

Diseño e implementación de la base de datos para una aplicación de control y gestión de riesgos empresariales

Miguel Ángel Carmona Quiñones

Grado de ingeniería Informática

Bases de datos

Jordi Ferrer Duran

Josep Cobarsí Morales

Junio 2023



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

FICHA DEL TRABAJO FINAL

Título del trabajo:	<i>Diseño e implementación de la base de datos para una aplicación de control y gestión de riesgos empresariales</i>
Nombre del autor:	<i>Miguel Ángel Carmona Quiñones</i>
Nombre del consultor/a:	<i>Jordi Ferrer Duran</i>
Nombre del PRA:	<i>Josep Cobarsí Morales</i>
Fecha de entrega (mm/aaaa):	06/2023
Titulación:	<i>Grado de Ingeniería Informática</i>
Área del Trabajo Final:	<i>Bases de datos</i>
Idioma del trabajo:	<i>Castellano</i>
Palabras clave	<i>TFG, Bases de datos y ERM</i>
<p>Resumen del Trabajo (máximo 250 palabras): <i>Con la finalidad, contexto de aplicación, metodología, resultados i conclusiones del trabajo.</i></p>	
<p>El presente trabajo de fin de grado tiene como primera finalidad diseñar una base de datos que permita a una gran empresa la gestión del riesgo empresarial y como segunda poner en práctica todo lo aprendido durante la realización del Grado de Informática en la UOC. En el actual contexto socioeconómico se pone de relevancia lo primordial que es incorporar la gestión del riesgo a los objetivos empresariales. Dado que es uno de los medios principales para dotar de resiliencia a las empresas antes el cambiante entorno.</p> <p>Para realizar esta base de datos se empleará la metodología de desarrollo <i>Waterfall</i>. Desarrollándose en el presente documento todas las fases habituales que la componen: planificación, seguimiento y control, análisis, diseño, implementación y validación. Como resultado de este trabajo se espera obtener un conjunto de entregables entre los que se encuentran: los scripts necesarios para la creación del sistema, procedimientos que aseguraran su correcto uso, creación y mantenimiento de una serie de indicadores indispensables para el negocio y la elaboración de un conjunto de pruebas para comprobar el correcto funcionamiento del sistema.</p> <p>Durante la realización del trabajo se han conseguido alcanzar la mayoría de los puntos solicitados. Quedando solo pendiente la implementación de una serie de elementos, propuestos a lo largo de la fase de diseño, que dotarían al sistema de un carácter más proactivo y del desarrollo de un conjunto de pruebas más específicas. Estos desarrollos no han tenido cabida dentro del proyecto debido a lo ajustado de los tiempos.</p>	

Abstract (in English, 250 words or less):

The first purpose of this end-of-degree project is to design a database that allows a large company to manage business risk and, secondly, to put into practice everything learned during the Computing Degree at the UOC. In the current socioeconomic context, the importance of incorporating risk management into business objectives becomes relevant. Since it is one of the main means to provide companies with resilience before the changing environment.

To create this database, the Waterfall development methodology will be used. Developing in this document all the usual phases that compose it: planning, monitoring and control, analysis, design, implementation, and validation. As a result of this work, it is expected to obtain a set of deliverables, among which are: the necessary scripts for the creation of the system, procedures that will ensure its correct use, creation, and maintenance of a series of essential indicators for the business and the elaboration of a set of tests to verify the correct functioning of the system.

During the execution of the work, most of the requested points have been achieved. Only pending the implementation of a series of elements, proposed throughout the design phase, which would give the system a more proactive nature and the development of a set of more specific tests. These developments have not had a place within the project due to tight times.

Índice

1. Introducción.....	10
1.1 Contexto y justificación del Trabajo	10
1.2 Objetivos del Trabajo.....	11
1.3 Enfoque y método seguido.....	11
1.4 Planificación del Trabajo	13
1.4.1 Alcance del proyecto.....	14
1.4.2 Descomposición de tareas.....	17
1.4.2.1 Planificación.....	17
1.4.2.2 Análisis y Diseño	19
1.4.2.3 Implementación y Validación	21
1.4.2.4 Documentación.....	24
1.4.2.5 Uso de la dedicación y tareas descartadas	25
1.4.2 Temporización del proyecto	26
1.4.3 Gestión de Riesgos.....	28
1.5 Breve resumen de productos obtenidos	30
1.6 Breve descripción de los otros capítulos de la memoria.....	30
2. Seguimiento y control	31
2.1 Seguimiento de la fase de Planificación	31
2.2 Seguimiento de la fase de análisis y diseño.....	32
2.3 Seguimiento de la fase de implementación y validación	33
3. Recogida y análisis de requisitos	34
3.1 Estudio del enunciado del problema	34
3.2 Estudio del contexto	36
3.3 Elaboración de requerimientos.....	37
3.3.1 Requisitos de producto	37
3.3.1.1 Requisitos funcionales.....	37
3.3.1.2 Requisitos no funcionales	40
3.3.2 Requisitos del proceso.....	41
4. Diseño	42
4.1 Diseño conceptual.....	43
4.1.1 ERM.....	43
4.1.2 LOG	49
4.1.3 DW.....	49
4.2 Diseño lógico.....	53
4.2.1 ERM.....	53
4.2.2 LOG.....	56
4.2.3 DW	56
4.3 Diseño físico.....	57
4.3.1 ERM	59
4.3.2 LOG.....	63
4.3.3 DW	64
5. Implementación	65
5.1 ERM	67
5.2 DW	76
6. Validación.....	79

6.1 ERM	79
6.2 DW	81
7. Conclusiones.....	84
8. Glosario	86
9. Bibliografía	87

Índice de figuras

Figura 1. Fases implementadas para el desarrollo en cascada del proyecto	12
Figura 2. Diagrama Gantt con la planificación del proyecto	27
Figura 3. Diagrama Gantt de seguimiento de la primera fase de planificación.	31
Figura 4. Diagrama Gantt de seguimiento de la fase de análisis y diseño	33
Figura 5. Gantt de seguimiento de la fase de implementación y validación	34
Figura 6. Captura de mana mental inicial sobre el enunciado del problema	35
Figura 7. Relación entre categoría e importancia.....	45
Figura 8. Diagrama del bloque funcional relacionado con departamento.....	46
Figura 9. Diagrama del bloque funcional relacionado con campaña	47
Figura 10. Diagrama del bloque funcional relacionado con riesgos	48
Figura 11. Diagrama del bloque funcional relacionado con el LOG	49
Figura 12. Diagrama del bloque funcional relacionado con DW.....	51
Figura 13. Diagrama conceptual completo del bloque ERM.....	52
Figura 14. Diagrama conceptual completo de los bloques LOG y DW.....	52
Figura 15. Relación de tablas que corresponden al bloque ERM.....	59
Figura 16. Relación de tablas que corresponden al bloque LOG	63
Figura 17. Relación de tablas que corresponden al bloque DW.....	64
Figura 18. Resultado de ejecución de batería de prueba de procedimientos...	80
Figura 19. Resultado de ejecución de validación de indicadores	83

Índice de tablas

Tabla 1. Temporización de las actividades de la asignatura	14
Tabla 2. Capacidad personal durante la primera entrega	15
Tabla 3. Capacidad personal durante la segunda entrega	15
Tabla 4. Capacidad personal durante la tercera entrega	15
Tabla 5. Capacidad personal durante la entrega final	15
Tabla 6. Capacidad de trabajo final	16
Tabla 7. Tareas que realizar durante la fase de planificación	17
Tabla 8. Tareas planificadas de la fase de diseño	20
Tabla 9. Tareas planificadas de la fase de implementación y validación	22
Tabla 10. Tareas planificadas de la fase final de documentación	24
Tabla 11. Distribución de dedicación por áreas de trabajo	25
Tabla 12. Tareas descartadas	25
Tabla 13. Lista de riesgos que pueden afectar al proyecto	28
Tabla 14. Nueva tarea acercamiento a Data Warehouse	35
Tabla 15. Requisitos funcionales del producto	37
Tabla 16. Requisitos no funcionales del producto	40
Tabla 17. Requisitos del proceso	41
Tabla 18. Conversiones de tipos efectuadas	57
Tabla 19. Definición de tablas que corresponden al bloque ERM	59
Tabla 20. Definición de tablas que corresponden al bloque LOG	63
Tabla 21. Definición de tablas que corresponden al bloque DW	64
Tabla 22. Definición de procedimientos y pruebas unitarias	68
Tabla 23. Definición de indicadores de DW	77
Tabla 24. Validación de indicadores de DW	81

1. Introducción

A la hora de abordar la realización del proyecto fin de carrera he considerado importante no solo dar protagonismo a la materia de este, las bases de datos. Sino además aplicar todos los conceptos aprendidos a lo largo de la carrera. Es decir, considero que este trabajo no solo es importante poner en relevancia el qué, sino además el cómo. Por lo que he decidido no solo restringirme a los procesos de construcción de un producto sino también incluir todos los procesos de gestión del proyecto y las posibles decisiones tomadas a la hora de realizarlo. Es por esto por lo que, no solo se incluirá en el documento final una descripción del producto obtenido, sino que además se mostrará todo el camino realizado para su consecución.

1.1 Contexto y justificación del Trabajo

Una de las cosas que ha quedado claro en los últimos años es que a las empresas no les queda más remedio que estar preparadas para adaptarse a lo inesperado si desean tener una continuidad. Esto ha ocasionado que la gestión de los riesgos empresariales haya pasado de ser una de las buenas prácticas de la gestión empresarial, a ser uno de los objetivos principales para la continuidad de un negocio. La búsqueda de esta resiliencia empresarial ha posicionado la gestión de los riesgos empresariales (ERM) como una de las principales herramientas para lograr esta meta. Esta gestión de riesgos es la encargada de identificar y analizar las posibles amenazas que puedan afectar a la gestión o los resultados de una compañía. Con la intención de minimizar su impacto en las operaciones de la empresa. Esta gestión de riesgos tiene que ser vista como un proceso a largo plazo que tiene como ultima finalidad la de ser incorporado a la cultura de la empresa. De manera que la información que nos brindan sus herramientas pueda ser utilizada como una más en el día a día de la toma de decisiones.

De todo esto se deduce la importancia que tiene todo este proceso para la gran empresa del sector financiero recogida en el enunciado del problema, “la empresa”. Esta empresa hasta el día actual solo realiza una gestión de riesgos de una manera manual y por criterio propio de cada uno de los departamentos que la componen. Por tanto, dado el contexto actual, esta empresa ha decidido realizar un giro en la cultura actual de la empresa para incluir entre sus valores la gestión del riesgo. Para ello se ha realizado un primer proceso de análisis a través de una consultora externa para definir el proceso de gestión de riesgos necesario a implementar en la empresa con el fin de mantener controlados los riesgos. En inicio, “la empresa” no tiene claro si se decantará por elegir alguna de las soluciones informáticas existentes ERM (*Enterprise Risk Management*) en el mercado para la gestión de riesgos empresariales o creará un modelo de gestión propio. Sin embargo, como resultado de esta primera fase, y con los aspectos principales del proceso detectado por la consultora, quiere implementar un modelo de BD para poder recoger toda la información relevante.

Este proyecto tiene una doble vertiente y se espera conseguir dos resultados a través de su elaboración. Por un lado, está enfocado a la obtención de un producto final. Una base de datos que cumpla los requerimientos de “la empresa”. Este producto será el punto de partida para que “la empresa” se inicie en la correcta gestión de su riesgo empresarial. Por otro lado, y debido al contexto en el que se realiza, un trabajo de fin de grado, se espera que el presente trabajo refleje una muestra de los conocimientos adquiridos a través de la realización de las diferentes asignaturas y más concretamente las implicadas en el área de trabajo de este, Base de datos. Por tanto, como resultado de este otro aspecto se espera obtener una serie de entregables propios de la realización de cualquier proyecto de este tipo: una planificación, un control de riesgos, una relación de requerimientos formales, un diseño (conceptual, lógico y físico) que recoja y soporte estas necesidades, una implementación final que posibilite la gestión y el acceso a la información y, por último, una serie de procesos que aseguren la trazabilidad de los datos y la validación del sistema.

1.2 Objetivos del Trabajo

A continuación, se recogen una serie de objetivos que se persiguen durante la elaboración del siguiente trabajo fin de grado:

- Poner en práctica los conocimientos adquiridos a través de la realización del grado de ingeniería informática y más concretamente los de las asignaturas de Bases de datos.
- Ampliar conocimientos tanto sobre el uso de nuevas herramientas como sobre un campo de estudio no conocido.
- Dado un documento de definición detectar cuales son las necesidades básicas que debe de cumplir un sistema.
- Proponer posibles funcionalidades que complementen o evolucionen las necesidades del sistema definido.
- Crear un diseño siguiendo la metodología de Bases de datos que cubra los requerimientos solicitados.
- Implementar un sistema final en SQL sobre el diseño definido que encapsule las funciones de acceso a los datos.
- Elaborar todo el proyecto siguiendo una metodología de trabajo definida y ajustándose a una planificación.

1.3 Enfoque y método seguido

A lo largo de la carrera durante la realización de distintas asignaturas (Gestión de proyectos, Ingeniería del software, Proyecto de desarrollo del software...) hemos podido comprobar la utilización de las distintas metodologías existentes para realizar un proyecto de desarrollo, tanto agiles como tradicionales. Pero dada la propia naturaleza del trabajo final de grado, y en mi opinión, la estrategia que más casa con sus características es la metodología tradicional de desarrollo en cascada (Waterfall)¹. Esta metodología tiene como principal desventaja que

¹ ¿Cuál es la metodología más adecuada para tu proyecto? [en línea]. Disponible en: <https://www2.deloitte.com/es/es/pages/technology/articles/waterfall-vs-agile.html>

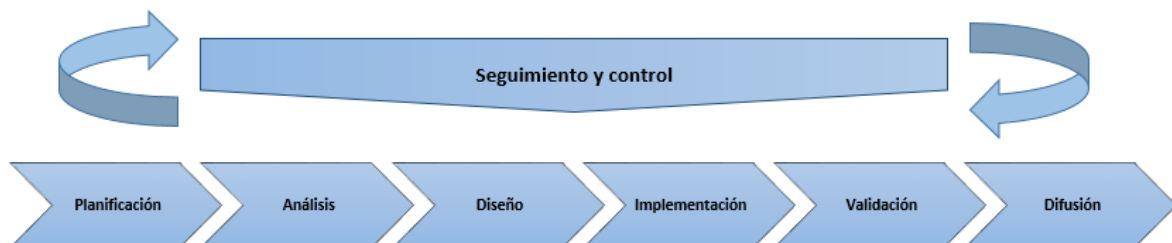
tiene una estructuración rígida y es poco ágil a la hora de afrontar los cambios. Es por esto por lo que durante su realización es necesario utilizar una planificación más exhaustiva y mantener un seguimiento constante de las tareas para intentar minimizar el impacto del cambio. Sin embargo, esta limitación en nuestro caso nos afecta de manera leve, dado que los requerimientos son estables y no hay una iteración con el cliente. En contraposición, una de las principales ventajas del uso de este método es su capacidad para concretar las fechas finales y fijar una temporización sobre cada una de las tareas. Lo que posibilita abordar el desarrollo del proyecto de una manera continuada y estable. Y, además, nos permitirá realizar un seguimiento preciso y anticipar posibles problemáticas que puedan comprometer la fecha de entrega.

Por todos estos motivos he decidido utilizar esta metodología de desarrollo en cascada (*Waterfall*). No es que no crea que el proyecto no pueda ser desarrollado mediante el uso de una metodología más ágil. Sino que dado que el cambio está muy acotado y solo hay un recurso que gestionar no creo que su uso me aportara muchos beneficios.

Una vez decidida la metodología deberemos de ver como se aplica en nuestro caso. Esta estrategia de trabajo puede ser aplicada en proyectos de cualquier índole tal como se recoge en la asignatura de Gestión de proyectos. En todos los casos descompone el trabajo en una serie de fases de carácter secuencial cuyos resultados sirven de base para la realización de las siguientes. Normalmente de manera paralela a todo el proceso se desarrolla una fase encargada de realizar un seguimiento y control durante todo el proyecto. En cierta medida existe una cierta retroalimentación entre cada una de las fases. Pero sobre todo con la fase de seguimiento y control. En mi caso, y dada la naturaleza del proyecto, he decidido que el conjunto de fases que implementaré durante el desarrollo serán muy similares a las propias de cualquier ciclo de vida de desarrollo de un sistema (*SDLC*).² Aunque he decidido particularizar algunas de ellas. En mi caso la fase de verificación se corresponderá exclusivamente con la fase validación. Omitiré la fase de mantenimiento dado que no existirá una continuidad en el producto. Y, sin embargo, incluiré una fase final centrada en la difusión del trabajo.

Veamos en detalle que fases se realizarán y cuál será su contenido.

Figura 1. Fases implementadas para el desarrollo en cascada del proyecto



- **Planificación.** Esta fase es una de las más importantes para esta metodología. En ella se deberá de concretar la estrategia que se utilizará durante todo el proyecto, se deberá de fijar el alcance, definir las tareas a abordar, realizar

² Pere Mariné Jové y José Ramón Rodríguez. Componentes de la gestión de proyectos: las áreas de conocimiento. Material de la asignatura Gestión de proyectos. UOC PID_00247940

una temporización de estas y elaborar un análisis de los posibles riesgos que afecten al proyecto. Esta fase se realizará durante la primera PEC1.

- **Análisis.** Esta parte está centrada en concretar el problema y extrapolar una serie de requerimientos que deben de ser cubierto por el producto final. Se deberá elaborar un análisis del enunciado del problema, realizar un estudio del contexto y elaborar una justificación de este. Como resultado de este proceso se deberán de obtener una lista de requerimientos tanto funcionales como no funcionales del sistema. Esta parte se realizará durante la segunda de las entregas de manera conjunta con la siguiente fase.
- **Diseño.** Esta es una de las fases de más peso durante la elaboración de un sistema de base de datos. En ella se partirá de los requerimientos encontrados en la fase anterior y a través de un proceso de refinado se irán elaborando una serie de modelos sucesivos (conceptual, lógico y físico). Hasta generar un modelo final que se podrá implementar en el sistema de bases de datos final. Esta parte se abordará durante la segunda de las entregas.
- **Implementación.** En esta etapa se creará el entorno de trabajo y según el diseño físico obtenido en la fase anterior se creará el sistema final. En este punto también se elaborarán también todos los elementos de programación necesarios para la utilización del sistema (procedimientos, funciones, disparadores...). Esta fase se realizará durante la tercera de las entregas de manera conjunta a la fase de validación.
- **Validación.** Como norma general en todo proceso existe una fase dedicada a la validación y verificación por parte del cliente del producto realizado. En nuestro caso la verificación final por el cliente no tendrá cabida en el proyecto. Pero si realizare todo el proceso de validación del sistema. Intentaré abordar la validación del sistema en todos los niveles posibles. Dado que considero que es una buena práctica el diseño orientado a las pruebas (*Test-driven development o TDD*) también intentare utilizarlo durante la realización de la implementación. Esta fase se temporizará con la anterior durante la tercera de las entregas.
- **Difusión.** Esta fase es ajena al propio ciclo de vida de un sistema normal. Pero en nuestro caso tiene gran importancia. Dado que durante ella se elaborará la entrega final del proyecto. En gran medida la nota final de la asignatura dependerá de la correcta difusión del trabajo realizado. Siendo necesaria en esta fase el refinado de la memoria y la elaboración de la presentación del proyecto. Esta fase se realizará durante la última parte del proyecto.

1.4 Planificación del Trabajo

De partida, es necesario indicar que el proyecto no sigue las líneas generales de un proyecto tradicional. En este caso tanto el marco de trabajo y como las fechas de entrega están claramente definidas. Es por esto por lo que para una correcta planificación del trabajo en inicio abordaré la definición del alcance. Dado que la capacidad de trabajo es la que fijara en gran medida las tareas que tendrán cabida en el proyecto.

1.4.1 Alcance del proyecto

Lo normal es que una de las primeras tareas a la hora de realizar un proyecto sea fijar el alcance del proyecto. De este modo nos aseguramos a través de su definición que el producto final se corresponderá con el requerido. A la hora de definir el alcance se debe de tener en cuenta, sobre todo si es un producto nuevo, que hay que diferenciar entre lo que es el alcance del proyecto y el alcance del producto. No es lo mismo que se espera a la larga de un producto con respecto a que es lo que se va a realizar en un proyecto concreto. Es decir, es básico para un proyecto concretar qué es lo que se va a realizar y que no se va a abordar. Esta definición de la naturaleza del trabajo requerido es la que nos suele determinar en gran medida la temporización de un proyecto. Aunque, esto no significa que sea el único criterio aplicable, existen otros como: costes, requerimientos de fechas por parte del cliente, disponibilidad del equipo... Todo depende de la naturaleza y el contexto del proyecto.

En este caso que nos me ocupa el calendario es fijo y las tareas no pueden exceder el tiempo definido. Por lo que, al contrario de lo que suele ser habitual, en nuestro caso deberá de ser la capacidad la que dicte en gran medida producto final. Y deberá de estar presente a la hora de fijar cuales son las tareas que abordar, y nos determinará que puede ser realizado y que no. Es por esto por lo que, para definir mi alcance la primera tarea que he realizado en la planificación del proyecto es definir la capacidad disponible.

La temporización de las actividades definidas para la asignatura son las siguientes:

Tabla 1. Temporización de las actividades de la asignatura

Actividades	Inicio	Entrega	Días	Calificación
PEC1: Plan de trabajo	02/03/2023	20/03/2023	19	28/03/2023
PEC 2	21/03/2023	24/04/2023	35	05/05/2023
PEC 3	25/04/2023	25/05/2023	31	29/05/2023
Entrega final	26/05/2023	23/06/2023	29	07/07/2023
Tribunal de evaluación virtual	26/06/2023	30/06/2023	5	
		Total días	119	

Omitiré de la temporización la última actividad, la defensa del proyecto, porque no presenta actividades que deban de ser planificadas. Veamos el resto de ellas.

Debido a mis circunstancias, las horas que puedo dedicar a la asignatura en este rango de días no tienen un carácter homogéneo. Dado que me encuentro actualmente trabajando y tengo además una serie de cargas familiares. Es por esto por lo que ante este calendario si queremos ser lo más realistas posibles es necesario descender a los días concretos para obtener el número de horas de mi capacidad. Como norma general, debido a mi jornada laboral entre semana mi capacidad suele estar limitada a dos horas tres días de la semana y de una hora los otros dos días. Durante los fines de semana y los festivos esta cantidad de tiempo queda compensada y puedo dedicar un mínimo de 6 horas por día.

Por lo que dada esta capacidad semanal para la realización de la **PEC1** tendré:

Tabla 2. Capacidad personal durante la primera entrega

Lu	Ma	Mi	Ju	Vi	Sa	Do	Lu	Ma	Mi	Ju	Vi	Sa	Do	Total
			02-mar	03-mar	04-mar	05-mar	06-mar	07-mar	08-mar	09-mar	10-mar	11-mar	12-mar	
			2	1	6	6	2	2	1	2	1	6	6	35
13-mar	14-mar	15-mar	16-mar	17-mar	18-mar	19-mar	20-mar							
2	2	1	2	1	6	6	2							22
														Total horas
														57

Para la segunda entrega **PEC2** y teniendo en cuenta que hay dos festivos:

Tabla 3. Capacidad personal durante la segunda entrega

Lu	Ma	Mi	Ju	Vi	Sa	Do	Lu	Ma	Mi	Ju	Vi	Sa	Do	Total
							21-mar	22-mar	23-mar	24-mar	25-mar	26-mar		
								2	1	2	1	6	6	18
27-mar	28-mar	29-mar	30-mar	31-mar	01-abr	02-abr	03-abr	04-abr	05-abr	06-abr	07-abr	08-abr	09-abr	
2	2	1	2	1	6	6	2	2	1	2	6	6	6	45
10-abr	11-abr	12-abr	13-abr	14-abr	15-abr	16-abr	17-abr	18-abr	19-abr	20-abr	21-abr	22-abr	23-abr	
6	2	1	2	1	6	6	2	2	1	2	1	6	6	44
24-abr														
2														2
														Total horas
														109

Para el siguiente hito **PEC3** con el festivo dispondré de:

Tabla 4. Capacidad personal durante la tercera entrega

Lu	Ma	Mi	Ju	Vi	Sa	Do	Lu	Ma	Mi	Ju	Vi	Sa	Do	Total					
							25-abr	26-abr	27-abr	28-abr	29-abr	30-abr	01-may	02-may	03-may	04-may	05-may	06-may	07-may
								2	1	2	1	6	6	6	2	1	1	6	6
08-may	09-may	10-may	11-may	12-may	13-may	14-may	15-may	16-may	17-may	18-may	19-may	20-may	21-may						
2	2	1	2	1	6	6	2	2	1	2	1	6	6	40					
22-may	23-may	24-may	25-may																
2	2	1	2											7					
														Total horas					
														89					

Y para la **Entrega final**:

Tabla 5. Capacidad personal durante la entrega final

Lu	Ma	Mi	Ju	Vi	Sa	Do	Lu	Ma	Mi	Ju	Vi	Sa	Do	Total			
							26-may	27-may	28-may	29-may	30-may	31-may	01-jun	02-jun	03-jun	04-jun	
								6	6	2	2	1	2	1	6	6	32
05-jun	06-jun	07-jun	08-jun	09-jun	10-jun	11-jun	12-jun	13-jun	14-jun	15-jun	16-jun	17-jun	18-jun				
6	2	1	2	1	6	6	2	2	1	2	1	6	6	44			
19-jun	20-jun	21-jun	22-jun	23-jun													
2	2	1	2											7			
														Total horas			
														83			

Es necesario indicar que en este último periodo existe un día festivo y que he marcado el último día como no disponible, dado que tengo un evento familiar. Por tanto deberé de realizar la entrega un día antes del máximo permitido. En suma, tendré unas 338 horas de dedicación al menos aseguradas.

