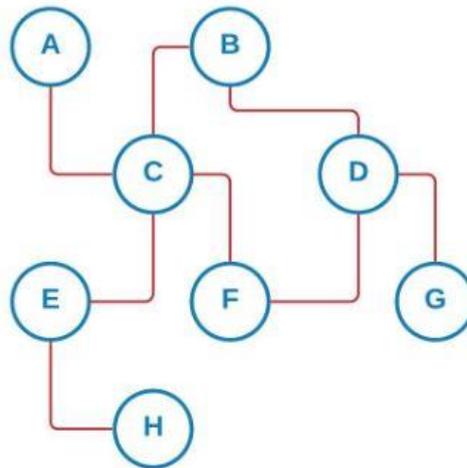


Figura 2. Base de datos de red

- **Orientadas a Objetos:** como propiamente su nombre indica, se trata de una base de datos que consta de objetos y son destinadas en implementaciones orientadas a objetos. Los datos son guardados en objetos junto con los métodos (funciones) y los atributos que los describen más en profundidad. Los objetos más semejantes pueden ser agrupados en una misma clase, definida como *object category*. Dentro de una clase dada, cada objeto que forma parte de ella se conoce instancia de este. Visualizando la **Figura 3** se desprende una idea de lo recogido anteriormente.

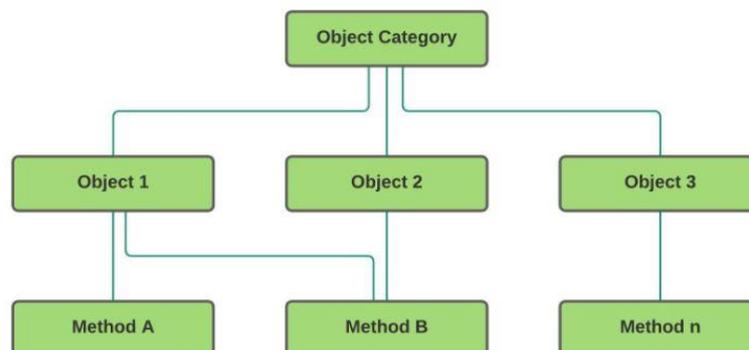


Figura 3. Base de datos orientada a objetos

- **Transaccionales:** se modelan con el objetivo de enviar y recibir datos a grandes velocidades a la vez que se conserva y mantiene la integridad de los datos que van siendo almacenados. No se constituyen específicamente para el análisis de datos. Son muy utilizadas en el ámbito de las cuentas bancarias y financieras.
- **Multidimensionales:** las bases de datos multidimensionales más comunes pueden ser vistas como cubos de datos (**Figura 4**), aunque pueden ser extendidas hasta N dimensiones, que representan diferentes perspectivas de todos los datos alojados. Su diseño y uso es debida al desarrollo de aplicaciones muy concretas.

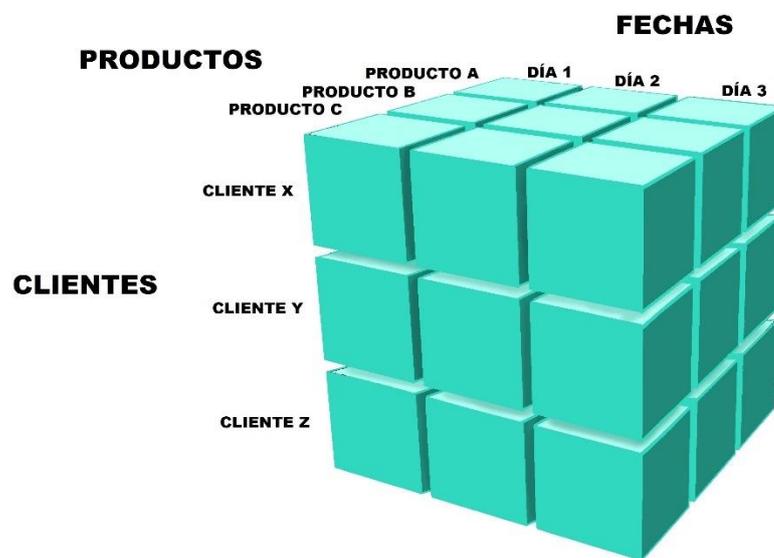


Figura 4. Base de datos multidimensional

- **Relacionales:** se trata del modelo de bases de datos más aceptado, popular e implementado en la actualidad. En la inmediata sección 2.3, se describe con más detenimiento y profundidad este tipo de base de datos, al cual pertenece la base de datos que se ha implementado para llevar a cabo el desarrollo y cometido de este TFG.

2.3 Base de Datos Relacionales

Las bases de datos relacionales están basadas en el modelo relacional, una forma sencilla e intuitiva de representar datos en tablas. Están constituidas por una colección de tablas donde quedan plasmados los datos, a la vez que quedan reflejadas las diferentes

relaciones entre ellas. Su disposición permite establecer interconexiones o relaciones entre los distintos datos que se almacenan en las tablas, y a través de dichas conexiones nos permite identificar y acceder a datos en relación con otro dato en la base de datos. La **Figura 5** recoge una ilustración de la estructura de una posible base de datos relacional.

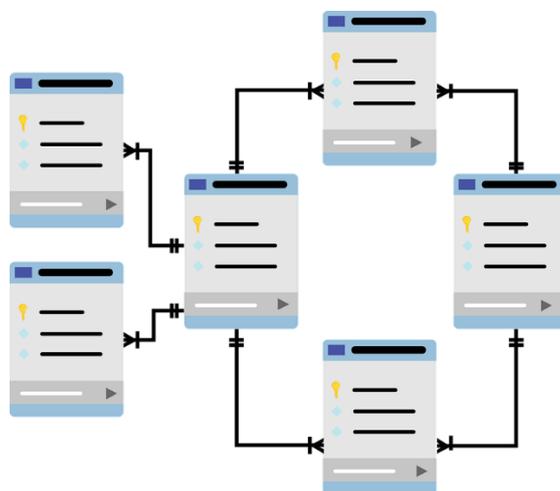


Figura 5. Base de datos relacional

Las bases de datos relacionales mantienen los datos en tablas, lo que proporciona una manera eficiente, intuitiva y flexible de almacenar y acceder a información estructurada.

Algunas de las ventajas más destacadas que presta el uso de una base de datos relacional son:

- **Simplicidad.** El modelo de bases de datos relacionales es más simple que el modelo empleado en las bases de datos jerárquicas y/o de red.
- **Facilidad de manejo y uso.** El modelo relacional es muy intuitivo y fácil de manejar, ya que las tablas (formadas por filas y columnas) son muy naturales y fáciles de entender.
- **Independencia estructural.** La conformación de una base de datos relacional solo se ocupa propiamente de los datos alojados y no de la estructura diseñada para su disposición. Esto mejora y aumenta el rendimiento.
- **Independencia de los datos.** La estructura de una base de datos relacional se puede modificar sin tener que realizar cambios en ninguna otra aplicación.

- Capacidad de consulta. Mediante el lenguaje de consulta de alto nivel como SQL, el más empleado en este tipo de bases de datos, evita una navegación compleja por la base de datos.
- Escalabilidad. La cantidad de filas o registros y la cantidad de columnas o atributos pueden verse aumentados o, en su defecto disminuidos, sin provocar ningún problema, para adaptarse y mejorar así su usabilidad.

En ocasiones, las bases de datos relacionales pueden ver aumentada su complejidad a medida que aumenta la cantidad de datos y relaciones entre sí. Otro de los aspectos negativos es que los sistemas de bases de datos relacionales con mayor complejidad pueden conducir a bases de datos aisladas donde la información no puede ser compartida de un sistema a otro.

2.3.1 Estructura

Como se anunciaba antes y hemos podido ver en la **Figura 5**, la estructura de las bases de datos de tipo relacional se alza principalmente sobre dos partes notablemente destacadas, tablas y relaciones. Aunque para la constitución de estas dos son necesarios también otros elementos que ahora conoceremos. Para ello, lo haremos apoyándonos en un ejemplo simplificado de base de datos relacional como se ilustra en la **Figura 6**, la cual perfectamente puede haber sido extraída de una empresa comercial, en la cual registraría las numerosas ventas que efectúa a sus clientes.

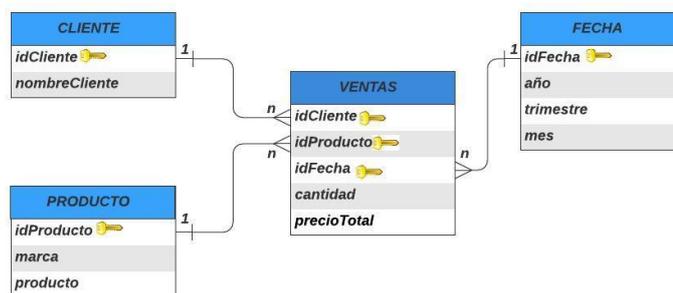


Figura 6. Ejemplo de base de datos relacional

En primer lugar y partiendo de un grado de abstracción mayor, para ir en sentido descendente, el conjunto de las cuatro tablas presentes (*Cliente*, *Fecha*, *Producto* y *Ventas*) son lo que conforman la base de datos. Se pueden apreciar también las relaciones que

ligarán y vincularán los datos almacenados en ellas. Si observamos, ligeramente encima de cada una de las líneas que muestran las relaciones creadas entre las tablas se aprecian dos valores: 1 y n . Estos valores indican si la relación es de uno (1) a muchos (n), de uno a uno, de muchos a muchos y en qué sentido. Por ejemplo, si observamos la orientación y sentido de la relación existente entre la tabla *Cliente* y la tabla *Ventas*, deducimos que la relación es de uno a muchos, lo que indica que un único cliente registrado, ya veremos más adelante cómo, puede tener asociadas numerosas ventas. De la misma manera, atendiendo a la relación de las tablas *Producto* y *Ventas* podemos extraer que un mismo producto puede estar presente en numerosas ventas diferentes.

Bajando un grado en la abstracción y entrando en detalles de cada una de las diferentes tablas, estas están organizadas en filas y columnas. Por columnas se conoce a las etiquetas que aparecen representadas dentro de las tablas. Por lo cual, la tabla *Fecha* presentará cuatro columnas, caracterizadas cada una de ellas por el nombre *idFecha*, *año*, *trimestre* y *mes* respectivamente. Siguiendo el mismo argumento, la tabla *Ventas* se compone de cinco columnas, la tabla *Producto* de tres y la tabla *Cliente* de dos. Cada columna solamente podrá registrar y almacenar los valores del tipo de dato que en ella sea especificado, es lo que se conoce como dominio. El concepto de fila hace referencia al cómputo global de los valores o datos almacenados en cada una de las columnas; esto es, una fila la representarían los datos asociados a todas las columnas. Por ejemplo y dando valores aleatorios, en la tabla *Fecha* una fila vendrá definida por los valores $idFecha = 1$, $año = 2020$, $trimestre = 4$ y $mes = Noviembre$.

Es de importante atención y se debe tener conocimiento que podemos referirnos a los mismos elementos, aunque estemos empleando distinto vocablo. Hasta ahora solo se habían empleado términos como, por ejemplo, columna o fila; pero existen otros que también hacen referencia a esos mismos. Se muestran recogidos en la siguiente **Tabla 1** y la manera de interpretarlos es que podemos referirnos al mismo elemento empleando los términos recogidos a la misma altura.

COLUMNA	↔	ATRIBUTO	↔	CAMPO
FILA	↔	TUPLA	↔	REGISTRO
CLAVE	↔	LLAVE		

Tabla 1. Sinónimos empleados en bases de datos relacionales

De esta manera, a lo largo del desarrollo de la memoria del TFG se hará uso de los diferentes sinónimos para referirnos al elemento que se quiera aludir en cuestión.

Anunciado este pequeño inciso, se prosigue con la descripción de la estructura. El siguiente de los elementos es de gran importancia en el diseño y estructura de las bases de datos relacionales, pues mediante ellos se crean las relaciones que asocian las diferentes tablas y se preserva la integridad de los datos. Se puede reconocer si observamos que junto a algunos de los atributos presentes en las tablas se identifica un icono de una llave, este refleja que ese atributo es clave de la tabla. En el caso de las tablas *Cliente*, *Producto* y *Fecha*, los atributos *idCliente*, *idProducto* e *idFecha* son claves respectivamente, en este caso claves primarias. Ahora veremos que significa eso.

La clave o llave es un atributo o conjunto de atributos, que se utiliza para identificar una fila en una tabla. Los distintos tipos de claves que pueden existir en las tablas de una base de datos relacional son los siguientes [4]:

- **Clave compuesta:** clave formada por más de un atributo o campo.
- **Superclave:** conjunto de columnas o atributos que identifica de forma única cada fila de las tablas. Las superclaves representan una restricción que evita que dos filas no puedan tener exactamente los mismos valores.
- **Clave candidata:** es una superclave mínima. Una clave candidata es un atributo o conjunto mínimo de atributos cuyos valores identifican de forma única las tuplas de la tabla correspondiente.
- **Clave única:** superclave empleada por la base datos para exigir la unicidad de su valor. Puede contener valor nulo.
- **Clave primaria:** es una clave candidata designada. Implícitamente la clave primaria se trata de una clave única, pero la clave principal nunca puede ser nula, debe contener valores.
- **Clave alternativa:** se trata de una clave candidata pero que no es clave primaria.
- **Clave secundaria:** clave empleada para buscar registros pero que no garantiza la unicidad de este.

- **Clave foránea o ajena:** columna o conjunto de atributos de una tabla (hija) que se utiliza para “hacer referencia” a una clave, y en consecuencia una fila concreta, en otra tabla (padre o referenciada).

Volviendo al ejemplo de la **Figura 6**, como ya adelantaba previamente, las columnas *idCliente*, *idProducto* e *idFecha* son las claves primarias simples correspondientes en las tablas *Cliente*, *Producto* y *Fecha* respectivamente. En el caso de la tabla *Ventas*, posee una clave primaria compuesta formada por sus atributos *idCliente*, *idProducto* e *idFecha*. Estos a su vez, pueden ser evidenciados individualmente como tres llaves foráneas haciendo referencia a las columnas con mismo nombre, pero alojadas en cada una de las tres tablas padre restantes, y el conjunto de ellas forman la clave primaria de la tabla *Ventas*.

Otro de los elementos característicos en las bases de datos relacionales son los índices. A simple vista en el ejemplo no pueden ser identificados dado que requeriría conocer en mayor nivel su implementación, no solo en disposición del diagrama relacional. Los índices son atributos designados cuyo cometido es poder ubicar y seleccionar filas de manera eficiente sin tener que rastrear y explorar toda la tabla. Pueden estar compuestos por un único atributo o un conjunto de ellos. Están muy ligados a las claves puesto que mediante las claves podemos realizar las mismas facilidades que nos ofrecen los índices, aunque en ocasiones se puede designar como índice una columna que no es parte de un clave. En el caso concreto de MySQL, clave e índice son sinónimos y, por lo tanto, se puede referir a los atributos o columnas declaradas con esa naturaleza empleando ambos términos. MySQL es un ejemplo de sistema de gestión de bases de datos que más adelante se detallará qué son y para qué se utilizan.

En resumen, las características fundamentales y más destacadas en la estructura de un modelo de base de datos relacional son las siguientes:

- Una base de datos relacional está compuesta por varias tablas o relaciones.
- Las tablas no pueden compartir un mismo nombre. Este debe ser distinto por cada una de ellas.
- Cada una de las tablas están formadas a su vez por filas y columnas.
- Cada fila de la tabla puede ser llamada también tupla o registro.

- A cada columna de la tabla se le conoce también como atributo o campo.
- Dentro de las tablas, las filas pueden estar en cualquier orden.
- En el interior de las tablas, los atributos pueden estar en cualquier orden.
- Por definición, todas las filas de una tabla son distintas. No puede haber dos filas exactamente iguales conteniendo los mismos valores.
- Las tablas deben tener una clave. Las claves pueden estar formadas por uno o varios atributos.
- Las claves primarias son clave principal de una tupla o fila y son las que permiten conseguir la integridad de los datos almacenados.
- Las relaciones entre las tablas padre y las tablas hijas se forman mediante las claves establecidas en ellas, claves primarias y foráneas o ajenas respectivamente.
- Las claves ajenas de las tablas hijas deben poseer el mismo tipo de datos y los mismos posibles valores que la clave primaria de la tabla padre.
- Para cada columna de una tabla se establece un tipo de dato que ostenta un conjunto de valores posibles llamado su dominio. El dominio contiene todos los valores posibles que pueden aparecer en esa columna.
- El grado de la tabla es el número de columnas o atributos establecidos en la tabla.
- La cardinalidad es el número de filas o tuplas dispuestas en la tabla.

2.3.2 Normalización

En general, uno de los resultados implícitos en el diseño de una base de datos relacional es generar un conjunto de tablas que permitan almacenar la totalidad de la información disponible en todas ellas, de tal forma que no exista duplicidad o redundancia alguna y, a la vez, están relacionadas entre sí mediante las claves establecidas en cada una de ellas, permitiendo así acceder y recuperarla fácilmente en el momento que sea estimado oportuno. Esto se consigue mediante el diseño de esquemas de relación que se hallen en la forma normal adecuada.

La normalización es una técnica de diseño o proceso de reorganización de las bases de datos relacionales en el que se aplican una serie de reglas para poseer una estructura de datos robusta y saneada. El objetivo principal de este proceso es evitar que se produzcan en la base de datos problemas reconocidos en su terminología como anomalías, tales como:

- Evitar la redundancia o duplicidad de los datos.
- Preservar y proteger la integridad de los datos.
- Optimizar el espacio de almacenamiento.
- Prevenir problemas de actualización en las tablas.
- Reforzar la veracidad y seguridad de los datos.
- Facilitar el acceso e interpretación de los datos.
- Reducir el tiempo y complejidad de supervisión de la base de datos.

La aplicación de la teoría de la normalización garantiza que el diseño es capaz de evitar cualquier anomalía indeseada. Es por ello por lo que las bases de datos relacionales son mucho más robustas y poseen menor vulnerabilidad ante los fallos, cumpliendo con lo que en informática se denomina ACID (Atomicidad, Consistencia, Aislamiento y Durabilidad) [5].

Para garantizar todo ello, como antes decía, existen diferentes niveles normales, que dependiendo del nivel de normalización que quiera adoptar una base de datos, sus tablas deben de cumplir las reglas presentes en cada uno de ellos. Cada uno de los niveles inmediatamente superior incluye con obligatoriedad que se haya adoptado y cumplido las reglas del nivel inferior. La imagen de la **Figura 7**, mostrada a continuación, nos ayuda a comprender mejor lo descrito en la anterior expresión.

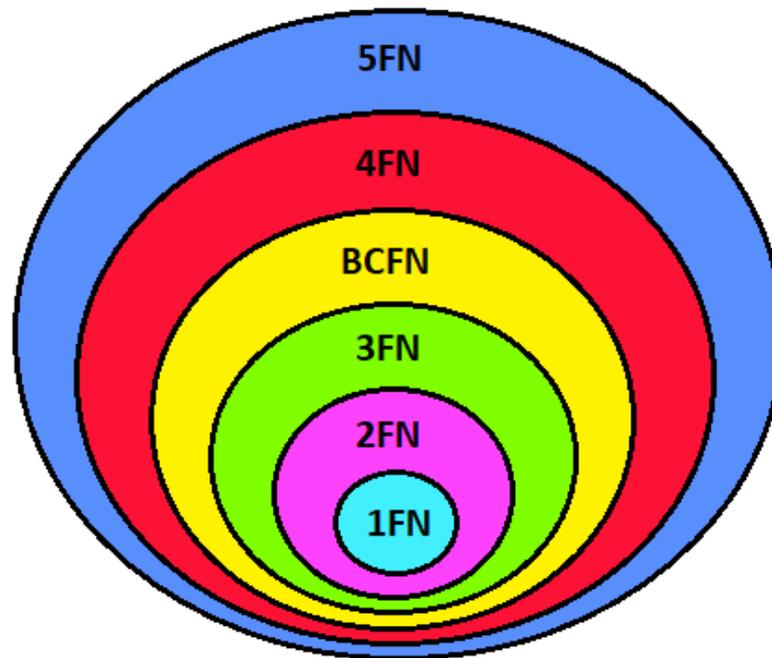


Figura 7. Niveles de normalización

A continuación, y sin entrar en detalles de exhaustiva profundidad, se especifican los niveles y sus correspondientes reglas [4]:

- **Primera forma normal (1FN)**

- Cada una de las columnas o atributos debe poseer nombre único.
- Cada columna o atributo debe albergar un único tipo de dato.
- Cada columna o atributo tiene que contener un único valor.
- Dos o más filas no pueden contener valor idéntico.
- El orden de las filas o columnas es irrelevante.
- Las columnas no pueden contener grupos repetidos.

- **Segunda forma normal (2FN)**

- Estar en primera forma normal (1FN).
- Todos los campos o atributos que no son clave dependen de todos los campos o atributos que son clave.

- **Tercera forma normal (3FN)**

- Estar en segunda forma normal (2FN).
- No puede contener dependencias transitivas.

Una dependencia transitiva se produce cuando el valor de un campo que no es clave depende del valor de otro campo que tampoco es clave.

- **Forma normal de Boyce-Codd (BCFN)**

- Estar en tercera forma normal (3FN).
- Todos los determinantes son a su vez claves candidatas.

Un determinante es un campo o atributo que determina el valor en otro de los campos.

- **Cuarta forma normal (4FN)**

- Estar en la forma normal de Boyce-Codd (BCFN).
- No puede contener una dependencia multivaluada no relacionada (también se le reconoce como dependencia multivaluada independiente).

Se conoce como atributo multivaluado o de valor múltiple a aquel atributo que puede contener dos o más valores para un determinado valor de la clave. Y se produce una dependencia multivaluada cuando dos o más atributos pueden adoptar múltiples valores para un mismo valor de clave y dichos atributos son independientes entre sí.

- **Quinta forma normal (5FN)**

- Estar en cuarta forma normal (4FN).
- Puede contener dependencias multivaluadas no relacionadas ya que la tabla no puede ser fragmentada más sin que dé lugar a pérdida de información.

Un esquema relacional de base de datos que satisface todos los niveles o formas de normalización da lugar a un esquema que arroja la suficiente claridad y donde cada

acontecimiento elemental producido en la misma estará correctamente definido y distinguido del resto.

No obstante, en algunos modelos o diseños se pueden implementar redundancias o agrupaciones de datos por motivos específicos, como pueda ser realizar una búsqueda más con mayor rapidez y aumentar así el rendimiento en la lectura, o necesidades de la propia empresa que la gestiona y violar así algunas de las reglas especificadas en los diferentes grados de normalización. A ese proceso se le conoce como desnormalización.