Sin embargo, tal como se comentará en el análisis de riesgos he decidido ser conservador y de este tiempo guardar un porcentaje de tiempo como colchón de seguridad, un 10%. Esto tiempo nos permitirá absorber un pequeño margen de descuadre. En otras circunstancias quizás hubiera preferido que el porcentaje

fuera mayor pero el calendario disponible no permite mucho rango de maniobrabilidad.

Teniendo en cuenta el porcentaje indicado mi alcance final será:

Tabla 6. Capacidad de trabajo final

Actividades	Inicio	Entrega	Días	Horas	Colchón 10%	Disponible
PEC1: Plan de trabajo	02/03/2023	20/03/2023	19	57	6	51
PEC 2	21/03/2023	24/04/2023	35	109	11	98
PEC 3	25/04/2023	25/05/2023	31	89	9	80
Entrega final	26/05/2023	23/06/2023	29	83	8	75
Tribunal de evaluación virtual	26/06/2023	30/06/2023	5			
		Total	119	338	34	304

Por tanto, en resumen de la capacidad total de 338 solo planificaré tareas por unas 304 horas. Esta capacidad nos fijará que mi producto mínimo viable *MVP* será aquel que cubra con todos los requerimientos fijados por el enunciado del proyecto y pueda ser realizado en 304 horas. En el caso de que el seguimiento determine que no es necesario este colchón o que dispongo de un conjunto horas sobrantes, replanificaré su uso en la realización de nuevas tareas que de partida fueron descartadas o para mejorar la calidad de alguno de los requerimientos fijados. A la hora de seleccionar estas tareas, he decidido focalizar todo el esfuerzo disponible en la parte del desarrollo funcional. Simplificando en la medida de lo posible todo lo que sea ajeno a la solución (arquitectura física de la solución, mantenimiento...)

Por esta limitación he desechado una serie de tareas que de partida me parecían interesantes incorporar a la solución planteada como:

- Montar un entorno de trabajo y pruebas a través de contenedores Docker.
- Automatizar la ejecución de las pruebas automáticas en un entorno de prueba.
- Elaboración un proyecto de pruebas que facilite la ejecución y futura automatización de las pruebas.
- Implementar herramientas automatizadas para controlar la calidad del código (que se cumplan reglas de salida, de validación...)
- Securización del sistema así como de la base de datos, posible uso de TDE.
- Elaboración de una política de control de desastres.
- Alta disponibilidad.
- Revisión de Rendimiento.

Incorporar estas características a la solución solo tendría sentido en un proyecto con más capacidad. Ya que requerirán un mayor tiempo de dedicación y algunas de estas concretamente de estudio. Además como resultado de su análisis, algunas posiblemente no tengan una solución lo suficientemente ágil para que realmente aporten al proyecto y deban de ser desechadas.

Con esto en mente, pasaré a concretar cuales son as tareas seleccionadas y, por tanto, en las que he dividido el proyecto.

1.4.2 Descomposición de tareas

Lo primero es recordar que uno de los puntos principales en esta metodología de trabajo es definir de manera muy clara las tareas que se han de abordar. Dado que estas son las que nos posibilitan realizar un seguimiento constante de la evolución del proyecto. Sin embargo, también es cierto que este conjunto de tareas no tiene por qué ser algo inmutable y la planificación ira evolucionando conforme las tareas se vayan concretando. Pudiendo incrementarse o decrementarse el número de tareas. Así como modificarse su estimación inicial de horas. Este proceso es algo que se suele refinar conforme a la experiencia previa en procesos similares. Cosa que no ocurrirá en este caso dada la naturaleza propia del proyecto.

Estructuraré el conjunto de tareas según las diferentes fases que ya he definido en la metodología de trabajo para la realización del proyecto. A la hora de realizar la descomposición de las tareas a realizar (planificación operativa) he visto que de manera natural ya casaban con las actividades fijadas para la asignatura (planificación estratégica). Por tanto, he tenido en cuenta el tiempo disponible en cada una de las entregas a la hora de realizar la descomposición y valoración. Ya que mantener esta sincronidad evitando solapamientos me pareció deseable, por que facilitara en parte la temporización del proyecto y valdrá para que el seguimiento de los hitos de seguimiento fijados por la asignatura tenga un interés más significativo. Por otra parte, he decidido no crear tareas específicas para el seguimiento y control dado que asumo que se deberá de realizar de manera constante y paralela a todo el proceso y por tanto no tiene sentido temporizarlas. Como resultado de este proceso deberían de existir una serie de entregables en los que se refleje el seguimiento de la planificación y de los riesgos a lo largo de la elaboración del proyecto. Los cuales podría tener sentido planificar. Pero al final asumiré la elaboración de estos entregable como una parte de la creación del documento final de cada una de las entregas, que ya se encuentra planificado.

Veamos cada una de las fases definidas por la metodología, como se temporizarán y en qué PEC estarán recogidas.

1.4.2.1 Planificación

De esta parte del desarrollo es de la que se ocupa el inicio del proyecto. Por tanto, existirá una correspondencia entre la **PEC1** y el proceso de planificación. Esta fase tiene como objetivo fijar la metodología de trabajo y poner en marcha las estructuras necesarias acorde a la metodología elegida para realizar el trabajo (Gantt, análisis de riesgos...). Además, en esta fase se realiza una descomposición del trabajo en paquetes actividades más pequeñas. Realizando una estimación del tiempo necesario para su ejecución e identifican las posibles dependencias que puedan afectar a su realización. Esto nos permitirá posteriormente concretar estas actividades en un calendario de trabajo.

Veamos al conjunto de tareas que se han planificado para esta fase.

Tabla 7. Tareas que realizar durante la fase de planificación

PL-001	Repaso de documentación de asignaturas años anteriores
--------	--

	Tipo	Investigación	Duración	9h
	Objetivo	Se realizan dos acciones. Por un lado, se repasarán las asignaturas cursadas para refrescar cualquier tipo de materia que tenga cabida en el proyecto. Por otro lado, se revisarán los recursos de las asignaturas: Uso de Bases de datos y Diseño de Bases. Dado que estas dos asignaturas las tengo convalidadas y será necesario asegurar las metodologías de trabajo.		
	Dependencia	<Sin dependencias>		
	Observaciones	Esta tarea surge de la elaboración del análisis de riesgos y tiene como intención paliar el riesgo RG-002 No cumplimiento de alguna de las metodologías de desarrollo de BD . Además de tener una correspondencia con el habitual proceso de documentación para la elaboración de cualquier proyecto.		
PL-002	Decidir metodología de trabajo			
	Tipo	Organización de trabajo	Duración	2h
	Objetivo	Se deben de revisar las diversas metodologías tradicionales y ágiles para fijar cuál de ellas es mejor utilizar para la elaboración del proyecto.		
	Dependencia	Lo lógico es que no se aborde hasta que sea completada la tarea PL-001 Repaso de documentación de asignaturas años anteriores		
	Observaciones	---		
PL-003	Generar portfolio de proyecto			
	Tipo	Organización de trabajo	Duración	1h
	Objetivo	Para la correcta elaboración del trabajo se ha decidido estructurar toda la información del proyecto en un portfolio.		
	Dependencia	<Sin dependencias>		
	Observaciones	Con motivo del análisis de riesgo RG-003 Fallo tecnológico se decide incluir esta carpeta de trabajo en el Google Drive personal del alumno. Y en caso de que por problemas de conectividad no se pueda realizar la sincronización se procederá a hacer una copia de seguridad en una unidad de disco externo después de cada sesión de trabajo según RG-006 Problemas de conectividad		
PL-004	Generar un Git para control de versiones			
	Tipo	Organización de trabajo	Duración	1h
	Objetivo	Dentro de la estructura de trabajo se diferenciará una de las partes dedicadas a almacenar los scripts generados en la realización del proyecto. En esta parte será deseable disponer de un control de versiones. Por lo que será necesario dar de alta un proyecto en Git para su gestión.		
	Dependencia	<Sin dependencias> Se podría posponer en caso de falta de tiempo hasta el uso de esta carpeta		
	Observaciones	Esta tarea surge de la elaboración del análisis de riesgos y tiene como intención paliar el riesgo RG-005 Destrucción parcial o pérdida de trabajo en desarrollo .		
PL-005	Calcular capacidad máxima del proyecto según calendario			
	Tipo	Planificación Tareas	Duración	4h
	Objetivo	Dado que en gran medida el alcance que es abordable del proyecto dependerá del tiempo disponible. Se deberá de calcular el tiempo que puede ser empleado en cada temporización.		
	Dependencia	<Sin dependencias>		
	Observaciones	Se tendrá en cuenta para su elaboración el porcentaje de seguridad indicado en el riesgo RG-001 Fallo en estimaciones .		
PL-006	Descomposición de tareas del proyecto			
	Tipo	Planificación Tareas	Duración	9h
	Objetivo	En este punto se procederá a generar una lista de cada una de las actividades que a priori se consideran necesarias para la ejecución del proyecto. Algunas de ellas serán fijadas por la propia metodología de trabajo y otras por la propia naturaleza de la asignatura. Además se incluirán toda posible idea de funcionalidad que pueda aportar al proyecto. Se intentará en la medida de lo posible determinar		

		cuáles serán optativas y cuáles deberán ser necesarias. Así como las posibles dependencias entre ellas.
Dependencia	Solo podrá ser abordada después de los puntos. PL-001 Repaso de documentación de asignaturas años anteriores PL-002 Decidir metodología de trabajo	
Observaciones	---	
PL-007	Fijar alcance	
Tipo	Planificación Tareas	Duración 4h
Objetivo	Dada la capacidad determinada para el proyecto y el conjunto de tareas necesarias y optativas. Se debe de concretar cuales, de partida, compondrán el producto mínimo viable MVP.	
Dependencia	Claramente dependerá de la realización de las tareas PL-005 Calcular capacidad máxima del proyecto según calendario y PL-006 Descomposición de tareas del proyecto	
Observaciones	---	
PL-008	Elaborar diagrama Gantt para el seguimiento de la planificación	
Tipo	Planificación Tareas	Duración 6h
Objetivo	Se deberá de concretar la herramienta a utilizar y elaborar un diagrama Gantt para el seguimiento de la planificación. Para su elaboración se tendrán en cuenta las dependencias localizadas durante la descomposición de tareas.	
Dependencia	Dependerá de lo determinado en las tareas PL-006 Descomposición de tareas del proyecto y PL-007 Fijar alcance	
Observaciones	Si es viable tener en cuenta tener una copia del instalador asociado según lo recogido en el plan de acción del Riesgo RG-003 Fallo tecnológico	
PL-009	Elaborar análisis de Riesgos	
Tipo	Análisis	Duración 3h
Objetivo	Elaborar un análisis de riesgos para el proyecto para minimizar el impacto de los acontecimientos adversos en el proyecto	
Dependencia	<Sin dependencias> Puede ser un punto abierto a ir trabajando conforme se vaya elaborando la primera PEC1	
Observaciones	---	
PL-010	Generar documento de entrega para la PEC1	
Tipo	Documentación	Duración 12h
Objetivo	Generar un documento de entrega que recoja las especificaciones recogidas en el enunciado del proyecto. Verificar que se cumplen los criterios definidos en el documento de validación suministrado.	
Dependencia	Cada una de las partes del documento está vinculada a la realización de las tareas recogidas en él por tanto tiene una dependencia con todo el trabajo realizado en esta fase. Sin embargo, esto solo limita su elaboración a la finalización de alguna de las tareas.	
Observaciones	Considero que uno de los aspectos más importantes de la elaboración del documento debe residir en la transmisión correcta del trabajo que se está realizando. De este modo se podrá evaluar si el camino elegido es el correcto y realizar alguna corrección en el caso de que el personal docente detecte que el enfoque no es el acertado. Es por esto por lo que he determinado que en esta fase inicial del proceso es importante focalizarme en ella y por lo que tiene reservado casi un 25% de la capacidad de esta fase para su elaboración. Se elaborará de manera paralela a esta fase.	

1.4.2.2 Análisis y Diseño

Esta parte engloba el mayor porcentaje de esfuerzo y es donde se elaborará la definición y la estructura a partir de la cual se desarrollará el resto del proyecto. Se alinea con la segunda de las entregas la **PEC2**. Aunque en realidad engloba dos

fases de nuestro ciclo de vida del proyecto. La fase de análisis y la fase de diseño. Durante la fase de análisis se intentará concretar el problema a resolver, su justificación, sus objetivos y en la medida de lo posible el contexto. Como resultado de esta fase se generará una serie de requisitos que el sistema debe de cumplir. Como norma general estos requisitos se deberían de recoger a través de una iteración con el cliente y de un posterior refinado. Sin embargo, para nuestro caso estos requisitos se extrapolarán de lo indicado en el enunciado. Durante la fase de diseño, según lo recogido por la asignatura de Diseño de Bases, se procederá a modelar un sistema en el que se recojan todas las necesidades detectadas durante el análisis. Durante esta fase se realizarán diseños sucesivos en los que se partirá de los requerimientos para acabar en una implementación real.

La lista de tareas a abordar en esta fase serán las siguientes:

Tabla 8. Tareas planificadas de la fase de diseño

DI-011 Revisión del enunciado y concretar el contexto y objetivos del trabajo			
Tipo	Análisis	Duración	8h
Objetivo	En esta tarea se realizará una lectura del enunciado y se intentará extrapolar todas las posibles funcionalidades necesarias a alto nivel. Se deberá resolver cualquier posible duda que surja del estudio del enunciado a través del personal docente. Por otra parte, se intentará concretar cuál es el contexto y justificación del trabajo. Ya que estas nos marcarán la dirección y objetivo de este.		
Dependencia	<Sin dependencias>		
Observaciones	---		
DI-012 Revisión de "estado del arte" de soluciones ERM			
Tipo	Investigación	Duración	6h
Objetivo	Intentar ampliar el contexto de la gestión de riesgos, dado que es un campo sobre el que no dispongo de conocimientos y, si es posible, analizar algunas de las soluciones informáticas (ERM) existentes.		
Dependencia	Esta tarea tiene como interés ampliar el rango de visión sobre el marco de trabajo. Luego debería de ser abordada después de la tarea DI-011 Revisión del enunciado y concretar el contexto y objetivos del trabajo		
Observaciones	Se debe de focalizar el trabajo en intentar completar o mejorar las funcionalidades que se han extrapolado del enunciado. Si surgen nuevas funcionalidades ajenas al documento de definición se deberán de ver si tienen cabida (tiempo, aportan al contexto...)		
DI-013 Recogida y análisis de requisitos			
Tipo	Análisis	Duración	12h
Objetivo	Dada la revisión del enunciado y lo que se haya podido sacar en claro de la investigación del contexto. Extrapolar el conjunto de requisitos funcionales y no funcionales que debe de cumplir nuestro sistema.		
Dependencia	Para su elaboración se deberán de haber completado las dos tareas anteriores. DI-011 Revisión del enunciado y concretar el contexto y objetivos del trabajo y DI-012 Revisión de "estado del arte" de soluciones ERM		
Observaciones	Esta fase se corresponde con la formalización de los requisitos.		
DI-014 Instalación de herramienta de modelado			
Tipo	Diseño	Duración	2h
Objetivo	Determinar la herramienta necesaria para elaborar el modelado. La herramienta sugerida para realizar el modelado (MagicDraw) presenta una limitación por un tema de licencia. Ver si esta puede realizar algún tipo de problema y si es así buscar una alternativa.		

	Dependencia	<Sin dependencias>		
	Observaciones	---		
DI-015		Diseño conceptual		
	Tipo	Diseño	Duración	28h
	Objetivo	Dados los requisitos se elaborará un esquema conceptual de alto nivel que debe de ser independiente de la tecnología final de implementación.		
	Dependencia	Dependerá de la elaboración de la tarea DI-013 Recogida y análisis de requisitos y de que se haya fijado la herramienta en el punto DI-014 Instalación de herramienta de modelado		
	Observaciones	Esta es la primera de las tareas de la fase de diseño y una de las más importantes.		
DI-016		Diseño lógico		
	Tipo	Diseño	Duración	14h
	Objetivo	Se transformará el modelo conceptual obtenido en la fase anterior en un modelo dependiente del tipo de sistema de gestor de base de datos elegido. En nuestro caso un sistema relacional. Como resultado se obtendrá un modelo lógico del producto.		
	Dependencia	Depende del modelo obtenido en DI-015 Diseño conceptual		
	Observaciones	Analizar si el modelo conceptual presenta alguna trampa de diseño y que se encuentra al menos en 3FN.		
DI-017		Diseño físico		
	Tipo	Diseño	Duración	16h
	Objetivo	Adaptar el diseño lógico resultante de la fase anterior a un diseño en <i>SQL Server</i> .		
	Dependencia	Es disponer del diseño lógico elaborado en DI-016 Diseño lógico		
	Observaciones	Tener en cuenta para su elaboración todo lo referido en los riesgos RG-004 Destrucción por error de la base datos de desarrollo elaborar un script de creación que permitan en todo momento una creación de la estructura mediante la ejecución de un solo script.		
DI-018		Generar documento de entrega para la PEC2		
	Tipo	Documentación	Duración	12h
	Objetivo	Generar un documento de entrega que recoja las especificaciones recogidas en el enunciado del proyecto. Verificar que se cumplen los criterios definidos en el documento de validación suministrado. En este documento debe de existir una parte dedicada al seguimiento del proyecto y otra parte que recoja los resultados de las fases de Análisis y Diseño		
	Dependencia	Como en el caso anterior tiene una dependencia directa con la elaboración de cada una de las partes que documenta. Pero puede ser realizado conforme estas partes se van completando.		
	Observaciones	Para su elaboración he reservado la misma cantidad de tiempo que para la elaboración del documento de la entrega anterior. Como para la otra fase se intentará realizar su elaboración de manera paralela a la ejecución de cada una de las partes del proceso. Aunque se reservará una parte del tiempo final a la elaboración del seguimiento.		

1.4.2.3 Implementación y Validación

En esta entrega **PEC3** se engloban las dos partes finales de la metodología de trabajo la parte dedicada a la implementación y la parte final de validación. Como norma general la parte de implementación suele corresponderse con el 40% del esfuerzo en el ciclo de vida de un proyecto.³ Sin embargo, vemos que para nuestro caso esta circunstancia no se cumple y que la mayor parte del peso es

³ Figura 6. José Ramón Rodríguez. La gestión de proyectos. Conceptos básicos. Material de la asignatura Gestión de proyectos. UOC PID_00247944

dedicada al diseño del sistema. Esto es así porque en este tipo de proyecto es uno de los puntos clave es la correcta definición del sistema. Por otra parte, en un proyecto exclusivo de base de datos no tienen cabida muchos otros aspectos a los que se debe de dar una solución en un proyecto de un sistema de información que incrementarían el peso de la parte de implementación (interfaz de usuario, usabilidad, accesibilidad...). En nuestra parte de implementación se limitará a la creación en un sistema físico real según lo concretado en la fase de diseño. Y en la elaboración de todos los elementos de programación necesarios para facilitar su correcto uso y asegurar la consistencia y coherencia de los datos almacenados. La parte final de la metodología, la validación, como norma general es un proceso a través del cual el usuario final asegura que el producto final se corresponde a los requerimientos acordados. Dada la naturaleza de nuestro proyecto esto no es posible. Es por esto por lo que esta validación se realizará a través de una serie de pruebas automatizadas que nos aseguraran el correcto funcionamiento de este. De partida, una de las primeras cosas que tuve claro a la hora de descomponer estas tareas es realizar esta parte de validación de manera que no fuera solo un proceso final al uso para confirmar el correcto funcionamiento del sistema. Sino enfocarlo como una buena práctica a la hora de implementar el desarrollo TDD. De manera que este esfuerzo pudiera ser utilizado en un futuro como base para cualquier refactorización o escalado del sistema original. Aportando un plus de seguridad en la evolución habitual del sistema y fijando también las características que debe de cumplir. Haciéndolo por tanto más robusto y estable. Es por esto por lo que en la parte de validación existirán pruebas de carácter unitario, funcionales e incluso de auditoria para el *datawarehouse*.

El conjunto de tareas para esta fase será:

Tabla 9. Tareas planificadas de la fase de implementación y validación

IV-019 Generar entorno de trabajo			
	Tipo	Implementación	Duración 4h
	Objetivo	Se deberán de concretar la versión a utilizar de SQL Server así como cualquier otra posible herramienta de trabajo.	
	Dependencia	<Sin dependencias>	
	Observaciones	Tener en cuenta lo indicado por el riesgo RG-003 Fallo tecnológico y guardar los posibles instaladores. Si no es posible por tema de espacio poner un enlace directo a la página de descarga.	
IV-020 Estructura. Creación de tablas, repositorio estadístico...			
	Tipo	Implementación	Duración 10h
	Objetivo	En esta tarea se pasará a crear físicamente la estructura de la base de datos fijada en el análisis. Si es necesario se realizará cualquier tipo de ajuste.	
	Dependencia	Depende de la creación del entorno de trabajo IV-019 Generar entorno de trabajo	
	Observaciones	Tal como se recoge en el riesgo RG-004 Destrucción por error de la base datos de desarrollo debe de ser posible recrear toda la estructura de trabajo con un solo script. Por tanto, buscar la manera más ágil de cara a las modificaciones.	
IV-021 Programación. Creación de procedimientos, funciones, disparadores...			
	Tipo	Implementación	Duración 8h
	Objetivo	Se deberán de crear todos los procedimientos, funciones y disparadores necesarios para otorgar funcionalidad a la estructura.	

	Dependencia	Debe de estar implementada la estructura de la base de datos por tanto debe de completarse la tarea IV020 Estructura. Creación de tablas, repositorio estadístico...		
	Observaciones	Este punto podrá ser abordado usando la metodología de desarrollo ágil TDD. Por tanto de manera paralela a la tarea IV-025 Creación de pruebas		
IV-022	Almacén de datos. Consultas			
	Tipo	Implementación	Duración	8h
	Objetivo	Generar las consultas sobre el repositorio estadístico. Se generarán al menos las propuestas del enunciado del proyecto. Determinar si es mejor implementarlas como procedimientos o como vistas para su posterior explotación.		
	Dependencia	Se podrán elaborar de manera independiente aunque sería más útil posponer su desarrollo hasta que se genere el conjunto de datos de pruebas IV-024 Generar conjunto de datos de prueba		
	Observaciones	Esta tarea podrá ser elaborada de manera paralela a la realización de las pruebas de auditoría del sistema de DW que validarán su funcionamiento.		
IV-023	Concretar la manera de implementar las pruebas y alcance			
	Tipo	Investigación	Duración	2h
	Objetivo	He creado esta tarea porque en el punto en el que me encuentro no tengo claro la viabilidad de elaborar las pruebas como un proyecto de prueba o como un conjunto de script. Dependiendo del estado de la planificación y de las comprobaciones de este punto deberé de concretar la manera de realizarlo. Por otra parte, de igual modo no tengo claro el alcance de las pruebas. Tengo claro al menos cubrir pruebas unitarias, funciones y de auditoría del DW. Pero no sé si dará tiempo a auditar la trazabilidad del sistema o abordar test de rendimiento. Se deberá por tanto concretar el alcance antes de su inicio.		
	Dependencia	<Sin dependencias> Esta tarea puede ser avanzada a otras fases del proyecto.		
	Observaciones	---		
IV-024	Generar conjunto de datos de prueba			
	Tipo	Validación	Duración	6h
	Objetivo	Se debe de elaborar un conjunto de datos de pruebas sobre los que realizar las validaciones del sistema. Se debe de asegurar la consistencia del conjunto de datos para que el resto de las pruebas sean correctas.		
	Dependencia	Depende de la creación de la IV020 Estructura. Creación de tablas, repositorio estadístico... y en parte de IV-023 Concretar la manera de implementar las pruebas y alcance		
	Observaciones	Dependiendo de lo determinado en la tarea IV-023 Concretar la manera de implementar las pruebas y alcance será necesario crear un juego de pruebas más o menos extenso. Estudiar la posibilidad de generar un script que genere un conjunto de datos aleatorios (asegurar consistencia).		
IV-025	Creación de pruebas			
	Tipo	Validación	Duración	16h
	Objetivo	Al menos abordar pruebas de carácter unitario, funcionales y de auditoría del DW. Abordar su realización de manera paralela a la parte de programación TDD		
	Dependencia	Depende de la tarea de investigación IV-023 Concretar la manera de implementar las pruebas y alcance y de la creación del conjunto de pruebas IV-024 Generar conjunto de datos de prueba . Lo correcto es abordarse de manera conjunta a la realización de la tarea IV-021 Programación. Creación de procedimientos, funciones, disparadores...		
	Observaciones	Su alcance será fijado en la tarea de investigación y dependerá del tiempo disponible.		
IV-026	Generar documento de entrega para la PEC3			
	Tipo	Documentación	Duración	12h
	Objetivo	Generar un documento de entrega que recoja las especificaciones recogidas en el enunciado del proyecto. Verificar que se cumplen los criterios definidos en el		

	documento de validación suministrado. En este documento debe de existir una parte dedicada al seguimiento del proyecto y otra parte que recoja los resultados de las fases de Implementación y validación
Dependencia	Como en los casos anteriores tiene una dependencia directa con la elaboración de cada una de las partes que documenta. Pero puede ser realizado conforme estas partes se van completando.
Observaciones	Para su elaboración he reservado la misma cantidad de tiempo que para la elaboración del documento de la entrega anterior. Como para la otra fase se intentará realizar su elaboración de manera paralela a la ejecución de cada una de las partes del proceso. Aunque se reservará una parte del tiempo final a la elaboración del seguimiento

1.4.2.4 Documentación

Esta fase tiene una correspondencia con la entrega final según lo recogido en la temporización de la asignatura. Esta fase estará centrada en la elaboración de los entregables finales del proyecto. Durante ella se deberán de concretar los medios a través de los que realizar la presentación y si es necesario el uso de alguna herramienta adicional para mejorar la correcta difusión del proyecto realizado. Una vez fijados se deberá elaborar una memoria y comprobar que cumple los criterios de valoración definidos. En gran parte esta documentación se construirá a partir de la documentación elaborada en cada una de las entregas. Por otra parte, se deberá de generar una presentación que resuma y transmita a ser posible de manera amena el trabajo elaborado.