2.3.3 Sistemas Gestores de Bases de Datos Relacionales (SGBDR)

Los sistemas de gestión de bases de datos relacionales (SGBDR) son servicios o herramientas software que permiten crear, gestionar y administrar las bases de datos relacionales, así como elegir o manejar las estructuras necesarias para almacenar y buscar los datos de la información contenida de la manera más eficiente a la vez que garantiza la seguridad y la integridad de estos. Mediante ellos podemos crear, eliminar, modificar, leer y actualizar (funciones más básicas en cualquier base de datos) la información en nuestra base de datos.

Algunos ejemplos de estos sistemas de gestión más populares y arraigados son: MySQL, Oracle Database, Microsoft SQL Server, PostgreSQL, IBM Db2, SQLite y Microsoft Access. Otro ejemplo es MariaDB. Este sistema está relacionado directamente con MySQL, ya que es una derivación del mismo y comparten la mayoría de las características, aunque incluye algunas extensiones más. Es propiedad de Oracle Corporation [6].

Dependiendo de las características que posea la base de datos que queramos administrar y gestionar, en función del ámbito en el que se vaya a englobar a la misma y reconociendo el abanico de criterios para el manejo de esta, se deberá decir por uno u otro. Aunque en muchos de los casos el principal motivo para decidir la elección entre unos y otros viene residido en el precio del coste que hay que abonar por hacer uso de este, dado que algunos de ellos carecen de gratuidad y exigen de pago.

La **Figura 8**, presente en la siguiente página, muestra un gráfico de la evolución en función de la popularidad de los mismos.

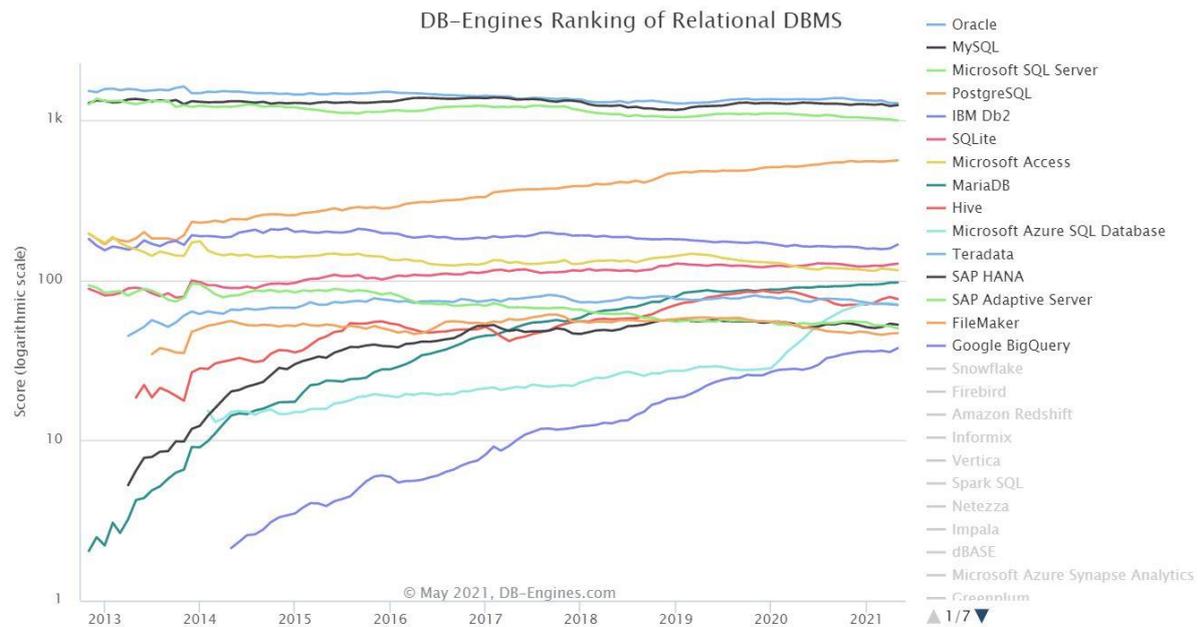


Figura 8. Popularidad de los sistemas gestores de bases de datos relacionales más utilizados
(Fuente: www.db-engines.com)

MySQL

Es un sistema de gestión de código abierto y cuenta con licencia pública y gratuita para poder disfrutar de su uso. En la actualidad es propiedad de Oracle Corporation, aunque cuando se creó se desarrolló como un proyecto independiente a esta sociedad, la cual luego lo absorbió. Su funcionamiento está basado en el modelo cliente-servidor. Hace uso del lenguaje de consulta estructurado SQL (*Structured Query Language*). Soporta compatibilidad multiplataforma con los principales sistemas operativos más utilizados, entre ellos los más destacados del mercado: Linux, Mac y Windows. Su uso no necesita de unos requisitos elevados en cuanto a propiedades de recursos clave del ordenador (CPU, disco duro, RAM...). Este gestor ofrece niveles de rendimiento y eficiencia de los más elevados del mercado, dado que posee una elevada velocidad para realizar las operaciones, a la vez que hace un uso bajo de los recursos. Se trata de un sistema muy flexible y fácil de usar, a la vez que nos proporciona un nivel muy elevado de seguridad. Ofrece una gran capacidad de integración con lenguajes como, por ejemplo, PHP o .NET para dar forma a la creación de innumerables aplicaciones web, la sinergia conseguida de la asociación Apache-MySQL-PHP es de las más utilizadas para ello. En su contra reside que no es un gestor recomendado para hacer uso del mismo con bases de datos de tamaño elevado [7].

Oracle Database

Es el otro principal competidor por alzarse con la corona del sistema de gestión de bases de datos relacionales más usado en la actualidad. Ha sido desarrollado por la compañía Oracle Corporation. Es multiplataforma, puede ser utilizado en los diferentes sistemas operativos instalados. La principal diferencia que existe con MySQL es que se trata de un sistema de gestión de bases de datos que para disfrutar de su amplia variedad de funciones requiere de suscripción por pago, ya que es un software con licencia privada y enfocado más al entorno empresarial, mientras que MySQL es un sistema de código abierto. Aunque actualmente ambos sistemas son propiedad de Oracle Corporation, se les suele reconocer como sistema de gestión gratuito (MySQL) y sistema de pago (Oracle Database) de la misma compañía (aunque MySQL también dispone de versiones de pago) [8].

Microsoft SQL Server

Es un sistema propiedad de Microsoft. Al igual que ocurre con Oracle, este también es un sistema gestor comercial enfocado al ámbito empresarial, por lo que requiere de pago para obtener la licencia de acceso para trabajar con él. Su uso siempre estuvo ligado con sistema operativo Windows, pero desde hace unos años también está disponible para Linux. Basa su funcionamiento en el modelo cliente-servidor [9].

PostgreSQL

Se trata de un sistema de gestión y administración de bases de datos relacionales, enfocado a bases de datos orientadas a objetos. Es multiplataforma, se puede utilizar en todos los principales sistemas operativos del mercado, además de ser un sistema con licencia gratuita y de código abierto. Posee varias interfaces de programación desarrolladas en diversos lenguajes de programación. Es un gestor que trabaja con mayor lentitud cuanto menor tamaño poseen las bases de datos administradas, puesto que su diseño está destinado a gestionar grandes volúmenes de datos [10].

2.3.3.1 Lenguaje SQL

Lo cierto es que cada día más casi la totalidad de los sistemas de gestión de bases de datos relacionales son más compatibles entre sí y comparten muchas de las

funcionalidades. Un motivo de que esto sea posible es que todos ellos comparten el mismo lenguaje estandarizado para realizar las operaciones.

Si nos fijamos, muchos de ellos comparten la palabra SQL en sus nombres. SQL es el lenguaje estándar de programación empleado por los sistemas de gestión de las bases de datos relacionales para realizar la multitud de operaciones que deseamos con los datos. Es un lenguaje estándar, el cual está creado y diseñado para la administración, gestión y recuperación de la información en cualquier base de datos relacional. Es un lenguaje de declaraciones, en el cual indicando las palabras de instrucción reservadas se especifican las órdenes e implícitamente en la declaración de la expresión se denota cual debe ser el resultado esperado [11].

2.3.3.2 Futuro

Actualmente y cada vez con más peso están irrumpiendo con fuerza los sistemas de bases de datos al completo basados en la nube, dejando atrás los sistemas de implementación software como tal. La principal razón asociada al desarrollo de este tipo no es otro que el menor coste para el almacenamiento y mantenimiento de estos, y la comodidad que el uso de ellos supone en los equipos remotos desde los cuales poder tener acceso completo al sistema. Microsoft Azure SQL y Google BigQuery son ejemplos de ellos. Si nos hemos detenido a observar la **Figura 8**, vemos como desde inicios de pasado año 2020, Microsoft Azure SQL ha experimentado un crecimiento más abrupto y mucho mayor al que pueda presentar cualquier otro de los involucrados en el gráfico.

3

La empresa ficticia ACME. Situación de partida y análisis de requisitos

3.1 Introducción

ACME es el nombre de la empresa ficticia que consideramos en el TFG. ACME realiza pruebas de calidad de productos, y quiere modificar e innovar su modo de almacenamiento de los resultados de dichas pruebas, así como las herramientas, de distinta índole, involucradas en ellas y las características o efectos que dichas valoraciones presentarán.

En este capítulo se va a recoger de forma detallada la situación en la que se encuentra la empresa ACME y toda la información del estado en el que actualmente trabaja.

3.2 La empresa ficticia ACME

3.2.1 Descripción General

La empresa ACME es una sociedad ficticia, como anunciaba antes, ubicada en el sector industrial y dedicada a la producción de bienes y prestación de servicios orientados en dicho sector. Actualmente, la empresa goza de presencia física en dos ciudades, ya que cuenta con dos factorías ubicadas en Salamanca y Valladolid, siendo esta última el emplazamiento base con mayor peso en la empresa.

En relación con la producción de bienes, ACME elabora sus propios productos, a los que etiqueta con el nombre de *Productos Propios*, y les realiza un estudio de calidad. Por otra parte, también ofrece la prestación de ese mismo servicio de evaluación de calidad a

productos remitentes de otros proveedores que así lo desean. Esos otros productos son etiquetados como *Productos Externos*.

A día de hoy, ACME colabora y trabaja simultáneamente con diez empresas distintas, a las que presta ese servicio de evaluación de calidad.

Esta labor o tarea de evaluación de la calidad, tanto a sus propios productos previamente elaborados, como a los productos procedentes de otros proveedores, es la actividad en que se centra y desarrolla a lo largo de este TFG. La tarea de producción de bienes y productos por parte de la empresa ACME no es competencia directa de este. Aunque, si bien es necesario, se haga alusión a ella en diversos puntos determinados, ya que, para evaluar la calidad con la que un producto ha sido elaborado, debe previamente pasar por el proceso de constitución del mismo.

Para el testeo de ambos tipos de productos (Productos Propios y Productos Externos), la empresa ACME dispone de una serie de máquinas para realizar el estudio de calidad. En cada factoría tiene una o varias máquinas dedicadas al análisis de calidad de sus propios productos, y una o varias máquinas, distintas de las anteriores, dedicadas al análisis de los productos elaborados por proveedores externos. En concreto, la suma global de las máquinas asciende a ocho, siendo cinco de ellas destinadas a la evaluación de los Productos Propios y las tres restantes para la de los Productos Externos. Si las ubicamos en la geografía empresarial, la factoría de Valladolid alberga un total de cinco máquinas, tres de ellas destinadas para el análisis de sus propios productos y las otras dos para la evaluación de los ajenos. Quedando alojadas en la factoría salmantina las otras tres, dos empleadas para el uso de los productos elaborados en sus propias factorías y la otra, por descontado, para la valoración de los productos remitentes de otros proveedores.

El personal con el que cuenta contratado ACME para la evaluación de la calidad se compone de 15 operadores distribuidos a lo largo de ambas factorías.

Todas y cada una de las máquinas, anteriormente citadas, en las que pueden llevarse a cabo la evaluación de los lotes de los distintos productos, pueden confeccionarse con diferentes configuraciones y softwares, de esta manera la ejecución y evaluación de los análisis no es realizada siempre de la misma forma. Concretamente, la empresa ACME posee cuatro configuraciones distintas para emplear. La mitad prevista para el caso de que las medidas de análisis sean realizadas a los lotes de sus propios productos y la otra mitad

para los lotes de productos derivados de otros fabricantes, los Productos Externos. De la misma manera, tiene en su haber cuatro softwares distintos. Dos para evaluar los Productos Propios y dos para los productos ajenos a su producción.

Las mediciones realizadas en las distintas pruebas de evaluación de la calidad están sometidas bajo diez posibles protocolos de medición. Teniendo, además, como se expondrá más adelante, uno de ellos, la posibilidad de realizar el análisis bajo una de las diez variantes de las que dispone habilitadas.

3.2.2 Descripción Específica. Política de identificación.

Dentro de la empresa se extiende una política interna para identificar a cada uno de los tests realizados, personal contratado, pasando por el material tecnológico empleado, tangible y no tangible, así como diversas configuraciones o sistemas de cálculo utilizados en las distintas valoraciones de la calidad realizadas en cada uno de los testeos. Se puede decir que utiliza una política de identificación globalizada, que ya veremos cómo es implantada en las siguientes líneas.

En primer lugar, cada uno de los tests de análisis de la calidad están identificados de manera única en la máquina en la cual han sido realizados, indistintamente del tipo de producto al que se le realice, ya sea propio o externo. En cada una de las máquinas, cada nuevo test que se realiza incrementa en una unidad el valor del test anterior. La variable que lo almacena, ID TEST, representa un número natural y comienza en valor 1 para la primera de las pruebas acometidas en dicha máquina. Como he dicho es un valor único, en la propia máquina, que identifica a la prueba de análisis realizada y que nunca va a ser reseteado, ya que cada uno de los tests serán posteriormente registrados y almacenados, y deberán permanecer patentes a lo largo del tiempo. En consecuencia, puede haber dos tests realizados con el mismo ID TEST, pero pertenecerán a dos máquinas distintas.

En el caso concreto de las máquinas, al contar con un total de ocho establecidas en ambas factorías, se realiza primeramente una distinción. Por un lado, las que analizarán los productos elaborados dentro de la misma empresa ACME y, por otro, las destinadas a la evaluación de los productos de otros proveedores. La identificación de las máquinas es globalizada, es decir, se representa mediante un número natural que recoge la variable ID MÁQUINA y que es único por cada máquina perteneciente a la empresa ACME y se va incrementando a medida que se van adquiriendo nuevas máquinas. De esta manera, las

máquinas que miden la calidad de los Productos Propios en la factoría de Valladolid son establecidas con los ID MÁQUINA del 1 al 3. Dejando los valores 4 y 5, que toma esa misma variable, para las dos máquinas restantes de la factoría en los que se van a analizar los Productos Externos. Si, por otro lado, nos situamos en la factoría salmantina, los valores 6 y 7 son establecidos para las máquinas que evaluarán a sus propios productos, quedando identificada con el valor 8 para hacer lo mismo, pero en este caso con los productos ajenos a la propia empresa.

Caso análogo ocurre con el personal contratado. Cada uno de los trabajadores están identificados bajo el valor que recibe la variable ID OPERADOR, que al igual que ocurre con las máquinas, se trata de un número natural. Actualmente, como ACME cuenta con 15 operadores y la política de identificación es también globalizada para el caso del personal contratado, estos tomarán dichos valores del 1 al 15, indistintamente de la factoría en la que se encuentren trabajando.

Lo que se ha descrito hasta ahora valdría de forma semejante con las configuraciones y los softwares que las máquinas pueden llevar implementados y que veremos ahora de forma detallada. En el caso de los softwares instalados y con los que operan las máquinas, están representados por el valor, un número natural, registrado en ID SOFTWARE. La empresa ACME dispone concretamente de cuatro variantes informáticas. De todas ellas, la mitad son empleadas para testear los Productos Propios, que se identifican con los valores 1 y 2, y el resto se utilizan para evaluar los Productos Externos, cuyos valores se corresponden con 3 y 4. Sigue siendo indiferente que esté implantado en cualquiera de las máquinas ubicadas en las dos factorías.

Caso especial ocurre con las configuraciones de las que se dispone y que se pueden aplicar a cada una de las máquinas. No especial, ni distinto, en tanto a la forma de identificarlas y distribuirlas, sino porque cada una de ellas lleva otros valores asociados, como veremos en el siguiente apartado de la memoria, denominados SISTEMA DE CÁLCULO, UMBRAL BAJO y UMBRAL ALTO. Son fundamentales para el resultado de la valoración del nivel de calidad de los lotes de productos analizados. La representación e identificación de las distintas configuraciones siguen en la misma línea que todos los descritos anteriormente. ID CONFIGURACIÓN se llama la variable que, en este caso, determinará que configuración ha sido aplicada. El valor sigue siendo un número natural. Hasta la fecha actual, la empresa cuenta con cuatro opciones de configuración distintas,

estableciendo los valores 1 y 2 para las configuraciones de análisis de los productos remitidos por los diferentes proveedores. Y los valores de ID CONFIGURACIÓN 3 y 4 para el caso en el que los productos valorados sean los suyos propios.

Otro de los valores identificados y representados mediante un número natural, y al cual he hecho alusión antes, es el valor de SISTEMA DE CÁLCULO. Determina qué ecuación y cómo van a ser ponderados en esa ecuación la cantidad y diferente tipología de los defectos de calidad encontrados. En este momento, la empresa ACME cuenta con tres sistemas que arrojarán los resultados pertinentes, esos resultados darán lugar a un nuevo valor registrado como NIVEL DE DEFECTOS. Volviendo a las tres posibilidades anteriores, están identificadas con los valores del 1 al 3:

- Si SISTEMA DE CÁLCULO toma el valor 1, la ecuación empleada para calcular el NIVEL DE DEFECTOS será: $NIVEL DE DEFECTOS = 20 \times dC + 5 \times dM + 2 \times dm$. Siendo dC el número de defectos críticos, dM el número de defectos mayores y dm el número de defectos menores.
- Si el valor que adopta dicho campo es el 2, la ecuación se corresponde con: $NIVEL DE DEFECTOS = 10 \times dC + 2 \times dM$. Este modelo de cálculo no tendrá en consideración los defectos menores.
- Por último, si este valor es 3, la ecuación empleada para el cálculo sería: $NIVEL DE DEFECTOS = 10 \times dC + 3 \times dM + dm$.

Ya veremos cómo es empleado más adelante.

Todos y cada uno de los análisis que se efectúan son realizados bajo la medición de un protocolo estrictamente concreto. En este aspecto, la empresa cuenta a su libre disponibilidad de los protocolos identificados por la variable PROTOCOLO MEDICIÓN y que puede tomar los valores: 1, 2, 3, 4 y 5. Teniendo la particularidad de que, si el protocolo empleado es el 1, este cuenta con diversas variantes. La variante elegida en concreto se recoge en la variable VARIANTE PROTOCOLO 1, y van desde la 1 hasta la 10.

Para identificar de qué fabricante o proveedor son los lotes de productos los cuales hay que valorar su calidad de producción, también se sigue un criterio de identificación.

En este caso el número natural que se almacena en la variable que los determina, está recogido en ID PROVEEDOR. Pudiendo tomar los valores comprendidos entre 1 y 10, distinguiendo cada valor a cada uno de los diez proveedores con los que actualmente ACME trabaja y quedando reservado el valor 0 para sus propios productos.

Finalmente, si en un momento dado, la empresa ve necesario, plantea y decide ampliar sus instalaciones, tecnologías, implementaciones, personal, etc., la empresa lo único que tendrá que hacer es ir incrementando el identificador al valor inmediatamente siguiente.

3.2.3 Funcionamiento de la actividad.

Como ya he dicho previamente, la labor de producción de la empresa no es competencia propia del TFG, de manera que no se entrará a detallar en profundidad el ejercicio de esta práctica. Centrémonos entonces ahora, en averiguar la forma de operar para valorar la calidad con la que los productos han sido confeccionados.

Para el desempeño de esta actividad, la empresa basa su desarrollo en las máquinas de las que dispone en cada una de sus factorías. Quedando patente en un registro tanto la FACTORÍA en la que se ha realizado la prueba de valoración, así como el CÓDIGO DE PAÍS en el cual está ubicada la factoría concreta. Se recoge de esta manera, dada una propuesta de valoración para posibilidad de expansión en primer lugar a países más cercanos, como Portugal o Francia, y no descartando en un mayor plazo ampliar las fronteras europeas.

Volviendo de nuevo a las máquinas alojadas en ellas, por ellas transcurren los distintos lotes de los diferentes productos una vez elaborados, si se trata de los suyos propios, o los que han sido enviados hasta sus instalaciones, si son destinados a ese proceso por parte de empresas ajenas. Las máquinas, deben estar configuradas con una determinada configuración, valga la redundancia, a la vez que trabajan con un software concreto.

Todos y cada uno de los tests que realizan los softwares de las máquinas, conforme a las configuraciones previstas para llevarlos a cabo, están supervisados por uno de los operadores presentes en la factoría con el fin de validar o no, dicho test. De este modo quedará reflejado el NOMBRE DE OPERADOR e ID OPERADOR del operario en cuestión que esté designado para realizar dicha labor en ese test. De esta manera, al

finalizar el proceso de valoración de la calidad, este revisa la medida de evaluación y si observa cualquier anomalía que pueda darse o cualquier razonamiento u objeción que no le convenza, debido a que pueda haberse dado el caso de producirse algún error, este puede realizar y comprobar nuevamente la medida manualmente. Así, de esta manera modificaría la previamente realizada por el software de la máquina. En consecuencia, un operador puede modificar o no las medidas que realice el software implementado en una máquina.

Una vez realizado el test, este quedará registrado mediante la fecha y hora en la que ha llevado a cabo su cometido, dichos valores se almacenan en el registro que tiene mismo nombre, FECHA Y HORA. A su vez, se detallará a cada uno de ellos con un nuevo identificador, denominado CÓDIGO DE TEST, compuesto en primer lugar, por los cuatro dígitos del año, seguidos por los dos dígitos del mes y continuando con los del día en el que se ha realizado. Finalmente, se les añadirá a los valores anteriores, el número de prueba que ha realizado la máquina a lo largo de ese día, en formato de tres dígitos y rellenando los valores a la izquierda con ceros si así fuese necesario. Para reconocerlo e identificarlo más claramente pondremos un ejemplo: supongamos que una máquina realiza el cuarto test del día 13 de Julio de 2020. Entonces ese test llevará asociado como CÓDIGO DE TEST el valor 20200713004.