Para completar esta fase se deberán realizar las tareas:

Tabla 10. Tareas planificadas de la fase final de documentación

EF-027	Revisar recursos y concretar herramientas para su elaboración		
Tipo	Investigación	Duración	4h
Objetivo	Comprobar los recursos disponibles y determinar cuáles serán los mejores para transmitir la información a través de la presentación.		
Dependencia	<Sin dependencias>		
Observaciones	---		
EF-028	Generar memoria		
Tipo	Generar entregables	Duración	41h
Objetivo	Se deberá elaborar un documento de memoria que recoja todo el trabajo realizado durante la elaboración del proyecto. Dicho documento debe cumplir todos los requerimientos indicados en el documento de validación del proyecto.		
Dependencia	De partida, aunque podría no depender de ninguna tarea, será mejor abordarlo una vez se complete la tarea EF-027 Revisar recursos y concretar herramientas para su elaboración . Ya que después de la investigación puede surgir alguna idea para mejorar la memoria (uso imágenes, infografía...)		
Observaciones	---		
EF-029	Crear presentación		
Tipo	Generar entregables	Duración	30h
Objetivo	Se deberá de elaborar una presentación que difunda el trabajo realizado durante la elaboración del proyecto y el producto obtenido		
Dependencia	Depende de la elaboración de la tarea EF-027 Revisar recursos y concretar herramientas para su elaboración		
Observaciones	---		

1.4.2.5 Uso de la dedicación y tareas descartadas

A modo de resumen, en este punto se recoge como se distribuirá la dedicación entre los tipos de tareas estimadas. De este modo podremos conocer que áreas de trabajo requerirán un mayor esfuerzo.

Tabla 11. Distribución de dedicación por áreas de trabajo

Tipo	Estimación	%
Análisis	23	8%
Diseño	60	20%
Documentación	36	12%
Generar entregables	71	23%
Implementación	44	14%
Investigación	21	7%
Organización de trabajo	4	1%
Planificación Tareas	23	8%
Validación	22	7%
Total general	304	

Vemos que la mayor parte del tiempo se dedicará a generar los entregables del proyecto. Esto es así dado que he respetado la propia temporización indicada por la asignatura y el último bloque está centrado exclusivamente en la difusión del proyecto. Las siguientes tareas que se estiman más importantes serán el diseño y la implementación del sistema. Lo cual es lo lógico en un proyecto de base de datos. Quedando los otros tipos de tareas con una menor dedicación. Como es el caso del análisis, lo cual es normal dado que la definición del problema está fijada por el propio enunciado. Por último, hay que indicar que no existen muchas tareas propias de investigación dado que me he centrado en las tareas que creo que son necesarias para cubrir la propia elaboración del núcleo del proyecto. Ya que por cuestiones de tiempo no creo que, en inicio, exista una posibilidad de una dedicación mayor a esta tipología de tareas. Conforme se desarrolle el proyecto puede ser que esto cambie y pueda dedicar algo de tiempo a alguna de las tareas descartadas en las cuales tiene un mayor peso la investigación. Solo a modo de anotación se recogen una serie de posibles tareas a desarrollar que han quedado fuera de la planificación.

Tabla 12. Tareas descartadas

Título	Descripción
Estado del sistema	Determinar una serie de KPIs centrados en el mantenimiento del sistema y elaborar un informe que sirvan como indicador del estado del sistema. (Tamaño, Cantidad de procesos ejecutados semanalmente...)
Uso de contenedores Docker	En lugar de realizar las instalaciones en local realizarlas en una imagen que contenga un sistema. Esto posibilitaría la realización de un sistema de integración y la distribución continuas (CI/CD). Y por otro lado, facilitaría la futura automatización de las pruebas.
Automatización ejecución de pruebas	Tendría cierta vinculación con de la anterior. Se correspondería con realizar las estructuras necesarias (<i>Pipeline</i>) para ejecutar pruebas en un entorno acotado de manera automatizada.
Proyecto pruebas	De partida sé que existe la posibilidad de generar un proyecto de Visual Studio donde es posible incorporar la elaboración de pruebas al igual que

	en un proyecto de código ⁴ . Esto facilitaría la ejecución de las pruebas y el análisis de resultados.
Calidad del código	Sé que existen herramientas para evaluar la calidad del código. Pero desconozco si existe un equivalente a nivel de SQL. Sería interesante su uso sobre todo si permite definir normas de estilo o fijar formatos (parámetro de salida RSP).
Securización del sistema	Definición de políticas específicas de seguridad, sistemas de encriptación de datos (TDE)...
Control de desastres	Implementación de una política de recuperación del sistema (<i>Backup</i>).
Alta disponibilidad	Comprobar las posibilidades de escalado del motor de base de datos utilizado.
Revisión de rendimiento	Esta tarea estudiaría como realizar pruebas de estrés en el sistema de manera automatizadas. La he recogido dado que es un área de la que no tengo conocimiento y me gustaría explorar las posibilidades que presenta.

1.4.2 Temporización del proyecto

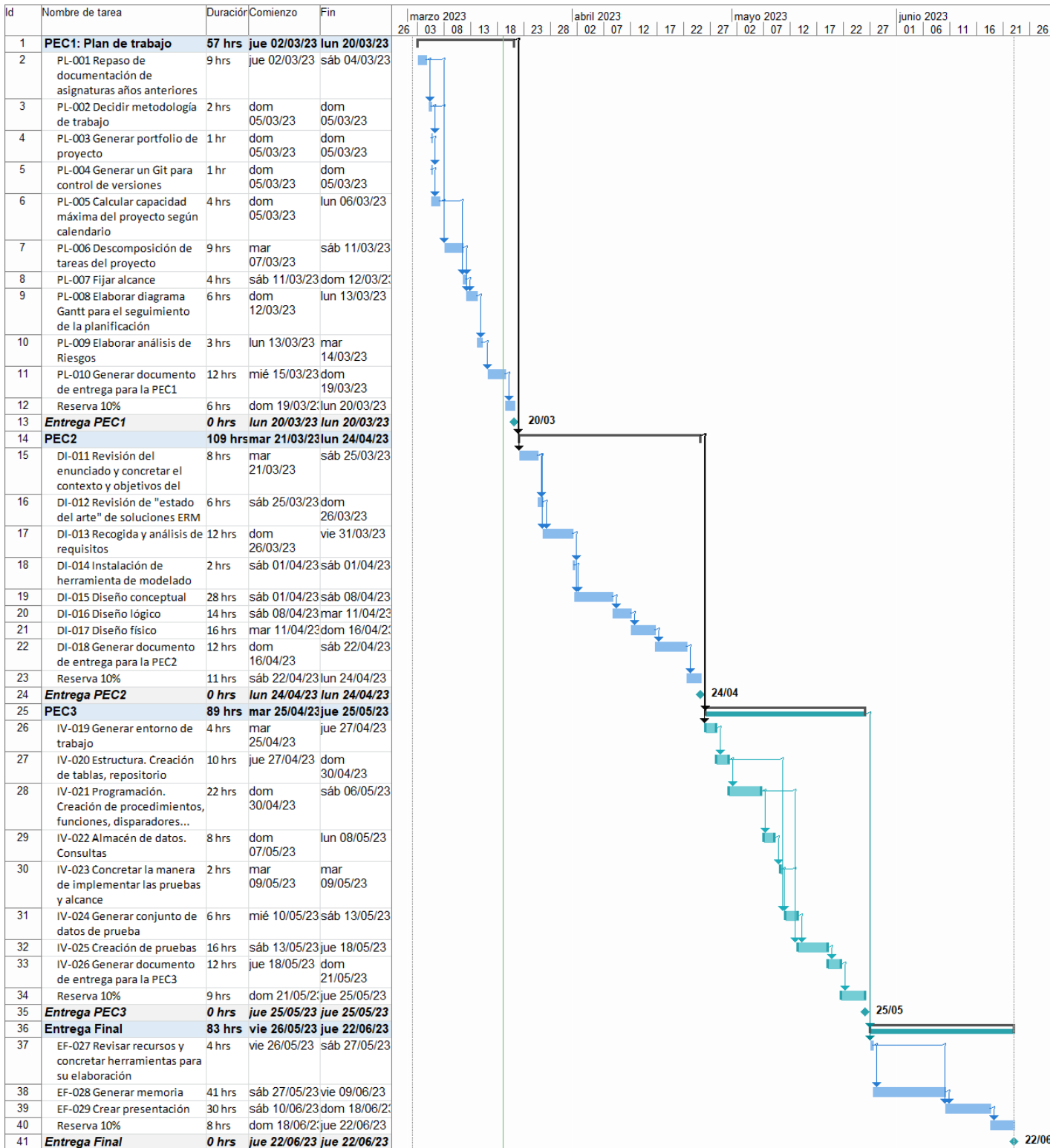
Para plasmar la temporización del proyecto he decidido utilizar como herramienta el Microsoft Project debido a que el paquete viene incluido en mi equipo de desarrollo y es la herramienta a la que estoy más habituado. Por este motivo no he podido localizar el instalador. Sin embargo, he anotado la versión en el portfolio de trabajo. Para elaborar el diagrama Gantt he dado de alta mi dedicación ya definida en el alcance del proyecto como un calendario personalizado y he incluido la dedicación especial para los días festivos que ocurren durante la temporización. Para facilitar el seguimiento he decidido no solapar el tratamiento de tareas. Aunque de partida durante la realización asumo que solaparé al menos la dedicación de las tareas de documentación con el resto de las realizas en los distintos periodos. Ya que después de realizar pruebas he comprobado que al fraccionar la capacidad del único recurso disponible y realizar el solapamiento se perdía un poco la valoración dado que esta se escalaba también. Esto sumado a que en realidad solo hay un recurso disponible y a que el uso principal que daré al diagrama es realizar un seguimiento para anticipar el posible descuadre. He preferido sacrificar el realismo para facilitar el seguimiento.

Para su realización he transcrito las tareas, valoraciones y dependencias fijadas durante la descomposición de tareas. He optado por programar de manera secuencial aquellas tareas que no tienen una dependencia definida. Además he incluido como una tarea adicional el 10% de dedicación reservada para imprevistos. Manteniendo de esta manera, en inicio, la realización de todas las tareas dentro de cada una de las entregas.

Dados los criterios indicados se obtiene la siguiente planificación.

⁴ Tutorial: Crear y ejecutar una prueba unitaria de SQL Server [en línea]. Disponible en: <https://learn.microsoft.com/es-es/sql/ssdt/walkthrough-creating-and-running-a-sql-server-unit-test?view=sql-server-ver16>

Figura 2. Diagrama Gantt con la planificación del proyecto



1.4.3 Gestión de Riesgos

Para empezar, es necesario concretar que un riesgo es todo lo que afecte cuantitativamente a uno de los parámetros del proyecto, como: tiempo, coste, alcance o calidad. Si afecta de manera positiva se conoce como oportunidad y si afecta de manera negativa se le denomina problemática. La existencia de estos riesgos aumenta la incertidumbre a la hora de la realización de un proyecto. El objetivo de la gestión de riesgos es intentar aumentar la probabilidad de que ocurran los acontecimientos positivos y disminuir la posibilidad de que sobrevengan los negativos. O en caso de que ocurran, reducir su impacto en el proyecto. Para el correcto tratamiento de los riesgos es necesario realizar un análisis para determinar cuales se pueden presentar durante el semestre y posteriormente elaborar un plan de contingencia. Como consecuencia de su elaboración pueden llegar a surgir una serie de medidas que será necesario implementar durante el proyecto para minimizar las problemáticas en caso de producirse. Para la correcta gestión de riesgos además será necesario durante la ejecución del proyecto realizar un seguimiento de los riesgos encontrados y comprobar su evolución. Así como de estar pendientes de detectar aquellos riesgos que en inicio no fueron identificados.

La lista de riesgos detectada es la siguiente.

Tabla 13. Lista de riesgos que pueden afectar al proyecto

RG-001		Fallo en estimaciones				
Área	Planificación	Impacto	MEDIO	Probabilidad	MEDIA	
Descripción	El proyecto incluye actividades que nunca han sido realizadas lo que puede ocasionar fallos en las estimaciones.					
Acción de prevención	Realizar un seguimiento exhaustivo del cronograma y asegurar el cumplimiento de los hitos acordados. Disponer de un margen de gestión a la hora de realizar la planificación para absorber las ocurrencias de carácter leve.					
Contramedida	Utilizar colchón de seguridad. Incrementar la capacidad de trabajo ampliando el horario habitual. Solicitar más tiempo al profesor. O incluso en circunstancias especiales intentar incrementar capacidad de trabajo solicitando un día de vacaciones si el trabajo lo permite.					
RG-002		No cumplimiento de alguna de las metodologías de desarrollo de BD				
Área	Calidad	Impacto	MEDIO	Probabilidad	MEDIA	
Descripción	A pesar de tener experiencia en el medio, esta se encuentra focalizada en el uso de un gestor de base de datos final. Esto unido a que tengo convalidadas las asignaturas: Uso de Bases de datos y Diseño de Bases. Podría ocasionar la no correcta implementación de alguna de las metodologías de diseño de BD.					
Acción de prevención	Revisar una revisión de los recursos disponibles de estas dos asignaturas para asegurar el correcto desarrollo del proyecto					
Contramedida	---					
RG-003		Fallo tecnológico				
Área	Planificación	Impacto	ALTO	Probabilidad	BAJA	
Descripción	Problemas con el equipo que utilizo para la elaboración del proyecto.					
Acción de prevención	Realizar un portafolio para facilitar la organización del proyecto y que este esté sincronizado en la nube. Tener preparado un equipo de emergencia que poder utilizar en caso de fallo. Mantener este ordenador sincronizado con la nube.					

		Mantener un control de los programas instalados y tener localizada la versión utilizada durante el desarrollo. De manera que sea fácil replicar el entorno de trabajo.				
	Contramedida	Usar el equipo de respaldo				
RG-004	Destrucción por error de la base datos de desarrollo					
	Área	Planificación	Impacto	MEDIO	Probabilidad	BAJA
	Descripción	Aunque es poco probable siempre existe la posibilidad de incapacitar o destruir la base de trabajo durante el desarrollo. Dado que de partida no contemplo la necesidad de implementar ningún proceso de copias de seguridad.				
	Acción de prevención	La mejor medida es implementar una correcta metodología de trabajo desde el inicio. Se elaborará una estructura que en todo momento permitirá la creación de la base de datos pasando un solo script y a la vez esto facilitará la entrega final. De manera que sea fácil recrear un nuevo sistema de desarrollo. En todo momento trabajar sobre un archivo en local donde ir guardando los cambios en lugar de sobre una sesión y posteriormente almacenar el trabajo.				
	Contramedida	---				
RG-005	Destrucción parcial o perdida de trabajo en desarrollo					
	Área	Planificación	Impacto	BAJA	Probabilidad	BAJA
	Descripción	Posible pérdida de parte del trabajo realizado en la base de datos. O que surja la necesidad de retomar una línea de acción descartada.				
	Acción de prevención	Implementar un sistema de control de versiones en la parte utilizada para el desarrollo GIT. Trabajar cambios sobre ficheros ubicados en esta parte y realizar guardados de manera regular				
	Contramedida	Recuperar el trabajo del fichero o revisar histórico de versiones				
RG-006	Problemas de conectividad					
	Área	Planificación	Impacto	ALTO	Probabilidad	BAJA
	Descripción	Puede afectar tanto durante la elaboración del trabajo como a la hora de realizar las entregas.				
	Acción de prevención	Disponer de una línea 4G adicional al utilizado habitualmente.				
	Contramedida	Realizar backup físico en disco físico en el caso de que no se pueda realizar sincronización. En el caso de afectar a una entrega entrar en contacto con el profesor para informar del incidente.				
RG-007	Baja por incapacidad de carácter transitorio					
	Área	Planificación	Impacto	ALTO	Probabilidad	BAJA
	Descripción	Siempre existe la posibilidad de que surja algún problema de carácter médico durante la elaboración del trabajo.				
	Acción de prevención	---				
	Contramedida	Si el impacto es pequeño las mismas que las empleadas en el riesgo RG-001 Fallo en estimaciones . Si no se puede compensar por estos medios avisar al profesor de la problemática.				
RG-008	Aumento de la carga de trabajo en el ámbito laboral					
	Área	Planificación	Impacto	ALTO	Probabilidad	BAJA
	Descripción	Existe la posibilidad de que venga un pico de trabajo en mi ámbito laboral que ocasione una disminución de mi capacidad de trabajo.				
	Acción de prevención	---				
	Contramedida	Mismas que en el caso anterior. Si el impacto es pequeño las mismas que las empleadas en el riesgo RG-001 Fallo en estimaciones . Si no se puede absorber por estos medios avisar al profesor de la problemática.				

1.5 Breve resumen de productos obtenidos

De partida con la elaboración del proyecto se esperan conseguir una serie de resultados estos se materializan en forma de entregables. Este proyecto tiene dos vertientes por un lado se corresponde con *un trabajo fin de grado* centrado en Bases de datos y por otro se corresponde con *la implementación de la base de datos para una aplicación de control y gestión de riesgos empresariales para "la empresa"*. Sin embargo, he obviado la relación de entregables con respecto a este segundo aspecto. Dado que la mayoría de entregables de carácter interno para la empresa, como: Planificación temporal del proyecto, Plan de gestión de riesgos..., están incluidos en la memoria del trabajo. Con respecto a los entregables más al uso, es decir, los externos: script de implementación del sistema, scripts de prueba..., se corresponderán también con una de las partes de los entregables propios del trabajo fin de grado. Por tanto, solo indicaré la relación de entregables que se corresponde al trabajo fin de grado. Veámoslos:

- **Memoria del trabajo:** Se corresponde con el presente documento. En él se recoge todo el trabajo realizado para la ejecución del proyecto, así como las decisiones tomadas durante su elaboración. Este documento sigue los criterios y aspectos formales fijados por la asignatura.
- **Producto:** diferenciaré dos bloques de *scripts*. Un primer conjunto de bloques destinado a la creación del producto. Y un segundo bloque encargado de testear la estructura creada.
- **Presentación:** este punto se corresponde con una presentación virtual en formato de video, que sintetizará y difundirá el trabajo realizado.
- **Autoinforme:** este entregable se corresponde con uno de los requerimientos de la asignatura y contiene una autoevaluación por parte del alumno.

1.6 Breve descripción de los otros capítulos de la memoria

En el primer punto de la memoria se ha fijado el contexto, justificación del trabajo y la planificación del proyecto. Se ha definido como se organizará el trabajo, que trabajo se abordará y cuando se deberá de ser realizada cada una de sus partes. Como resultado de esta fase han surgido una serie de herramientas de control que nos ayudarán a gestionar la correcta realización del proyecto (análisis de Riesgos y planificación).

Los siguientes apartados se centrarán en la elaboración del proyecto. Para ello he decidido hacer corresponder cada uno de los apartados siguientes de la memoria con las consiguientes fases del proyecto. El primero de ellos se encargará de la fase transversal de seguimiento y control. Por tanto, iré completándolo de manera secuencial conforme se vaya completando cada una de las entregas del proyecto. El siguiente capítulo se encargará de la definición del problema y se corresponderá con la fase de análisis. A continuación desarrollaré la fase de diseño. Estructurándola en tres partes que son las propias de un proyecto de base de datos: diseño conceptual, diseño lógico y diseño físico.

Posteriormente pasará a la fase de implementación. Donde se creará el producto propiamente dicho. Y por último a la de validación, donde se verificará que el producto cumple con lo definido.

Como ya se comentó anteriormente existe una cierta realimentación en el proyecto. Por tanto, se producirán una serie de modificaciones en lo definido conforme se vayan concretando la elaboración. He decidido anotar estas modificaciones en cada una de las fases. Ya que es el lugar donde tienen más sentido. Pero diferenciándolas de lo realizado durante la planificación. De este modo, se podrá reflejar la evolución del trabajo y poner en relevancia el propio proceso de desarrollo, en lugar de incorporarlas directamente y plasmar solo la solución final.

2. Seguimiento y control

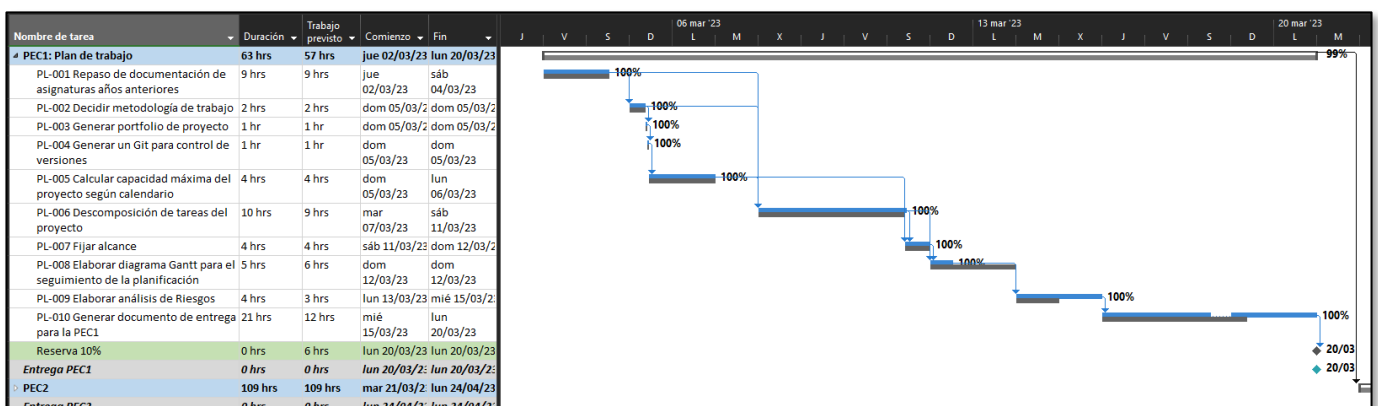
Este capítulo, tal como ya se ha comentado, se completará al inicio de cada una de las fases. Y se corresponderá con un resumen de lo ocurrido durante la elaboración de cada una de las fases del proyecto con respecto a la planificación original. En él se recogerá cada una de las rectificaciones o las modificaciones que se han tendido que realizar sobre las temporizaciones. Así como las posibles nuevas tareas o riesgos que surjan.

2.1 Seguimiento de la fase de Planificación

Como primera tarea de la fase de análisis y diseño se realiza un resumen de lo ocurrido durante la temporización de la fase anterior de planificación. Para ello he modificado la vista de trabajo del diagrama utilizado a Gantt de seguimiento y he fijado la planificación inicial definida como línea base del proyecto. Para facilitar su interpretación he agregado además una columna que indica el tiempo previsto, que se podrá cotejar con la duración real de las tareas.

Como resultado final tenemos el siguiente Gantt de seguimiento.

Figura 3. Diagrama Gantt de seguimiento de la primera fase de planificación



De partida, vemos que en inicio se fue cumpliendo lo recogido en la planificación. Se produjo un descuadre leve de una hora durante la realización de la tarea [PL-006](#) Descomposición de tareas del proyecto que pudo ser compensado en la [PL-008](#) Elaborar diagrama Gantt. La dedicación para la tarea [PL-009](#) Elaborar análisis de Riesgos también fue una hora superior de lo esperado lo que obligo a

compensar este tiempo con el tiempo de reserva planificado. Sin embargo, vemos que para la realización de la última de las tareas [PL-010](#) Generar documento de entrega para la PEC1 fueron necesarias 21 horas en lugar de las 12 horas planificadas. Esto casi duplica el coste inicial que ya era suficientemente alto. Lo que ocasionó que se consumieran el colchón de seguridad definido en el riesgo [RG-001](#) Fallo en estimaciones. Y además se tuviera que implementar otra de las contramedidas definidas para este riesgo. Incrementándose la capacidad de trabajo en los días anteriores a la entrega para poder compensar este tiempo. Por tanto, tal como se comprueba en el diagrama de seguimiento la dedicación total prevista en esta fase se ha tenido que incrementar en 6 horas. Este fallo de estimaciones en la tarea [PL-010](#) lo achacado en parte a 2 motivos. Por un lado, el tiempo planificado para el desarrollo de esta tarea era insuficiente. Por otro lado, deje la redacción del texto para la fase final en lugar de ir realizándola conforme iba realizando las tareas. En cuando al segundo de los motivos se anota como punto de mejora para la siguiente fase de desarrollo. Se deberán de ir avanzando los textos conforme se vayan realizando cada una de las tareas. En lo que respecta al primero. Por ahora he preferido mantener la planificación inicial y no incrementar la dedicación de las tareas de documentación. Ya que considero que esta primera fase de elaboración ha sido un poco más caótica de lo esperado y no se dispuso de una planificación a la que realizar un seguimiento desde el inicio.

Por último, durante la realización de esta fase se han completado la creación de un porfolio de trabajo y un control de versiones. Por lo que para el correcto seguimiento del trabajo por el profesorado incluyo enlaces a los mismos.