Otra de las características relacionadas con las marcas temporales de los tests, es la franja horaria en la que han sido desempeñados, dado que, unido a la política temporal de la empresa, en función del instante de tiempo en el que se desarrolle el test, pertenecerá a uno de los tres turnos establecidos. Los tests realizados entre las 07:00:00 horas, incluida esta, y las 15:00:00 horas de un mismo día pertenecen, y quedarán reflejados, al TURNO de mañana. Si han sido efectuados a partir de las 15:00:00 horas, hora válida para este turno, pero antes de las 23:00:00 horas se corresponden con el TURNO de tarde. Y finalmente si han sido llevados a cabo y están registrados desde las 23:00:00 horas, hora incluida, hasta antes de las 07:00:00 horas, su ubicación es el TURNO de noche.

El propósito de los tests puede estar destinado en base a tres tipos de los que puede ser objeto: análisis, investigación o producción. Que se almacenará o recogerá en un campo denominado TIPO DE TEST.

Como se dijo anteriormente, previamente a los análisis de evaluación de la calidad, los lotes de los diferentes productos son emplazados en las diversas máquinas. Cada uno

de esos lotes de productos lleva consigo un identificador de lote asociado, ID LOTE, y poder ser así reconocida su partida de fabricación. De la misma manera ocurre con los productos, cada producto concreto con el que se conforman los lotes posee su identificador individual, este será ID PRODUCTO. Un lote está formado por un conjunto de elementos. La política de identificadores de lotes y de productos dependen de las empresas que los generan. Veámoslo de forma más intuitiva con un ejemplo: supongamos que tenemos una fabricación de tornillos. En este caso los tornillos serán nuestro producto concreto y tienen como ID PRODUCTO = 4C. La producción de tornillos de un día concreto (1000 cajas de 100 tornillos cada una) constituye un lote, y cada caja es lo que denominaremos elemento. Así pues, por ejemplo, la producción de tornillos (producto con ID PRODUCTO = 4C) del 4 de enero de 2021 constituye el lote con identificador ID LOTE = 12AB de dicho producto. Este lote está constituido a su vez por 1000 elementos (las 1000 cajas).

Para llevar a cabo la evaluación de la calidad, se realiza un análisis de un número concreto de elementos (cajas en nuestro ejemplo) que componen el lote de un determinado producto, que se registrará en el campo NÚMERO DE ELEMENTOS ANALIZADOS, pero ese número concreto de elementos a analizar puede variar en los diferentes tests ejecutados, de la misma manera que pueden variar los elementos que conforman el lote del producto de partida. El análisis de esos elementos puede realizarse de forma aleatoria u ordenada. Si el modo aplicado, el cual quedará reflejado en otro campo denominado MODO DE ANÁLISIS, es el aleatorio, se seleccionan al azar el número de elementos que previamente anunciábamos de entre todos los elementos que conforman el lote y se lleva a cabo el análisis de estos (por ejemplo, se seleccionan aleatoriamente 20 de las cajas de tornillos que constituyen cada lote). Si el modo elegido es el ordenado, el análisis se realizará de los primeros elementos del lote (en el ejemplo anterior, las 20 primeras cajas).

En las distintas pruebas de evaluación de los productos se van analizando y valorando el número de defectos presentes en cada uno de los elementos que forman el lote y de que naturaleza son. La cantidad y tipología de los posibles defectos presentes se gradúa en una escala como la siguiente: NÚMERO DE DEFECTOS MENORES (por ejemplo, cuántos tornillos de la caja tienen ligeras imperfecciones), NÚMERO DE DEFECTOS MAYORES (por ejemplo, cuántos tornillos tienen defectos apreciables, pero pueden utilizarse sin problemas) y NÚMERO DE DEFECTOS CRÍTICOS (tornillos con defectos que impiden su uso).

Una vez contabilizados y clasificados los defectos detectados, se procede a calcular el valor global resultante de la prueba de análisis, que quedará almacenado en el campo NIVEL DE DEFECTOS.

Para establecer el criterio de evaluación, las diferentes configuraciones llevan asociados directamente tres parámetros que darán forma al resultado del test, cómo bien anunciaba en el punto previo. El primero de ellos es el SISTEMA DE CÁLCULO. En función de él, nos determinará qué ecuación va a ser aplicada para obtener el NIVEL DE DEFECTOS. Los otros dos valores asociados son dos umbrales, UMBRAL BAJO y UMBRAL ALTO. Estos dos valores limitarán y determinarán la CALIDAD resultante de la fabricación del lote de productos.

Es importante destacar que a mayor valor de NIVEL DE DEFECTOS peor habrá sido el proceso de elaboración, y, en consecuencia, peor nivel de CALIDAD otorgado. Y viceversa, si esa cantidad es menor, nos estará indicando que se ha detectado menor presencia de defectos y, por lo tanto, mejor CALIDAD presenta.

El procedimiento posterior, es comparar el valor obtenido de NIVEL DE DEFECTOS con los umbrales anteriormente definidos. Dependiendo de la comparación con los dos umbrales, UMBRAL BAJO y/o UMBRAL ALTO, el resultado arrojado será superior, inferior o quedará limitado entre ambos. Lo cual nos determinará la CALIDAD resultante con la que han sido elaborados los diferentes lotes de productos. El método de comparación es el siguiente:

- Se compara el valor de NIVEL DE DEFECTOS con el UMBRAL BAJO establecido y si el resultado es menor al valor del umbral, la CALIDAD con la que ha sido elaborado ese lote de productos es alta.
- Si lo que ocurre es lo contrario, se compara el valor de NIVEL DE DEFECTOS con el otro umbral, UMBRAL ALTO. De esta nueva comparación, se pueden obtener dos resultados, que el valor sea mayor, o que sea menor al umbral. Si ocurre el primero de los resultados, la CALIDAD determinada será automáticamente baja. Si no es así, y el valor es menor al UMBRAL ALTO, entonces el valor quedará limitado entre ambos umbrales, lo que nos hace indicar que la CALIDAD en este caso será media.

- Reflejar que, si el valor de NIVEL DE DEFECTOS adopta el mismo valor que cualquiera de los dos umbrales, la CALIDAD será media.

No obstante, si no se está de acuerdo con la resolución o procedimiento llevado a cabo en cada uno de los tests realizados, cada uno de ellos tiene la posibilidad de verse repetido con otro nuevo test. En este caso la nueva prueba tendrá una mayor duración en su procesado, pero arrojará una mayor fiabilidad en los resultados. Esta posibilidad quedará patente en un campo, llamado TEST EXTENDIDO, que puede tomar los valores de sí o no.

Finalmente, por cada test realizado existe también un campo en el que se puede anotar y describir cualquier comentario, objeción o explicación que haya dado lugar en el desarrollo del test. Este campo está identificado con el nombre de COMENTARIOS.

3.2.4 Almacenamiento de datos. Archivo Excel.

Durante el proceso de la actividad anteriormente descrita, entran en escena todas las variables y campos presentados, tomando valores concretos en la realización de los sucesivos análisis de calidad.

Actualmente, tanto los resultados obtenidos, como las configuraciones, softwares, preferencias o las diferentes opciones que han sido seleccionadas a la hora de realizar las pertinentes pruebas de análisis y valoración del nivel de calidad con el que han sido elaborados los productos resultantes, tanto propios como externos, son recogidos y almacenados en un archivo Excel.

Como ya he anunciado en anterior ocasión, parte del propósito global de este TFG es que todos estos datos almacenados y recogidos en dicho archivo Excel, sean migrados y, por lo tanto, estén presentes en la base de datos diseñada y creada.

Una vez dicho lo anterior, y entrando en detalle en lo que nos concierne en este punto de la memoria, veamos cuál es la estructura, distribución y contenido del archivo Excel.

Sabemos ya que la empresa realiza pruebas o tests de evaluación del nivel de calidad a productos elaborados por ellos mismos o bien someten a dichas valoraciones a productos derivados de otras empresas ajenas. Entrando a detallar la estructura del archivo, en él existen varias hojas en las que se albergan únicamente las pruebas realizadas a los

Productos Propios y otras en las que se reúnen las que son efectuadas a los Productos Externos, diferenciando también en esas mismas hojas las factorías en las que han sido ejecutadas. En concreto, el archivo consta de cuatro hojas. La primera de ellas refleja las pruebas sometidas a los Productos Propios que se han realizado en la factoría de Valladolid. La siguiente hoja recoge los tests efectuados al mismo tipo de productos pero que han sido llevados a cabo en la factoría salmantina. En la tercera de las hojas, los tests almacenados son los realizados por las tecnologías de las instalaciones situadas en la factoría vallisoletana, esta vez, a los Productos Externos. Para finalizar con la cuarta de las hojas registrando los análisis a los productos provenientes de otras empresas y que han sido llevados a cabo en la factoría de Salamanca.

La estructura y distribución de cada una de las hojas es siempre la misma, estableciendo en la primera de las filas el nombre o enunciado del campo que, a continuación, en la segunda fila y sucesivas, tomará el valor concreto que haya sido decretado, ya sea como parámetro de configuración, de preferencia, de consecuencia... o arrojado, si es un valor resultado de la prueba ejecutada. Es decir, la primera fila alberga los enunciados de los campos por lo que quedarán registrados todos los componentes que son partícipes, directa e indirectamente, en la realización de los tests de evaluación de la calidad de los productos elaborados. Posteriormente, todos los valores de una misma fila reflejarán y detallarán las identificaciones involucradas en un mismo test.

Para entender de manera más intuitiva lo dicho anteriormente, a continuación, se detalla un breve resumen y dos tablas en las que se recogen y muestran todos los valores a los que hacía referencia en el párrafo anterior:

El fichero Excel contiene cuatro hojas, correspondientes a PRODUCTOS PROPIOS, una hoja por cada factoría, y a PRODUCTOS EXTERNOS, de la misma manera una hoja por cada factoría. De esta forma el fichero Excel está distribuido de la siguiente manera:

1. 1ª Hoja: **PRODUCTOS PROPIOS** en la factoría de **VALLADOLID**.
2. 2ª Hoja: **PRODUCTOS PROPIOS** en la factoría de **SALAMANCA**.
3. 3ª Hoja: **PRODUCTOS EXTERNOS** en la factoría de **VALLADOLID**.
4. 4ª Hoja: **PRODUCTOS EXTERNOS** en la factoría de **SALAMANCA**.

La distribución de las hojas Excel en el caso de los PRODUCTOS PROPIOS, sería cómo se recoge en la siguiente **Tabla 2**:

COLUMNA	CAMPO	DESCRIPCIÓN	VALORES
A	FACTORÍA	Nombre de la factoría en la que se realiza el test.	VALLADOLID / SALAMANCA
B	CÓDIGO DE PAÍS	Código del país donde está ubicada la factoría.	ES
C	ID MÁQUINA	Número natural. Identifica la máquina que realiza el test. Es único a nivel global de la empresa. No hay dos máquinas con el mismo identificador.	1, 2, 3...
D	ID TEST	Número natural correlativo que identifica la prueba de análisis. No se resetea jamás. En cada máquina empieza desde 1.	1, 2, 3...
E	CÓDIGO DE TEST	Ej: Formato 191216003 (3ª prueba hecha el 16/12/19). Nótese que distintas máquinas pueden tener los mismos códigos de test.	20210518001 - 20210522004
F	FECHA Y HORA	Fecha y hora de realización del test.	18/05/2021 00:02:45 - 22/05/2021 21:16:35
G	OPERADOR	Nombre y apellidos de la persona que supervisa la prueba de evaluación.	NOMBRE APELLIDOS
H	ID OPERADOR	Número natural. Único por cada operador a nivel global de la empresa.	1, 2, 3...
I	TURNO	Mañana, tarde, noche.	MAÑANA / TARDE / NOCHE
J	TIPO DE TEST	Investigación, Producción, Análisis.	INVESTIGACIÓN / PRODUCCIÓN / ANÁLISIS
K	ID SOFTWARE	Número natural. Identifica el software con el que opera la máquina.	1, 2, 3...
L	ID CONFIGURACIÓN	Número natural. Refleja la configuración concreta de la máquina.	1, 2, 3...
M	ID PRODUCTO	Valor alfanumérico. Identifica el producto.	1A / 2B / 3C / 4D / 5E / 6F / 7G / 8H / 9I
N	ID LOTE	Valor alfanumérico. Identifica el lote de producción.	12AB / 34CD / 56EF / 78GH / 90IJ

COLUMNA	CAMPO	DESCRIPCIÓN	VALORES
O	PROTOCOLO DE MEDICIÓN	1, 2, 3, 4, 5.	1 - 5
P	VARIANTE PROTOCOLO 1	Número natural. En el caso anterior de emplearse el protocolo 1, este puede haber utilizado una de sus variantes 1, 2, ..., 10	1 - 10
Q	MODO DE ANÁLISIS	Ordenado, Aleatorio. Si se eligen los elementos a analizar de forma correlativa (ej. primeros 10 elementos) o aleatoria (10 elementos al azar)	ORDENADO / ALEATORIO
R	NÚMERO DE ELEMENTOS ANALIZADOS	Número de elementos del lote que se analizan.	1, 2, 3, ...
S	NÚMERO DE DEFECTOS MENORES	Número total de defectos menores detectados en el test.	0, 1, 2, ...
T	NÚMERO DE DEFECTOS MAYORES	Número total de defectos mayores detectados en el test.	0, 1, 2, ...
U	NÚMERO DE DEFECTOS CRÍTICOS	Número total de defectos críticos detectados en el test.	0, 1, 2, ...
V	NÚMERO TOTAL DE DEFECTOS	Suma total de todos los tipos de defectos anteriores.	0, 1, 2, ... (Suma de los tres anteriores)
W	SISTEMA DE CÁLCULO DE CALIDAD	Número natural que identifica qué fórmula se utiliza para calcular la calidad del lote en función de los defectos encontrados y su tipología.	1, 2, 3...
X	NIVEL DE DEFECTOS	Valor numérico real calculado en función del tipo de defectos y el sistema de cálculo de calidad.	Número real
Y	CALIDAD	ALTA/MEDIA/BAJA Compara el nivel de defectos con dos umbrales definidos.	ALTA / MEDIA / BAJA
Z	TEST EXTENDIDO	SI/NO. La prueba extendida tiene mayor duración que la estándar, pero tiene más fiabilidad en los resultados.	SI / NO

COLUMNA	CAMPO	DESCRIPCIÓN	VALORES
AA	COMENTARIOS	Texto. Observaciones y comentarios relacionados con el test.	AAAA - ZZZZ

Tabla 2. Distribución en Excel para Productos Propios

La distribución de las hojas del archivo Excel en el caso de que los productos valorados sean los PRODUCTOS EXTERNOS, es análoga a la anterior, pero se incluye al final de los anteriores un campo adicional que es: ID PROVEEDOR, toma valores 1, 2, 3... e identifica al proveedor externo que ha fabricado el producto testeado (Tabla 3):

COLUMNA	CAMPO	DESCRIPCIÓN	VALORES
...
AB	ID PROVEEDOR	Número natural. Identifica al proveedor al que pertenece el lote analizado.	1, 2, 3...

Tabla 3. Distribución en Excel para Productos Externos

3.2.5 Información adicional.

Conjuntamente al archivo Excel anteriormente descrito, existen determinados datos de configuración, software y proveedores asociados a la empresa, recogidos paralelamente en un cuaderno electrónico de notas del que también hace uso la empresa. En él recoge características de descripción y parámetros concretos asociados a cada uno de los softwares o configuraciones que posee y puede implantar en las máquinas de la empresa en un momento dado, a la vez que hace lo mismo para identificar el nombre del proveedor con su identificador asociado. A algunos de ellos ya he hecho alusión en anteriores ocasiones.

Los valores asociados a los diferentes softwares son:

1. **ID SOFTWARE:** Número natural que identifica un software concreto.
2. **NOMBRE:** Apelativo descriptivo del software.
3. **VERSIÓN:** Identificación de la versión del software.
4. **PARÁMETRO A:** Valor de ajuste del software empleado.
5. **PARÁMETRO B:** Valor de ajuste del software empleado.

Por parte de la configuración vienen definidos los siguientes:

1. **ID CONFIGURACIÓN:** Número natural que identifica una configuración concreta.
2. **NOMBRE:** Apelativo descriptivo de la configuración.
3. **SISTEMA DE CÁLCULO:** Número natural que identifica el procedimiento de evaluación empleado para calcular la calidad final del lote analizado.
4. **UMBRAL BAJO:** Valor de referencia fijado para la valoración de calidad como alta o media.
5. **UMBRAL ALTO:** Valor de referencia fijado para la valoración de calidad como media o baja.

En el caso de los proveedores asociados los valores anotados en el cuaderno electrónico son:

1. **ID PROVEEDOR:** Número natural que identifica un proveedor concreto.
2. **NOMBRE PROVEEDOR:** Apelativo descriptivo del proveedor.

De manera que a la hora de almacenar la información relativa a la realización de las diversas pruebas en el archivo Excel, lo que aparece reflejado en él haciendo referencia a todo lo aquí arriba plasmado son los valores de ID CONFIGURACIÓN, ID SOFTWARE e ID PROVEEDOR los cuales llevan asociados directamente e identifican cada una de las descripciones anteriores.

3.2.6 Implementaciones futuras

En este apartado se reflejarán algunos de los planteamientos y valoraciones llevadas a cabo por la empresa ACME y que ha decidido implantar para llevar a cabo la nueva forma de almacenamiento, en la nueva base de datos creada, y extender el funcionamiento anteriormente expuesto en la sección 3.2.3.

La primera de ellas es que todas las modificaciones, o no alteraciones, que cualquier operador realice quedarán reflejadas en otra variable la cual almacenará un carácter. Esta

variable tendrá el nombre de **VALIDACIÓN** y puede recoger cualquiera de los siguientes valores: V (Validado por el operador tras revisar el test), M (Modificado por el operador tras revisar el test), N (No revisada por el operador) y NULL.

- Si el carácter presente en esta variable es V, quiere decir que el test ha sido llevado a cabo por el software y el operador lo ha validado dando por buenas las medidas de análisis producidas en él.
- Si, en lugar de presentar el valor anterior, presenta una M, nos estará indicando que dicha valoración ha sido realizada por el software, pero posteriormente se ha visto reevaluada por el operario allí presente y, en consecuencia, ha sido modificada.
- Si el valor recogido es el carácter N, determina que la prueba de análisis ha concluido y ha sido ejecutada por el software, pero en este caso no ha sido revisada por ningún operador.
- Por último, aquellos tests de los que se desconozca la naturaleza y supervisión de los mismos, serán reconocidos con el valor NULL. Este último valor, será el que reflejen todos los tests almacenados en el archivo Excel a la hora de ser migrados y recogidos en la nueva base de datos diseñada, dado que hasta la fecha del nuevo diseño era una consideración que no había sido llevada a cabo.

Otra de las posibilidades que de cara al futuro se va a implementar, es el desglose o distinción de cada uno de los defectos y del tipo al que pertenezca, identificados en cada uno de los elementos analizados en los tests realizados. Esto significa que todos los elementos analizados de un mismo lote serán identificados y cada uno de ellos llevará asociado el número y tipo de defectos que le han sido detectados. Hasta ahora, el registro de la cuantía y tipo de defectos detectados en el test de análisis reflejaba el sumatorio global de todos ellos, pero no identificaba de ninguna manera los elementos en los que habían sido hallados. Esto es lo que se pretende con esta nueva futura implantación, indistintamente de cómo sea el modo de análisis llevado a cabo en disco test. Veamos más claro con un ejemplo, lo que se reconoce a día de hoy y hasta dónde se quiere extender y destacar exhaustivamente en cada análisis. Imaginemos que en la máquina con ID MÁQUINA = 3 se ha llevado a cabo un test de análisis de 64 elementos, identificado con

ID TEST = 8, el cual ha arrojado como resultados: NÚMERO DE DEFECTOS MENORES = 3, NÚMERO DE DEFECTOS MAYORES = 3 y NÚMERO DE DEFECTOS CRÍTICOS = 1. Estos serían los números o datos que se reflejarían del resultado del test tal y como viene haciendo históricamente ACME, junto con el resto de parámetros e identificadores, expuestos a lo largo de los anteriores apartados 3.2.2, 3.2.3 y 3.2.4, que describen los detalles del mismo. Pues bien, aplicando esta nueva funcionalidad, ahora se reflejará cada uno de los elementos analizados en el test, identificados mediante una variable recogida bajo el nombre de ID ELEMENTO y llevarán asociados el número de defectos ligados a la tipología que presenta cada uno de ellos. Siguiendo con el ejemplo anterior, ahora el test ID TEST = 8, realizado en la máquina ID MÁQUINA = 3, llevará asociado consigo cada uno de los ID ELEMENTO (ID ELEMENTO = 1, ID ELEMENTO = 2... ID ELEMENTO = 64) que identifican a los 64 elementos que conforman el análisis y cada uno de ellos presentará su valor correspondiente a NÚMERO DE DEFECTOS MENORES, NÚMERO DE DEFECTOS MAYORES y NÚMERO DE DEFECTOS CRÍTICOS que hayan sido revelados en él.

De tal manera, que sumando el global de todos los defectos presentes en cada elemento, se obtendrá el mismo valor que presentaban anteriormente las variables NÚMERO DE DEFECTOS MENORES = 3, NÚMERO DE DEFECTOS MAYORES = 3 y NÚMERO DE DEFECTOS CRÍTICOS = 1.

En resumen, cada elemento analizado será identificado con un **ID ELEMENTO**, compartiendo los mismos ID MÁQUINA e ID TEST que corresponda, y llevará consigo asociado un NÚMERO DE DEFECTOS MENORES, NÚMERO DE DEFECTOS MAYORES y NÚMERO DE DEFECTOS CRÍTICOS que reflejarán la cantidad de estos que han sido hallados en cada uno de los elementos que han participado en el análisis global del test.

4

Diseño de la Base de Datos

4.1 Introducción

Una vez conocida la forma y el modo en cómo opera la empresa, se pueden abstraer de todo ello las directrices y principales particularidades para afrontar el siguiente paso. Este no es otro que el diseño de la base de datos para almacenar todos los datos actualmente registrados en el archivo Excel y también los establecidos en el cuaderno de notas.

Tras un primer análisis superficial, lo primero que se puede deducir es que existe una importante relación directa entre muchos de los datos involucrados y arrojados en cada una de las pruebas de análisis. Como ya he anunciado en anterior ocasión el diseño de la base de datos estará destinado a crear una base de datos relacional.

Para el diseño y desarrollo de creación de una base de datos, ya sea relacional o de cualquier otro tipo, previamente es aconsejable que se marquen una serie de fases o etapas, mediante las cuales se va confeccionando y perfilando la misma. Para este caso concreto, las etapas llevadas a cabo vienen dadas de la siguiente manera y orden (**Figura 9**) [3] [4] [5]:

1. **Planteamiento realizable.** Análisis de viabilidad y posibilidad de llevar a cabo la creación de la base de datos.
2. **Análisis de requisitos.** Estudio de las necesidades, condiciones y capacidades operacionales que debe poseer o cumplir la base de datos.
3. **Diseño.** Creación de prototipos de esta en función de las ideas extraídas en la etapa previa y decisión por uno de ellos. También se podría decidir continuar trabajando sobre varios, aunque lo recomendable y más razonable es que finalmente se decante por la implantación de una base de datos y no varias.

4. **Implementación.** Puesta en marcha del diseño creado en la etapa previa en el marco operativo.
5. **Validación y pruebas.** Acciones de evaluación y testeo para comprobar y conocer el nivel de robustez del diseño implementado. En esta etapa también se van perfilando y puliendo errores o fallos posibles, antes de su implantación final.
6. **Implantación y funcionamiento.** Establecimiento, instalación y puesta en ejecución de la base de datos.

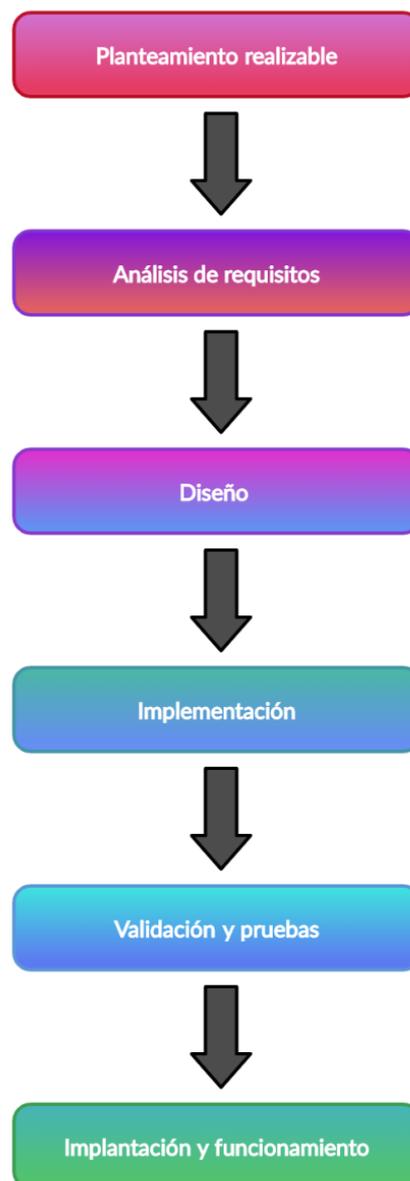


Figura 9. Fases de diseño y desarrollo de una base de datos

En todas y cada una de las etapas de elaboración de la base de datos, se ha de tener en valor y perdurar de una a otra la eficiencia, integridad, seguridad, aplicabilidad y flexibilidad de las operaciones en la misma.

4.2 Despliegue de requisitos

Hasta ahora ACME está trabajando con una manera de actuar bien definida, pero tiene previsto extender e implantar otras funcionalidades como hemos sabido de todo ello con anterioridad en el Capítulo 3. Por ello se debe tener en cuenta previamente un análisis de esas funcionalidades, actuales y futuras, para posteriormente valorar y definir cómo serán aplicadas en el diseño de la base de datos a desarrollar.

4.2.1 Funcionalidades actuales

Si estudiamos más profundamente la política de identificación que emplea de manera interna la empresa ACME, podemos extraer varias conclusiones:

- Todas y cada una de las máquinas con independencia de estar alojadas en cualquiera de las factorías y del tipo de producto que analicen (*Productos Propios* o *Productos Externos*) llevan un identificador único a nivel global de la empresa, ID MÁQUINA.
- Cada uno de los operadores está identificado de manera global mediante una variable denominada ID OPERADOR, con independencia de la factoría en la que se halle trabajando. Una aplicación lógica implementable es la asociación del nombre y apellidos del operador concreto directamente a cada uno de los valores que tome la variable identificativa ID OPERADOR.
- Otra de las determinaciones globales que emplea la empresa ACME es el reconocimiento de los distintos softwares mediante el valor de ID SOFTWARE. Si, además, recordamos que hacía uso de un cuaderno electrónico de notas para apuntar los parámetros y características adicionales asociados a cada uno de los softwares identificados, se puede crear una relación más directa en la que queden patentes todas estas asociaciones.

- El mismo caso que nos planteaba el software, lo presenta la configuración. Esta vez identificado por ID CONFIGURACIÓN. La solución planteada es idéntica a la que se ha definido previamente con el software.
- Finalmente, y como en los dos casos anteriores, la identificación de los distintos proveedores que envían sus productos a la empresa ACME sigue la misma línea anterior, esta vez representada por ID PROVEEDOR.

Otra de las particularidades es que cada máquina es independiente de la otra y en cada una de ellas se realizan diferentes tests de evaluación, con independencia de los que se hayan desarrollado en el resto de las máquinas. Los tests son registrados individualmente con un valor natural, empezando en 1, y van aumentando consecutivamente en una unidad por cada uno de ellos realizado.

Si enfocamos nuestra atención en los protocolos de medición de los cuales ACME puede hacer uso para realizar sus diversas pruebas de valoración de la calidad, nos encontramos con que de los cinco que posee, el identificado con el valor 1, tiene 10 variantes posibles, de las cuales una de ellas será la elegida. Por lo tanto, cuando el protocolo aplicado sea este concreto, es también un factor para tener en cuenta a la hora de desarrollar el diseño.

Entre las pruebas de evaluación podemos diferenciar entre las que realizan a los propios productos que ACME previamente elabora, o si esos tests han sido generados a raíz de analizar los productos derivados de otros proveedores. Todo ello podemos averiguarlo si observamos las máquinas que han procesado esos análisis, ya que unas están destinadas a un tipo de productos y otras al otro. Pero una distinción importante dentro de los tests realizados al segundo tipo de productos antes mencionado, Productos Externos, es determinar a qué proveedor pertenece dicho lote de productos. Por ello es importante asociar y relacionar la variable ID PROVEEDOR con cada uno de ellos.

Siguiendo en la misma línea de los ejemplos anteriormente expuestos, se debe de tener en cuenta también para diseñar las tablas de la base de datos dónde registrar y almacenar las características, variables descriptivas, prestaciones... que describen y caracterizan la totalidad de los términos relacionados con un mismo test. Ejemplos de ello pueden ser: FECHA Y HORA, TURNO, ID LOTE, ID PRODUCTO, PROTOCOLO DE

MEDICIÓN, MODO DE ANÁLISIS, NÚMERO DE ELEMENTOS ANALIZADOS, etc...

Todo ello se reconocerá y entenderá de mejor manera a la hora de plasmar el diseño de la propia base de datos, más adelante.

4.2.2 Funcionalidades futuras

Como se anunciaba previamente en el capítulo anterior, ACME ha decidido después de estudiar y valorar, extender y profundizar en el modo de analizar los productos. Dichas consideraciones son: la diferenciación entre los tests con las medidas de análisis realizadas automáticamente, pero revisadas por el operador (que pueden ser validadas o modificadas), de las no revisadas por él. La otra se trata de la identificación de cada elemento que ha formado parte del análisis del conjunto global del test y la determinación del número y grado al que pertenecen los defectos padecidos y detectados en cada uno de dichos elementos.

En la primera de las consideraciones se pretende registrar la validación, modificación o falta de supervisión y revisión por parte de los operadores al mando a la hora de que las medidas de análisis llevadas a cabo en los tests sean determinadas por el software ejecutado en la máquina. Para ello debemos dejar constancia en nuestro diseño de la base de datos, de manera que cada prueba realizada lleve asociado uno de entre los valores posibles que pueda darse en los diferentes casos y que ya se expusieron en el capítulo anterior, más en concreto, en el apartado 3.2.6.

La otra nueva implantación es la separación e identificación de la cantidad de defectos y el grado al que pertenecen, de todos los defectos que aparezcan y estén presentes en cada elemento que forme parte de cada uno de los análisis realizados. Para ello deberemos introducir un nuevo atributo o variable, el cual identifique por sí solo a cada uno de los elementos del lote que acabará siendo objeto de análisis. Aunque ya ha sido anunciado en alguna ocasión, vuelvo a adelantar que se tratará del campo ID ELEMENTO. Esta nueva segregación se tendrá en cuenta tanto para las medidas de análisis realizadas por el software y validadas directamente como para las que no han seguido el mismo camino y han tenido que ser modificadas y/o corregidas. Hasta ahora nada de esto había ocurrido, y solo estaban registrados los totales de la suma global por cada grado o tipo de defecto del total de elementos analizados.

4.3 Diseño

Una vez realizado el estudio y la valoración de las consideraciones previas o requisitos de partida, estamos en condiciones de abordar la siguiente fase que no es otra que comenzar a esbozar y realizar el diseño que adoptará la base de datos. A continuación, veremos de forma más clara e intuitiva todo lo recogido en el punto anterior y la manera de relacionar todo ello mediante la presentación del diseño llevado a cabo.

Teniendo en cuenta las consideraciones y particularidades anteriores, se ha optado por realizar un diseño de la base de datos normalizado en tercera forma normal (3FN).

El caso más común y popular para almacenar y recoger los datos en algunos de los diversos campos extendidos a lo largo de todas las tablas es el de cadenas de caracteres de diferente longitud. A la hora de considerar el almacenamiento de los diferentes datos contemplados en cadena de caracteres se ha determinado seguir el siguiente criterio por motivos de eficiencia en el rendimiento y la rapidez de realizar las consultas posteriormente a los datos recogidos en dichos campos y en cuanto al ahorro del espacio de la memoria empleada para su almacenamiento [12]. Primando el rendimiento ya que el objetivo y fin de la base de datos es dar soporte a un cuadro de mando, el cual deberá ejecutar innumerables consultas para realizar las diferentes operaciones que se quieran llevar a cabo, pero sin descuidar un gasto excesivo y en vano de la memoria en el momento de almacenar dichos datos. Tratando de conseguir el equilibrio entre ambas justificaciones.

- Si la cantidad de caracteres que se van a almacenar en el campo de un registro es fija, siempre se utilizará CHAR para ese atributo.
- Si la cantidad de caracteres a almacenar en dicho campo del registro varía, entonces se usará VARCHAR.

Es notable que los diferentes identificadores de los distintos registros sean almacenados como un número natural (UNSIGNED INT), dado que no tiene mucho sentido que estos mismos tomen valores negativos. De la misma manera ocurre con los valores de aquellos datos a almacenar cuyo valor negativo no tenga ningún sentido y significado dentro del marco operativo dónde se desarrolla la actividad de ACME, para cuyo objetivo es el diseño de la presente base de datos.

En el caso de que el valor o parámetro a registrar sea un número decimal, la naturaleza del atributo para recoger dicho dato será de tipo “coma flotante” o decimal (FLOAT).

La manera de ilustrar el diseño será una explicación individual de cada una de las tablas que albergaran los numerosos registros con sus correspondientes datos y que en conjunto conforman nuestra base de datos, para posteriormente exponer el resultado global de todas ellas y las relaciones que se atañen entre ellas y que hacen posible la robustez y trazabilidad del diseño de la base.

4.3.1 Tablas

La base de datos se compone de diez tablas independientes, de tal manera que al vincularse y crearse las relaciones pertinentes entre ellas se conforma finalmente nuestra base de datos, relaciones que más adelante se verán. De momento vamos a centrarnos en cómo están estructuradas cada una de las tablas.

La manera de presentar cada una de ellas seguirá el siguiente orden:

1. Una breve descripción de la tabla junto con la representación de la misma.
2. Una tabla resumen en la que se recogen las columnas o atributos que forman parte de ella junto con los comentarios pertinentes y explicativos sobre alguno de ellos concretos que sea necesario subrayar. En la tabla se especificará el nombre del atributo, una breve descripción de este, el tipo de dato que recoge y si puede tomar valor nulo. En la misma tabla junto al nombre aparecerá un icono de llave primaria si es que ese atributo es o forma parte de ella.
3. Finalmente se repetirá el mismo procedimiento, pero en lugar de los atributos, los protagonistas ahora serán los índices y claves de cada una de las tablas partícipes.

4.3.1.1 Tabla tests

En primer lugar, empezaremos por la tabla principal, desde la cual surgen o giran en torno a ella el resto. Esta tabla está titulada con el nombre de *tests*, ya que en esta tabla se almacenarán todos y cada uno de los tests que han sido llevados a cabo en las diferentes

máquinas y que están recogidos en el archivo Excel debido al desempeño de la actividad. Recogerá las principales características y particularidades que describen la realización de las diferentes pruebas y las consecuentes prestaciones que han arrojado los resultados de las mismas. Podemos verla a continuación en la **Figura 10**.

acme tests	
id_maquina	: int(11) unsigned
id_test	: int(11) unsigned
fecha_hora	: datetime
codigo_test	: char(11)
id_operador	: int(11) unsigned
id_software	: int(11) unsigned
id_configuracion	: int(11) unsigned
turno	: varchar(6)
tipo_test	: varchar(13)
id_proveedor	: int(11) unsigned
id_producto	: varchar(20)
id_lote	: varchar(20)
protocolo_medicion	: varchar(3)
modo_analisis	: varchar(9)
numero_elementos_analizados	: int(11) unsigned
defectos_menores	: int(11) unsigned
defectos_mayores	: int(11) unsigned
defectos_criticos	: int(11) unsigned
nivel_defectos	: float
calidad	: varchar(5)
validacion	: char(1)
test_extendido	: char(2)
comentarios	: text

Figura 10. Tabla tests

Como se observa, la tabla dará cabida para albergar los 23 atributos que conforman los distintos registros y que veremos descritos brevemente en la siguiente tabla resumen, **Tabla 4**.

TABLA TESTS (ATRIBUTOS)				
#	CAMPO	DESCRIPCIÓN	TIPO DATO	NULO
1	id_maquina	Identifica la máquina en la que ha sido efectuado el test.	unsigned int(11)	No
2	id_test	Identifica el test realizado.	unsigned int(11)	No
3	fecha_hora	Fecha y hora en la que se ha realizado el test.	datetime	No
4	codigo_test	Código asociado al test.	char(11)	No
5	id_operador	Identifica al operador de supervisado del test.	unsigned int(11)	No
6	id_software	Identifica el software instalado en la máquina.	unsigned int(11)	No

TABLA TESTS (ATRIBUTOS)				
7	id_configuracion	Identifica la configuración establecida en la máquina.	unsigned int(11)	No
8	turno	Turno en el que se ha realizado el test. (Mañana / Tarde / Noche)	varchar(6)	No
9	tipo_test	Objeto del test. (Análisis / Investigación / Producción)	varchar(13)	No
10	Id_proveedor	Identifica el proveedor de los productos evaluados en el test.	unsigned int(11)	No
11	id_producto	Identifica el producto evaluado.	varchar(20)	No
12	id_lote	Identifica el lote de productos evaluado.	varchar(20)	No
13	protocolo_medicion	Protocolo empleado en el análisis. (1 / 2 / 3 / 4 / 5)	varchar(3)	No
14	modo_analisis	Forma de analizar los elementos. (Aleatorio / Ordenado)	varchar(9)	No
15	numero_elementos_analizados	Cantidad de elementos del lote objeto de análisis.	unsigned int(11)	No
16	defectos_menores	Cantidad de defectos de tipología menor detectados en el test.	unsigned int(11)	No
17	defectos_mayores	Cantidad de defectos de tipología mayor detectados en el test.	unsigned int(11)	No
18	defectos_criticos	Cantidad de defectos de tipología crítica detectados en el test.	unsigned int(11)	No
19	nivel_defectos	Valor numérico real calculado en función de la cantidad y tipo de defectos y del sistema de cálculo de calidad.	float	No
20	calidad	Resultado de comparar el nivel de defectos con dos umbrales. (Alta / Media / Baja)	varchar(5)	No
21	validacion	Atributo de validación, comprobación, modificación... por parte del operador al mando de la supervisión del test. (N / M / V / NULL)	char(1)	Sí

TABLA TESTS (ATRIBUTOS)				
22	test_extendido	Campo de señalización para una nueva prueba de análisis. (NO / SI)	char(2)	No
23	comentarios	Descripción adicional del test.	text	No

Tabla 4. Registros contenidos en tabla tests

De los atributos recogidos en cada uno de los registros que se almacenarán en la tabla, quizás el que más llame la atención en su forma de almacenar sea *codigo_test*, ya que se trata de un número compuesto por los dígitos de la fecha en el que el test fue efectuado seguido del número de test realizado por la máquina en esa fecha, como ya fue explicado en la sección 3.2.3. Pero a pesar de tratarse de un número se almacena como un conjunto de once caracteres en la forma *char(11)*, dado que si se almacena como un entero de dichos once dígitos, el espacio de almacenamiento que consumiría cada uno de ellos sería mucho mayor. Puesto que los últimos 3 caracteres se utilizan para indicar el número de test realizado ese día en esa máquina, dicha codificación admite un máximo de 1000 tests cada día. Se asume que ese valor es razonable, incluso con vistas al futuro, por lo que se decide mantener ese tipo de codificación.

TABLA TESTS (ÍNDICES)				
NOMBRE	DESCRIPCIÓN	PRIMARIO	ÚNICO	ÍNDICE
PRIMARY	Índice de clave primaria compuesta. Formada por <i>id_maquina</i> e <i>id_test</i> .	Sí	Sí	Sí
<i>id_maquina</i>	Identificador de máquina	No	No	Sí
<i>id_software</i>	Identificador de software	No	No	Sí
<i>id_configuracion</i>	Identificador de configuración	No	No	Sí
<i>id_operador</i>	Identificador de operador	No	No	Sí
<i>id_proveedor</i>	Identificador de proveedor	No	No	Sí

Tabla 5. Índices de la tabla tests

La anterior **Tabla 5** recoge los índices involucrados en la consecución de esta primera tabla descrita. La tabla *tests* posee una clave primaria compuesta formada por *id_maquina* e *id_test*, de manera que queden identificados de manera única e independiente cada uno de los tests efectuados en la totalidad de las diferentes máquinas. Así mismo, se reflejan otros cinco índices, *id_maquina*, *id_software*, *id_configuracion*, *id_operador* e *id_proveedor*, necesarios para crear las relaciones con otras de las tablas contenidas en la

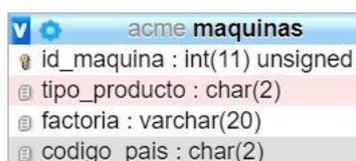
base de datos, como son: *maquinas*, *software*, *configuracion_calidad*, *operadores* y *proveedores*, respectivamente.

Las relaciones creadas y que vinculan entre sí las diferentes tablas se expondrán de manera más extendida y completa más adelante en la sección 4.3.2 titulada Relaciones.

4.3.1.2 Tabla maquinas

En primer lugar, que la palabra *maquinas* no esté acentuada no es a causa de un error ortográfico, sino que al diseñar la base de datos el nombramiento de cada una de las tablas y atributos no deben contener acentos, caracteres especiales o de tipo raro. Por ello que sucederá lo mismo con alguna tabla más.

Dicho lo anterior, el diseño y estructura de esta tabla está realizado de tal forma que, al volcar y realizar la migración de los datos, únicamente van a quedar almacenadas en ella las máquinas en las que se han llevado a cabo los diferentes tests registrados, el tipo de producto a los que efectúa los análisis dicha máquina, la factoría en la que está situada y el código de país relacionado con la ubicación de la factoría (ver **Figura 11**). Repárese que en el archivo Excel existirán innumerables tests con los mismos valores repetidos y que pertenecen a los registros antes citados. En esta tabla de la base de datos no ocurrirá así, ya que el diseño de esta tabla está realizado de tal forma que solo serán registrados una única vez.



acme maquinas	
id_maquina	: int(11) unsigned
tipo_producto	: char(2)
factoria	: varchar(20)
codigo_pais	: char(2)

Figura 11. Tabla maquinas

Los diferentes atributos que conformarán los distintos registros y le darán forma al completar la tabla son los mostrados en la **Tabla 6**. De esta tabla quiero destacar el atributo *tipo_producto*, el cual es de tipo *char* y tiene capacidad para dos caracteres, *char(2)*. Ya que como se muestra en la tabla puede reflejar dos posibles combinaciones de caracteres: PP o PE, dependiendo de si la máquina efectúa tests de análisis a Productos Propios o Productos Externos, respectivamente. Por otro lado, en esta tabla, con el propósito de simplificar la base de datos, se ha decidido desnormalizar al recoger la factoría y el código de país directamente en esta tabla.

TABLA MAQUINAS (ATRIBUTOS)				
#	CAMPO	DESCRIPCIÓN	TIPO DATO	NULO
1	 id_maquina	Identifica la máquina en la que se ejecutan los tests.	unsigned int(11)	No
2	tipo_producto	Tipo de producto que analiza la máquina. (PP / PE)	char(2)	No
3	factoria	Localización de la factoría.	varchar(20)	No
4	codigo_pais	Código asociado al país en el que se ubica la factoría.	char(2)	No

Tabla 6. Registros contenidos en tabla maquinas

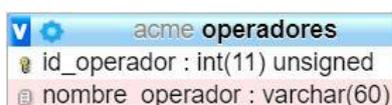
TABLA MAQUINAS (ÍNDICES)				
NOMBRE	DESCRIPCIÓN	PRIMARIO	ÚNICO	ÍNDICE
PRIMARY	Índice de clave primaria. id_maquina	Sí	Sí	Sí

Tabla 7. Índices de la tabla maquinas

La tabla *maquinas* estará consternada por una clave primaria la cual es referenciada por el índice *id_maquina*. Nos lo muestra la anterior **Tabla 7**. Este índice será el que diferencie indistintamente a cada una de las máquinas propiedad de la empresa en las que se hayan llevado a cabo los tests plasmados en el archivo Excel.

4.3.1.3 Tabla operadores

Otra de las tablas involucradas será la que albergue a todos los operadores de la empresa ACME que vengán anotados en el archivo Excel al haber estado supervisando las pruebas de valoración de la calidad de los diferentes tests. Dicha tabla será la que ilustra la **Figura 12**.



acme operadores	
	id_operador : int(11) unsigned
	nombre_operador : varchar(60)

Figura 12. Tabla operadores

En ella se asociará en cada tupla acumulada en la tabla, cada identificador de cada operador con su respectivo nombre y apellidos como se representa en la **Tabla 8**. En este caso se ha decidido simplificar al máximo y colocar el nombre y apellidos del operador en un único campo y no recoger información adicional. En un escenario real, debería existir una tabla mucho más completa con todos los datos relativos al operador.