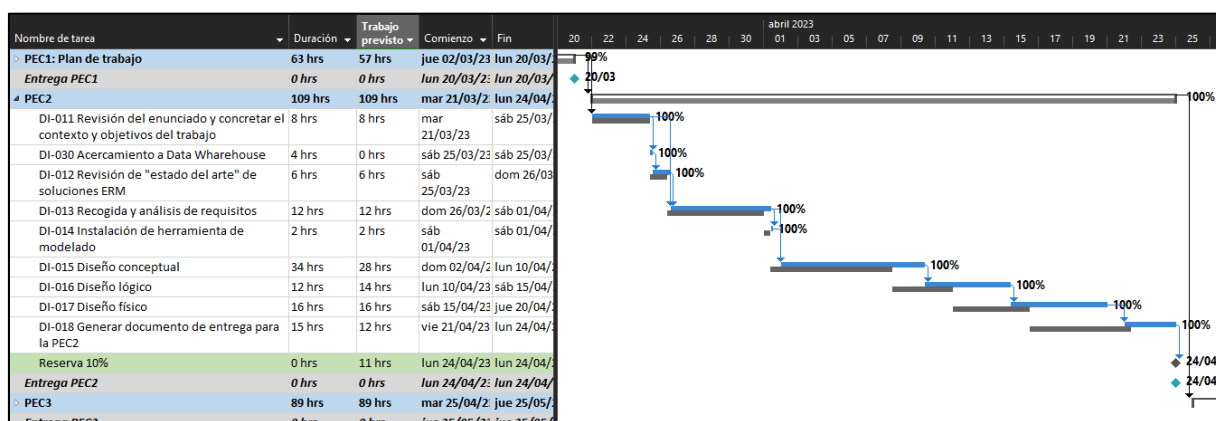
- **Porfolio:** <https://drive.google.com/drive/folders/1gOFaqtAd73dNZLTgmaE9P8og0UU0-a3P?usp=sharing>
- **Control de versiones:** <git@gitlab.com:mcarmonag/tgb-bd.git>

En el caso del control de versiones si el profesorado lo requiere se le puede dar acceso, aunque su contenido se encuentra también incluido en el porfolio de trabajo.

2.2 Seguimiento de la fase de análisis y diseño

La primera tarea de la fase de implementación y validación consiste en plasmar en el documento lo ocurrido durante la fase anterior de análisis y diseño. El diagrama de Gantt de seguimiento para esta fase se corresponde con:

Figura 4. Diagrama Gantt de seguimiento de la fase de análisis y diseño



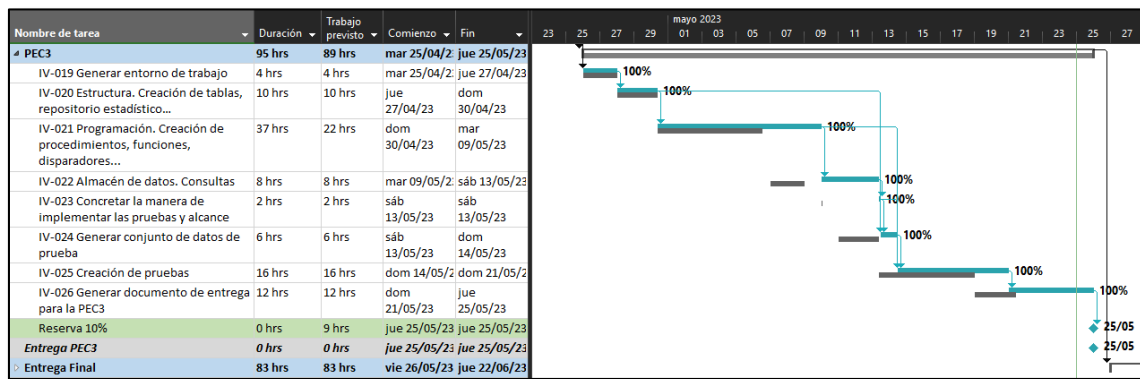
Durante la realización del primero de las tareas [DI-011](#) Revisión del enunciado y concretar el contexto y objetivos del trabajo. Me vi en la necesidad de crear y planificar una nueva tarea [DI-030](#) Acercamiento a Data Warehouse. Esto fue debido a que después del estudio del enunciado determine que no tenía más allá de un conocimiento al uso de los almacenes de datos. Y que para abordar de manera correcta su realización debería de documentarme más sobre ellos. Planifique esta tarea en 4 horas que debían de salir de la reserva de tiempo de esta fase. Para las tres siguientes tareas podemos comprobar que no se produjo ninguna variación con lo planificado. Sin embargo, para la realización de la tarea [DI-015](#) Diseño conceptual tuve que emplear 6 horas más de lo planificado esto fue debido a que realice varios modélos sucesivos que no me convencieron hasta que me decante por el que finalmente expongo. Parte del exceso pudo ser compensado durante la realización de la tarea [DI-016](#) Diseño lógico la cual fue mucho más mecánica de lo que en principio había previsto. Sin embargo, al final el resto de las horas disponibles en la reserva fue consumido durante la elaboración de la tarea final de [DI-018](#) Generar documento de entrega para la PEC2. Como en la planificación anterior he comprobado que el tiempo de seguridad era consumido por plasmar lo realizado en el documento. En este caso no ha sido necesario activar ninguna de las contramedidas adicionales definidas por el riesgo [RG-001](#) Fallo en estimaciones. Por lo que no realizaré ninguna modificación en la planificación de la siguiente fase. Ya que los descuadres por ahora son absorbidos por el colchón de seguridad.

2.3 Seguimiento de la fase de implementación y validación

En este caso he completado al final de la fase de implementación y validación el seguimiento de la fase. Para ello como en los casos anteriores he actualizado el diagrama de Gantt de seguimiento con lo ocurrido durante la realización de la fase. Podemos comprobar que la tarea [IV-021](#) Programación. Creación de procedimientos, funciones, disparadores... no solo ha consumido el tiempo de la reserva de 9 horas sino que además ha sido necesario incrementar el tiempo planificado de desarrollo en 6 horas (según lo definido en el [RG-001](#) Fallo en estimaciones). Esto ha sido debido a que la cantidad de procedimientos a desarrollar ha sido muy elevada. Incluso como se indica en la fase al final ha sido necesario acotar su cantidad a 42 + 1 procedimiento de cierre en lugar de los 60

necesarios. Con respecto, la parte de tiempo reservada para la implementación de las pruebas [IV-025](#) Creación de pruebas es necesario indicar que no se ha incrementado. Esto ha sido debido a que se ha automatizado parte de su ejecución. De modo que el tiempo para su realización se ha compensado aunque el número de pruebas finales hayan sido 208. Por último, en este caso el tiempo necesario para generar el documento final [IV-026](#) Generar documento de entrega para la PEC3 vemos que tampoco se ha incrementado. Esto ha sido debido a que conforme iba realizando los procedimientos iba documentándolos. Por tanto, el posible trabajo extra ocasiona por el número final de procedimientos tampoco se ha visto reflejado en esta parte y todo el incremento ha quedado asociado a la única tarea que se dispara. El diagrama final es:

Figura 5. Gantt de seguimiento de la fase de implementación y validación



3. Recogida y análisis de requisitos

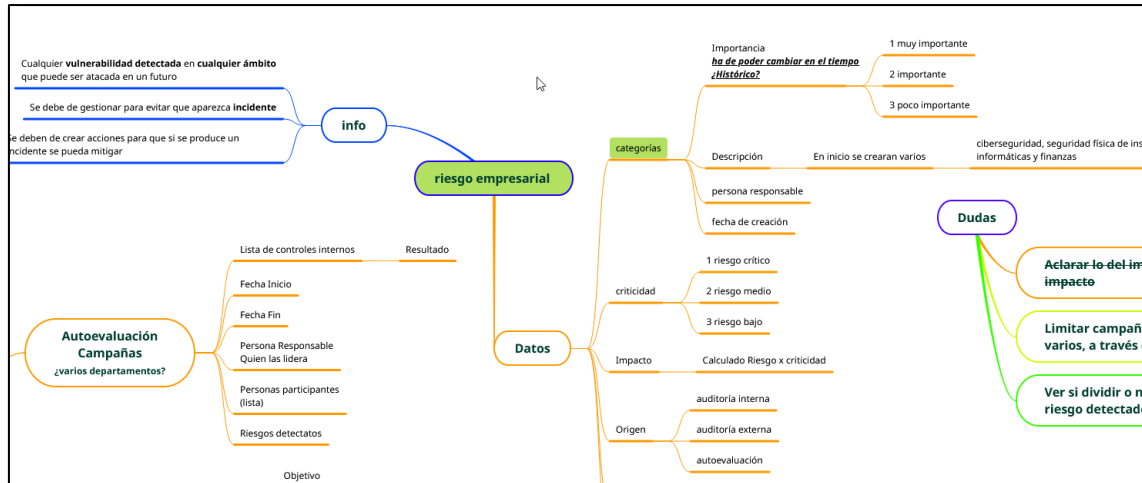
Esta parte del proceso tiene una correspondencia con una de las fases habituales en el diseño de una base de datos recogida y análisis de requisitos. En esta fase normalmente se realizan una serie de tareas centradas en concretar requerimientos que debe de cumplir el sistema (recogida de requisitos, estructuración y refinamiento de los requisitos y formalización de los requisitos). En nuestro caso gran parte de este proceso no es necesario. Dado que ya en la parte del enunciado del problema donde se indica "Descripción del trabajo a realizar" se nos da una descripción de los requisitos. Sin embargo, esta descripción en formato texto no tiene un carácter formal por lo que deberemos de realizar un estudio del enunciado del problema para extraer las necesidades, posteriormente deberemos de complementar esta información a través de preguntas al profesorado y realizando un análisis del estado del arte de la materia del trabajo. Con toda esta información como base podremos extrapolar los requisitos que nuestro sistema debe de cubrir y recogerlos de un modo más formal.

3.1 Estudio del enunciado del problema

Este punto se corresponde con la tarea planificada [DI-011](#) Revisión del enunciado y concretar el contexto y objetivos del trabajo. Esta tarea como cometido la realización de una primera revisión del enunciado. Intentado organizar desde el

inicio la información que recoge el documento. Aunque el interés principal era concretar el contexto y los objetivos del trabajo. Revolviendo cualquier posible duda a través del personal docente en una fase temprana del proyecto, para evitar problemas durante el desarrollo. De partida, para un primer acercamiento al problema se elaboró un mapa mental para intentar recoger y estructurar la información suministrada. Este es accesible a través del siguiente enlace.

Figura 6. Captura de mana mental inicial sobre el enunciado del problema



<https://www.mindomo.com/mindmap/9f347f2795cc4f9d9c002ae2969dd964>

Durante su elaboración ya empezaron a aparecer las principales entidades que deberían de ser modeladas en la fase de diseño. Como resultado del estudio realizado en esta tarea. Me vi la necesidad de crear y planificar una nueva tarea. Dado que uno de los puntos principales que indicaba el enunciado que se debía resolver es la elaboración de un almacén de datos (DW) para satisfacer una serie de consultas. Esta área era un tema sobre el que no tenía conocimiento muy formal y que de partida no sabía cuál debería de ser el método correcto para el modelado de su estructura.

Tabla 14. Nueva tarea acercamiento a Data Warehouse

DI-030	Acercamiento a Data Warehouse		
Tipo	Investigación	Duración	4h
Objetivo	En esta tarea se realizará un primer estudio sobre el modelado y uso de los almacenes de datos (DW). Centrado en que técnicas serán necesarias para modelar de manera correcta lo recogido en el problema.		
Dependencia	<Sin dependencias>		
Observaciones	--		

Realmente el tiempo dedicado a esta tarea solo sirvió más que para generar nuevas dudas sobre lo recogido en el enunciado del problema. Dado que en él se incluían una serie de restricciones (resultados actualizados en todo momento, operaciones en con un solo SELECT...) que no casaban con la metodología habitual de un DW al uso. En este tipo de sistemas las operaciones tienen un cierto tiempo de retraso dado que la mayoría de las veces el almacén de datos reside en un servidor distinto. Existiendo toda una parte de la metodología

centrada en la obtención y transformación de los datos (ETL). Por otra parte, los datos se organizan en una estructura bastante diferenciada a través de tablas de datos y de dimensiones. Sin embargo, esta estructura imposibilitaba el uso de operaciones en un solo SELECT como se recoge en el enunciado. También, una de las propiedades principales de un DW era el hecho de que los datos almacenados deben de ser “Non-volatile” y por tanto de solo lectura. Esto chocaba con el hecho de que al tener que realizar una sola consulta para obtener los datos. Dado que si es una sola consulta, debe de ser realizada en una sola ubicación. Lo que obliga a realizar actualizaciones sobre los datos existentes para mantener en todo momento un valor válido. Para resolver estas dudas no me quedo más remedio que realizar una consulta al personal docente. La respuesta me indico que el interés de lo expuesto en el trabajo no es la realización de un sistema de DW al uso y que los procesos de ETL se encuentran más allá del ámbito de este proyecto. El interés sobre todo era intentar modelar como deberían de resolverse estos procesos con estas limitaciones que estaban enfocados a brindar una información a negocio. Por tanto, el proceso se debería centrar más en estructurar y mantener este repositorio estadístico. Por tanto, lo solicitado estaría más enfocando en realizar un proceso de *Data Warehousing* (DWH) para extraer los datos con estas condiciones, que como el modelado de un almacén de datos estándar (DW).

3.2 Estudio del contexto

El principal objetivo de este punto era obtener una base de conocimientos sobre un campo de trabajo que me era desconocido. De manera que este conocimiento posibilitara el correcto modelado de la solución del problema. De partida, encontré mucha información de carácter comercial. Sobre todo enfocada a vender el uso de determinados productos ERM y sus posibles beneficios empresariales, más que a la definición de propio proceso de gestión de riesgos. ([Oracle Enterprise Resource Planning \(ERP\)](#)⁵, [SAP Risk Management](#)⁶, [Pelican](#)⁷...) Sin embargo, después de una búsqueda más focalizada acabé encontrando lo que se conoce como [Modelo Coso](#)⁸ (*Committee of Sponsoring Organizations of the Tradeway Commission*) que es un estándar metodológico creado por una serie de organismos privados enfocado en la gestión del riesgo empresarial. Que me dio un conocimiento más amplio de todo el proceso.

A raíz del estudio del contexto llegué a la conclusión que lo habitual es definir una estrategia que fije los límites de riesgo asumibles para la empresa. Para ello se definen a priori una serie de riesgos que pueden afectar a la empresa. Sobre estos se crean acciones encaminadas, según la probabilidad de que ocurra el riesgo, a evitarlos o mitigarlos. Una vez establecidas estas acciones sobre ellas se crean controles para asegurar su correcta implementación. Sin embargo, esto chocaba de lo que se desprende del enunciado. En él, toda la gestión del riesgo

⁵ Oracle Enterprise Resource Planning (ERP) [en línea]. Disponible en:

<https://www.oracle.com/es/erp/risk-management/what-is-enterprise-risk-management/>

⁶ SAP Risk Management [en línea]. Disponible en: <https://winterhawk.com/sap-grc/risk-management/>

⁷ Pelican [en línea]. Disponible en: <https://www.vosesoftware.com/products/pelican/>

⁸ ¿QUÉ ES EL MODELO COSO? [en línea]. Disponible en: <https://www.globalsuitesolutions.com/es/que-es-modelo-coso/>

estaría centrada en un enfoque de carácter reactivo. No se contempla ningún origen de los riesgos que no sea a través de revisiones (internas, externas o autoevaluaciones). Por tanto, solo se trabajarán los riesgos a posteriori una vez que hubieran sido detectados. Definiendo sobre ellos acciones para mantenerlos controlados. Me pareció interesante dotar a la solución final de un enfoque más realista y a raíz de comentar esta circunstancia con el profesor y con su beneplácito he decidido ampliar un poco la funcionalidad que se recoge en el texto. Creando un nuevo origen de datos para los riesgos. El del departamento de gestión de riesgos. E introduciendo una serie de modificaciones en el diseño que posibilitarán la existencia de los riesgos independiente de los métodos de detección. Por otra parte, se ha ampliado un poco el contexto de estos riesgos añadiendo una serie de entidades para recoger información sobre ellos (Causas, Consecuencias y ocurrencias). Sin embargo, el sentido de estas modificaciones ha sido ampliar la funcionalidad recogida en el documento. Por lo que no he desarrollado un sistema que no permita recoger lo que ya está incluido en el documento de definición, y he evitado realizar implementaciones que irían en contra de lo recogido en el enunciado del problema. Más allá, de la indicada creación de un nuevo origen para los riesgos.

3.3 Elaboración de requerimientos

Como resultado del estudio del estudio del enunciado y teniendo en cuenta la información obtenida de la revisión del contexto de la gestión de riesgos podemos extraer una serie de requerimientos que nuestro sistema debe de cumplir. He agrupado estos requerimientos según lo indicado en la asignatura de Ingeniería de requisitos. Para representarlos he optado por utilizar un enfoque ágil⁹. Dado que el uso de otras especificaciones más exhaustivas (casos de uso, escenarios...) alargaría mucho la memoria del trabajo y focalizaría el proceso en un punto que no es la materia principal del presente trabajo.

Por un lado, tendremos una serie de requisitos que serán centrados en el producto. Que podrán ser requisitos funcionales o no funcionales. Y por otro lado, tendremos una serie de requisitos centrados en el proceso. Veamos cuales serán y concretemos estas categorías.

3.3.1 Requisitos de producto

Estos requisitos definen aquellas necesidades o restricciones que el producto debe de resolver.

3.3.1.1 Requisitos funcionales.

Estos requisitos deben de recoger las funcionalidades que el sistema debe de proporcionar y que datos debe gestionar. Es decir, nos indicarán como se deben de calcular sus valores, que datos se mantienen, como se manipulan...

Tabla 15. Requisitos funcionales del producto

RF001	Registrar riesgos
--------------	--------------------------

⁹ Jordi Pradel Miquel y Jose Raya Martos. Documentación de requisitos. Material de la asignatura Ingeniería de requisitos. UOC PID_00191264

Como "la empresa"
Quiero poder registrar riesgos
Para tenerlos identificados y poder reaccionar a ellos para evitar que impacten en la gestión y en los resultados
Criterios de aceptación
Deben de pertenecer a una de las categorías según lo recogido en el RF003
Deben tener una criticidad que va de 1 (riesgo crítico) a 3 (riesgo bajo)
Presentarán un impacto que será el producto de la criticidad del riesgo por la importancia de la categoría a la cual el riesgo está asociado. (Cuanto más pequeño sea este valor más impacto tendrá).
Tendrán tres posibles orígenes: auditoría interna, auditoría externa o autoevaluación. Este criterio tal como se comenta más adelante se adapta para incluir un cuarto origen "gestión de riesgos". Que se corresponderá con los riesgos detectados a priori por la gestión del departamento de riesgos.
Dispondrán de los datos: categoría, fecha de creación, descripción, origen, criticidad, impacto...
Enunciado
... "la empresa", necesita mejorar la gestión de sus riesgos para poder tener una mejor reacción ante cualquier riesgo que se pueda detectar.
...riesgos que, dependiendo de su criticidad, pueden impactar en la gestión y en los resultados.
...En cualquier caso, el tema más importante es cómo la empresa reacciona delante de los riesgos identificados con tal de minimizar, tanto como sea posible, su impacto en las operaciones de la empresa.
...que haya un riesgo no implica que haya pasado ningún incidente relacionado con el correspondiente riesgo, sólo es importante gestionarlo cuanto antes mejor para que el riesgo desaparezca antes de tener un incidente.
Párrafo: "Los riesgos tendrá... criticidad, impacto..."
RF002 Definir planes de acción (acciones mitigadoras)
Como "la empresa"
Quiero definir planes de acción (acciones mitigadoras) sobre los riesgos
Para evitarlos o reducir el impacto de los posibles incidentes
Criterios de aceptación
Serán definidos por la persona responsable de los riesgos detectados
Vinculados a la gestión indicada en el RF008
Deben tener 5 estados: definida, en curso, implementada con el riesgo corregido, implementada con el riesgo mitigado o descartada
Se debe de almacenar la información básica: número, descripción, responsable, fecha de creación, fecha estimada de ejecución...
Enunciado
Si un riesgo no se puede eliminar del todo, como mínimo, se deberían definir acciones mitigadoras para reducir el impacto de posibles incidentes.
Párrafo: "El último tema... para eliminar o mitigar el riesgo."
Párrafo: "Las acciones definidas... estado de descartada."
RF003 Categorizar riesgos
Como "la empresa"
Quiero poder categorizar los riesgos
Para poder agrupar los de características similares
Criterios de aceptación
Deben de tener una característica importancia con valores 1 (muy importante) a 3 (poco importante).
El sistema de BD ha de permitir almacenar tantas categorías como se quiera.
Inicialmente se deben de crear las categorías: ciberseguridad, seguridad física de instalaciones, aplicaciones informáticas y finanzas
Se deberá guardar la fecha de creación en el sistema, los datos de la persona responsable y su importancia (que ha de poder cambiar en el tiempo).
Enunciado

	Todo el párrafo: "Dentro del proceso de gestión de riesgos se han definido diferentes categorías ... su importancia (que ha de poder cambiar en el tiempo)"
RF004	Definir controles
	<i>Como</i> equipo de control interno <i>Quiero</i> poder definir una serie de controles <i>Para</i> que los departamentos puedan revisarlos
	Criterios de aceptación
	El sistema de BD ha de permitir guardar toda la información relacionada con los controles definidos
	Enunciado
	Párrafo: "El equipo de control interno... los riesgos detectados."
RF005	Crear campañas
	<i>Como</i> equipo de control interno <i>Quiero</i> definir campañas <i>Para</i> para periódicamente comprobar que los diferentes departamentos cumplen los criterios definidos
	Criterios de aceptación
	Las campañas deben de almacenar cuando empiezan y acaban, quien las lidera, quien participa, los resultados que se obtienen...
	Deben de recoger toda la información relacionada con los riesgos detectados
	Enunciado
	Párrafo: "El equipo de control interno... los riesgos detectados."
RF006	Realizar auditorías internas
	<i>Como</i> equipo de auditoría interna <i>Quiero</i> revisar los diferentes departamentos <i>Para</i> asegurar que trabajen siguiendo los procesos y las políticas definidas
	Criterios de aceptación
	Tendrán un objetivo
	Deben de almacenar el departamento y personas implicadas
	Deben de recoger toda la información relacionada con los riesgos detectados
	Enunciado
	Párrafo: "La empresa dispone también de un equipo de auditoría interna... se debe guardar en la BD."
RF007	Realizar auditorías externas
	<i>Como</i> "la empresa" <i>Quiero</i> que una empresa auditora externa revise las operativas de los departamentos <i>Para</i> asegurar que los procesos de la empresa están correctamente garantizados, es decir, con los riesgos controlados
	Criterios de aceptación
	Deben de anotarse toda la información relacionada con los riesgos detectados
	Es necesario guardar la información correspondiente a cada intervención (empresa, auditora, fechas de inicio y finalización, coste, resultados, personas implicadas...)
	Enunciado
	Párrafo: "Con tal de poder... personas implicadas..."
RF008	Gestionar los riesgos detectados en las revisiones
	<i>Como</i> "la empresa" <i>Quiero</i> gestionar los riesgos detectados en las revisiones <i>Para</i> para eliminar o mitigar el riesgo
	Criterios de aceptación
	Deben de tener una persona responsable
	Como mínimo deben de tener una acción que no esté descartada
	Se considera corregido cuando tenga alguna acción asociada en estado de implementada con el riesgo corregido y el resto de las acciones como descartadas, o tenga todas las acciones asociadas en estado de implementada con el riesgo corregido.

	Se considera mitigado cuando alguna de las acciones asociadas esté en estado de implementada con el riesgo mitigado
	Se considera abierto cuando no está corregido ni mitigado
Enunciado	
	Párrafo: "El último tema... para eliminar o mitigar el riesgo."
	Párrafo: "Las acciones definidas...esté en estado de descartada."
	Párrafo: "Un riesgo se considerará corregido...se considerará como abierto."
RF009	Guardar información de los muestreos
	Como "la empresa" Quiero que se guarden todos los muestreos realizados en cualquier inspección (auditoría interna, por auditoría externa o por las autoevaluaciones) Para poder hacer un seguimiento a futuro y que todo el proceso sea transparente
Criterios de aceptación	
	<Se cumpla lo indicado por el requerimiento>
Enunciado	
	Párrafo: "Un tema importante... al muestreo realizado."
RF010	El repositorio estadístico tiene que cubrir una serie de consultas importantes
	Como "la empresa" Quiero definir una serie de consultas a priori sobre el repositorio estadístico Para asegurar que el sistema resuelve una serie de indicadores importantes para el desempeño de la empresa
Criterios de aceptación	
	Lo indicado en la lista de consultas del documento de requerimientos
Enunciado	
	Párrafo: "El estudiante podrá definir...porcentaje de acciones descartadas."
RF011	Se debe de almacenar en un log todas las acciones realizadas sobre la base de datos
	Como "la empresa" Quiero que todas las acciones realizadas sobre la base de datos se almacenen en un log Para facilitar el mantenimiento del sistema
Criterios de aceptación	
	Se debe de registrar almacenando el número del procedimiento ejecutado y los parámetros de entrada y de salida
Enunciado	
	Párrafo: "Para el tema del log de las acciones realizadas...tratamiento de excepciones."

3.3.1.2 Requisitos no funcionales

Los requisitos funcionales se corresponden con los requisitos del producto que no se refieren a la funcionalidad. Es decir, las calidades esperadas del sistema. Se referirán a aspectos como: fiabilidad, mantenibilidad, rendimiento o usabilidad.