TABLA OPERADORES (ATRIBUTOS)				
#	CAMPO	DESCRIPCIÓN	TIPO DATO	NULO
1	 id_operador	Identifica al operador.	unsigned int(11)	No
2	nombre_operador	Nombre y apellidos del operador.	varchar(60)	No

Tabla 8. Registros contenidos en tabla operadores

TABLA OPERADORES (ÍNDICES)				
NOMBRE	DESCRIPCIÓN	PRIMARIO	ÚNICO	ÍNDICE
PRIMARY	Índice de clave primaria. id_operador	Sí	Sí	Sí

Tabla 9. Índices de la tabla operadores

Como es de esperar y ocurre así en la mayoría de las tablas de la base de datos y de la misma forma que ocurría anteriormente en la tabla *maquinas*, en este caso el índice que representa la clave primaria de la tabla *operadores* será *id_operador*. De tal manera que cada uno de los operadores quede claramente diferenciado de otro al poseer cada operador un *id_operador* único. Esto es lo que viene reflejado en la **Tabla 9**.

4.3.1.4 Tabla configuracion_calidad

La siguiente de las tablas es la que reflejará todos los valores relacionados con las diferentes configuraciones que le han sido establecidas y, por lo tanto, con las que han trabajado las diferentes máquinas. En ella los diferentes registros tendrán asociados el valor de identificación de la configuración, el nombre que recibe esta, el número (identificador) que representa el sistema de cálculo que empleará dicha configuración para obtener el nivel de defectos y los valores de los dos umbrales con los que comparar el valor anterior y obtener así la calidad de producción del lote analizado.

Si recordamos y echamos memoria hacia atrás, la empresa ACME solo tenía presente en el archivo Excel, el identificador de configuración empleado en el desarrollo del test. El resto de los valores estaban anotados en el cuaderno de notas externo con el que también trabaja la empresa. De esta manera, al diseñar esta tabla, representada en la **Figura 13**, ACME tendrá un acceso más globalizado y directo a todo ello realizando un solo *click*.

Los atributos antes mencionados y que darán lugar a la formación de las sucesivas tuplas recogidas en la tabla *configuracion_calidad* son los que vienen recogidos en la **Tabla 10**.

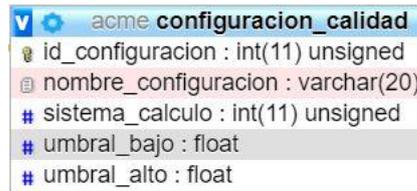


Figura 13. Tabla configuracion_calidad

TABLA CONFIGURACION_CALIDAD (ATRIBUTOS)				
#	CAMPO	DESCRIPCIÓN	TIPO DATO	NULO
1	 id_configuracion	Identifica la configuración a establecer en la máquina.	unsigned int(11)	No
2	nombre_configuracion	Nombre de la configuración.	varchar(20)	No
3	sistema_calculo	Identifica el sistema de cálculo empleado para obtener el nivel de defectos.	unsigned int(11)	No
4	umbral_bajo	Valor de frontera para el nivel de defectos y distinguir entre calidad alta o media.	float	No
5	umbral_alto	Valor de frontera para el nivel de defectos y distinguir entre calidad media o baja.	float	No

Tabla 10. Registros contenidos en tabla configuracion_calidad

La columna o atributo *sistema_calculo* tomará valores de un número natural, ya que ACME actualmente solo trabaja con tres valores distintos identificados por los valores del 1 al 3. En el futuro la empresa podría plantearse ampliarlos.

TABLA CONFIGURACION_CALIDAD (ÍNDICES)				
NOMBRE	DESCRIPCIÓN	PRIMARIO	ÚNICO	ÍNDICE
PRIMARY	Índice de clave primaria. id_configuracion	Sí	Sí	Sí

Tabla 11. Índices de la tabla configuracion_calidad

Esta nueva tabla estará franqueada por un índice de clave primaria fundado en *id_configuracion* (ver **Tabla 11**), el cual nos permitirá distinguir, caracterizar, ordenar, etc, cada una las configuraciones con las que haya operado la empresa y que se almacenarán en esta tabla.

4.3.1.5 Tabla software

El diseño de la tabla presentada a continuación (**Figura 14**), tabla *software*, brota de una cuestión idéntica a la anterior, dado que en el archivo Excel solamente viene recogido

el identificador de los distintos softwares instalados en las máquinas y obteniendo adicionalmente los valores asociados a ese identificador en el cuaderno de notas empleado.

En consecuencia, en esta nueva tabla recogida bajo el nombre de software, quedarán registrados todos los valores vinculados al identificador del software instalado, nombre que recibe dicho software empleado, versión que monta y los valores que emplean los parámetros de ajuste de configuración de software utilizados, parámetro A y parámetro B. Sobre esta cuestión, también es importante destacar que se supone que todos los softwares utilizados por ACME emplean esos dos parámetros. Si tuvieran otras características sería necesario replantearse el diseño.

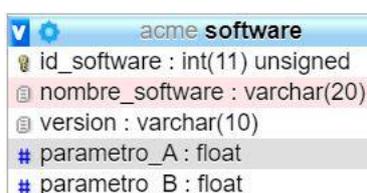


Figura 14. Tabla software

De esta manera los diferentes registros que se plasmarán en esta tabla quedarán caracterizados por los atributos anunciados antes y recogidos en la posterior **Tabla 12**.

TABLA SOFTWARE (ATRIBUTOS)				
#	CAMPO	DESCRIPCIÓN	TIPO DATO	NULO
1	 id_software	Identifica el software de instalación en la máquina.	unsigned int(11)	No
2	nombre_software	Nombre del software.	varchar(20)	No
3	version	Versión del software.	varchar(10)	No
4	parametro_A	Parámetro de ajuste del software.	float	No
5	parametro_B	Parámetro de ajuste del software.	float	No

Tabla 12. Registros contenidos en tabla software

TABLA SOFTWARE (ÍNDICES)				
NOMBRE	DESCRIPCIÓN	PRIMARIO	ÚNICO	ÍNDICE
PRIMARY	Índice de clave primaria. id_software	Sí	Sí	Sí

Tabla 13. Índices de la tabla software

El índice de clave primaria de la tabla *software* estará representado por *id_software*, tal y como se plasma en la **Tabla 13** anterior. Este será el que nos permita reconocer,

organizar, excluir, etc., indistintamente a cada uno de los softwares con los que opera y trabaja la empresa ACME.

4.3.1.6 Tabla proveedores

Observando el nombre de esta nueva tabla podemos reconocer o hacernos una idea de a qué podrá hacer alusión. Esta tabla recopila la lista de proveedores a cuyos productos se realizan tests. Tal y como ya se ha comentado, el identificador 0 se reservará para la propia empresa ACME, y el resto de los valores para los demás proveedores. La tabla *proveedores*, se representa en la **Figura 15**.

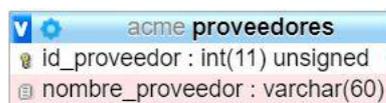


Figura 15. Tabla proveedores

Los registros quedarán reflejados por los campos *id_proveedor* y *nombre_proveedor*. Todos esos registros tendrán la misma apariencia que la marcada por la siguiente **Tabla 14**.

TABLA PROVEEDORES (ATRIBUTOS)				
#	CAMPO	DESCRIPCIÓN	TIPO DATO	NULO
1	 id_proveedor	Identificador del proveedor.	unsigned int(11)	No
2	nombre_proveedor	Nombre del proveedor.	unsigned int(11)	No

Tabla 14. Registros contenidos en tabla proveedores

TABLA PROVEEDORES (ÍNDICES)				
NOMBRE	DESCRIPCIÓN	PRIMARIO	ÚNICO	ÍNDICE
PRIMARY	Índice de clave primaria. id_proveedor	SÍ	SÍ	SÍ

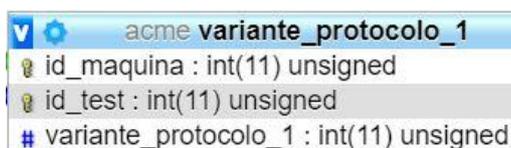
Tabla 15. Índices de la tabla proveedores

Nos encontramos con la presencia en esta tabla de un índice de clave primaria que identifica exclusiva e inequívocamente a cada proveedor, *id_proveedor*.

4.3.1.7 Tabla variante_protocolo_1

La tabla *variante_protocolo_1* surge como consecuencia de que en las pruebas de análisis de evaluación de la calidad de la composición de los distintos lotes de productos se realicen bajo el protocolo de medición 1. Por lo tanto, en esta tabla solamente se

almacenarán los tests que han sido efectuados en las distintas máquinas y en los cuales el protocolo de medición establecido como he dicho anteriormente es el identificado como 1. Esta última indicación como tal, haciendo referencia al protocolo de medición empleado nos aparecerá referenciado en la tabla tests. En esta tabla junto con el mismo *id_maquina* e *id_test*, se recogerá la variante del protocolo 1 utilizada, la cual sera registrada en el atributo *variante_protocolo_1*, tal y como aparece en la **Figura 16**.



```

acme variante_protocolo_1
id_maquina : int(11) unsigned
id_test : int(11) unsigned
# variante_protocolo_1 : int(11) unsigned
    
```

Figura 16. Tabla variante_protocolo_1

Los atributos anunciados previamente y recogidos en las tuplas de los tests realizados bajo el protocolo de medición número 1, se pueden consultar en la **Tabla 16**.

TABLA VARIANTE_PROTOCOLO_1 (ATRIBUTOS)				
#	CAMPO	DESCRIPCIÓN	TIPO DATO	NULO
1	 id_maquina	Identifica la máquina en la que ha sido efectuado el test.	unsigned int(11)	No
2	 id_test	Identifica el test realizado.	unsigned int(11)	No
3	variante_protocolo_1	Variante del protocolo 1 empleada en el análisis del test.	unsigned int(11)	No

Tabla 16. Registros contenidos en tabla variante_protocolo_1

TABLA VARIANTE_PROTOCOLO_1 (ÍNDICES)				
NOMBRE	DESCRIPCIÓN	PRIMARIO	ÚNICO	ÍNDICE
idMaquinaEidTest	Índice de clave foránea compuesta. Formada por <i>id_maquina</i> e <i>id_test</i> .	No	Sí	Sí

Tabla 17. Índices de la tabla variante_protocolo_1

Abandonando la tónica anterior de establecer un índice de clave primaria por cada tabla presentada, en la tabla actual y las tablas posteriores, los índices estarán designando una clave foránea compuesta formada en la mayoría de los casos por *id_maquina* e *id_test*, pero que en alguna ocasión pueden verse asociados a otro atributo, que ya se explicará cuando ello ocurra. La clave foránea señalada en estas tablas vendrá haciendo alusión o referencia a las filas de la tabla padre o referenciada, valga la redundancia. Esta tabla es la tabla principal expuesta en primer lugar, la tabla *tests*.

En este caso y como se describe en la **Tabla 17**, la formación de este nuevo índice que da lugar a la clave foránea recibe el nombre de *id_MaquinaEid_Test* y está formado por los atributos *id_maquina* e *id_test*.

4.3.1.8 Tabla medidas_consolidadas

Las tres próximas y últimas tablas están orientadas para las funcionalidades futuras, ya anunciadas en 4.2.2, y que ACME pretende implantar próximamente. De esta manera, a la hora de realizar la migración de los datos recogidos en el archivo Excel a esta nueva base de datos, las tres tablas presentarán un aspecto vacío, dado que obviamente al no trabajar actualmente con estas funcionalidades, el archivo Excel no presenta datos con los cuales poder trabajar.

El diseño de estas se basa en el registro y almacenamiento de los resultados arrojados en cada uno de los tests de análisis llevados a cabo. En ellas se registrarán el número de defectos clasificados dependiendo de su tipología de cada uno de los elementos analizados y que forman parte de los diferentes lotes de productos evaluados en la diversidad de máquinas de que dispone ACME.

En primer lugar, el diseño de la tabla *medidas_consolidadas* está realizado de tal manera que se plasmen en ella todos y cada uno de los tests efectuados. Estos contienen los resultados obtenidos de forma automática y tras ser corregidos (en su caso) por el operador. La información que aparece en la tabla *tests* será coherente con la que aparezca en esta nueva tabla mostrada en la **Figura 17**.

acme medidas_consolidadas	
id_maquina	: int(11) unsigned
id_test	: int(11) unsigned
id_elemento	: int(11) unsigned
num_defectos_menores	: int(11) unsigned
num_defectos_mayores	: int(11) unsigned
num_defectos_criticos	: int(11) unsigned

Figura 17. Tabla *medidas_consolidadas*

Cada una de las filas o tuplas que se escriban en la misma, quedarán registradas atendiendo a los siguientes atributos: *id_maquina*, identificador de la máquina en la que se efectuó el test; *id_test*, identificador del número de test realizado en esa máquina; *id_elemento*, identifica el elemento analizado; *num_defectos_menores*, cantidad de defectos de tipo menor detectados en dicho elemento; *num_defectos_mayores*, cantidad de

defectos de tipo mayor detectados en el elemento; *num_defectos_criticos*, cantidad de defectos de tipo crítico detectados en el elemento analizado. Vienen recogidos más detalles en la **Tabla 18**.

TABLA MEDIDAS_CONSOLIDADAS (ATRIBUTOS)				
#	CAMPO	DESCRIPCIÓN	TIPO DATO	NULO
1	 id_maquina	Identifica la máquina en la que ha sido efectuado el test.	unsigned int(11)	No
2	 id_test	Identifica el test realizado.	unsigned int(11)	No
3	 id_elemento	Identifica el elemento analizado.	unsigned int(11)	No
4	num_defectos_menores	Cantidad de defectos de tipología menor detectados en el elemento.	unsigned int(11)	No
5	num_defectos_mayores	Cantidad de defectos de tipología mayor detectados en el elemento.	unsigned int(11)	No
6	num_defectos_criticos	Cantidad de defectos de tipología crítica detectados en el elemento.	unsigned int(11)	No

Tabla 18. Registros contenidos en tabla medidas_consolidadas

TABLA MEDIDAS_CONSOLIDADAS (ÍNDICES)				
NOMBRE	DESCRIPCIÓN	PRIMARIO	ÚNICO	ÍNDICE
idMaquinaEidTestEidElemento	Índice de clave foránea compuesta. Formada por <i>id_maquina</i> e <i>id_test</i> . Asociada junto con <i>id_elemento</i> .	No	Sí	Sí

Tabla 19. Índices de la tabla medidas_consolidadas

En la tabla previa, *variante_protocolo_1*, el índice de clave foránea venía representado y formado por los campos *id_maquina* e *id_test*. Para esta tabla en particular y como podemos observar en la **Tabla 19** encima de estas líneas, a esos dos atributos se les asociará también el *id_elemento* para así de esta forma tener identificado y caracterizado en todo momento la máquina que desarrolló el test, el número de test realizado por la máquina correspondiente y el elemento que presenta dichos defectos.

4.3.1.9 Tabla medidas_software_no_aceptadas

Esta otra tabla también está destinada para las próximas nuevas funcionalidades y reunirá a aquellos tests que hayan sido revisados, por algún motivo, por el operador y en los que éste haya discrepado con los valores obtenidos por el software y decida corregirlos.

En la tabla *medidas_software_no_aceptadas* únicamente se almacenarán esos tests de lotes de productos cuyos valores determinados por el software hayan sido verificados y no aceptados por el operador. Es decir, si un test no ha sido revisado por un operador o si ha sido revisado por un operador y todo es correcto, no aparecerá información sobre ese test en esta tabla, pero sí aparecerá tanto en la tabla *tests* como en la tabla *medidas_consolidadas*. Si un test ha sido revisado por el operador y éste realiza modificaciones, entonces los valores nuevos corregidos aparecerán en las tablas *tests* y *medidas_consolidadas*, mientras que los valores originalmente medidos por el software vendrán recogidos en la tabla *medidas_software_no_aceptadas* y en la tabla *medidas_software_por_elemento_no_aceptadas*, que se detallará posteriormente.

Los atributos que caracterizarán los diferentes registros de la tabla *medidas_software_no_aceptadas* se pueden ver en la **Figura 18** y con más detalle en la posterior **Tabla 20**. Estos vendrán determinados por la máquina en la que se llevo a cabo el test, el identificador de test en la propia máquina y el número total de defectos de cada tipo (menor, mayor y crítico) detectados en la realización automática del test de un lote.

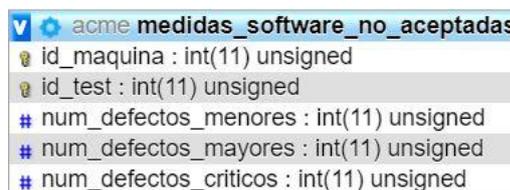


Figura 18. Tabla *medidas_software_no_aceptadas*

TABLA MEDIDAS_SOFTWARE_NO_ACEPTADAS (ATRIBUTOS)				
#	CAMPO	DESCRIPCIÓN	TIPO DATO	NULO
1	id_maquina	Identifica la máquina en la que ha sido efectuado el test.	unsigned int(11)	No
2	id_test	Identifica el test realizado.	unsigned int(11)	No
3	num_defectos_menores	Cantidad de defectos de tipología menor detectados en el test.	unsigned int(11)	No
4	num_defectos_mayores	Cantidad de defectos de tipología mayor detectados en el test.	unsigned int(11)	No
5	num_defectos_criticos	Cantidad de defectos de tipología crítica detectados en el test.	unsigned int(11)	No

Tabla 20. Registros contenidos en tabla *medidas_software_no_aceptadas*

TABLA MEDIDAS_SOFTWARE_NO_ACEPTADAS (ÍNDICES)				
NOMBRE	DESCRIPCIÓN	PRIMARIO	ÚNICO	ÍNDICE
idMaquinaEidTest	Índice de clave foránea compuesta. Formada por id_maquina e id_test.	No	Sí	Sí

Tabla 21. Índices de la tabla medidas_software_no_aceptadas

Para lograr esta distinción en dicha tabla, nos encontramos con un índice de clave foránea, cuya tabla referenciada sigue siendo la tabla *tests*. Nuevamente, como ocurría en la *tabla variante_protocolo_1* y se puede apreciar en la **Tabla 21**, está formado por el identificador de la máquina, *id_maquina*, junto con el identificador del test en dicha máquina, *id_test*.

4.3.1.10 Tabla medidas_software_por_elemento_no_aceptadas

En último lugar, nos encontramos con la tabla *medidas_software_por_elemento_no_aceptadas*, provista también para la implantación de las futuras funciones en el desempeño de la actividad. Esta tabla no es más que una extensión de la tabla anterior, pero en este caso detallando el nivel de análisis y detección de defectos por elemento analizado. Esto es, identificando la cantidad y tipo de defecto por cada elemento analizado del lote de productos proporcionado a analizar en la máquina pertinente. Por ello que en esta tabla se incluye un atributo más a los que idénticamente presentaba la tabla *medidas_software_no_aceptadas*. Ese atributo es *id_elemento*, mediante el cual se identificará al elemento analizado en cuestión y con el que se asociarán los defectos detectados y de la tipología que son, manteniendo y compartiendo todos ellos el mismo *id_maquina* e *id_test*.

Podemos ver la anterior tabla *medidas_software_no_aceptadas* como una tabla resumen de esta nueva tabla (*medidas_software_por_elemento_no_aceptadas*, **Figura 19**), para realizar operaciones de identificación, búsqueda o consulta de manera más rápida y eficiente, y esta otra para realizar otras operaciones con mayor grado de detalle.

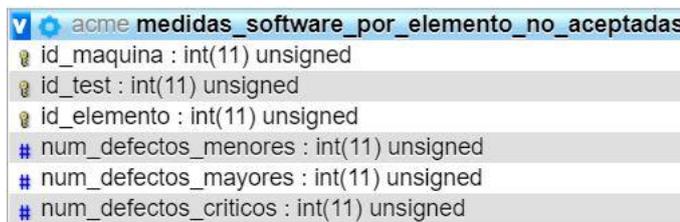


Figura 19. Tabla *medidas_software_por_elemento_no_aceptadas*

Los atributos agrupados en la tabla *medidas_software_por_elemento_no_aceptadas* se detallan en la siguiente **Tabla 22**.

TABLA MEDIDAS_SOFTWARE_POR_ELEMENTO_NO_ACEPTADAS (ATRIBUTOS)				
#	CAMPO	DESCRIPCIÓN	TIPO DATO	NULO
1	id_maquina	Identifica la máquina en la que ha sido efectuado el test.	unsigned int(11)	No
2	id_test	Identifica el test realizado.	unsigned int(11)	No
3	id_elemento	Identifica el elemento analizado.	unsigned int(11)	No
4	num_defectos_menores	Cantidad de defectos de tipología menor detectados en el elemento.	unsigned int(11)	No
5	num_defectos_mayores	Cantidad de defectos de tipología mayor detectados en el elemento.	unsigned int(11)	No
6	num_defectos_criticos	Cantidad de defectos de tipología crítica detectados en el elemento.	unsigned int(11)	No

Tabla 22. Registros contenidos en tabla *medidas_software_por_elemento_no_aceptadas*

TABLA MEDIDAS_SOFTWARE_POR_ELEMENTO_NO_ACEPTADAS (ÍNDICES)				
NOMBRE	DESCRIPCIÓN	PRIMARIO	ÚNICO	ÍNDICE
idMaquinaEidTestEidElemento	Índice de clave foránea compuesta. Formada por id_maquina e id_test. Asociada junto con id_elemento.	No	Sí	Sí

Tabla 23. Índices de la tabla *medidas_software_por_elemento_no_aceptadas*

De nuevo, como ocurría en la tabla *medidas_consolidadas*, para alcanzar este tipo de detalles en el análisis de cada elemento debemos asociar al índice de clave foránea formado por *id_maquina* e *id_test*, el atributo *id_elemento* (ver **Tabla 23**). De esta manera podemos reconocer con el mismo identificador de máquina e identificador de test, la prueba de evaluación que se ha llevado a cabo, y distinguir mediante el identificador de elemento, el elemento concreto que se quiera consultar o detallar.

4.3.2 Relaciones

Previamente a proceder con la explicación de las relaciones asociativas y que mantienen ligadas y vinculadas unas tablas con otras para conseguir la robustez y preservar la integridad de los datos, es importante no entender el término *relación* del título de este epígrafe como haciendo referencia la acepción de relación como tabla de una base de datos, ya que *relación* también puede ser utilizado como término formal con el que se reconoce a una tabla de una base datos.

En este caso haremos alusión del término *relación* con un uso más general del mismo como es la acepción de “*relacionado con*”, “*conexión, correspondencia de algo con otra cosa*” [Real Academia Española, *Diccionario de la Lengua Española*, 2021. <https://dle.rae.es/relación>] al referirnos a las relaciones o conexiones entre las diferentes tablas de la base de datos.

Primeramente, para que dos tablas se puedan relacionar o vincular deben presentar alguna de ellas un campo el cual sea clave primaria de la misma. Ambos campos, clave y no clave, deben ser del mismo tipo de dato y tamaño para así coincidir y poder recoger el mismo tipo de valores. El nombre que reciba el atributo o los atributos que vayan a formar la relación entre las tablas no tienen por qué compartir el mismo nombre en las dos. Por ejemplo, no podemos relacionar dos tablas si una dispone de un campo que es clave de tipo dato *int* recogido bajo el nombre de *id_persona* y la otra recoge un atributo con tipo de dato *varchar* apodado con el nombre de *id_persona*. Para ello deberían ser los dos atributos o bien tipo *int*, recogidos bajo el mismo o distinto nombre en cada tabla, o de tipo *varchar*, con el mismo o distinto nombre en cada tabla de igual manera.

Por lo tanto, teniendo en cuenta estas consideraciones previas y llevándolas hasta el diseño de las tablas con las que queremos formar nuestra base de datos, a la vez que mantenemos en mente la estructura y formación de las claves e índices presentes en cada tabla y que se han presentado recientemente en el punto anterior 4.3.1 para cada una de ellas, estamos en condiciones de ir creando las diferentes relaciones que vincularán con la presencia de los posteriores datos migrados y almacenados en los correspondientes registros la totalidad de las tablas de base de datos.

4.3.2.1 Relaciones de clave simple

Para ir dándole forma y detalle al diseño global, se comenzarán a plasmar las relaciones que incluyen las tablas con presencia de claves simples. Esto es, aquellas tablas que presentan un único atributo como clave primaria de la misma. Las tablas que poseen esta condición de clave primaria simple son: *maquinas*, *operadores*, *software*, *configuracion_calidad* y *proveedores*. En cada una de ellas nos encontramos con que los atributos *id_maquina*, *id_operador*, *id_software*, *id_configuracion* e *id_proveedor* están designadas claves primarias de cada una de las anteriores tablas respectivamente. Como se detalló anteriormente la tabla principal o matriz la cual albergará los valores de las características y particularidades implicadas, y los resultados arrojados en la ejecución de las pruebas de valoración de la calidad será la tabla *tests*. Algunas de esas particularidades recogidas en esta última tabla son la máquina que está realizando el test (atributo *id_maquina*), el operador al supervisado del test (*id_operador*), tanto el software (*id_software*) como la configuración (*id_configuracion*) montada en el desarrollo de la prueba evaluativa o el proveedor al cual pertenecen los productos que están siendo evaluados (*id_proveedor*).

En este caso el nombre de los atributos de las diferentes tablas coincide, pero insisto nuevamente que esta condición no es necesaria. Como el tipo de dato que guardarán también son el mismo en ambas tablas a relacionar, podemos crear la relación entre la tabla *tests* y las tablas mencionadas al principio. Quedando el diseño relacional como el mostrado en la **Figura 20** (siguiente página).

De esta manera, atendiendo a los valores registrados en los atributos vinculados de la tabla *tests*, podemos conocer y detallar el resto de los datos contenidos en los atributos restantes de las diversas tablas y así obtener mayor grado de información asociada al respecto de los valores establecidos en los identificadores en el desarrollo de los sucesivos tests. Por ejemplo, si durante el desarrollo de un test, el campo *id_operador* ha registrado un determinado valor, observando la tabla *operadores* podemos averiguar qué nombre y apellidos corresponden al operador con ese *id_operador*. Y actuar de forma idéntica con el resto de los identificadores de las otras tablas.

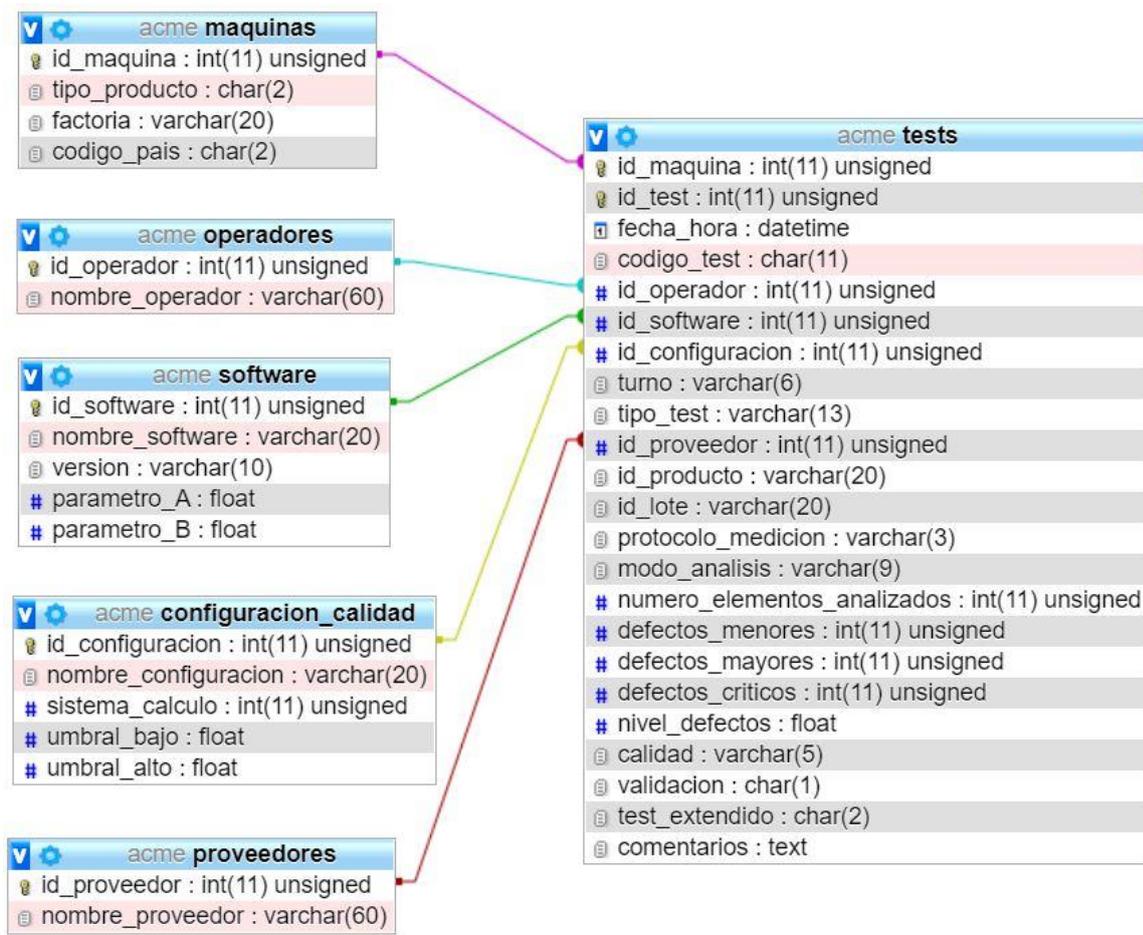


Figura 20. Relaciones de clave simple

4.3.2.2 Relaciones de clave compuesta

De igual manera que ocurre con las claves simples, se crearán también relaciones con otras tres tablas, que veremos ahora, a través de la clave primaria compuesta de la tabla *tests*, formada por los atributos *id_maquina* e *id_test*. La clave primaria de la tabla *tests*, nos permitirá identificar inequívocamente cada uno de los tests realizados en la máquina concreta en la que se ha llevado a cabo. Hay que recordar que esta clave de ser de naturaleza compuesta, ya que el campo *id_test* comenzará con valor 1 para la primera prueba realizada en cada una de las máquinas y se va incrementando por cada nuevo test que se haya efectuado en la misma, por lo que podemos tener, por ejemplo, un *id_test* repetido con valor 3, pero nunca podrá estar asociado a un *id_maquina* idéntico. De esta manera, al asociar *id_test* e *id_maquina*, podemos averiguar y distinguir que se trata del test con *id_test* el que sea y ha sido efectuado en la máquina con *id_maquina* que corresponda.

Con la formación de esta clave primaria compuesta en la tabla *tests* se derivan otras claves foráneas de los mismos atributos, en las cinco tablas restantes:

variante_protocolo_1, *medidas_software_no_aceptadas*, *medidas_consolidadas* y *medidas_software_por_elemento_no_aceptadas*. Siendo estas últimas las tablas hijas o referenciantes y la tabla de origen *tests*, la maestra o referenciada.

Aunque podemos realizar una pequeña distinción entre las tres primeras y las dos últimas. De momento en este apartado, nos centraremos en las dos primeras (*variante_protocolo_1*, y *medidas_software_no_aceptadas*). Si se observa más adelante la **Figura 21**, se puede entender de forma más visual lo explicado en el párrafo anterior, se pueden ver plasmadas las relaciones de claves foráneas involucradas.

Para el caso de la tabla *variante_protocolo_1*, la relación de clave foránea nos permitirá establecer en dicha tabla todos los tests efectuados en las distintas máquinas, los cuales únicamente se hayan realizado bajo el protocolo de medición 1, pudiendo observar en ella la variante de este protocolo utilizada (campo *variante_protocolo_1*). Siguiendo la misma interpretación, en la tabla *medidas_software_no_aceptadas*, diseñada para las funcionalidades venideras, podemos reconocer que ese test ha sido descartado posteriormente al análisis realizado por el software de la máquina y contabilizar los defectos globales de cada tipo detectados en los test descartados.

4.3.2.1 Relaciones de clave compuesta asociada a *id_elemento*

Las dos tablas que restan de la totalidad, *medidas_consolidadas* y *medidas_software_por_elemento_no_aceptadas*, siguen la misma explicación que las tres anteriores en el efecto de la vinculación de clave foránea con la tabla *tests*, nada más que presentan un pequeño matiz. Este matiz es que la clave foránea presente en estas dos tablas lo hace también compartiendo clave de la misma tabla con el atributo *id_elemento*, tal y como se plasma en la **Figura 22**. La razón no es otra que identificar de manera inequívoca dentro del mismo test, a cada uno de los elementos analizados del lote evaluado y conocer así individualmente los defectos que presenta y a que tipología pertenecen. Este plan de análisis más detallado y estricto era una funcionalidad no aplicada en la actualidad, pero para considerar su implantación cercanamente.

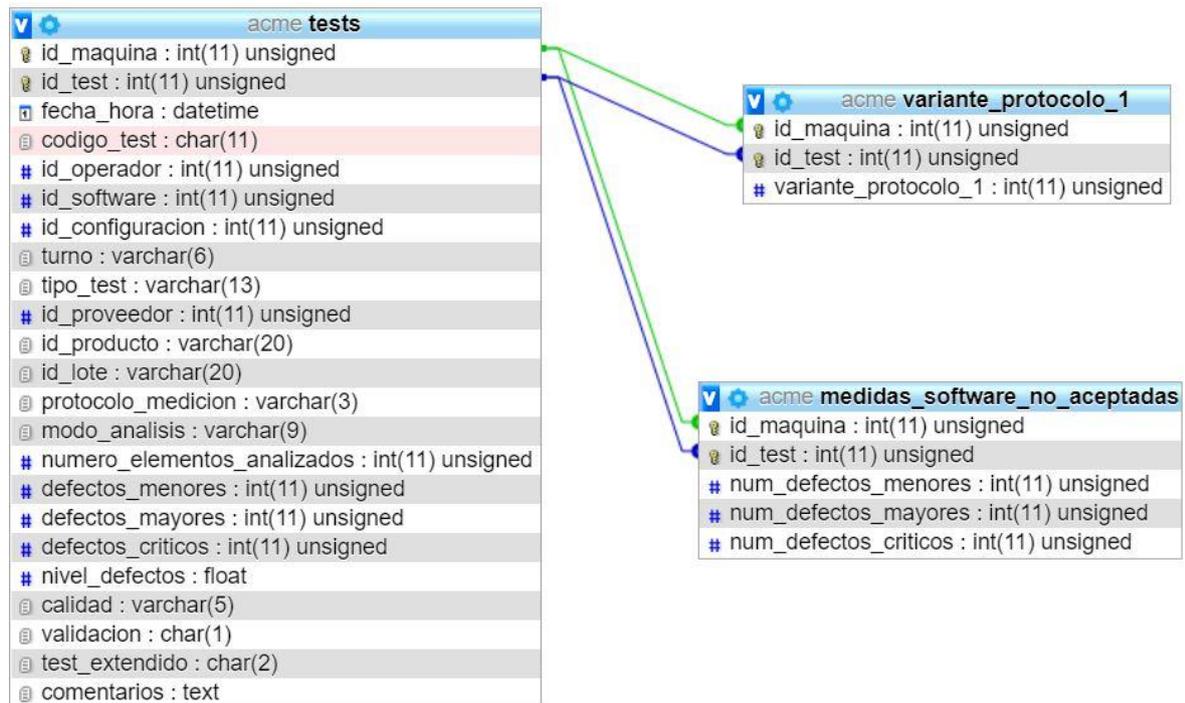


Figura 21. Relaciones de clave compuesta

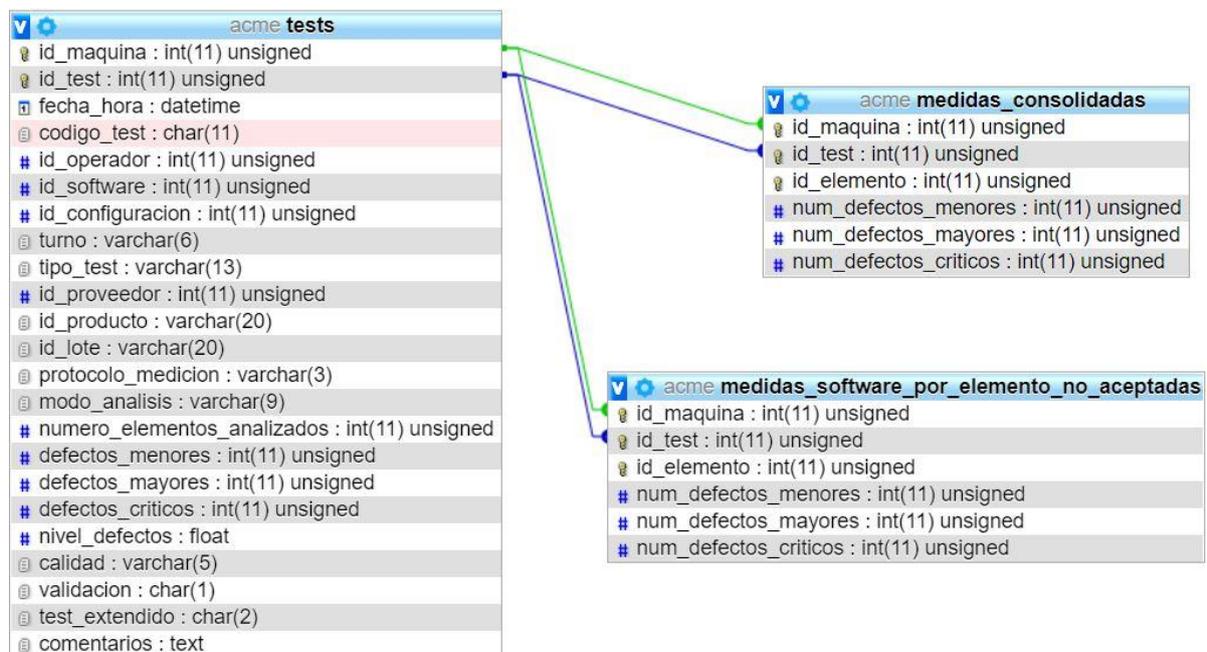


Figura 22. Relaciones de clave compuesta asociados con id_elemento

De esta forma se podrán reconocer los tests que han sufrido correcciones con respecto a los valores determinados inicialmente y poder acceder tanto a los valores corregidos como a los originales en caso de ser necesarios. Los primeros serán englobados en la tabla *medidas_consolidadas* y los segundos en la tabla *medidas_software_por_elemento_no_aceptadas*.

4.3.3 Base de datos (conjunto global)

Después de plantear y describir cada una de las tablas individualmente, detallar las relaciones creadas y que vinculan las tablas entre sí, únicamente queda por ver el aspecto del resultado global de la base de datos. Ello se puede apreciar en la **Figura 23**, reservando la página siguiente al completo para plasmarla y obtener así una mejor visualización de esta.

A la hora de realizar la migración de los datos contenidos en el archivo Excel a la misma, las tablas irán recogiendo los valores de sus correspondientes registros y se establecerán las relaciones antes mencionadas mediante los valores asociados que contemplen los atributos seleccionados para ello.

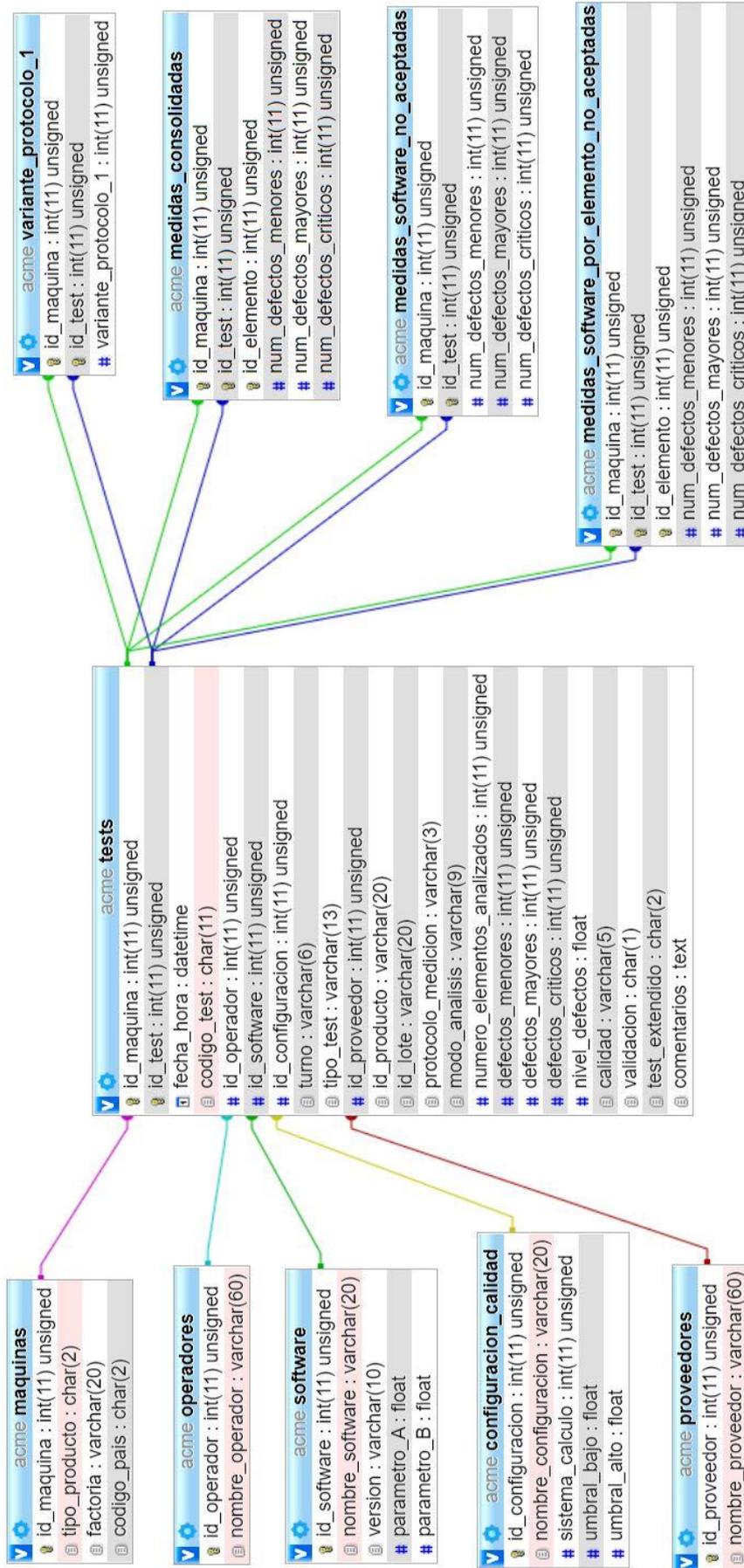


Figura 23. Diagrama Relacional de la Base de datos

5

Herramienta automatizada de migración. Resultados

5.1 Herramienta automatizada para la migración

5.1.1 Introducción

Una vez presentada la estructura de la base de datos y anteriormente descrito el archivo Excel en el que están contenidos los datos registrados, es hora de detallar cómo es la herramienta que hace posible la migración y volcado de los datos presentes en el archivo hasta completar los registros de las diferentes tablas de la base de datos.

Pensando lo laborioso que sería ir recorriendo cada celda del archivo Excel, tomando valor a valor de cada una de ellas y copiándolos en cada lugar correspondiente en la base de datos, se ha diseñado complementariamente una herramienta que realice esa tarea de manera automatizada.

5.1.2 Descripción

Se trata de un *script* realizado en lenguaje PHP [13], recogido bajo el nombre de *migracionExcelACME.php*. Se ha realizado en ese lenguaje dado que previamente se ha pensado su desarrollo para una posible implementación en una aplicación web, en la que posteriormente se puedan añadir otra serie de desarrollos con el mismo objetivo y disponer de multitud de posibilidades con las que realizar otra serie de operaciones y cometidos con los datos antes o después de ser volcados. Un ejemplo de ello podrían ser la creación de interfaces y formularios con los que acceder a los datos para realizar visualizaciones, consultas, modificaciones, borrados..., creación previamente al acceso de usuarios con diferentes permisos de operación con los datos almacenados, etc...

El *script* que desarrolla la herramienta de migración propiamente dicha está secundado en otros tres adicionales, *conexionACME.php*, *cierreConexionACME.php* y *baseDatosACME.php*, en los cuales se recoge información complementaria y necesaria para el funcionamiento de este. En concreto, el primero de ellos recoge las instrucciones para realizar la conexión con la base de datos. El segundo alberga la instrucción de desconexión. Y el que resta acoge los valores de las variables de la base de datos, estos son: el nombre de la misma, el servidor en el que se encuentra alojado y el usuario y contraseña de acceso a la misma. Por lo que tan pronto como sea necesario alguno de los procedimientos o informaciones anteriores, únicamente lo que habrá que hacer es invocar al *script* correspondiente durante el desarrollo dentro del principal.