Tabla 16. Requisitos no funcionales del producto

RNF001	Restringir el acceso a la BD solo por procedimientos
	Como "la empresa" Quiero que toda la gestión y acceso a la información se hará mediante procedimientos de BD Para asegurar la integridad de los datos
Criterios de aceptación	
	<Se cumpla lo indicado por el requerimiento>
Enunciado	
	Párrafo: "Nuestro equipo será el encargado...integridad que se consideren relevantes..."
RNF002	Implementar y describir los procedimientos de ABM (Alta + Baja + Modificación) de entidades más relevantes

	<p>Como "la empresa"</p> <p>Quiero implementar y describir los procedimientos de ABM (Alta + Baja + Modificación) de entidades más relevantes</p> <p>Para asegurar en un primer momento el tratamiento de todos los datos relevantes</p>
	Criterios de aceptación
	<Se cumpla lo indicado por el requerimiento>
	Enunciado
	A nivel de procedimientos, se deberá implementar y describir con detalle los procedimientos de ABM (Alta + Baja + Modificación) de todas las entidades (o clases) que se consideren relevantes.
RNF003	Diseñar la aplicación para cualquier volumen de datos
	<p>Como "la empresa"</p> <p>Quiero que la aplicación funcione para cualquier volumen de datos</p> <p>Para no se decrezca el rendimiento del sistema cuando crezcan los datos</p>
	Criterios de aceptación
	Las consultas estadísticas deben de ser lo más eficientes posibles en tiempos de respuesta
	Enunciado
	Párrafo: "Se deberá tener en cuenta...tiempo de respuesta."
	La BD deberá de ser escalable para poder ir incorporando progresivamente todas aquellas necesidades que surjan durante su vigencia.
RNF004	Restricción, los resultados sobre el repositorio estadístico en un tiempo constante 1
	<p>Como "la empresa"</p> <p>Quiero que los resultados del repositorio estadísticos se definan en un tiempo constante 1</p> <p>Para asegurar la calidad del sistema obtenido</p>
	Criterios de aceptación
	<Se cumpla lo indicado por el requerimiento>
	Enunciado
	Párrafo: "La única restricción...de su ejecución."
RNF005	Todos los procedimientos deben de cumplir un estándar de salida
	<p>Como "la empresa"</p> <p>Quiero que todos los procedimientos cumplan con un estándar de salida llamado RSP</p> <p>Para asegurar la calidad y homogeneidad del sistema obtenido</p>
	Criterios de aceptación
	Dispondrán de un parámetro de salida llamado RSP, de tipo Sting, que indicará si la ejecución ha finalizado correctamente (valor 'OK') o si ha fracasado (valor 'ERROR+TIPO DE ERROR')
	Dispondrán de tratamiento de excepciones.
	Enunciado
	Párrafo: "Para el tema del log de las acciones realizadas...tratamiento de excepciones."

3.3.2 Requisitos del proceso

Este conjunto de requisitos se refiere a restricciones que están al margen de la funcionalidad del producto y afectan al propio desarrollo del producto. Estas suelen centradas en aspectos como: la metodología del proceso, a las herramientas utilizadas, tiempo de desarrollo... Uno de los requisitos más importantes en esta categoría suele ser el coste. Pero en nuestro caso no aplica.

Tabla 17. Requisitos del proceso

RP001	Realizar un modelo conceptual mediante un diagrama UML o E/R
	<p>Como "la empresa"</p> <p>Quiero se deberá realizar el modelo conceptual utilizando el diagrama UML o E/R, incluyendo una lista con todas las restricciones de integridad que se consideren relevantes.</p> <p>Para asegurar que se cumple la metodología durante el proceso de realización del producto</p>
	Criterios de aceptación

<Se cumpla lo indicado por el requerimiento>	
Enunciado	
Párrafo: "Nuestro equipo será el encargado...integridad que se consideren relevantes."	
RP002	Realizar una inicialización de datos y un juego de pruebas
<i>Como</i> "la empresa"	
<i>Quiero</i> hacer la inicialización con un conjunto de datos y presentar un juego de pruebas exhaustivo	
<i>Para</i> garantizar el correcto funcionamiento de las funcionalidades implementadas, así como el control de errores y de situaciones de excepción	
Criterios de aceptación	
<Se cumpla lo indicado por el requerimiento>	
Enunciado	
Párrafo: "Además de presentar...situaciones de excepción."	
RP003	La base de datos debe de ser realizada en el gestor relacional SQL Server
<i>Como</i> "la empresa"	
<i>Quiero</i> que el sistema se desarrolle en SQL Server	
<i>Para</i> asegurar la velocidad de desarrollo	
Criterios de aceptación	
<Se cumpla lo indicado por el requerimiento>	
Enunciado	
No se recoge en el enunciado. Se desprende de lo indicado en la primera fase de planificación	
RP004	Se debe de cumplir las fechas finales del proceso
<i>Como</i> "la empresa"	
<i>Quiero</i> que el sistema se desarrolle en el tiempo definido	
<i>Para</i> cumplir con las fechas indicadas en el calendario	
Criterios de aceptación	
<Se cumpla lo indicado por el requerimiento>	
Enunciado	
No se recoge en el enunciado. Pero es de vital cumplimiento para poder presentar el TFG a tiempo	

4. Diseño

Este bloque engloba la parte principal del esfuerzo del proyecto y para su realización deberé basarme en los requerimientos extraídos del documento de definición y del conocimiento del entorno proporcionado por el estudio del contexto. Con ellos realizaré un primer diseño centrado en los datos que deben de ser recogidos en el sistema. Este diseño conceptual estará enfocado en recoger cómo se estructuran los datos y como se relacionan entre sí. Y en él, se sintetizarán en un modelo visual todas las necesidades y restricciones indicadas por el enunciado. El modelo obtenido debe de ser independiente de la tipología de base de datos que se utilice para su representación: relacional, orientada a objetos, documental...

Una vez fijado este primer modelo. Deberemos de usarlo como base para la elaboración del siguiente paso del proceso, el diseño lógico. En esta fase como norma general deberemos de concretar el tipo de sistema que soportará los datos. En nuestro caso, este punto ya está definido por el enunciado del problema. Será un sistema de base de datos del tipo relacional. Con este destino en mente en esta fase aplicaremos una serie de reglas y criterios ya definidos

que nos permitirán ir realizando la conversión al nuevo diagrama. Sin embargo, este proceso no será del todo automático y deberemos de decidir sobre el modo en que alguna de las estructuras del diseño conceptual acaba representándose en el sistema lógico.

Por último, en la última fase del diseño, deberemos lo primero concretar cuál será el sistema relacional concreto sobre el que implementaremos nuestra base de datos. Como en la fase anterior esta decisión ya fue tomada. En este caso en la fase de planificación donde se indicó que nuestro sistema relacional final será SQL Server. El siguiente paso será Transformar el diseño lógico en un diseño físico y por último implementarlo. Para ello será necesario fijar una correspondencia entre los tipos utilizados en los diseños anteriores con los tipos existentes en SQL Server. Fijaremos el tamaño de cada campo y si se admiten valores nulos o no. Crearemos las claves primarias y acabaremos plasmando las claves foráneas del modelo anterior. Por último, crearemos las restricciones que deban de ser necesarias las relaciones.

He elegido separar la lógica del sistema en tres bloques diferenciados funcionalmente y me basaré en esta división para ir explicando cada uno de los diseños. Por un lado un bloque principal que he denominado ERM que contendrá toda la lógica del sistema. Por otra parte un bloque centrado en la parte de la trazabilidad del sistema LOG. Y por último, un bloque encargado de almacenar todos los indicadores solicitados por negocio DW. Estos bloques tienen un cierto carácter independiente. Aunque de partida he decidido que por simplicidad todos deberían estar alojados en un mismo servidor. Con el tiempo el propio escalado del sistema podría aconsejar que residieran en servidores distintos o incluso que fueran transformados a otro sistema de almacenamiento no relacional.

4.1 Diseño conceptual

Primeramente, es necesario indicar que el hecho de modelar para representar el conocimiento otorga relevancia a lo que quieres que se refleje en el sistema final. Por tanto, hay muchos diagramas que representan al enunciado que en gran medida dependerán del punto de vista en el que se focalice el diseñador. Es por esto por lo que aunque expongo en inicio el siguiente diagrama, este no será el único posible ni tendrá un carácter inamovible. Como el de cualquier otro sistema ira evolucionando a la lo largo del ciclo de vida del producto. Dependiendo de en donde se centren las necesidades del negocio (escalabilidad, rendimiento, trazabilidad...) Pudiendo incluso modificarse algún punto durante el desarrollo de alguna de las fases del proyecto. Intentaré de partida centrar el primer diseño en recoger la lógica de negocio descrita de la manera más abierta posible. Permitiendo de este modo la posible evolución natural del sistema. Discriminaré el diseño conceptual para cada uno de los bloques cada uno de los bloques que he definido.

4.1.1 ERM

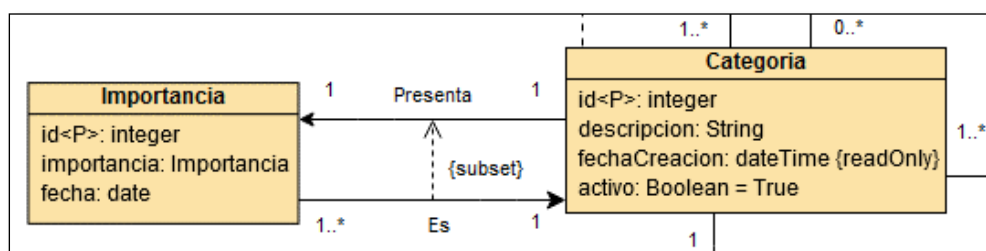
El primero de ellos ERM engloba la funcionalidad principal de la aplicación. Tal como ya he comentado, después de la revisión del enunciado y del análisis del

contexto de la gestión de riesgos. Surgen una serie de dudas con respecto al enfoque planteado en el enunciado del proyecto. A raíz de ponerlas en común con el profesor y con su beneplácito he decidido ampliar un poco la funcionalidad que se recoge en el texto. Para ello, lo primero ha sido incluir un nuevo origen para los riesgos detectados “gestión de riesgos”. Esto es debido a que, tal como comentaba con el profesor, lo habitual no es que un riesgo se detecte en una campaña de control, en ese momento se analice y se generen las acciones oportunas. Sino que exista un departamento dedicado a la gestión de riesgos que sea el encargado de tratarlos. Este departamento se encarga de definir cuáles serán los impactos que se podrán dar en la empresa, con que probabilidad y cuáles serán las consecuencias en el caso de que ocurran. Todo esto a priori. Es decir, con antelación a realizar ninguna gestión para detectarlos. Una vez localizados los impactos, sobre ellos se generan las posibles acciones para evitarlos o mantenerlos controlados. Sobre los riesgos y las acciones que se desprenden es sobre los que se definen una serie de chequeos para que se mantengan controlados. Por todo esto me parecía necesario ampliar el diseño conceptual para que permita recoger esta realidad. Esto permitirá tener una gestión de riesgos con un carácter más proactivo, en lugar de reactivo. En términos generales este nuevo origen de riesgos es la única regla que rompe con lo definido en el enunciado. El resto de las licencias se corresponden a ampliaciones de lo definido en el contexto e inclusión de valores que no se indican en el enunciado. Pasemos a concretar, solo donde sea necesario, cada uno de los criterios que me han llevado a modelar el siguiente diseño. Me iré centrado en ciertas entidades para ir revisando los distintos bloques funcionales.

Categorías

He decidido sacar la importancia de la categoría para dar relevancia a que era un valor que podía cambiar en el tiempo. De partida, se podría pensar en intentar incluir en la relación entre categoría y riesgo el valor de tiempo para recoger lo que cita el enunciado. Pero en realidad lo que indica el enunciado es que varía la importancia de una categoría en el tiempo, no el resto de sus valores o la asignación con el riesgo. Por otro lado, el hecho de que sea una propiedad ya permite que pueda ser modificada en el tiempo, por lo que no necesitaría ser externalizada de la entidad. Sin embargo, para permitir la trazabilidad de esta modificación y facilitar el seguimiento de sus valores a lo largo del tiempo, una de las premisas del enunciado. He decidido crear una entidad al margen que permitirá recoger todos los valores que ha ido tomando una categoría.

Figura 7. Relación entre categoría e importancia



Para ello modelo una doble relación. Por un lado, la relación que indica que una categoría presenta una sola importancia en un momento dado. Y por otro lado, la relación que recoge que los valores recogidos en la entidad importancia son los valores de una determinada categoría.

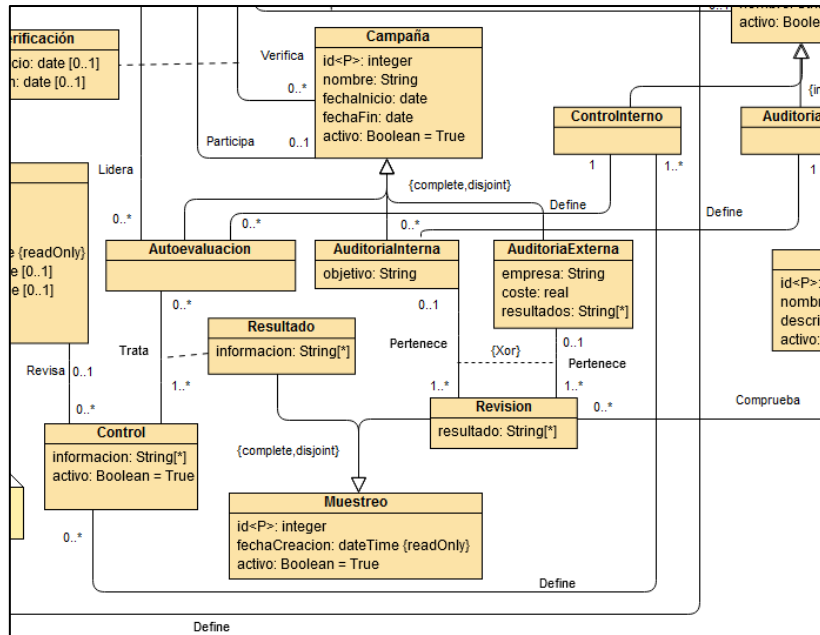
También, he decidido incluir una relación entre categoría y departamento. Ya que aunque no se comenta de manera explícita en el enunciado, si existe un punto donde se indicaba que los controles se definirán acorde a los departamentos implicados en una determinada categoría. Me pareció interesante incluir esta relación en el modelo, que de partida yo creo que existirá, dado que me casaba más con el enfoque proactivo. De este modo el departamento de riesgos se podrá centrar en realizar acciones sobre una categoría de riesgos definiendo campañas. Y, al tener modelada esta relación dada una categoría se sabrá sobre cuales departamentos se deberá de trabajar.

Departamento

Lo primero que llama la atención es que he indicado una especialización de la clase donde queda reflejada la existencia de un departamento de gestión de riesgo. La cual he realizado para poder en relevancia el nuevo enfoque y poder reflejar en el diagrama ciertas acciones que realizará que este departamento define los riesgos. Como crear las categorías y riesgos. Por otra parte, he incluido una relación con las operativas de los departamentos, la cual he disgregado en: procesos, políticas y herramientas. Ya que se cita en el enunciado y me ha parecido interesante recogerlas dado que serán el objeto de trabajo para las auditorías. El hecho de tenerlas definidas y tipificadas podría facilitar la creación de nuevas revisiones facilitando por tanto las auditorías. También, he decidido modelar la existencia de los equipos de control interno y auditoría de la empresa. Dado que así podre indicar que son los encargados de definir las autoevaluaciones y auditorías internas. Estos equipos podrían estar relacionados los dos con el departamento de gestión de riesgos, o al menos el de control interno. Pero en el modelo he preferido dejarlo abierto y no reflejarlo. Dado que podría existir una separación entre los que definen los riesgos y los que los gestionan. En cualquier caso el diagrama actual permitiría ambos supuestos dado que los equipos cuelgan de la generalización y por tanto podrían pertenecer al departamento de gestión de riesgos. Por otro lado, estos equipos vemos que estarán formados por personas y he indicado que serán un subconjunto de las que contiene el departamento. Se podría pensar en un modelo en el que los equipos no fueran una entidad débil con respecto a los departamentos y tuvieran un carácter transversal o incluso paralelo. Pudiendo personas de diversos

relevancia a este concepto y se realizaba en el texto la misma generalización. Sobre la naturaleza de los datos a registrar en estos puntos se indica poca cosa en el enunciado, por lo que he incluido un campo texto multivaluado para recoger la información. En otro tipo de proyecto sería un punto por desarrollar y concretar con el cliente.

Figura 9. Diagrama del bloque funcional relacionado con campaña



Riesgos

Este será el último conjunto de entidades. Es el bloque principal del proyecto y por tanto el que se desarrolla más en el enunciado. Es por esto por lo que sus entidades presentan una mayor cantidad de atributos. Además de las entidades que se mencionan he incluido una serie de entidades que no son recogidas en el enunciado. Estas son originadas del estudio del análisis de riesgos y tienen como interés complementar la información sobre los riesgos pero sin variar la funcionalidad indicada en el documento de definición. Por un lado, tenemos las causas, un riesgo ocurre normalmente debido a unas determinados motivos. Por ejemplo, un accidente en una planta de producción puede ocasionar que se materialice el riesgo de una mala imagen de la empresa. Incluso habrá causas que ocasionarán más de un riesgo, por ejemplo “el COVID” puede ocasionar que se materialicen varios riesgos. Por otro lado, una vez que ocurre un riesgo tiene una serie de consecuencias. Por ejemplo, una mala imagen de la empresa puede tener un impacto financiero. Estas tres entidades están intrínsecamente relacionadas. Y la probabilidad de que ocurra un riesgo dependerá de la probabilidad de la ocurrencia de sus causas en gran medida y las consecuencias tendrán una probabilidad de ocurrencia dependiendo de la probabilidad de ocurrencia de los riesgos. Sobre esta base habitualmente es la que trabajará un sistema de la gestión de riesgos. Según la criticidad de los riesgos, las consecuencias que ocasionan si se materializan y las probabilidades de que ocurran, la gestión de riesgos deberá de definir las acciones para intentar eliminarlos o controlarlos. Estas acciones serán de corrección cuando quieran

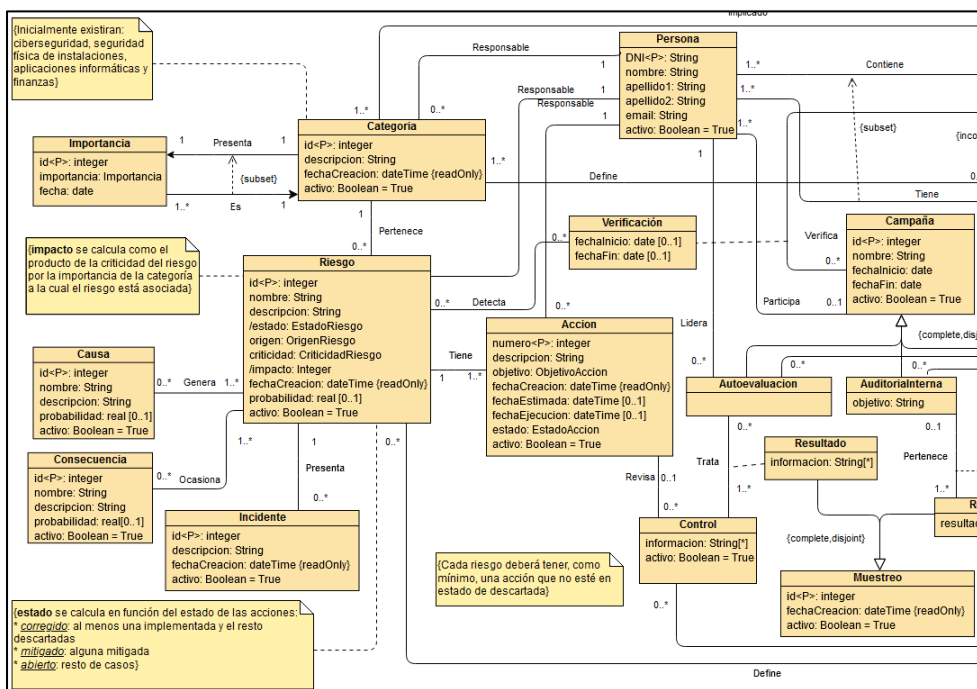
afectar a las causas que ocasionan los riesgos y de mitigación cuando quieran tratar las consecuencias. Por lo que, existe una relación directa entre las acciones de control de los riesgos y las causas y consecuencias de ellos.

Sin embargo, si revisamos el diagrama vemos que esta relación no se recoge. Esto es debido a que ya se ha definido en el enunciado un tratamiento de estados complejo entre los riesgos y las acciones. Es por esto por lo que no he querido complicar demasiado el diagrama referenciando las acciones a las causas y las consecuencias. Sobre todo teniendo en cuenta que no era solicitado en el enunciado y me obligaría a romper quizás con lo recogido en el enunciado en varios puntos. Por lo que he preferido incluir estas entidades solo para mejorar la información que recoge de los riesgos y le he otorgado un carácter opcional. Es por esto por lo que se permite registrar riesgos sin causas o consecuencias. Eso sí he incluido un campo objetivo en la acción que nos determinará si tiene como intención corregir o mitigar un riesgo. Este punto es un punto posible de evolución del sistema. Que se podría replantear en un futuro para modelizar un sistema con un carácter más proactivo. Por otra parte he incluido una entidad para registrar los posibles incidentes que se den de un determinado riesgo. Ya que me ha parecido un punto interesante para facilitar un posible seguimiento.

Otro matiz interesante de resaltar es que vinculado los riesgos a la entidad asociativa verificación que existe entre las campañas y los departamentos. Dado que este es el punto normal, según el enunciado de detección de un riesgo. Además será necesaria para satisfacer uno de los indicadores solicitados para la parte de DW.

Por último, hay que indicar que he decidido relacionar los controles con las acciones. De esta manera se podrá definir a priori controles con la intención de revisar la implementación de las acciones en las campañas de autoevaluación.

Figura 10. Diagrama del bloque funcional relacionado con riesgos

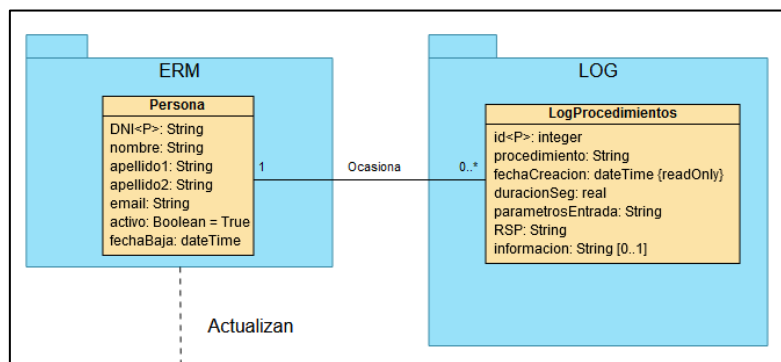


Además de lo indicado durante la fase de desarrollo se ha visto la necesidad de agregar nuevos campos (activo) a la mayoría de las entidades con el fin de realizar un borrado lógico de los elementos y no físico. De este modo se mejorará la trazabilidad de todo el sistema.

4.1.2 LOG

Veamos el segundo de los bloques es el que corresponde al Log de la aplicación. En esta parte poco hay que decir. Simplemente que he agregado un parámetro para recoger información adicional. Ya que tal el parámetro RSP en el caso de que la operación se ha realizado correctamente solo debe de contener el valor OK. Por tanto me ha parecido necesario un parámetro adicional de texto para poder registrar cualquier otra información relativa a la operación correcta. Posiblemente esta parte se amplíe a la hora de realizar la siguiente fase del proyecto.

Figura 11. Diagrama del bloque funcional relacionado con el LOG



Durante la fase de implementación se decide agregar un parámetro adicional a la tabla de log para almacenar la duración de la ejecución. De este modo se podrá detectar en el futuro puntos donde sea necesario realizar una optimización de los procedimientos.

4.1.3 DW

El siguiente bloque se refiere a la parte que comprende el DW. En este caso el diseño estará supeditado en gran parte a la funcionalidad que debe de soportar y no solo a los datos que debe de almacenar. Dado que se solicita algunos de los indicadores deben de presentar en todo momento valores correctos y se restringe la posibilidad de a través de una consulta en los datos existentes el único camino posible es mantener los indicadores y los datos necesarios para su cálculo a través de actualizaciones cuando ocurra una acción que afecte a su valor. Se pueden abordar estas actualizaciones de manera automática o de manera específica. En el caso de optar por automatizar su mantenimiento trataremos sobre las modificaciones que afecten a las entidades implicadas en el cálculo de los indicadores. Para ello deberíamos de desarrollar una serie de disparadores (*triggers*). De partida, no esto muy a favor de su empleo. Ya que suelen ocultar parte de la lógica de los sistemas y en el día a día se suele olvidar que se encuentra presente en el sistema. Lo cual complica muchas veces detectar y solventar problemas de rendimiento, sobre todo si incorporan una lógica

compleja. El otro camino es gestionar específicamente los procedimientos que afectan a las tablas en lugar de las tablas en sí. Este camino es viable porque en la definición de problema se indica que los valores solo podrán ser modificados mediante el uso de procedimientos. Pospondré esta decisión a la fase de implementación y realizaré pruebas antes de decidirme.

Otro de los problemas que ocasiona tener que mantener los valores de los indicadores a través de actualizaciones es que se debe tener en cuenta la concurrencia de procesos. Para cumplir con los requerimientos debo de mantener en todo momento los valores necesarios para realizar el cálculo de ciertos indicadores. Por tanto, los procesos que los gestionen deberán de obtener el valor de trabajo que exista, realizar el cálculo que toque sobre ellos y actualizar por último los valores existentes. Por tanto, si dos procesos ocurren a la vez. Los dos obtendrán los valores de trabajo, cada uno de ellos realizará su cálculo y actualización de manera paralela por lo que a la hora de actualizar se perderá uno de los valores. Para solucionar este problema se puede optar por un enfoque pesimista u optimista del problema. En el pesimista se bloquearían los registros a la hora de trabajarlos. Lo que ocasionaría el error o la espera del resto de procesos. En el optimista, se presupone que este problema no suele ocurrir muy habitualmente por lo que se gestiona controlando en el caso de que ocurra (realizando un reintento...). A la larga los dos sistemas pueden ocasionar problemas de rendimiento conforme se escale el sistema si se aumenta la concurrencia de los procesos. En este caso lo mejor sería realizar una cola de trabajo donde acumular las actualizaciones de los indicadores y delegar a un solo proceso encargado del cálculo de sus valores. De esta manera serializaríamos la actualización y evitaríamos el problema. Sin embargo, esta opción chocaría con la inmediatez de alguno de los indicadores. De partida, y dado el contexto de la solución donde creo que la concurrencia no será un problema, elegiré el enfoque optimista. Manteniendo por tanto la premisa de inmediatez del enunciado. Aunque a la larga creo que se deberán de desconectar más los sistemas y poner una capa con colas para el tratamiento.

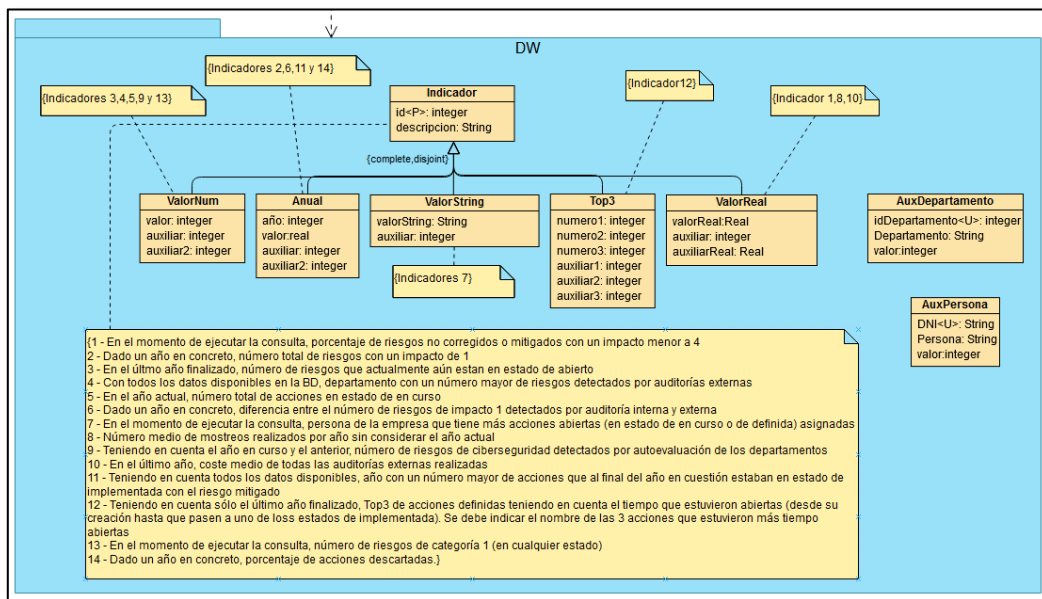
Una vez tratados estos puntos funcionales que podrían afectar al diseño de esta parte. Pasaré a concretar la estructura donde se almacenarán los datos. Para ello he realizado un análisis de los indicadores basándose en sus características. Hay indicadores que tienen un carácter anual, por ejemplo el 8, 9... Estos indicadores se calcularán al cierre del año. Por tanto, como no necesitan ser calculados con cada actualización, no necesitan tener almacenados ningún valor auxiliar a la hora de realizar el cálculo. Para poder dar el resultado del resto de indicadores de carácter inmediato será necesario mantener no solo el valor de este, sino en muchos casos una serie de valores auxiliares necesarios para realizar sus cálculos. Por ejemplo, para calcular el valor del primer indicador: *“En el momento de ejecutar la consulta, porcentaje de riesgos no corregidos o mitigados con un impacto menor a 4”*. Necesitaré dos valores auxiliares uno que mantenga el número total de riesgos y otro que mantenga el número de riesgos que cumple la condición. De manera, que cada modificación sobre los riesgos pueda contabilizar sobre el valor auxiliar que le corresponda y el indicador resultante será el resultado de realizar la división. En dos de los indicadores (4 y 7) he visto

que no será solo necesario utilizar valores auxiliares sino que necesitaré mantener toda una entidad auxiliar para almacenar los datos. Estas entidades las he incluido en el diseño pero no he agregado ninguna relación por no complicar más la comprensión del diagrama.

Por otra parte, para poder modelar el diseño del DW he necesitado tener en además la tipología de valores a almacenar. En algunos casos habrá valores que deberán de almacenar un valor entero, en otros casos una cadena e incluso en uno de los casos deberán de contener una serie de datos asociados a un año.

Teniendo en cuenta todo lo dicho anteriormente he clasificado cada uno de los valores de características similares y especializando cada uno de los valores desde la generalización de indicador. Como resultado tendremos el siguiente diseño.

Figura 12. Diagrama del bloque funcional relacionado con DW



Como resultado del desarrollo de la fase de implementación ha sido necesario incluir algún parámetro adicional en alguna de las tablas para poder usarlo como parámetro auxiliar. Por otra parte, alguno de los indicadores ha sido a alguna tabla distinta a la inicial para almacenar su valor.

El diseño final será el siguiente.

Figura 13. Diagrama conceptual completo del bloque ERM

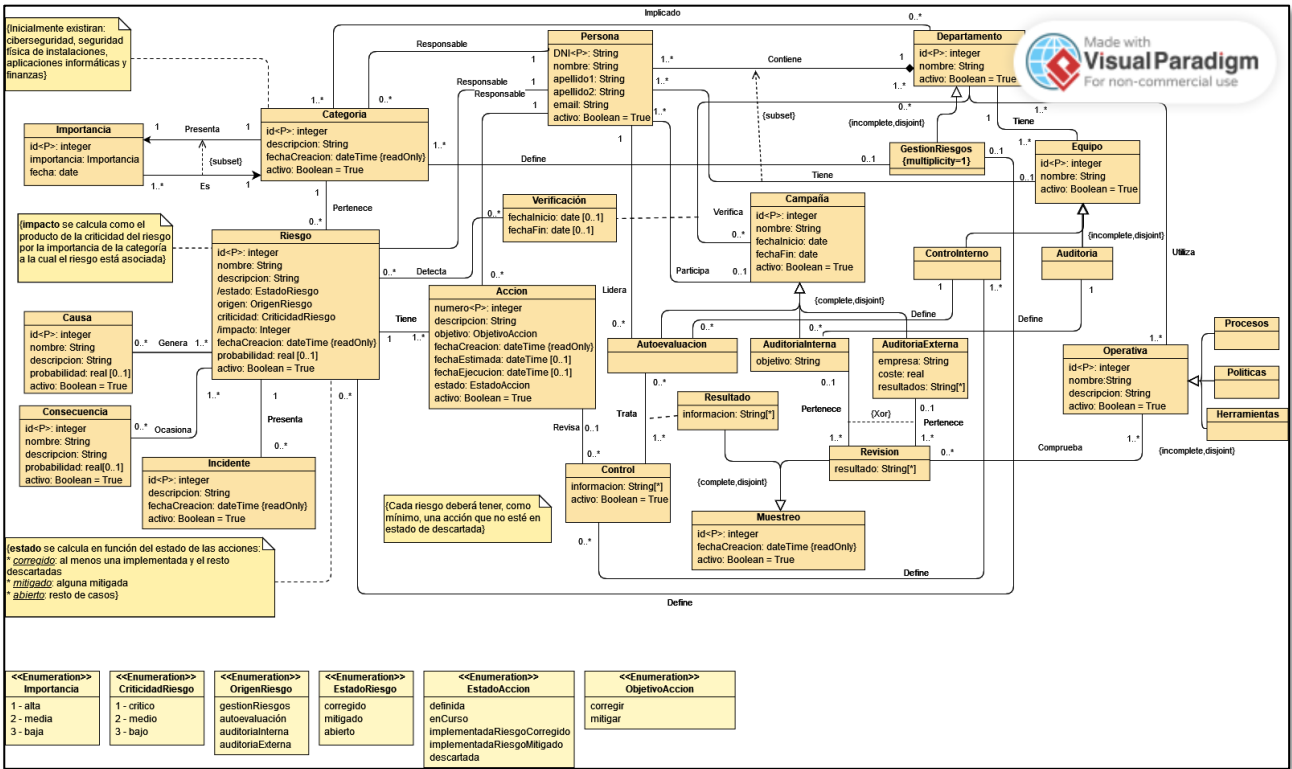
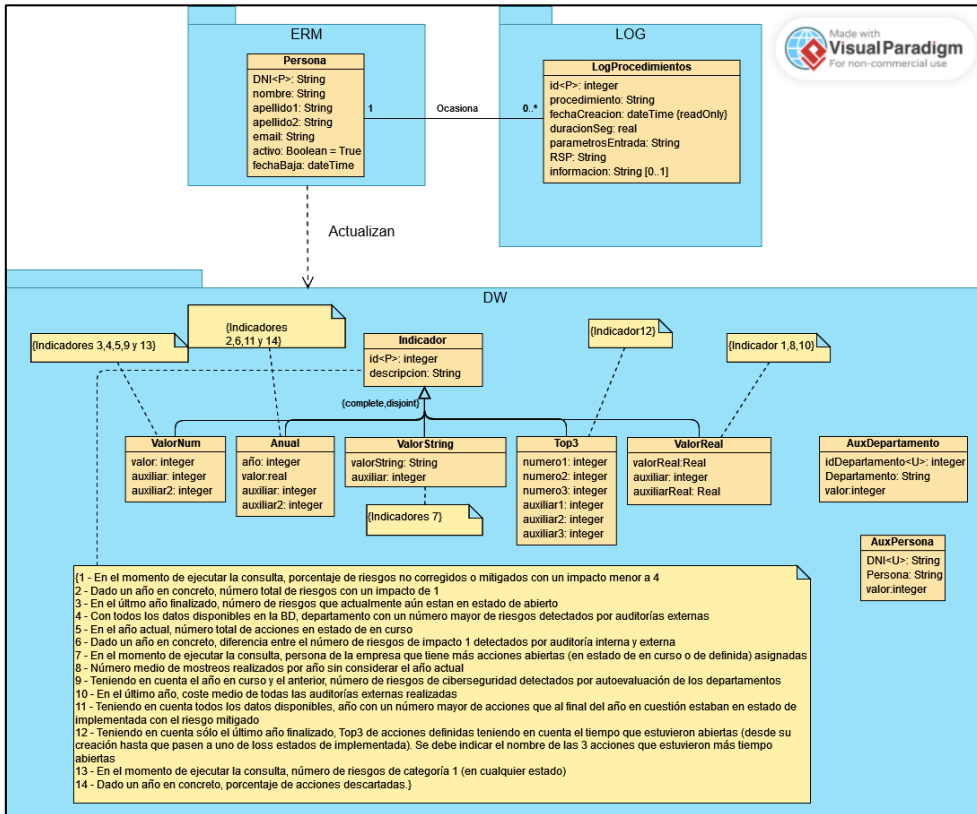


Figura 14. Diagrama conceptual completo de los bloques LOG y DW



4.2 Diseño lógico

En esta fase deberemos de basarnos en el diseño anterior e ir realizando una serie de conversiones muy estandarizadas para llevar nuestro esquema lógico a una base de datos relacional. Omitiré las reglas de conversión dado que no aportan nada pero éstas están descritas en el módulo 3 de la asignatura de Diseño de Bases de datos que está incluido en la bibliografía. Por tanto, pasare a describir cada uno de los bloques ya diferenciados en la fase anterior comentando si es necesario concretar algún punto.

4.2.1 ERM

Veamos el primero de los bloques ERM

Departamento

Lo primero a comentar es la entidad *GestionRiesgos* esta entidad era una especialización de Departamento. Y se modelo para poder representar que este departamento podía crear *Categorias* y *Riesgos*. Esta representación tenía relevancia en el diseño conceptual y tendrá utilidad en el caso de querer utilizar este diagrama para el diseño de procesos. Sin embargo, tiene poca utilidad mantenerla en el modelo de datos. Dado que este departamento no tiene ninguna propiedad que lo distinga del resto de departamentos. Por otro lado si analizamos la multiplicidad de sus relaciones vemos que se debería de crear una referencia al identificador de riesgos dentro de las entidades de *Categoria* y *Riesgo*. Dado que solo hay un departamento que las puede crear este indicador siempre contendrá el mismo valor. Por tanto, no tiene sentido representarlo. Por tanto, asumiré que esta entidad estará recogida en la entidad *Departamento* y obviaré las relaciones.

Pasará algo parecido con respecto a los equipos de control interno y auditoría. No tienen de partida ninguna propiedad que los deba diferenciar del resto de equipos. Y sus relaciones tendrán sentido tenerlas en cuenta en un diseño de procesos. Porque solo el equipo de control interno definirá autoevaluaciones y controles. Por lo que en el caso de arrastrar un identificador a estas tablas siempre tendrá un valor fijo. Por lo tanto, se omitirá en el diseño lógico. Si es necesario en un futuro será fácil de agregar.

Por último, en cuanto a las operativas también he decidido agrupar la generalización en una sola entidad. Dado que en el enunciado solo se comentan y no se define nada que las diferencien.

Departamento (id, nombre, activo)

Equipo (id, nombre, idDepartamento, activo)
{idDepartamento} is foreing key to Departamento

Operativa (id, nombre, descripcion, activo)

DepartamentoOperativa (idDepartamento, idOperativa)
{idDepartamento} is foreing key to Departamento
{idOperativa} is foreing key to Operativa

Persona (DNI, nombre, apellido1, apellido2, email,
idDepartamento, idEquipo, activo)
{idDepartamento} is foreing key to Departamento
{idDepartamento} is foreing key to Equipo

Categoría

Analizando el diseño elegido para modelar estas entidades vemos que se produce lo que se conoce como abrazo mortal entre la entidad *Categoría* e *Importancia*. Dado que la creación de cada una de ellas referencia a la anterior. Lo que nos generaría un problema a la hora de realizar la inserción. Para resolverlo en este caso he optado por la solución más simple que es omitir una de las relaciones. He decidido mantener la relación que representa que toda categoría tiene un estado. Por otro lado, para facilitar su creación uno de los indicadores que relacionan las tablas debe admitir el valor nulo. En mi caso he optado por el existente en la categoría. De este modo insertaré primero la categoría y posteriormente su estado. Actualizando su valor después en la categoría del estado creado. Aunque se decida no realizar la clave foránea está será mantenida a nivel de procedimiento. Por tanto tendremos:

Categoria (id, descripcion, fechaCreacion, importanciaActual,
idResponsable, activo)
{importanciaActual} is foreing key to Importancia
{idResponsable} is foreing key to Persona

Importancia (id, importancia, fecha, idCategoria)

CategoriaDepartamento (idCategoria, idDepartamento)
{idCategoria} is foreing key to Categoria
{idDepartamento} is foreing key to Departamento

Riesgos

Este bloque no necesita ninguna aclaración. Veamos cómo queda todo:

Riesgo (id, nombre, descripcion, estado, origen, criticidad,
impacto, fechaCreacion, probabilidad, idCategoria, activo)
{idCategoria} is foreing key to Categoria

Causa (id, nombre, descripcion, probabilidad, activo)

RiesgoCausa (idRiesgo, idCausa)
{idRiesgo} is foreing key to Riesgo
{idCausa} is foreing key to Causa

Consecuencia (id, nombre, descripcion, probabilidad, activo)

RiesgoConsecuencia (idConsecuencia, idRiesgo)
{idConsecuencia} is foreing key to Consecuencia
{idRiesgo} is foreing key to Riesgo

Incidente (id, descripcion, fechaCreacion, idRiesgo, activo)
{idRiesgo} is foreing key to Riesgo

Accion (**numero, descripcion, objetivo, fechaCreacion, fechaEstimada, fechaEjecucion, estado, idRiesgo, idResponsable, activo**)
{idRiesgo} is foreign key to Riesgo
{idResponsable} is foreign key to Persona

Campaña

En este bloque existen dos generalizaciones. Con respecto a la primera de ellas campaña he decidido no realizar ninguna agrupación. Tienen propiedades y relaciones propias por lo que he preferido representarla tal como está. Con respecto al muestreo pasa lo mismo. Manteniendo las clases separadas podré discriminar fácilmente sus orígenes. En este bloque existen una serie de atributos multivaluados, que agregue según mi propio criterio para recoger la información referente a los resultados. En principio sobre estos valores se debería determinar si generar entidades sobre ellos, dejarlos como están o incrementar el número de ocurrencias. Avanzándome al siguiente diagrama he decidido en este caso mantenerlos como valores únicos y transformarlos luego en SQL Server con campos TEXT que admiten hasta 2GB de datos. Como norma general se debería de analizar su uso y modelar el contenido. En el caso de contener un conjunto de datos muy grande si esta no es susceptible de ser explotada por el sistema quizás se debería de buscar métodos alternativos para su almacenamiento. Dado que el hecho de contener campos con un tamaño grande en la base de datos relacional podría penalizar a la hora de ser utilizado en consultas. En mi caso he decidido mantenerlas. En un futuro si esto se produce siempre se podrá realizar un particionamiento horizontal de la tabla para que solo afecte la penalización de estas propiedades solo en las consultas que realmente lo necesitan. Veamos cómo queda:

Campaña (id, nombre, fechaInicio, fechaFin, activo)

Verificacion (idDepartamento, idCampaña, fechaInicio, fechaFin)
{idDepartamento} is foreign key to Departamento
{idCampaña} is foreign key to Campaña

Deteccion (idDepartamento, idCampaña, idRiesgo)
{idDepartamento, idCampaña} is foreign key to Verificacion
{idRiesgo} is foreign key to Riesgo

Autoevaluacion (id, idLider)
{id} is foreign key to Campaña
{idLider} is foreign key to Persona

AuditoriaInterna (id, objetivo)
{id} is foreign key to Campaña

AuditoriaExterna (id, empresa, coste, resultados)
{id} is foreign key to Campaña

Control (id, informacion, idAccion, idAutoevaluacion, activo)
{idAccion} is foreign key to Accion

{idAutoevaluacion} is foreing key to Campaña
{idAutoevaluacion} is foreing key to Autoevaluacion

Muestreo (id, fechaCreacion, activo)

Revision (id, resultado, idAuditoria)
{id} is foreing key to Muestreo
{idAuditoria} is foreing key to Campaña
{idAuditoria} is foreing key to AuditoriaInterna
{idAuditoria} is foreing key to AuditoriaExterna

Resultado (id, idAutoevaluacion, idControl, informacion)
{id} is foreing key to Muestreo
{idAutoevaluacion} is foreing key to Campaña
{idAutoevaluacion} is foreing key to Autoevaluacion
{idControl} is foreing key to Control

Es necesario indicar que durante la fase de implementación se han agregado un nuevo campo denominado “activo” a la mayoría de las entidades principales para poder realizar una baja lógica en lugar de una baja física de las ocurrencias. Por otra parte se ha decidido replantear los nombres de las entidades que surgen de las relaciones. En principio se les había otorgado un nombre particular para cada una de ellas. Pero durante el desarrollo de la fase se ha decidido transformar ese nombre en uno compuesto de las dos entidades a las que relaciona. Con el interés de facilitar la interpretación de este

4.2.2 LOG

Esta parte no tiene nada que comentar la conversión es directa.

LogProcedimientos (id, procedimiento, fechaCreacion, duracionSeg
parametrosEntrada, RSP, información, idUsuario)
{idUsuario} is foreing key to Persona

4.2.3 DW

Con respecto a la generalización de este bloque. Se podría de partida pensar en realizar una tabla que contenga todos los valores. Sin embargo, vemos que solo alguno de los valores tiene carácter anual. Por lo que he preferido mantener la especialización. El mantener o no la tabla padre es opcional. En mi caso me he decantado por mantenerla porque me parece interesante el hecho de tener guardada una descripción de lo que recoge el indicador en la base de datos. Veamos:

Indicador (id, descripcion)

ValorNum (id, valor, auxiliar, auxiliar2)
{id} is foreing key to Indicador

Anual (id, año, valor, auxiliar, auxiliar2)
{id} is foreing key to Indicador

ValorString (id, nombre, auxiliar)
{id} is foreing key to Indicador

Top3 (id, numero1, numero2, numero3, auxiliar1, auxiliar2, auxiliar3)
{id} is foreing key to Indicador

ValorReal (id, valorReal, auxiliar, auxiliarReal)
{id} is foreing key to Indicador

AuxDepartamento (idDepartamento, Departamento, valor)
{idDepartamento} is foreing key to

AuxPersona (DNI, Persona, valor)
{DNI} is foreing key to Persona

Tal como ya se ha comentado en la fase de diseño durante la realización del proyecto se han agregado nuevos campos auxiliares a los planificados en el diseño original.

4.3 Diseño físico

Esta es la última fase de nuestro proceso de diseño. En esta fase deberemos de transformar nuestro diseño lógico en uno que pueda ser aplicado en nuestro gestor relacional SQL Server. El primer punto será transformar las tipologías de los datos utilizados en el diseño conceptual en unos equivalentes en el gestor seleccionado¹⁰. Esto nos da la siguiente tabla de conversión.

Tabla 18. Conversiones de tipos efectuadas

Tipo	SQL Server
integer	int
String	varchar
dateTime	datetime
real	decimal
String[*]	text
Enumeration	varchar+CHECK o Tabla

En esta tabla la mayoría de las conversiones son lo suficientemente claras. Comentemos las dos últimas. He optado por transformar la cadena de texto multivaluada en un campo TEXT este campo admite hasta algo más de 2 GB de longitud máxima. Esto debería ser más que suficiente para almacenar toda la información necesaria. Incluso si esta se encuentra formateada. De partida el uso de este campo me parece un camino lógico. Aunque como ya he comentado anteriormente en este punto lo mejor sería realizar un análisis con el cliente para concretar la información que se debe de guardar. Ya que, no aporta mucho tener esta información en formato texto en el servidor relacional a no ser que se vaya a explotar. Se podría almacenar en formato de ficheros en un *fileserv* o si se quiere explotar sería almacenarla en almacén de datos más especializado en su

¹⁰ Tipos de datos (Transact-SQL) [en línea]. Disponible en: <https://learn.microsoft.com/es-es/sql/t-sql/data-types/data-types-transact-sql?view=sql-server-ver16>

tratamiento, como un servidor no relacional. En cualquier caso si se externalizan estos datos estos campos deberían de ser reemplazados por una referencia a la ubicación de la información. Por otro lado, no existe en SQL Server la posibilidad de ponerle a un campo la propiedad *readOnly*. Por tanto, esta circunstancia deberá de ser controlada manualmente.

En cuanto a los campos que se corresponden con alguna de las enumeraciones definidas. He comprobado que no existe una correspondencia directa en SQL Server. En este caso se suele optar por dos posibles opciones. Si se espera que la enumeración sufra algún tipo de modificación en el futuro se pueden transformar sus valores en una entidad y transformar el campo en un campo relacionado con esta tabla. Esta opción además nos permitirá guardar más información en el caso de ser necesaria asociada a estos valores. La otra opción será reemplazar la enumeración por una cadena de texto y agregar una restricción que asegure que los valores se corresponden con los esperados. En mi caso utilizaré los dos métodos. Transformare las enumeraciones *Importancia* y *CriticidadRiesgo* en dos nuevas entidades. Dado que tienen un valor numérico que usare para calcular el impacto y un texto asociado. Y el resto las implementaré como restricciones.

En cuanto a las claves primarias y foráneas referidas en el diseño lógico. Se deberán de implementar su correspondencia en el diseño físico. En este caso la correspondencia es directa. Sobre este punto solo es necesario indicar que se optará por que los campos sean autoincrementales en el caso de que no sea valores que vengan de la transformación de una relación en tabla y por tanto sean una clave compuesta.