De la misma manera, el *script* principal se apoya en otra librería denominada *PhpSpreadsheet* [14], la cual ha sido necesaria importar y adjuntar junto al resto. *PhpSpreadsheet* es una biblioteca escrita en PHP puro y que proporciona un conjunto de clases que permiten leer, escribir y realizar otra serie de operaciones en diferentes formatos con los datos albergados en las diferentes celdas de un archivo de hoja de cálculo, como Excel. Tal y como sucedía antes con los otros *scripts* complementarios, cuando sea necesario hacer uso de alguna de las funciones incluidas en esta última librería anunciada, tan solo debemos aludir a ella y esta llevará a cabo su cometido, dado que al principio del *script* de migración lleva implementada la invocación a la biblioteca.

El desarrollo del *script* para la migración y volcado está diseñado de la siguiente manera:

1. Conexión con la base datos.
2. Instrucción de carga y apertura para la lectura del archivo Excel.
3. Lectura de las distintas celdas que componen las sucesivas filas del archivo Excel.
4. Instrucciones y peticiones de escritura y registro de los datos obtenidos en las diferentes tablas de la base de datos.
5. Desconexión de la base de datos.

6. Instrucciones de comprobación y verificación que todo ha funcionado correctamente.

5.1.3 Ejecución

Para el correcto funcionamiento y ejecución del mismo, se debe disponer o tener alojado un servidor local, mediante el cual puedan generarse las pertinentes conexiones y operaciones descritas en este. Al final de la memoria, se adjunta un anexo en el que se detalla un ejemplo-guía de ejecución completa (**Anexo. Guía de ejecución**).

5.2 Resultados

5.2.1 Introducción

Llegados a este punto, una vez descritas y teniendo claras las tres partes definidas anteriormente por separado (archivo Excel, base de datos y *script* de migración), ha llegado la hora de ejecutar y poner en práctica el trabajo conjunto desarrollado. Para ello nos apoyaremos en las pautas descritas en **Anexo. Guía de ejecución**.

5.2.1 Situación previa

Como se puede pensar y si hemos seguido el desarrollo del TFG, en la situación de punto de partida, la empresa ACME partirá de la disponibilidad de sus datos recogidos y presentes en el archivo Excel después de haber llevado a cabo diferentes tests. Al tratarse de una empresa ficticia, no obstante, se ha tenido que crear un archivo Excel en el que simular una copia idéntica con los campos distribuidos del archivo Excel e insertar multitud de datos al azar en consonancia y guardando la correlación absoluta entre los campos que así han sido especificados. Ese archivo luego ha sido empleado para realizar las posteriores y sucesivas ejecuciones de migración de los datos e ir perfilando las diferentes pruebas evolutivas y ver así que el conjunto global desarrollado funciona correctamente. Se ilustrará de manera gráfica en las tres siguientes figuras, **Figura 24**, **Figura 25** y **Figura 26**, las cuales corresponden a tres fragmentos de la primera de las hojas del archivo Excel creado.

Capítulo 5: Herramienta automatizada para la migración. Resultados

A	B	C	D	E	F	G	H	I	J	K	L	
1	FACTORIA	COD. PAIS	ID MÁQUINA	ID TEST	COD. TEST	FECHA Y HORA	OPERADOR	ID OPERADOR	TURNO	TIPO TEST	ID SOFTWARE	ID CONFIGURACIÓN
2	VALLADOLID	ES	1	1	20200518001	18-5-21 0:02	JOSE MULAS	11	NOCHE	ANÁLISIS	1	3
3	VALLADOLID	ES	1	2	20200518002	18-5-21 7:18	ANTONIO NÁRVAEZ	1	MAÑANA	PRODUCCIÓN	1	3
4	VALLADOLID	ES	1	3	20200518003	18-5-21 13:56	EUGENIO BIGAS	2	MAÑANA	PRODUCCIÓN	1	3
5	VALLADOLID	ES	1	4	20200518004	18-5-21 20:34	ROBERTO PRIETO	15	TARDE	INVESTIGACIÓN	1	3
6	VALLADOLID	ES	1	5	20200519001	19-5-21 3:12	ÁLVARO MATEOS	13	NOCHE	ANÁLISIS	1	3
7	VALLADOLID	ES	1	6	20200519002	19-5-21 9:50	LETICIA SANTOS	3	MAÑANA	INVESTIGACIÓN	1	3
8	VALLADOLID	ES	1	7	20200519003	19-5-21 16:28	ROBERTO PRIETO	15	TARDE	INVESTIGACIÓN	1	3
9	VALLADOLID	ES	1	8	20200519004	19-5-21 23:06	ANTONIO NÁRVAEZ	1	NOCHE	ANÁLISIS	1	3
10	VALLADOLID	ES	1	9	20200520001	20-5-21 5:44	CRISTINA ANTÚNEZ	10	NOCHE	INVESTIGACIÓN	2	4
11	VALLADOLID	ES	1	10	20200520002	20-5-21 12:22	ENRIQUE BAJO	12	MAÑANA	ANÁLISIS	2	4
12	VALLADOLID	ES	1	11	20200520003	20-5-21 19:00	ISABEL MUÑOZ	9	TARDE	PRODUCCIÓN	2	4
13	VALLADOLID	ES	1	12	20200521001	21-5-21 1:38	ENRIQUE BAJO	12	NOCHE	PRODUCCIÓN	2	4
14	VALLADOLID	ES	1	13	20200521002	21-5-21 8:16	ISABEL MUÑOZ	9	MAÑANA	PRODUCCIÓN	2	4
15	VALLADOLID	ES	1	14	20200521003	21-5-21 14:54	ISABEL MUÑOZ	9	MAÑANA	ANÁLISIS	2	4
16	VALLADOLID	ES	1	15	20200521004	21-5-21 21:32	CRISTINA ANTÚNEZ	10	TARDE	ANÁLISIS	2	4
17	VALLADOLID	ES	2	1	20200518001	18-5-21 2:14	ENRIQUE BAJO	12	NOCHE	INVESTIGACIÓN	1	3
18	VALLADOLID	ES	2	2	20200518002	18-5-21 9:31	ÁLVARO MATEOS	13	MAÑANA	PRODUCCIÓN	1	3
19	VALLADOLID	ES	2	3	20200518003	18-5-21 16:09	ISABEL MUÑOZ	9	TARDE	INVESTIGACIÓN	1	3
20	VALLADOLID	ES	2	4	20200518004	18-5-21 22:47	LORENA FINCIAS	6	TARDE	INVESTIGACIÓN	1	3
21	VALLADOLID	ES	2	5	20200519001	19-5-21 5:25	ANTONIO NÁRVAEZ	1	NOCHE	PRODUCCIÓN	1	3
22	VALLADOLID	ES	2	6	20200519002	19-5-21 12:03	ENRIQUE BAJO	12	MAÑANA	PRODUCCIÓN	1	3
23	VALLADOLID	ES	2	7	20200519003	19-5-21 18:41	ANTONIO NÁRVAEZ	1	TARDE	ANÁLISIS	1	3
24	VALLADOLID	ES	2	8	20200520001	20-5-21 1:19	ANTONIO NÁRVAEZ	1	NOCHE	PRODUCCIÓN	2	4
25	VALLADOLID	ES	2	9	20200520002	20-5-21 7:57	ROBERTO PRIETO	15	MAÑANA	PRODUCCIÓN	2	4

Figura 24. Primer fragmento de la hoja 1 del archivo Excel

M	N	O	P	Q	R	S	T	
1	ID PRODUCTO	ID LOTE	PROTOCOLO MEDICIÓN	VARIANTE PROTOCOLO	MODO DE ANÁLISIS	NUM. ELEMENTOS ANALIZADOS	NUM. DEFECTOS MENORES	NUM. DEFECTOS MAYORES
2	9I	12AB	5		ALEATORIO	6	1	2
3	6F	34CD	4		ORDENADO	9	1	0
4	1A	12AB	1	9	ALEATORIO	20	5	1
5	2B	12AB	2		ORDENADO	10	2	1
6	7G	12AB	1	4	ORDENADO	19	0	0
7	2B	34CD	5		ALEATORIO	11	4	0
8	6F	78GH	5		ALEATORIO	6	4	3
9	8H	12AB	3		ALEATORIO	1	4	2
10	3C	90IJ	2		ORDENADO	18	1	0
11	1A	56EF	5		ORDENADO	12	0	1
12	2B	56EF	4		ORDENADO	10	2	3
13	6F	56EF	1	3	ALEATORIO	11	3	2
14	7G	90IJ	2		ORDENADO	2	1	2
15	8H	34CD	2		ORDENADO	6	5	1
16	3C	78GH	1	10	ORDENADO	20	1	0
17	7G	56EF	1	8	ALEATORIO	12	4	2
18	7G	78GH	1	5	ALEATORIO	19	4	2
19	6F	34CD	3		ALEATORIO	8	3	3
20	1A	34CD	1	7	ORDENADO	17	4	0
21	1A	90IJ	3		ALEATORIO	17	3	0
22	4D	56EF	4		ALEATORIO	8	5	2
23	8H	56EF	3		ALEATORIO	2	4	0
24	4D	12AB	2		ORDENADO	20	1	3
25	7G	90IJ	3		ORDENADO	3	4	3

Figura 25. Segundo fragmento de la hoja 1 del archivo Excel

U	V	W	X	Y	Z	AA	
1	NUM. DEFECTOS CRÍTICOS	NUM. TOTAL DE DEFECTOS	SISTEMA DE CÁLCULO DE CALIDAD	NIVEL DE DEFECTOS	CALIDAD	TEST EXTENDIDO	COMENTARIOS
2	1	5	1	28	MEDIA	SI	AAAA
3	1	4	1	26	MEDIA	SI	ZZZZ
4	0	3	1	12	MEDIA	SI	RRRR
5	0	6	1	15	MEDIA	NO	Oooo
6	0	6	1	18	MEDIA	SI	XXXX
7	0	5	1	19	MEDIA	NO	AAAA
8	1	2	1	22	MEDIA	SI	YYYY
9	0	4	1	11	MEDIA	NO	UUUU
10	1	7	3	22	BAJA	SI	ZZZZ
11	1	4	3	15	MEDIA	NO	XXXX
12	0	1	3	3	ALTA	NO	EEEE
13	0	5	3	11	MEDIA	SI	UUUU
14	0	5	3	9	MEDIA	SI	UUUU
15	1	6	3	17	BAJA	SI	CCCC
16	1	5	3	16	BAJA	NO	EEEE
17	0	4	1	14	MEDIA	SI	CCCC
18	1	9	1	45	BAJA	NO	JJJJ
19	1	6	1	36	BAJA	SI	JJJJ
20	0	2	1	4	ALTA	NO	EEEE
21	0	4	1	14	MEDIA	NO	CCCC
22	0	6	1	18	MEDIA	NO	RRRR
23	1	7	1	35	BAJA	SI	Oooo
24	0	6	3	8	ALTA	NO	EEEE
25	1	7	3	22	BAJA	NO	PPPP

Figura 26. Tercer fragmento de la hoja 1 del archivo Excel

El otro escenario de comienzo es en disposición del diseño e implementación de la base de datos, pero con una situación previa de la misma a la ejecución del *script* de migración donde el contenido de la totalidad de sus tablas es nulo o vacío.

5.2.2 Migración

La etapa intermedia entre el escenario anterior y el escenario en el que desemboca, en el que todos esos datos se habrán visto volcados en los distintos registros de todas las tablas, es el marco de la migración de los mismos. Este escenario no es más que la ejecución del procedimiento de migración descrito e implantado en el *script* anunciado en 5.1.2, y que ahora se verá más explícitamente.

El punto de partida de ejecución del *script* es la posesión del archivo que contiene los datos, para ello se indica cuál es el archivo que queremos migrar. Después se cargan las librerías necesarias para realizar las diferentes operaciones que se harán con el archivo Excel y a continuación, se abre el archivo y obtiene el número de hojas con datos del mismo. Posteriormente, se establecerá conexión con la base de datos para la inminente lectura y volcado de los datos a esta. Las operaciones antes descritas se ilustran en la siguiente **Figura 27**, la cual es un fragmento perteneciente al *script*.

```
10  /*Nombre del archivo a cargar (se recomienda poner la ruta absoluta si no está junto al script)*/
11  $rutaArchivo = "ACME.xlsx";
12
13  /*Carga las librerías necesarias para la lectura y extracción de datos del archivo Excel*/
14  require_once "vendor/autoload.php";
15
16  /*IOFactory: permite abrir diferentes tipos de archivo*/
17  use PhpOffice\PhpSpreadsheet\IOFactory;
18
19  /*Abre el documento indicado anteriormente*/
20  $documento = IOFactory::load($rutaArchivo);
21
22  /*Calcula el número de hojas que tiene el archivo Excel y lo almacena en la variable $totalDeHojas*/
23  $totalDeHojas = $documento->getSheetCount();
24
25  /*Abre y establece conexión con la base de datos para realizar la migración*/
26  require_once "baseDatos_acme.php";
27  require "conexion_acme.php";
```

Figura 27. Fragmento del script de migración (I)

Sabiendo internamente las hojas que contienen datos, se procede entonces a reconocer y leer esos datos para posteriormente extraerlos y copiarlos en las correctas tablas que debidamente tienen que recoger cada uno de ellos. Para ello, se realiza una búsqueda automatizada por las hojas y se determina el número de fila y columna que contienen datos, a partir de las cuales no existen más datos en esa hoja. Una vez obtenidos dichos valores,

estos van a ser usados nuevamente para realizar la lectura automatizada de las celdas del archivo. La **Figura 28** muestra estas operaciones.

```

29      /*Itera por cada una de las hojas del documento*/
30      /* Tomo $totalDeHojas - 1 , porque la ultima hoja es auxiliar y recoge la fórmula aleatoria de los operadores*/
31      for ($indiceHoja = 0; $indiceHoja < $totalDeHojas - 1; $indiceHoja++)
32      {
33
34          /*Calcula el índice de la hoja en la que actualmente se dispone a leer*/
35          $hojaActual = $documento->getSheet($indiceHoja);
36
37          /*Calcula tanto el máximo valor de la fila (será un número entero) y el máximo valor de la columna (será una
38          letra) que contienen datos almacenados. Es decir, el límite de la filas y columnas del Excel para no seguir
39          leyendo filas vacías*/
40          $numeroMayorDeFila = $hojaActual->getHighestRow(); // Número
41          $letraMayorDeColumna = $hojaActual->getHighestColumn(); // Letra
42          /*Convierte la letra de la máxima columna con datos al número entero equivalente*/
43          $numeroMayorDeColumna = \PhpOffice\PhpSpreadsheet\Cell\Coordinate::columnIndexFromString($letraMayorDeColumna);
44
45          /*Itera y recorre fila a fila.
46          Empieza en la fila 2, porque la fila 1 recoge los títulos o encabezados de los datos que almacena esa misma
47          columna*/
48          for ($indiceFila = 2; $indiceFila <= $numeroMayorDeFila; $indiceFila++)
49          {
50              //Las columnas del archivo Excel están en orden descrito: Factoria, Codigo País, ID Test, Codigo Test...
51              $columnaA = $hojaActual->getCellByColumnAndRow(1, $indiceFila);
52              $factoria = $columnaA->getCalculatedValue();
53
54              $columnaB = $hojaActual->getCellByColumnAndRow(2, $indiceFila);
55              $codPais = $columnaB->getCalculatedValue();
56
57              $columnaC = $hojaActual->getCellByColumnAndRow(3, $indiceFila);
58              $idMaquina = $columnaC->getCalculatedValue();

```

Figura 28. Fragmento del script de migración (II)

Leídos y obtenidos los datos, el proceso siguiente es definir las instrucciones y peticiones de registro en la base de datos. En la **Figura 29** se pueden observar algunos ejemplos de dichas sentencias de instrucción.

```

310      /*-----*/
311      /*      MÁQUINAS      */
312      /*-----*/
313      /*INSTRUCCIÓN DE REGISTRO EN LA TABLA MAQUINAS DE LA BBDD*/
314      $insercionTablaMaquinas = "INSERT INTO maquinas (id_maquina, tipo_producto, factoria, codigo_pais)
315      values('$idMaquina', '$tipoProducto', '$factoria', '$codPais')";
316
317      /*PETICIÓN DE REGISTRO EN LA TABLA MAQUINAS DE LA BBDD*/
318      $volcadoMaquinas = mysqli_query($conexion, $insercionTablaMaquinas);
319
320
321
322      /*-----*/
323      /*      OERADORES      */
324      /*-----*/
325      /*INSTRUCCIÓN DE REGISTRO EN LA TABLA OPERADORES DE LA BBDD*/
326      $insercionTablaOperadores = "INSERT INTO operadores (id_operador, nombre_operador)
327      values('$idOperador', '$nombreOperador')";
328
329      /*PETICIÓN DE REGISTRO EN LA TABLA OPERADORES DE LA BBDD*/
330      $volcadoOperadores = mysqli_query($conexion, $insercionTablaOperadores);

```

Figura 29. Fragmento del script de migración (III)

Por último, una vez se haya completado el conjunto de las instrucciones y peticiones de registro para todas las tablas de la base de datos, el cometido a llevar a cabo ahora será el dar por finalizada y cerrar la conexión que en un comienzo se había originado

con la misma. Finalmente, si el proceso ha ido según lo esperado y no ha surgido ningún error, se muestra un mensaje reflejando el éxito de la migración en pantalla.

5.2.3 Resultados

Manifestada la situación de partida y con conocimiento del modo que se ha llevado a cabo para ejecutar cada una de las operaciones del *script* de migración, solo queda, por tanto, llevar a cabo dicha actuación y que el volcado de los datos de una ubicación a otra se haga efectivo. Finalmente, y después de todo ello, se habrá logrado conseguir el propósito y objetivo principal del TFG, realizar la migración automatizada de los datos contenidos en el archivo Excel a la base de datos diseñada e implementada. Estando en condiciones de dar soporte y cobertura al cuadro de mando que le sea asociado.

El aspecto final de algunas de las tablas contenidas en la base de datos cambiará de apariencia y presentarán en ellas los datos que previamente había almacenados en el archivo Excel. Muestra de ello son la **Figura 30** y **Figura 31** correspondiéndose con los registros ahora contenidos en la tabla *tests*. La **Figura 32** hace lo propio con la tabla *maquinas*. Las dos siguientes figuras, **Figura 33** y **Figura 34**, pertenecen a los datos registrados y contenidos en las tablas *configuracion_calidad* y *operadores*, respectivamente. Por último, en la **Figura 35** se recoge un fragmento de la totalidad de los tests que se almacenan en la tabla *variante_protocolo_1*.

id_maquina	id_test	fecha_hora	codigo_test	id_operador	id_software	id_configuracion	turno	tipo_test	id_proveedor	id_producto	id_lote	protocolo_medicion
1	1	2021-05-18 00:02:45	20200518001	11	1	3	NOCHE	INVESTIGACION	0	2B	12AB	1
1	2	2021-05-18 07:18:54	20200518002	11	1	3	MAÑANA	INVESTIGACION	0	8H	34CD	3
1	3	2021-05-18 13:56:51	20200518003	7	1	3	MAÑANA	PRODUCCION	0	1A	90IJ	4
1	4	2021-05-18 20:34:48	20200518004	11	1	3	TARDE	ANALISIS	0	2B	90IJ	3
1	5	2021-05-19 03:12:45	20200519001	13	1	3	NOCHE	INVESTIGACION	0	3C	56EF	4
1	6	2021-05-19 09:50:42	20200519002	13	1	3	MAÑANA	PRODUCCION	0	2B	56EF	5
1	7	2021-05-19 16:28:39	20200519003	4	1	3	TARDE	INVESTIGACION	0	6F	56EF	1
1	8	2021-05-19 23:06:36	20200519004	4	1	3	NOCHE	INVESTIGACION	0	1A	78GH	1
1	9	2021-05-20 05:44:33	20200520001	14	2	4	NOCHE	ANALISIS	0	5E	12AB	4
1	10	2021-05-20 12:22:30	20200520002	14	2	4	MAÑANA	INVESTIGACION	0	1A	78GH	5
1	11	2021-05-20 19:00:27	20200520003	6	2	4	TARDE	PRODUCCION	0	6F	12AB	2
1	12	2021-05-21 01:38:24	20200521001	1	2	4	NOCHE	ANALISIS	0	9I	56EF	4
1	13	2021-05-21 08:16:21	20200521002	13	2	4	MAÑANA	PRODUCCION	0	5E	56EF	1
1	14	2021-05-21 14:54:18	20200521003	12	2	4	MAÑANA	PRODUCCION	0	7G	12AB	2
1	15	2021-05-21 21:32:15	20200521004	1	2	4	TARDE	ANALISIS	0	8H	34CD	4
2	1	2021-05-18 02:14:04	20200518001	10	1	3	NOCHE	PRODUCCION	0	4D	90IJ	5
2	2	2021-05-18 09:31:33	20200518002	15	1	3	MAÑANA	INVESTIGACION	0	6F	56EF	1
2	3	2021-05-18 16:09:30	20200518003	5	1	3	TARDE	INVESTIGACION	0	1A	34CD	4
2	4	2021-05-18 22:47:27	20200518004	3	1	3	TARDE	ANALISIS	0	6F	34CD	1
2	5	2021-05-19 05:25:24	20200519001	4	1	3	NOCHE	ANALISIS	0	5E	12AB	1
2	6	2021-05-19 12:03:21	20200519002	13	1	3	MAÑANA	PRODUCCION	0	7G	12AB	1
2	7	2021-05-19 18:41:18	20200519003	14	1	3	TARDE	INVESTIGACION	0	8H	56EF	4
2	8	2021-05-20 01:19:15	20200520001	13	2	4	NOCHE	INVESTIGACION	0	3C	90IJ	1