Mostraré el diseño final mediante diagramas obtenidos de la implementación de la estructura en SQL Server. El script de creación es accesible a través del porfolio de trabajo o del enlace:

- [01 Creación de tablas.sql](#)

Incluiré de partida el diagrama y la definición de las estructuras creadas. Veamos por cada uno de los bloques:

Persona**ERM**

Descripción	Tipo	Null	Rest	Observaciones
DNI	varchar(10)	Not Null	PK	
nombre	varchar(50)	Not Null		
apellido1	varchar(50)	Not Null		
apellido2	varchar(50)	Not Null		
email	varchar(50)	Not Null		
idDepartamento	int	Not Null	FK	Con la tabla Departamento
idEquipo	int	Null	FK	Con la tabla Equipo
activo	bit	Not Null	DF=1	Se agrega en fase de diseño. Baja lógica

ECriticidadRiesgo**ERM**

Descripción	Tipo	Null	Rest	Observaciones
id	int	Not Null	PK	
descripcion	varchar(50)	Not Null		

Eimportancia**ERM**

Descripción	Tipo	Null	Rest	Observaciones
id	int	Not Null	PK	
descripcion	varchar(50)	Not Null		

Importancia**ERM**

Descripción	Tipo	Null	Rest	Observaciones
id	int IDENTITY (1,1)	Not Null	PK	
idImportancia	int	Not Null	FK	Con la tabla Eimportancia
fecha	varchar(500)	Not Null		
idCategoria	varchar(40)	Not Null		FK no representada

Categoria**ERM**

Descripción	Tipo	Null	Rest	Observaciones
id	int IDENTITY (1,1)	Not Null	PK	
descripcion	varchar(50)	Not Null		
fechaCreacion	datetime	Not Null		
importanciaActual	int	Null	FK	Con la tabla Importancia
idResponsable	varchar(10)	Not Null	FK	Con la tabla Persona
activo	bit	Not Null	DF=1	Se agrega en fase de diseño. Baja lógica

CategoriaDepartamento**ERM**

Descripción	Tipo	Null	Rest	Observaciones
idCategoria	int	Not Null	PK,FK	Con la tabla Categoria
idDepartamento	int	Not Null	PK,FK	Con la tabla Departamento

Riesgo**ERM**

Descripción	Tipo	Null	Rest	Observaciones
id	int IDENTITY (1,1)	Not Null	PK	
nombre	varchar(50)	Not Null		
descripcion	varchar(500)	Not Null		
estado	varchar(40)	Not Null	CHECK	IN('corregido','mitigado','abierto')
origen	varchar(18)	Not Null	CHECK	IN('gestionRiesgos', 'autoevaluación', 'auditoriaInterna', 'auditoriaExterna')
idCriticidad	int	Not Null	FK	Con la tabla ECriticidadRiesgo
impacto	int	Not Null		
fechaCreacion	datetime	Not Null		
probabilidad	decimal(3,3)	Null		

idCategoria	int	Not Null	FK	Con la tabla Categoria
activo	bit	Not Null	DF=1	Se agrega en fase de diseño. Baja lógica

Causa ERM

Descripción	Tipo	Null	Rest	Observaciones
id	int IDENTITY (1,1)	Not Null	PK	
nombre	varchar(50)	Not Null		
descripcion	varchar(500)	Not Null		
probabilidad	decimal(3,3)	Null		
activo	bit	Not Null	DF=1	Se agrega en fase de diseño. Baja lógica

RiesgoCausa ERM

Descripción	Tipo	Null	Rest	Observaciones
idRiesgo	int	Not Null	PK,FK	Con la tabla Riesgo
idCausa	int	Not Null	PK,FK	Con la tabla Causa

Consecuencia ERM

Descripción	Tipo	Null	Rest	Observaciones
id	int IDENTITY (1,1)	Not Null	PK	
nombre	varchar(50)	Not Null		
descripcion	varchar(500)	Not Null		
probabilidad	decimal(3,3)	Null		
activo	bit	Not Null	DF=1	Se agrega en fase de diseño. Baja lógica

RiesgoConsecuencia ERM

Descripción	Tipo	Null	Rest	Observaciones
idConsecuencia	int	Not Null	PK,FK	Con la tabla Consecuencia
idRiesgo	int	Not Null	PK,FK	Con la tabla Riesgo

Incidente ERM

Descripción	Tipo	Null	Rest	Observaciones
id	int IDENTITY (1,1)	Not Null	PK	
descripcion	varchar(500)	Not Null		
fechaCreacion	datetime	Not Null		
idRiesgo	int	Not Null	FK	Con la tabla Riesgo
activo	bit	Not Null	DF=1	Se agrega en fase de diseño. Baja lógica

Acción ERM

Descripción	Tipo	Null	Rest	Observaciones
numero	int IDENTITY (1,1)	Not Null	PK	
descripcion	varchar(500)	Not Null		
objetivo	varchar(100)	Not Null		
fechaCreacion	datetime	Null		
fechaEstimada	datetime	Null		
fechaEjecucion	datetime	Not Null		
estado	varchar(30)	Not Null	CHECK	IN('definida','enCurso', 'implementadaRiesgoCorregido', 'implementadaRiesgoMitigado', 'descartada')
idRiesgo	int	Not Null	FK	Con la tabla Riesgo
idResponsable	varchar(10)	Not Null	FK	Con la tabla Persona
activo	bit	Not Null	DF=1	Se agrega en fase de diseño. Baja lógica

Campaña ERM

Descripción	Tipo	Null	Rest	Observaciones
id	int IDENTITY (1,1)	Not Null	PK	

nombre	varchar(100)	Not Null		
fechalnicio	datetime	Not Null		
fechaFin	datetime	Not Null		
activo	bit	Not Null	DF=1	Se agrega en fase de diseño. Baja lógica

Verificacion ERM

Descripción	Tipo	Null	Rest	Observaciones
idDepartamento	int	Not Null	PK,FK	Con la tabla Departamento
idCampaña	int	Not Null	PK,FK	Con la tabla Campaña
fechalnicio	datetime	Null		
fechaFin	datetime	Null		

Deteccion ERM

Descripción	Tipo	Null	Rest	Observaciones
idDepartamento	int	Not Null	PK,FK	Con la tabla Verificacion
idCampaña	int	Not Null	PK,FK	Con la tabla Verificacion
idRiesgo	int	Not Null	PK,FK	Con la tabla Riesgo

Autoevaluacion ERM

Descripción	Tipo	Null	Rest	Observaciones
id	int	Not Null	PK,FK	Con la tabla Campaña
idLider	varchar(10)	Not Null	FK	Con la tabla Persona

AuditorialInterna ERM

Descripción	Tipo	Null	Rest	Observaciones
id	int	Not Null	PK,FK	Con la tabla Campaña
objetivo	varchar(500)	Not Null		

AuditoriaExterna ERM

Descripción	Tipo	Null	Rest	Observaciones
id	int	Not Null	PK,FK	Con la tabla Campaña
empresa	varchar(100)	Not Null		
coste	decimal(9,3)	Not Null		
resultados	text	Null		

Control ERM

Descripción	Tipo	Null	Rest	Observaciones
id	int IDENTITY (1,1)	Not Null	PK	
informacion	text	Not Null		
idAccion	int	Null	FK	Con la tabla Accion
idAutoevaluacion	int	Null	FK	Con la tabla Autoevaluacion
activo	bit	Not Null	DF=1	Se agrega en fase de diseño. Baja lógica

Muestreo ERM

Descripción	Tipo	Null	Rest	Observaciones
id	int IDENTITY (1,1)	Not Null	PK	
fechaCreacion	datetime	Not Null		
activo	bit	Not Null	DF=1	Se agrega en fase de diseño. Baja lógica

Revision ERM

Descripción	Tipo	Null	Rest	Observaciones
id	int	Not Null	PK,FK	Con la tabla Muestreo
resultado	text	Not Null		
idAuditoria	int	Not Null	FK	Con la tabla Campaña

Resultado ERM

Descripción	Tipo	Null	Rest	Observaciones
id	int	Not Null	PK	
idAutoevaluacion	int	Not Null	FK	Con la tabla Autoevaluacion
idControl	int	Not Null	FK	Con la tabla Control
informacion	text	Not Null		

4.3.2 LOG

Además de la tabla de log para los procedimientos se ha agregado una nueva tabla. Esta surge debido a la fase de validación y tiene como único cometido facilitar el rápido desarrollo de las pruebas. No se incluye en los anteriores diagramas, dado que por su particular uso no tendrá cabida más allá de en este desarrollo físico en particular.

Figura 16. Relación de tablas que corresponden al bloque LOG

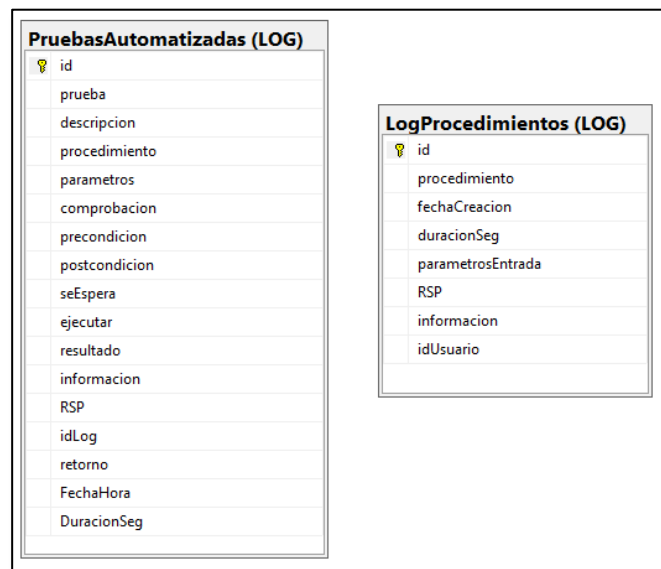


Tabla 20. Definición de tablas que corresponden al bloque LOG

PruebasAutomatizadas LOG

Descripción	Tipo	Null	Rest	Observaciones
id	int IDENTITY (1,1)	Not Null	PK	
prueba	varchar(50)	Not Null		Indicador numérico de la prueba
descripcion	varchar(500)	Null		Texto opcional para definir la prueba
procedimiento	varchar(50)	Not Null		Procedimiento que se ejecuta
parametros	varchar(4000)	Null		Parámetros necesarios para la ejecución
comprobacion	varchar(4000)	Null		Comprobación que se debe de cumplir al finalizar la ejecución
precondicion	varchar(4000)	Null		Campo para definir operaciones anteriores a la ejecución (definido para futuras mejoras del procedimiento)
postcondicion	varchar(4000)			Campo para definir operaciones posteriores a la ejecución (definido para futuras mejoras del procedimiento)
seEspera	varchar(2)	Not Null		Indica si el resultado debe ser OK o KO
ejecutar	bit	Not Null	DF=1	Determina si la prueba debe ser ejecutada
resultado	varchar(500)	Null		Correcta o Incorrecta

informacion	varchar(4000)	Null		Indica si el error es debido a una comprobación
RSP	varchar(4000)	Null		RSP resultante
idLog	int	Null		Identificador del log generado
retorno	int	Null		Retorno de la ejecución
FechaHora	datetime	Null		Fecha y hora de la ejecución
DuracionSeg	float	Null		Duración de la ejecución

LogProcedimientos LOG

Descripción	Tipo	Null	Rest	Observaciones
id	int IDENTITY (1,1)	Not Null	PK	
procedimiento	varchar(100)	Not Null		
fechaCreacion	datetime	Not Null		
duracionSeg	float	Not Null		
parametrosEntrada	text	Not Null		
RSP	varchar(4000)	Not Null		
información	varchar(4000)	Null		
idUsuario	varchar(10)	Not Null	FK	Con la tabla Persona

4.3.3 DW

Figura 17. Relación de tablas que corresponden al bloque DW

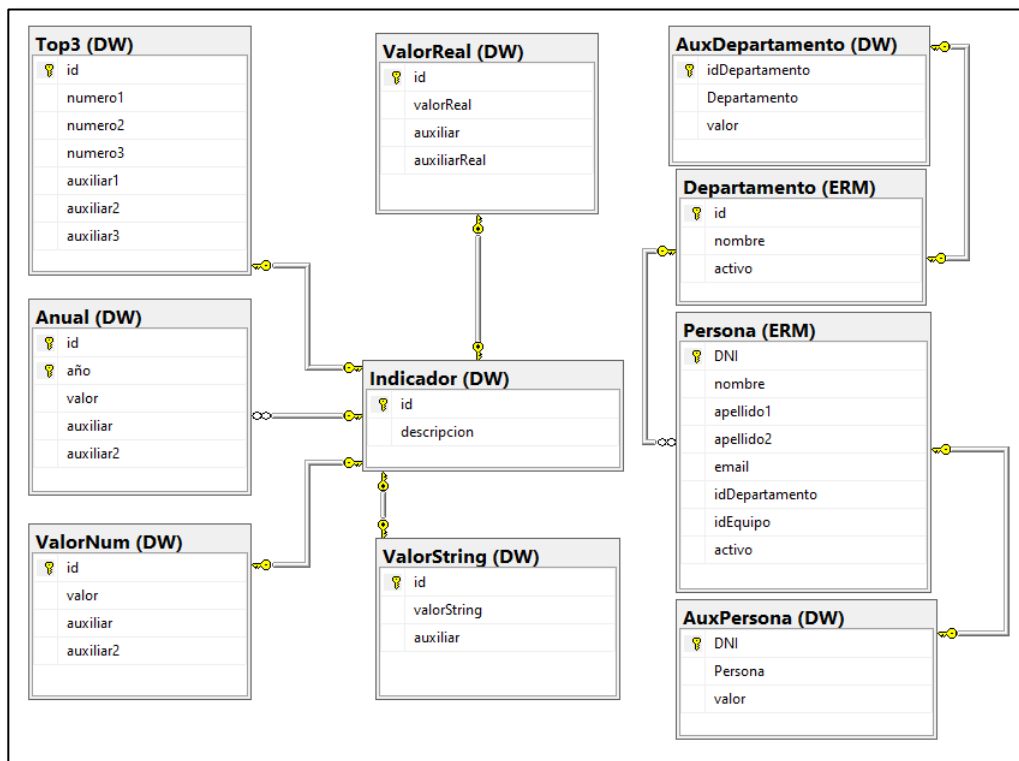


Tabla 21. Definición de tablas que corresponden al bloque DW

Indicador DW

Descripción	Tipo	Null	Rest	Observaciones
id	int	Not Null	PK	
descripcion	varchar(4000)	Not Null		

ValorNum DW

Descripción	Tipo	Null	Rest	Observaciones
-------------	------	------	------	---------------

id	int	Not Null	PK,FK	Con la tabla Indicador
valor	int	Not Null		
auxiliar	int	Not Null		
auxiliar2	int	Not Null		

Anual DW

Descripción	Tipo	Null	Rest	Observaciones
id	int	Not Null	PK,FK	Con la tabla Indicador
año	int	Not Null		
valor	float	Not Null		
auxiliar	int	Not Null		
auxiliar2	int	Not Null		

ValorString DW

Descripción	Tipo	Null	Rest	Observaciones
id	int	Not Null	PK,FK	Con la tabla Indicador
nombre	varchar(200)	Not Null		
auxiliar	int	Not Null		

Top3 DW

Descripción	Tipo	Null	Rest	Observaciones
id	int	Not Null	PK,FK	Con la tabla Indicador
numero1	varchar(200)	Not Null		
numero2	varchar(200)	Not Null		
numero3	varchar(200)	Not Null		
auxiliar1	int	Not Null		
auxiliar2	int	Not Null		
auxiliar3	int	Not Null		

ValorReal DW

Descripción	Tipo	Null	Rest	Observaciones
id	int	Not Null	PK,FK	Con la tabla Indicador
valorReal	decimal(9,3)	Not Null		
auxiliar	int	Not Null		
auxiliarReal	decimal(9,3)	Not Null		

AuxDepartamento DW

Descripción	Tipo	Null	Rest	Observaciones
idDepartamento	int	Not Null	PK,FK	Con la tabla Departamento
Departamento	varchar(50)	Not Null		
valor	int	Not Null		

AuxPersona DW

Descripción	Tipo	Null	Rest	Observaciones
DNI	varchar(10)	Not Null	PK,FK	Con la tabla Persona
Persona	varchar(200)	Not Null		
valor	int	Not Null		

5. Implementación

Durante esta fase deberemos de crear todos los elementos necesarios para poder trabajar con la base de datos diseñada en el apartado anterior. He decidido restringirme a los procedimientos ABM (Alta + Baja + Modificación), aunque lo

lógico es que también se decidiera crear las operaciones de consulta. De este modo se podría realizar un correcto aislamiento de la base de datos. Lo que nos permitiría asegurar su consistencia y el correcto rendimiento del sistema. Dado que en caso de ser necesario podríamos realizar optimizaciones sobre las consultas. Ya que de otro modo quedarían fuera de nuestro ámbito de trabajo.

Para poder cubrir de manera correcta las 27 tablas creadas me surgen un total de 60 procedimientos a desarrollar con sus correspondientes pruebas. Este número es menor que el esperado debido a que existen tablas sobre las que no se creará ningún procedimiento. Como en el caso de las tablas que representan enumerados (ECriticidadRiesgo y Eimportancia). O tablas como en el caso de Campaña, que representa una generalización de tipo completa. Por lo que solo permitiré realizar las operaciones ABM sobre sus hijos y serán estos los que realicen las modificaciones oportunas sobre el padre. Conforme se ha ido desarrollando esta fase he visto que realizar la cantidad total de procedimientos necesarios excedería el tiempo planificado para la entrega. Comentado esta circunstancia con el profesor surgían dos caminos. Por un lado, la posibilidad de alargar la entrega y, por otro, acotar la cantidad de procedimientos. Siguiendo las indicaciones del profesor he tenido que reducir la cantidad de procedimientos a desarrollar. Por lo que, he decidido no realizar todos los procedimientos asociados a las nuevas tablas agregadas para extender la funcionalidad y que no se recogen en el enunciado del problema. No desarrollar estos procedimientos en una primera fase no perjudicará a “la empresa” dado que no estaban recogidos en los requerimientos. Y por otra parte son bastante simples y solo tienen como intención extender la información que se recoge de los riesgos. Por tanto, a nivel funcional no tendrán ningún tipo de implicación el desarrollarlos más adelante. Las tablas afectadas serán: Causa, RiesgoCausa, Consecuencia, RiesgoConsecuencia, Incidente, Operativa y DepartamentoOperativa. Posponiendo la realización de este bloque evitaré implementar en esta fase 18 de los procedimientos. Por lo que el desarrollo final solo deberá cubrir 42 del total inicial.

Para la creación de estos procedimientos la primera tarea que realice fue la creación de una serie de procedimientos modelos sobre cada una de las operaciones. Estandarizando así el uso de las transacciones e integrando de manera homogénea la lógica del control de errores. Constan de una primera parte donde se realiza una validación de los parámetros de entrada de la operación y si no se produce ningún error se ejecuta una segunda parte donde se realiza la operación solicitada. Todas las modificaciones sobre los datos son realizadas dentro de una transacción y si se produce un error esta se deshace. Por defecto he utilizado un criterio de bloqueo optimista sobre las tablas incluyendo la cláusula de (*NoLock*) en las operaciones. En ninguno de los procedimientos implementados así como de los scripts entregados se hace servir el uso de cursores. Dado que de partida considero una mala praxis su utilización.

El discriminado dos partes en esta fase. Una primera ERM que se encargará de la gestión de las operaciones sobre la base de datos y su materia de trabajo serán los procedimientos. Y una segunda DW que tratará sobre el mantenimiento de los indicadores. En esta parte la mayoría del trabajo se desarrollará a través de

la implementación de disparadores. Aunque también se incluirá un procedimiento encargado de realizar el cierre del año. Que deberá de programarse su ejecución en la temporización más inicial posible en el primer día del año. Pasemos a tratar al primero de los bloques.

5.1 ERM

De partida una de las primeras decisiones que tome a la hora de implementar las operaciones es no realizar una baja física de las entidades principales sino una baja lógica. Admitiendo solo el borrado en dos tablas que mantienen relaciones. Esto me obliga a incluir una serie de campos que he denominado “activo” que en principio no estaban recogidos durante la fase de diseño. De este modo mejoraremos la trazabilidad del sistema que es uno de los requerimientos del enunciado. He optado por desarrollar una programación defensiva implementando una lógica lo más simple posible, que a mi modo de ver es la que será más fácil de gestionar. La mayoría de las operaciones afectan a un solo elemento o a los elementos de su ámbito. Es decir, si tengo que crear un departamento y dar de alta al conjunto de personas que lo componen. Se deberá de realizar una llamada a `uspDepartamentoAlta` y tantas como sea necesarias al procedimiento `uspPersonaAlta`. Por tanto, se evita también realizar operaciones en cascada sobre los elementos y se opta por delegar estas operaciones al cliente. Generando mensajes de error en el caso de que alguna de las operaciones no sea posible. Más adelante si es necesario siempre se podrán crear procedimientos más específicos para ciertas operaciones si se considera necesario. La única excepción a esta regla es el procedimiento de `uspRiesgoAlta`. Para cumplir con lo indicado en los requerimientos este procedimiento creará tanto un nuevo Riesgo como una Acción y en caso de ser necesario la verificación y detección oportuna. Dado que es uno de los requerimientos del sistema.

Por otra parte, para todas las tablas se han creado las tres operaciones posibles ABM (Alta + Baja + Modificación). Salvo en dos casos. El primero de ellos es sobre la tabla de relación `CategoriaDepartamento` esta tabla solo contiene dos campos que se corresponden con el identificador de categoría y el de departamento. En este caso he obviado la operación de modificación. Como norma general esta operación no ocurrirá dado que implicaría la reasignación de una categoría a un departamento diferente o viceversa. Por tanto, la evito dado que creo que puede ser fuente de errores. En cualquier caso de ser necesaria, que no lo creo, siempre se podrá realizar una operación de borrado y una posterior inserción. El segundo de ellos es en la tabla de detección. Esta tabla almacena 3 indicadores `idDepartamento`, `idCampaña` e `idRiesgo`. Por el mismo caso que en la anterior no he desarrollado la operación de modificación. Y para este caso particular no he desarrollado ninguna operación de borrado. Dado que ocasionaría que un riesgo pudiera estar dado de alta con origen una campaña quedara sin una vinculación a ella. En el caso de que se realizará el alta por error siempre se podría dar de baja el riesgo implicado y su correspondiente acción. Me ha parecido menos mala esta opción que admitir el borrado.

Veamos la lista de procedimientos y de paso indicaremos el número de prueba unitaria asociada.

Tabla 22. Definición de procedimientos y pruebas unitarias

SP-001		uspDepartamentoAlta	
	Descripción	Da de alta un departamento	
	Parámetros (RSP):	@nombre varchar(50), @idUser varchar(10)	
	Precondición (Sino retorna RSP="KO" y deja traza en LOG):		Prueba
	El @nombre debe de tener un valor		SP001.TS001
	El @nombre no puede existir		SP001.TS002
	El @idUser debe de existir y estar activo		SP001.TS003
	Resultado (Retorna RSP="OK"):		Prueba
	Se da de alta con los datos indicados por los parámetros. Retorna además el nuevo ID		SP001.TS004
SP-002		uspDepartamentoBaja	
	Descripción	Realiza la baja lógica del departamento	
	Parámetros (RSP):	@idDepartamento int, @idUser varchar(10)	
	Precondición (Sino retorna RSP="KO" y deja traza en LOG):		Prueba
	El @idDepartamento debe de existir		SP002.TS005
	No puede haber ninguna persona ni equipo activo con el departamento indicado		SP002.TS006
	El @idUser debe de existir y estar activo		SP002.TS007
	Resultado (Retorna RSP="OK"):		Prueba
	Campo Activo=0		SP002.TS008
SP-003		uspDepartamentoModificacion	
	Descripción	Se modifica el nombre del departamento	
	Parámetros (RSP):	@idDepartamento int, @nombre varchar(50)	
	Precondición (Sino retorna RSP="KO" y deja traza en LOG):		Prueba
	El @idDepartamento debe de existir		SP003.TS009
	El @nombre debe de tener un valor		SP003.TS010
	El @nombre nuevo no puede existir		SP003.TS011
	El @idUser debe de existir y estar activo		SP003.TS012
	Resultado (Retorna RSP="OK"):		Prueba
	Se modifica según los datos indicados en los parámetros		SP003.TS013
SP-004		uspEquipoAlta	
	Descripción	Se da de alta un nuevo equipo en el sistema.	
	Parámetros (RSP):	@descripcion varchar(50), @idDepartamento int, @idUser varchar(10)	
	Precondición (Sino retorna RSP="KO" y deja traza en LOG):		Prueba
	El @descripcion no puede existir		SP004.TS014
	El @descripcion debe de tener un valor		SP004.TS015
	El @idDepartamento debe existir y estar activo		SP004.TS016
	El @idUser debe de existir y estar activo		SP004.TS017
	Resultado (Retorna RSP="OK"):		Prueba
	Se da de alta con los datos indicados por los parámetros. Retorna además el nuevo ID		SP004.TS018
SP-005		uspEquipoBaja	
	Descripción	Realiza la baja lógica del equipo	
	Parámetros (RSP):	@idEquipo int, @idUser varchar(10)	
	Precondición (Sino retorna RSP="KO" y deja traza en LOG):		Prueba
	El @idEquipo debe de existir		SP005.TS019
	No puede haber ninguna persona en el equipo indicado		SP005.TS020
	El @idUser debe de existir y estar activo		SP005.TS021
	Resultado (Retorna RSP="OK"):		Prueba
	Campo Activo=0		SP005.TS022
SP-006		uspEquipoModificacion	
	Descripción	Se modifica los datos del equipo	
	Parámetros (RSP):	@idEquipo int, @descripcion varchar(50), @idDepartamento int, @idUser varchar(10)	
	Precondición (Sino retorna RSP="KO" y deja traza en LOG):		Prueba
	El @idEquipo debe de existir		SP006.TS023
	El @descripcion debe de tener un valor		SP006.TS024

	El @descripcion no puede existir	SP006.TS025
	El @idDepartamento debe existir y estar activo	SP006.TS026
	El @idUserario debe de existir y estar activo	SP006.TS027
	Resultado (Retorna RSP="OK"):	Prueba
	Se modifica según los datos indicados en los parámetros	SP006.TS028
SP-007 uspPersonaAlta		
	Descripción	Da de alta un persona
	Parámetros (RSP):	@DNI varchar(10), @nombre varchar(50), @apellido1 varchar(50), @apellido2 varchar(50), @email varchar(50), @idDepartamento int, @idEquipo int, @idUserario varchar(10)
	Precondición (Sino retorna RSP="KO" y deja traza en LOG):	Prueba
	El @DNI no puede existir y debe de cumplir las reglas de validación	SP007.TS029
	Debe indicarse un @nombre, @apellido1,@apellido2	SP007.TS030
	El @email debe existir y cumplir las reglas de validación	SP007.TS031
	El @idDepartamento debe de existir	SP007.TS032
	El @ idEquipo de indicarse debe de existir	SP007.TS033
	El @idUserario debe de existir y estar activo	SP007.TS034
	Resultado (Retorna RSP="OK"):	Prueba
	Se da de alta con los datos indicados por los parámetros. Retorna además el nuevo ID	SP007.TS035
SP-008 uspPersonaBaja		
	Descripción	Realiza la baja lógica de la persona
	Parámetros (RSP):	@DNI varchar(10) , @idUserario varchar(10)
	Resultado (Retorna RSP="OK"):	Prueba
	El @DNI debe de existir	SP008.TS036
	No puede ser el responsable de ninguna categoría activa	SP008.TS037
	No puede ser el responsable de ninguna acción activa	SP008.TS038
	No puede ser el líder de una autoevaluación activa	SP008.TS039
	El @idUserario debe de existir y estar activo	SP008.TS040
	Resultado: (Retorna RSP="OK")	Prueba
	Campo Activo=0	SP008.TS041
SP-009 uspPersonaModificacion		
	Descripción	Se modifica los datos de la persona indicada
	Parámetros (RSP):	@DNI varchar(10), @nombre varchar(50), @apellido1 varchar(50), @apellido2 varchar(50), @email varchar(50), @idDepartamento int, @idEquipo int, @idUserario varchar(10)
	Resultado (Retorna RSP="OK"):	Prueba
	El @DNI debe de existir	SP009.TS042
	Debe indicarse un @nombre, @apellido1, @apellido2	SP009.TS043
	El @email debe existir y cumplir las reglas de validación	SP009.TS044
	El @idDepartamento debe de existir	SP009.TS045
	El @ idEquipo de indicarse debe de existir	SP009.TS046
	El @idUserario debe de existir y estar activo	SP009.TS047
	Resultado (Retorna RSP="OK"):	Prueba
	Se modifica según los datos indicados en los parámetros	SP009.TS048
SP-010 uspCategoriaAlta		
	Descripción	Da de alta una categoría y le crea una importancia asociada
	Parámetros (RSP):	@descripcion varchar(50), @importancia int, @idResponsable varchar(10), @idUserario varchar(10)
	Precondición (Sino retorna RSP="KO" y deja traza en LOG):	Prueba
	El @descripcion debe tener un valor	SP010.TS049
	La @importancia debe de existir	SP010.TS050
	El @idResponsable debe de existir y estar activo	SP010.TS051
	El @idUserario debe de existir y estar activo	SP010.TS052
	Resultado (Retorna RSP="OK"):	Prueba
	Se da de alta con los datos indicados por los parámetros. Retorna además el nuevo ID	SP010.TS053
SP-011 uspCategoriaBaja		
	Descripción	Realiza la baja lógica de la categoría