Figura 30. Primer fragmento del aspecto final de la tabla tests

modo_analisis	numero_elementos_analizados	defectos_menores	defectos_mayores	defectos_criticos	nivel_defectos	calidad	validacion	test_extendido	comentarios
ALEATORIO	7	0	3	0	15	MEDIA	NULL	SI	ZZZZ
ORDENADO	8	3	3	0	21	MEDIA	NULL	NO	PPPP
ALEATORIO	13	3	0	1	26	MEDIA	NULL	NO	MMMM
ALEATORIO	11	5	1	1	35	BAJA	NULL	SI	EEEE
ORDENADO	8	0	0	1	20	MEDIA	NULL	NO	BBBB
ORDENADO	7	1	2	1	32	BAJA	NULL	SI	IIII
ORDENADO	3	4	1	1	33	BAJA	NULL	NO	KKKK
ALEATORIO	13	1	3	1	37	BAJA	NULL	SI	BBBB
ALEATORIO	7	1	0	0	1	ALTA	NULL	NO	RRRR
ORDENADO	11	5	3	1	24	BAJA	NULL	NO	AAAA
ORDENADO	15	1	2	0	7	ALTA	NULL	SI	VVVV
ORDENADO	3	3	0	1	13	MEDIA	NULL	NO	JJJJ
ORDENADO	18	5	0	0	5	ALTA	NULL	SI	JJJJ
ORDENADO	13	4	3	0	13	MEDIA	NULL	SI	JJJJ
ALEATORIO	13	5	0	1	15	MEDIA	NULL	NO	JJJJ
ORDENADO	14	3	3	1	41	BAJA	NULL	NO	EEEE
ALEATORIO	2	3	2	0	16	MEDIA	NULL	NO	QQQQ
ORDENADO	14	2	1	0	9	ALTA	NULL	NO	NNNN
ORDENADO	11	5	3	1	45	BAJA	NULL	NO	XXXX
ORDENADO	18	0	0	0	0	ALTA	NULL	SI	QQQQ
ALEATORIO	7	3	0	1	26	MEDIA	NULL	NO	LLLL
ALEATORIO	4	5	0	0	10	MEDIA	NULL	NO	UUUU
ALEATORIO	3	2	1	0	5	ALTA	NULL	NO	NNNN

Figura 31. Segundo fragmento del aspecto final de la tabla tests

id_maquina	tipo_producto	factoria	codigo_pais
1	PP	VALLADOLID	ES
2	PP	VALLADOLID	ES
3	PP	VALLADOLID	ES
4	PE	VALLADOLID	ES
5	PE	VALLADOLID	ES
6	PP	SALAMANCA	ES
7	PP	SALAMANCA	ES
8	PE	SALAMANCA	ES

Figura 32. Aspecto final de la tabla maquinas

id_configuracion	nombre_configuracion	sistema_calculo	umbral_bajo	umbral_alto
1	PE - ESTRICTO	1	10	30
2	PE - SUAVE	2	7	12
3	PP - ESTRICTO	1	10	30
4	PP - SUAVE	3	9	15

Figura 33. Aspecto final de la tabla configuracion_calidad

id_operador	nombre_operador
1	ANTONIO NÁRVAEZ
2	EUGENIO BIGAS
3	LETICIA SANTOS
4	ALBERTO MARTÍN
5	CARMEN HERNÁNDEZ
6	LORENA FINCIAS
7	MARÍA QUEVEDO
8	ÁNGEL DOLORES
9	ISABEL MUÑOZ
10	CRISTINA ANTÚNEZ
11	JOSE MULAS
12	ENRIQUE BAJO
13	ÁLVARO MATEOS
14	CELIA QUIJADA
15	ROBERTO PRIETO

Figura 34. Aspecto final de la tabla operadores

id_maquina	id_test	variante_protocolo_1
1	1	1
1	7	10
1	8	6
1	13	2
2	2	6
2	4	7
2	5	7
2	6	8
2	8	1
2	9	6
2	12	5
2	15	6
3	5	5
3	11	4
3	12	4
4	6	6
4	9	10
4	12	4

Figura 35. Fragmento del aspecto final de la tabla variante_protocolo_1

6

Conclusiones y Líneas Futuras

El último de los capítulos que recoge esta memoria de Trabajo de Fin de Grado está reservado para presentar las conclusiones que se han extraído a lo largo del desarrollo del presente proyecto y también poder exponer u ofrecer posibles líneas de actuación futura para seguir perfilando y mejorándolo.

6.1 Conclusiones

La primera de las conclusiones no es otra que el objetivo general de este TFG se ha cumplido. La base de datos ha sido creada y ha dado cabida para albergar a los datos migrados desde un archivo Excel, los cuales han sido trasferidos gracias a la ayuda del *script* implementado.

Pero a medida que el trabajo iba progresando se ha podido ir abstrayendo algunas otras más que se exponen a continuación.

Alojar los datos en una base de datos es más práctico, organizado, escalable y manejable que tenerlos almacenados en un archivo Excel.

Además, son infinidad de posibilidades las que se pueden ofrecer a raíz de poseer la información almacenada en una base de datos. En este caso concreto, al servir de soporte para un cuadro de mando o *dashboard* (desarrollado en un TFM realizado paralelamente) y derivar la multitud de información con la cual poder monitorizar, operar y obtener decisiones, forman un conjunto global potente a la par que muy útil.

Obviamente, cuando se dispone de una gran cantidad de datos para realizar una migración de estos a otro lugar donde almacenarlos, será mejor recurrir a un proceso

automatizado que hacerse de forma manual. El proceso será más breve y la probabilidad de cometer errores presentará un carácter más nulo.

Antes de abordar un planteamiento, cuestión o problema, se recomienda analizar la información de conocimiento desde la que se parte. Posteriormente, proponer o diseñar un plan de etapas y actuación, y los objetivos que se quieren conseguir en cada una de ellas. Tener en cuenta en la realización de las mismas las diferentes posibilidades y herramientas que se pueden emplear para conseguir dichos propósitos. Ir comprobando y evaluando etapa a etapa si se ha conseguido lo propuesto. En su defecto, realizar las modificaciones pertinentes. En disposición de conseguir y lograr dar por finalizada la última de las etapas, comprobar y validar el conjunto global desarrollado y si se ha dado solución al planteamiento, cuestión o problema inicial.

Otro de las reflexiones que destaca es que, aunque en un momento determinado, en cualquier contexto, se requiera de la decisión de uso de una determinada alternativa a la hora de presentar un abanico de ellas, no siempre la que se tiene que utilizar es la más potente, la más ampliamente utilizada, la más recomendada, etc., sino que depende de las prioridades o preferencias que se establezcan con anterioridad en cada caso.

Actualmente, no se requiere crear bases de datos desde cero. La razón es que existen diversidad de *frameworks*, los cuales son herramientas que facilitan modelos conceptuales y estructurados a partir de los que ir adaptando, modificando y desarrollando nuestro software o aplicación como se requiera o desee.

6.2 Líneas Futuras

La base de datos es capaz de almacenar los datos recogidos y de los que previamente se partía en el archivo Excel. En disposición de la misma, se me ocurren algunas otras funcionalidades posibles con las que podemos extender y mejorar su práctica.

Una vez creada e implementada la base de datos, una de las posibilidades que se podría realizar es crear una aplicación web junto con la cual poder instaurar la base de datos para que la empresa pueda seguir desempeñando su trabajo y realizando sus pertinentes tests de evaluación de los lotes de productos facilitados para que, a su vez,

todos los parámetros, variables, factores y resultados involucrados en los mismos queden registrados directamente en la base de datos.

Otra opción posible sería configurar en la misma aplicación buscadores que ayudaran a filtrar los tests de evaluación llevados a cabo y almacenados en la base de datos, en función del parámetro, factor o motivo que sea necesario o preferible en un momento dado.

Un nuevo aspecto funcional que se le podía incluir es la creación de perfiles o tipo de usuarios que posean diferentes privilegios para operar con la base de datos. Es decir, un perfil de usuario que tengan acceso de lectura únicamente, otros que puedan acceder tanto para leer como para modificar datos de la misma, etc...

En relación con el *script* automatizado para la migración de los datos, dadas las limitaciones que hay hasta el momento, ya que se debe de ingresar en el código del mismo el nombre del archivo que se quiere importar y alojarlo en una ubicación concreta... Un aspecto más práctico y que reduciría esa limitación, podría ser el de conseguir que mediante dicha herramienta nos presentara un posible cuadro de diálogo en el que nos preguntara o permitiera realizar una búsqueda por los directorios y seleccionar el archivo Excel con el que se quisiera realizar la migración.

7

Referencias

- [1] P. Marina Boillos, *“Desarrollo de cuadros de mando (dashboard) para la Industria 4.0”*, Trabajo Fin de Máster (Máster en Ingeniería de Telecomunicación). Universidad de Valladolid, ETSI de Telecomunicación, 2020. <http://uvadoc.uva.es/handle/10324/43254>
- [2] R. Bourdon, *Página web oficial del software WampServer*. 2021. <https://www.wampserver.com/en/>
- [3] A. Silberschatz, H. F. Korth & S. Sudarshan, *“Fundamentos de Bases de Datos”*. 6ª Edición. MacGraw-Hill ,2014.
- [4] R. Stephens, *“Diseño de Bases de Datos”*. Anaya, 2009.
- [5] R. Elmasri & S. B. Navathe, *“Fundamentals of Database Systems”*. 7ª Edición. Pearson Addison-Wesley, 2016.
- [6] MariaDB, *Documentación oficial de MariaDB*. Página web oficial de MariaDB, 2021. <https://mariadb.org/documentation/>
- [7] Oracle Corporation, *Documentación oficial de MySQL*. Página web oficial de MySQL, 2021. <https://dev.mysql.com/doc/>
- [8] Oracle Corporation, *Documentación oficial de Oracle database*. Página web oficial de Oracle, 2021. <https://docs.oracle.com/en/database/oracle/oracle-database/index.html>

- [9] Microsoft, *Documentación oficial de Microsoft SQL Server*. Página web oficial de Microsoft, 2021. <https://docs.microsoft.com/es-es/sql/sql-server/?view=sql-server-ver15>
- [10] PostgreSQL Global Development Group, *Documentación oficial de PostgreSQL database*. Página web oficial de PostgreSQL, 2021. <https://www.postgresql.org/docs/13/index.html>
- [11] W3schools, *Documentación y manual de uso del lenguaje SQL*. Página web oficial de w3schools, 2021. <https://www.w3schools.com/sql/default.asp>
- [12] Oracle Corporation, *Los tipos CHAR y VARCHAR (Documentación oficial de MySQL)*. Página web oficial de MySQL, 2021. <https://dev.mysql.com/doc/refman/5.7/en/char.html>
- [13] PHP, *Documentación y manual de uso oficial de lenguaje PHP, 2021*. <https://www.php.net/manual/es/>
- [14] PHPspreadsheet, *Documentación y manual de usuario oficial de PHPspreadsheet, 2021*. <https://phpspreadsheet.readthedocs.io/>

8

Anexo. Guía de ejecución

Para poder ejecutar y que nuestra herramienta automatizada funcione correctamente y que la migración de datos del archivo Excel a la base de datos se haga efectiva, debemos tener en cuenta **dos requerimientos** previos a la realización del volcado de los datos. Estos son:

- I. En primer lugar, tener alojado o disponer de un **servidor local** en nuestro equipo. Existen varios ejemplos que nos pueden prestar este servicio, algunos de ellos son: Apache (el más común y utilizado), Ngix, Internet Information Services (IIS) (implementado en Windows Server), Cherokee, Tomcat...
- II. El otro sería la habilitación de un **sistema gestor de bases de datos (SGBD)** con el que podamos implementar nuestro diseño de la base de datos y opera con ella. Destacan entre todos ellos: MySQL, MariaDB, PostgreSQL, Microsoft SQL Server, Oracle...

Una vez reconocidas las condiciones anteriores, podría darse el caso de encontrarnos en uno de los cuatro escenarios posibles:

- a. **No disponer de nada.** Ni de servidor, ni de sistema gestor de bases de datos.
- b. Que **sí** tengamos alojado un **servidor local**, pero **no** el **sistema gestor de bases de datos**.
- c. **No** tener montado nuestro **servidor local**, pero **sí** nuestro **sistema gestor de bases de datos**.
- d. **Disponibilidad** de **servidor local** y **sistema gestor de bases de datos habilitado**.

La siguiente cuestión una vez identificado el escenario en el que nos hallamos, sería qué hacer para poder disponer de ambos, en caso de que careciéramos de uno o de los dos, o cómo actuar, si es que ya disponemos de ellos o, en su defecto, de uno de ellos. A continuación, voy a sugerir los pasos a seguir, dependiendo del escenario anterior en el que nos encontremos. Digo sugerir porque como ya he dicho antes, existen multitud de ejemplos o posibilidades tanto para la elección del servidor, como para la del sistema gestor de bases de datos. Aquí expondré, más abajo, un caso concreto.

- Si nos encontramos en la situación y con las necesidades del **escenario a** debemos seguir los pasos, señalizados más delante y numerados **del 8.1 al 8.4, desde el principio hasta el final**. Obviamente, si no disponemos de nada, nos encontramos en el peor de los casos anteriormente descritos.
- Si el **escenario** al que nos enfrentamos es el **b**, ya en posesión de un servidor, lo que debemos hacer es implementar o importar el diseño de la base de datos, mediante el sistema gestor que nos presta dicha función. Podemos emplear el que consideremos más adecuado para ello. Si queremos seguir este ejemplo, para ello debería obviar los dos primeros pasos y comenzar en el **punto 8.3**.
- El **escenario c**, nos ubica con capacidad de disposición de sistema gestor de bases de datos, pero no así de servidor. Por lo tanto, debería seguir **los pasos 8.1 y 8.2**. Seguidamente, mediante el sistema de gestión de bases de datos del que se dispondría, debería implementar el diseño de la base de datos (**paso 8.3**), para finalizar la migración realizando lo anunciado en el **paso 8.4**.
- Si en el mejor de los casos disponemos de ambas facilidades y, por lo tanto, nos encontramos en **la escena descrita en** [¡Error! No se encuentra el origen de la referencia.](#), deberemos afrontar únicamente el **último de los puntos, el 8.4**.

Considerados cada uno de los escenarios presentes en base a los requerimientos de los que se precisa con anterioridad, veamos pues los pasos de actuación.

8.1 Descarga de WampServer

Descarga e instalación del software **WampServer**, a veces presentado también como **WAMP** (soporte de sistema operativo **Windows**, monta servidor **Apache**, **MySQL**, **MariaDB** (sistemas gestores de bases de datos) y otras tecnologías y lenguajes muy convenientes para el uso de nuestra herramienta, **PHP**, **phpMyAdmin**...). Existen otras posibilidades equivalentes para otros sistemas operativos: **LAMP** (Linux), **MAMP** (Mac OS), **XAMPP** (multiplataforma: Windows, Linux, Mac...).

- 1) Accede a la página oficial de **WampServer**, <https://www.wampserver.com/en/>
- 2) En el menú principal, que aparece arriba en horizontal, hacer *click* en **DOWNLOAD**.
- 3) Nos remite más abajo, en la misma página, y nos presenta dos versiones del software, para escoger la que es compatible con la arquitectura de nuestro ordenador, x64 ó x32 bits. Preferentemente escoger **Wampserver 3.2.0 64 bit x64**.
- 4) Pulsando en él, nos presenta un formulario de registro para envío de notificaciones e información, **NO** es necesario **completar ningún campo**. Debajo, nos dice textualmente: **WARNING : Vous devez avoir installé Visual Studio 2012 : VC 11 vcredist_x64/86.exe: <http://www.microsoft.com/enus/download/details.aspx?id=30679>. Debemos descargar e instalarnos esa librería** la cual es necesaria previamente para la posterior instalación de la aplicación. Para ello **accedemos al enlace** y nos **derivará a la página de Microsoft**, en la cual ya disponemos de la descarga.
- 5) Una vez instalada, en la misma ventana emergente que nos mostraba el formulario y la advertencia anteriores, al final del primer párrafo previo al formulario, tenemos **un enlace** que anuncia: **you can download it directly**. Si pulsamos en él, serás **redirigido a SourceForge y la descarga comenzará automáticamente**.
- 6) Los pasos posteriores una vez se haya descargado el instalador serán: en primer lugar, **localiza el archivo descargado** en la carpeta que tengas adjudicada para tus descargas. Normalmente esa ubicación será la carpeta Descargas. Una vez localizado, **ejecútalo**. Si te pide permisos de administrador, acepta.

- 7) El siguiente paso para continuar con la instalación será **seleccionar el idioma**. Una vez escogido, pulsa **OK**.
- 8) Posteriormente, nos mostrará la ventana para aceptar los términos del acuerdo de licencia. Selecciona **"I accept the agreement"** y pulsa **Next**.
- 9) La ventana que se mostrará a continuación nos presenta los requisitos de WampServer y componentes que necesitamos tener instalados para que este funcione correctamente. Son básicamente los módulos o librerías de Visual C++ (VC++) Redistributable, como por ejemplo el citado en el apartado 4). De no ser así, la instalación finalizará, pero WAMP no funcionará. Hacer *click* en **Next** para seguir con el proceso de instalación.
- 10) Ahora, escribe el nombre del directorio, sin contener espacios, en el que se va a instalar WampServer. Importante que este, se instale en una carpeta del directorio raíz de tu disco duro, en lugar de otras dependencias. **Se recomienda dejar la ruta y el nombre que viene por defecto (C:\wamp64)**. Hacer *click* en **Next** para continuar.
- 11) Seguidamente, nos solicitará que introduzcamos la carpeta de menú de inicio en la que queramos que se creen los accesos directos a WAMP. **Dejar por defecto y pulsar Next**.
- 12) Llegados hasta aquí, **empezará la instalación**, pero **durante el proceso** tendremos que **seleccionar** tanto **el navegador** que se quiera que WampServer use por defecto, como **el editor de texto** con el que mostrará sus archivos de configuración cuando estos sean requeridos. Los valores que introduce para dicho uso por defecto WampServer son Internet Explorer y bloc de notas, respectivamente. En este caso es más **recomendable usar el navegador Chrome y el editor de texto Notepad++** (u otro IDE específico que dispongamos en nuestro equipo). Podemos encontrarlos en estas rutas: **C:\Program Files (x86)\Google\Chrome\Application** y seleccionar el archivo **chrome.exe**, para el navegador y **C:\Program Files (x86)\Notepad++**, escogiendo **Notepad++.exe**, en este caso, para el editor de texto.

- 13) Finalmente, **el proceso de instalación se habrá completado**. Pulsaremos en **Finish** para cerrar el instalador y estar en condiciones de usar nuestra nueva aplicación.

8.2 Ejecución de WampServer

- 1) Podemos **iniciar WAMP** desde el **acceso directo**, si este fue creado, o desde el **menú de inicio de Windows**. Si es de esta segunda manera, buscamos por la palabra WampServer y nos aparecerá la aplicación. Pulsando sobre ella, emitirá una pestaña en la que debemos de conceder los permisos requeridos, **pulsa Sí**.
- 2) Una vez se haya iniciado, en nuestra **barra de herramientas del menú de inicio de Windows** (abajo, a la derecha), se **añadirá el icono de la aplicación**. Si el color de este icono es verde, significa que todos los servicios se están ejecutando correctamente. Si ese color es rojo, ningún servicio está en funcionamiento. Cuando los servicios se estén iniciando su color será naranja.
- 3) Pulsando con el botón izquierdo sobre el icono de la aplicación podemos acceder a todos los servicios que presta, si por el contrario presionamos con el botón derecho tenemos acceso a las herramientas y configuraciones de esta.

8.3 Creación de la Base de Datos (phpMyAdmin)

- 1) **Acceder a phpMyAdmin**. Para ello, abrir el **navegador** e introducir en él, la **dirección: <http://localhost/phpmyadmin/>**. También existe la posibilidad de acceder mediante el **icono de WAMP** disponible en la barra de herramientas del menú de inicio de Windows, pulsando con el **botón izquierdo** sobre él, de entre todos los servicios pulsamos sobre **phpMyAdmin** y automáticamente nos la abrirá en una pestaña del navegador.
- 2) Una vez hayamos accedido a la dirección, elegimos el idioma con el que queramos trabajar y nos pedirá las credenciales de usuario y contraseña. Si no hemos modificado el archivo de configuración en la instalación de WampServer, el

- usuario** es **root** y la **contraseña** estará vacía. Por lo tanto, estos valores se presentarán por defecto y lo que tenemos que hacer es presionar en **Continuar**.
- 3) Ya dentro de phpMyAdmin, arriba a la izquierda, nos presenta la posibilidad de crear una **Nueva** base de datos. Hacemos *click* y nos pedirá el **nombre de la base de datos**. Ponemos el que queramos, aunque nosotros estamos trabajando con el nombre **acme**. A la derecha nos presenta las preferencias de **cotejamiento** que va a usar nuestra nueva base de datos, pinchamos en el menú despegable y escogemos **utf8_spanish_ci**.
 - 4) Creada ya la base de datos, aunque se encuentre vacía, nos aparecerá junto con el resto, en el lateral izquierdo. Debemos escoger y **seleccionarla** para a continuación trabajar con ella.
 - 5) Una vez seleccionada, en la parte superior entre las distintas opciones con las que podemos operar, seleccionamos **Importar**.
 - 6) A continuación, nos pedirá **seleccionar nuestro archivo .sql**. Debemos buscar y seleccionar el archivo **acme.sql** que le he proporcionado. El resto de las opciones las dejamos por defecto y al final pulsamos en **Continuar**.
 - 7) Finalmente, ya tenemos nuestra base de datos completada con las diferentes tablas y relaciones que deben implementar sus características.

8.4 Migración del archivo Excel

- 1) Primeramente, debemos **acceder al directorio creado automáticamente** cuando instalamos **WampServer**. Este deberá estar en el disco dónde instalamos la aplicación y con el nombre que la denominamos en la instalación. Si todo se hizo o dejó por defecto, este directorio debería tener esta ruta: **C:\wamp64**.
- 2) Una vez dentro del mismo, debemos acceder al directorio **www**.
- 3) Dentro de él podemos crear tantos directorios como proyectos con los que trabajar. Por lo tanto, para llevar a cabo el nuestro, **descomprimos en este directorio** el

archivo comprimido **ACME.zip**, lo que dará lugar a un nuevo directorio con el nombre de **ACME**.

- 4) En consecuencia, si accedemos al mismo (la **ruta completa** en la que deberíamos estar llegado este punto sería **C:\wamp64\www\ACME**) debemos comprobar la existencia de los siguientes archivos: **ACME.xlsx**, **baseDatos_acme.php**, **conexion_acme.php**, **cierreConexion_acme.php**, **migracionExcel_acme.php** y la carpeta **vendor**.
- 5) Disponemos ya de todo, lo único que nos queda es ejecutar nuestra herramienta para completar nuestra migración y poblar nuestra base de datos creada previamente. Para ello, debemos abrir nuestro navegador e introducir en él la dirección: **http://localhost/ACME/migracionExcelACME.php**
- 6) Una vez ejecutada la instrucción anterior, se mostrará un mensaje de éxito en la pantalla, lo que indica que todo se ha ejecutado correctamente y ya **podemos consultar nuestra base de datos completada** en la aplicación **phpMyAdmin**, accediendo a ella tal y como se explicó en los primeros puntos de la sección anterior 8.3. Observaremos que la migración se habrá realizado correctamente y que los datos están almacenados en sus correspondientes tablas.