	Parámetros (RSP):	@idCategoria int, @idUserario varchar(10)	
	Precondición (Sino retorna RSP="KO" y deja traza en LOG):		Prueba
	El @idCategoria debe de existir		SP011.TS054
	El @idUserario debe de existir y estar activo		SP011.TS055
	Resultado (Retorna RSP="OK"):		Prueba
	Campo Activo=0		SP011.TS056
SP-012	uspModificacion		
	Descripción	Se modifica los datos de la categoría y se recalculan el impacto de los riesgos asociados a esa categoría	
	Parámetros (RSP):	@idCategoria int, @descripcion varchar(50), @importancia int, @idResponsable varchar(10), @idUserario varchar(10)	
	Precondición (Sino retorna RSP="KO" y deja traza en LOG):		Prueba
	El @idCategoria debe de existir		SP012.TS057
	El @descripcion debe tener un valor		SP012.TS058
	La @importancia debe de existir		SP012.TS059
	El @idResponsable debe de existir y estar activo		SP012.TS060
	El @idUserario debe de existir y estar activo		SP012.TS061
	Resultado (Retorna RSP="OK"):		Prueba
	Se modifica según los datos indicados en los parámetros		SP012.TS062
SP-013	uspCategoriaDepartamentoAlta		
	Descripción	Da de alta una relación entre departamento y categoría	
	Parámetros (RSP):	@idCategoria int, @idDepartamento int, @idUserario varchar(10)	
	Precondición (Sino retorna RSP="KO" y deja traza en LOG):		Prueba
	El @idCategoria debe de existir		SP013.TS063
	El @idDepartamento debe de existir		SP013.TS064
	El departamento no puede estar ya asociado a la categoría		SP013.TS065
	El @idUserario debe de existir y estar activo		SP013.TS066
	Resultado (Retorna RSP="OK"):		Prueba
	Se da de alta con los datos indicados por los parámetros. Retorna además el nuevo ID		SP013.TS067
SP-014	uspCategoriaDepartamentoBaja		
	Descripción	Realiza el borrado físico de la relación entre departamento y categoría	
	Parámetros (RSP):	@idCategoria int, @idDepartamento int, @idUserario varchar(10)	
	Precondición (Sino retorna RSP="KO" y deja traza en LOG):		Prueba
	Debe existir con elemento con el @idCategoria y @idDepartamento indicados		SP014.TS068
	El @idUserario debe de existir y estar activo		SP014.TS069
	Resultado (Retorna RSP="OK"):		Prueba
	El registro es borrado		SP014.TS070
SP-015	uspAccionAlta		
	Descripción	Da de alta una acción asociada a un riesgo. Si es necesario recalcula el estado del riesgo	
	Parámetros (RSP):	@descripcion varchar(500), @objetivo varchar(100), @fechaEstimada datetime, @estado varchar(30), @idRiesgo int, @idResponsable varchar(10), @idUserario varchar(10)	
	Precondición (Sino retorna RSP="KO" y deja traza en LOG):		Prueba
	Los campos @descripcion, @objetivo y @estado son obligatorios		SP015.TS071
	El @objetivo debe de ser: corregir o mitigar		SP015.TS072
	El @estado debe ser: definida, enCurso, implementadaRiesgoCorregido, implementadaRiesgoMitigado o descartada		SP015.TS073
	El @idRiesgo debe de existir		SP015.TS074
	El @idResponsable debe de existir y estar activo		SP015.TS075
	El @idUserario debe de existir y estar activo		SP015.TS076
	Resultado (Retorna RSP="OK"):		Prueba
	Se da de alta con los datos indicados por los parámetros. Retorna además el nuevo ID		SP015.TS077
SP-016	uspAccionBaja		
	Descripción	Realiza la baja lógica de la acción indicada. Si es necesario recalcula el estado del riesgo	
	Parámetros (RSP):	@idAccion int, @idUserario varchar(10)	

Precondición (Sino retorna RSP="KO" y deja traza en LOG):		Prueba
La @idAccion debe de existir		SP016.TS078
El @idUserario debe de existir y estar activo		SP016.TS079
No se puede dar de baja una acción si el riesgo asociado queda sin acciones activas		SP016.TS080
Resultado (Retorna RSP="OK"):		Prueba
Campo Activo=0		SP016.TS081
SP-017	uspModificacion	
Descripción	Se modifica los datos de un riesgo asociado a una acción. Si es necesario recalcula el estado del riesgo	
Parámetros (RSP):	@idAccion int, @descripcion varchar(500), @objetivo varchar(100), @fechaEstimada datetime, @estado varchar(30), @idResponsable varchar(10)	
Precondición (Sino retorna RSP="KO" y deja traza en LOG):		Prueba
La @idAccion debe de existir		SP017.TS082
Los campos @descripcion, @objetivo y @estado son obligatorios		SP017.TS083
El @objetivo debe de ser: corregir o mitigar		SP017.TS084
El @estado debe ser: definida, enCurso, implementadaRiesgoCorregido, implementadaRiesgoMitigado o descartada		SP017.TS085
No se puede descartar esta acción dado que el riesgo quedaría sin acciones		SP017.TS086
El @idResponsable debe de existir y estar activo		SP017.TS087
El @idUserario debe de existir y estar activo		SP017.TS088
Resultado (Retorna RSP="OK"):		Prueba
Se modifica según los datos indicados en los parámetros		SP017.TS089
SP-018	uspRiesgoAlta	
Descripción	Da de alta un riesgo y una acción asociada a él. Si no se indica un departamento y una campaña, se crea con origen gestionRiesgos. Si se indica según el tipo de campaña se asocia el origen oportuno. En este caso también crea la detección y si es necesario la verificación	
Parámetros (RSP):	@nombre varchar(50), @descripcion varchar(500), @idCriticidad int, @probabilidad decimal(4,3), @idCategoria int, @idDepartamento int = 0, @idCampaña int = 0, @accionDescripcion varchar(500), @accionObjetivo varchar(100), @accionFechaEstimada datetime, @accionIdResponsable varchar(10), @idUserario varchar(10)	
Precondición (Sino retorna RSP="KO" y deja traza en LOG):		Prueba
Los campos @nombre, @descripcion, @idCriticidad, @idCategoria, @accionDescripcion y @accionObjetivo son obligatorios		SP018.TS090
@idCriticidad debe de existir		SP018.TS091
@idCategoria debe de existir		SP018.TS092
El @accionIdResponsable debe de existir y estar activo		SP018.TS093
El @idUserario debe de existir y estar activo		SP018.TS094
El @idDepartamento debe de existir		SP018.TS095
El @idCampaña debe de existir		SP018.TS096
Si el riesgo está asociado a una Campaña debe indicar un identificador de Campaña y de Departamento		SP018.TS097
La @probabilidad debe de estar comprendida [0,1]		SP018.TS098
Resultado (Retorna RSP="OK"):		Prueba
Se da de alta con los datos indicados por los parámetros. Si no se indica un departamento y una campaña, se crea con origen gestionRiesgos. Retorna además el nuevo ID		SP018.TS099
Se da de alta con los datos indicados por los parámetros. Si se indica el departamento y la campaña con el origen igual a la campaña y se crea la detección. Retorna además el nuevo ID		SP018.TS100
Se da de alta con los datos indicados por los parámetros. Si se indica el departamento y la campaña con el origen igual a la campaña y se crea la detección. Si no existe además se da de alta también la verificación. Retorna además el nuevo ID		SP018.TS101
SP-019	uspRiesgoBaja	
Descripción	Realiza la baja lógica del riesgo	
Parámetros (RSP):	@idRiesgo int, @idUserario varchar(10)	
Precondición (Sino retorna RSP="KO" y deja traza en LOG):		Prueba
El @idRiesgo debe de existir		SP019.TS102

	El @idUserario debe de existir y estar activo	SP019.TS103
	Resultado (Retorna RSP="OK"):	Prueba
	Campo Activo=0	SP019.TS104
SP-020	uspRiesgoModificacion	
	Descripción	Se modifica los datos del riesgo si es necesario se recalcula su importancia
	Parámetros (RSP):	@idRiesgo int, @nombre varchar(50), @descripcion varchar(500), @idCriticidad int, @probabilidad decimal(4,3), @idCategoria int, @idUserario varchar(10)
	Precondición (Sino retorna RSP="KO" y deja traza en LOG):	Prueba
	El @idRiesgo debe de existir	SP020.TS105
	Los @campos nombre, @descripcion, @idCriticidad y @idCategoria son obligatorios	SP020.TS106
	La @probabilidad debe de estar comprendida [0,1]	SP020.TS107
	La @Criticidad debe de existir	SP020.TS108
	La @Categoria debe de existir	SP020.TS109
	El @idUserario debe de existir y estar activo	SP020.TS110
	Resultado (Retorna RSP="OK"):	Prueba
	Se modifica según los datos indicados en los parámetros	SP020.TS111
SP-021	uspAuditoriaExternaAlta	
	Descripción	Da de alta un campaña y una auditoría externa asociada a ella
	Parámetros (RSP):	@nombre varchar(100), @fechaInicio datetime, @fechaFin datetime, @empresa varchar(100), @coste decimal(9,3), @resultados text, @idUserario varchar(10)
	Precondición (Sino retorna RSP="KO" y deja traza en LOG):	Prueba
	Los campos @nombre, @fechaInicio, @fechaFin, @empresa y @coste son obligatorios	SP021.TS112
	El @idUserario debe de existir y estar activo	SP021.TS113
	Resultado (Retorna RSP="OK"):	Prueba
	Se da de alta con los datos indicados por los parámetros. Retorna además el nuevo ID	SP021.TS114
SP-022	uspAuditoriaExternaBaja	
	Descripción	Realiza la baja lógica de la campaña asociada a la auditoría externa
	Parámetros (RSP):	@idAuditoriaExterna int, @idUserario varchar(10)
	Precondición (Sino retorna RSP="KO" y deja traza en LOG):	Prueba
	La @idAuditoriaExterna debe de existir	SP022.TS115
	El @idUserario debe de existir y estar activo	SP022.TS116
	Resultado (Retorna RSP="OK"):	Prueba
	Campo Activo=0 de la campaña	SP022.TS117
SP-023	uspAuditoriaExternaModificacion	
	Descripción	Se modifica los datos de la auditoría externa y de su campaña asociada
	Parámetros (RSP):	@idAuditoriaExterna int, @nombre varchar(100), @fechaInicio datetime, @fechaFin datetime, @empresa varchar(100), @coste decimal(9,3), @resultados text, @idUserario varchar(10)
	Precondición (Sino retorna RSP="KO" y deja traza en LOG):	Prueba
	Los campos @nombre, @fechaInicio, @fechaFin, @empresa y @coste son obligatorios	SP023.TS118
	La @idAuditoriaExterna debe de existir	SP023.TS119
	El @idUserario debe de existir y estar activo	SP023.TS120
	Resultado (Retorna RSP="OK"):	Prueba
	Se modifica según los datos indicados en los parámetros en la campaña o en la auditoría externa	SP023.TS121
SP-024	uspAuditorialInternaAlta	
	Descripción	Da de alta un campaña y una auditoría interna asociada a ella
	Parámetros (RSP):	@nombre varchar(100), @fechaInicio datetime, @fechaFin datetime, @objetivo varchar(500), @idUserario varchar(10)
	Precondición (Sino retorna RSP="KO" y deja traza en LOG):	Prueba
	Los campos @nombre, @fechaInicio, @fechaFin y @objetivo son obligatorios	SP024.TS122
	El @idUserario debe de existir y estar activo	SP024.TS123
	Resultado (Retorna RSP="OK"):	Prueba
	Se da de alta con los datos indicados por los parámetros. Retorna además el nuevo ID	SP024.TS124
SP-025	uspAuditorialInternaBaja	

	Descripción	Realiza la baja lógica de la campaña asociada a la auditoría interna	
	Parámetros (RSP):	@idAuditoriaInterna int, @idUser varchar(10)	
	Precondición (Sino retorna RSP="KO" y deja traza en LOG):		Prueba
		El @idAuditoriaInterna debe de existir	SP025.TS125
		El @idUser debe de existir y estar activo	SP025.TS126
	Resultado (Retorna RSP="OK"):		Prueba
		Campo Activo=0	SP025.TS127
SP-026	uspAuditoriaInternaModificacion		
	Descripción	Se modifica los datos de la auditoría interna y de su campaña asociada	
	Parámetros (RSP):	@idAuditoriaInterna int, @nombre varchar(100), @fechaInicio datetime, @fechaFin datetime, @objetivo varchar(500), @idUser varchar(10)	
	Precondición (Sino retorna RSP="KO" y deja traza en LOG):		Prueba
		Los campos @nombre, @fechaInicio, @fechaFin y @objetivo son obligatorios	SP026.TS128
		El @idAuditoriaInterna debe de existir	SP026.TS129
		El @idUser debe de existir y estar activo	SP026.TS130
	Resultado (Retorna RSP="OK"):		Prueba
		Se modifica según los datos indicados en los parámetros	SP026.TS131
SP-027	uspAutoevaluacionAlta		
	Descripción	Da de alta un campaña y una autoevaluación asociada a ella	
	Parámetros (RSP):	@nombre varchar(100), @fechaInicio datetime, @fechaFin datetime, @idLider varchar(10), @idUser varchar(10)	
	Precondición (Sino retorna RSP="KO" y deja traza en LOG):		Prueba
		Los campos @nombre, @fechaInicio y @fechaFin son obligatorios	SP027.TS132
		El @idLider debe de existir y estar activo	SP027.TS133
		El @idUser debe de existir y estar activo	SP027.TS134
	Resultado (Retorna RSP="OK"):		Prueba
		Se da de alta con los datos indicados por los parámetros. Retorna además el nuevo ID	SP027.TS135
SP-028	uspAutoevaluacionBaja		
	Descripción	Realiza la baja lógica de la campaña asociada a la autoevaluación	
	Parámetros (RSP):	@idAutoevaluacion int, @idUser varchar(10)	
	Precondición (Sino retorna RSP="KO" y deja traza en LOG):		Prueba
		El @idAutoevaluacion debe de existir	SP028.TS136
		El @idUser debe de existir y estar activo	SP028.TS137
	Resultado (Retorna RSP="OK"):		Prueba
		Campo Activo=0	SP028.TS138
SP-029	uspAutoevaluacionModificacion		
	Descripción	Se modifica los datos de la autoevaluación y de su campaña asociada	
	Parámetros (RSP):	@idAutoevaluacion int, @nombre varchar(100), @fechaInicio datetime, @fechaFin datetime, @idLider varchar(10), @idUser varchar(10)	
	Precondición (Sino retorna RSP="KO" y deja traza en LOG):		Prueba
		Los campos @nombre, @fechaInicio y @fechaFin son obligatorios	SP029.TS139
		El @idAutoevaluacion debe de existir	SP029.TS140
		El @idLider debe de existir y estar activo	SP029.TS141
		El @idUser debe de existir y estar activo	SP029.TS142
	Resultado (Retorna RSP="OK"):		Prueba
		Se modifica según los datos indicados en los parámetros	SP029.TS143
SP-030	uspControlAlta		
	Descripción	Da de alta un control vinculado a una acción o una autoevaluación	
	Parámetros (RSP):	@informacion text, @idAccion int, @idAutoevaluacion int, @idUser varchar(10)	
	Precondición (Sino retorna RSP="KO" y deja traza en LOG):		Prueba
		@informacion debe tener valor	SP030.TS144
		Un control debe de estar vinculado a una acción o a una autoevaluación	SP030.TS145
		@idAccion debe de existir	SP030.TS146
		@idAutoevaluacion debe de existir	SP030.TS147
		El @idUser debe de existir y estar activo	SP030.TS148

	Resultado (Retorna RSP="OK"):	Prueba
	Se da de alta con los datos indicados por los parámetros. Retorna además el nuevo ID	SP030.TS149
SP-031	uspControlBaja	
	Descripción	Realiza la baja lógica del control
	Parámetros (RSP):	@idControl int, @idUserario varchar(10)
	Precondición (Sino retorna RSP="KO" y deja traza en LOG):	Prueba
	El @idControl debe de existir	SP031.TS150
	El @idUserario debe de existir y estar activo	SP031.TS151
	Resultado (Retorna RSP="OK"):	Prueba
	Campo Activo=0	SP031.TS152
SP-032	uspControlModificacion	
	Descripción	Se modifica los datos del control
	Parámetros (RSP):	@idControl int, @informacion text, @idAccion int, @idAutoevaluacion int, @idUserario varchar(10)
	Precondición (Sino retorna RSP="KO" y deja traza en LOG):	Prueba
	@informacion debe tener valor	SP032.TS153
	El @idControl debe de existir	SP032.TS154
	Un control debe de estar vinculado a una acción o a una autoevaluación	SP032.TS155
	@idAccion debe de existir	SP032.TS156
	@idAutoevaluacion debe de existir	SP032.TS157
	El @idUserario debe de existir y estar activo	SP032.TS158
	Resultado (Retorna RSP="OK"):	Prueba
	Se modifica según los datos indicados en los parámetros	SP032.TS159
SP-033	uspVerificacionAlta	
	Descripción	Da de alta una verificación asociada a una campaña
	Parámetros (RSP):	@idDepartamento int, @idCampaña int, @fechalnicio datetime, @fechaFin datetime, @idUserario varchar(10)
	Precondición (Sino retorna RSP="KO" y deja traza en LOG):	Prueba
	El @idDepartamento debe de existir	SP033.TS161
	La @idCampaña debe de existir	SP033.TS162
	La @fechalnicio no puede ser superior a la fecha de fin de la campaña	SP033.TS163
	La @fechaFin no puede ser anterior a la fecha de inicio de la campaña	SP033.TS164
	El @idUserario debe de existir y estar activo	SP033.TS165
	Resultado (Retorna RSP="OK"):	Prueba
	Se da de alta con los datos indicados por los parámetros. Retorna además el nuevo ID	SP033.TS166
SP-034	uspVerificacionBaja	
	Descripción	Realiza el borrado de la verificación
	Parámetros (RSP):	@idDepartamento int, @idCampaña int, @idUserario varchar(10)
	Precondición (Sino retorna RSP="KO" y deja traza en LOG):	Prueba
	Debe existir una verificación asociada al departamento y campaña indicados	SP034.TS167
	No se puede eliminar la verificación porque tiene una detección asociada	SP034.TS168
	El @idUserario debe de existir y estar activo	SP034.TS169
	Resultado (Retorna RSP="OK"):	Prueba
	Elimina la verificación	SP034.TS170
SP-035	uspVerificacionModificacion	
	Descripción	Se modifica la fecha de inicio o de fin de la verificación indicada
	Parámetros (RSP):	@idDepartamento int, @idCampaña int, @fechalnicio datetime, @fechaFin datetime, @idUserario varchar(10)
	Precondición (Sino retorna RSP="KO" y deja traza en LOG):	Prueba
	Debe existir una verificación asociada al departamento y campaña indicados. Error departamento.	SP035.TS171
	Debe existir una verificación asociada al departamento y campaña indicados. Error campaña.	SP035.TS172
	La @fechalnicio no puede ser superior a la fecha de fin de la campaña	SP035.TS173
	La @fechaFin no puede ser anterior a la fecha de inicio de la campaña	SP035.TS174
	El @idUserario debe de existir y estar activo	SP035.TS175

	Resultado (Retorna RSP="OK"):	Prueba
	Se modifica según los datos indicados en los parámetros	SP035.TS176
SP-036	uspResultadoAlta	
	Descripción	Da de alta un muestreo y el resultado en el sistema
	Parámetros (RSP):	@idAutoevaluacion int, @idControl int, @informacion text, @idUserario varchar(10)
	Precondición (Sino retorna RSP="KO" y deja traza en LOG):	Prueba
	Debe de indicar una información que describa el resultado	SP036.TS177
	La @idAutoevaluacion debe de existir	SP036.TS178
	El @idControl debe de existir	SP036.TS179
	Ya existe un resultado asociado a esa autoevaluación y control	SP036.TS180
	El @idUserario debe de existir y estar activo	SP036.TS181
	Resultado (Retorna RSP="OK"):	Prueba
	Se da de alta con los datos indicados por los parámetros. Retorna además el nuevo ID	SP036.TS182
SP-037	uspResultadoBaja	
	Descripción	Realiza la baja lógica del muestreo asociado al resultado
	Parámetros (RSP):	@idResultado int, @idUserario varchar(10)
	Precondición (Sino retorna RSP="KO" y deja traza en LOG):	Prueba
	El @idResultado debe de existir	SP037.TS183
	El @idUserario debe de existir y estar activo	SP037.TS184
	Resultado (Retorna RSP="OK"):	Prueba
	Campo Activo=0	SP037.TS185
SP-038	uspResultadoModificacion	
	Descripción	Se modifica la información asociada al resultado
	Parámetros (RSP):	@idResultado int, @informacion text, @idUserario varchar(10)
	Precondición (Sino retorna RSP="KO" y deja traza en LOG):	Prueba
	Debe de indicar una información que describa el resultado	SP038.TS186
	El @idResultado debe de existir	SP038.TS187
	El @idUserario debe de existir y estar activo	SP038.TS188
	Resultado (Retorna RSP="OK"):	Prueba
	Se modifica según los datos indicados en los parámetros	SP038.TS189
SP-039	uspRevisionAlta	
	Descripción	Da de alta un muestreo y una revisión asociada al muestreo
	Parámetros (RSP):	@resultado text, @idAuditoria int, @idUserario varchar(10)
	Precondición (Sino retorna RSP="KO" y deja traza en LOG):	Prueba
	Debe de indicar un @resultado de la revisión	SP039.TS190
	@idAuditoria no reconocida	SP039.TS191
	Una revisión solo puede ser asociada a una Auditoria de tipo interna o externa	SP039.TS192
	El @idUserario debe de existir y estar activo	SP039.TS193
	Resultado (Retorna RSP="OK"):	Prueba
	Se da de alta con los datos indicados por los parámetros. Retorna además el nuevo ID	SP039.TS194
SP-040	uspRevisionBaja	
	Descripción	Realiza la baja lógica del muestreo asociado a la revisión
	Parámetros (RSP):	@idRevision int, @idUserario varchar(10)
	Precondición (Sino retorna RSP="KO" y deja traza en LOG):	Prueba
	El @idRevision debe de existir	SP040.TS195
	El @idUserario debe de existir y estar activo	SP040.TS196
	Resultado (Retorna RSP="OK"):	Prueba
	Campo Activo=0	SP040.TS197
SP-041	uspRevisionModificacion	
	Descripción	Se modifica el resultado de la revisión
	Parámetros (RSP):	@idRevision int, @resultado text, @idUserario varchar(10)
	Precondición (Sino retorna RSP="KO" y deja traza en LOG):	Prueba
	Debe de indicar un @resultado de la revisión	SP041.TS198
	El @idRevision debe de existir	SP041.TS199

	El @idUserario debe de existir y estar activo	SP041.TS200
	Resultado (Retorna RSP="OK"):	Prueba
	Se modifica según los datos indicados en los parámetros	SP041.TS201
SP-042	uspDeteccionAlta	
	Descripción	Da de alta una detección solicitada
	Parámetros (RSP):	@idDepartamento int, @idCampaña int, @idRiesgo int, @idUserario varchar(10)
	Precondición (Sino retorna RSP="KO" y deja traza en LOG):	Prueba
	El @idDepartamento debe de existir	SP042.TS202
	La @idCampaña debe de existir	SP042.TS203
	El @idRiesgo debe de existir	SP042.TS204
	Ya existe la Detección que intenta registrar	SP042.TS205
	La @idVerificacion debe de existir	SP042.TS206
	El @idUserario debe de existir y estar activo	SP042.TS207
	Resultado (Retorna RSP="OK"):	Prueba
	Se da de alta con los datos indicados por los parámetros. Retorna además el nuevo ID	SP042.TS208

5.2 DW

En principio he de indicar que esta ha sido una de las partes que he elaborado con más cuidado. Esto es así porque al final me he decidido a utilizar disparadores para su elaboración. Tal como indicaba en la fase de diseño no estoy muy a favor de su uso. Dado que como norma general en el día a día el programador se suele olvidar de su existencia y, por tanto, ocultan parte de la lógica de negocio. De manera que complican la gestión de errores o de problemas de rendimiento. Por tanto, para acotar su impacto he intentado en la medida de lo posible trabajar solo con las tablas de destino, las tablas de trabajo y los registros afectados. Me decidí a usarlos dado que de por sí todo el enunciado de almacén de datos parece encaminado a este fin. Esta opción tiene la ventaja de que al no tener que ser implementada de manera directa sobre los procedimientos implicados hace que la gestión del DW se realice de una manera independiente y transparente al resto de desarrollos. Simplemente cuando se realice una operación sobre una de las tablas se reflejará en el indicador.

Por otro lado, la otra complejidad inherente a la elaboración de estos indicadores es la propia concurrencia de los procesos. Que dependiendo de lo alta que sea debería fijar el estilo de desarrollo a la hora de elaborarlas. Si tenemos una alta concurrencia, quizás lo mejor sería optar por sacrificar parte de la inmediatez de estos indicadores. Anotando los cambios en una cola para ser tratados a posteriori por un proceso de modo secuencial. Sin embargo, tal como ya comenté en la fase de diseño he optado por hacer primar la inmediatez y además por utilizar un enfoque optimista a la hora de bloquear los elementos. Para paliar el efecto de la concurrencia he optado siempre que ha sido posible que el cálculo de estos indicadores se realice mediante una sola operación de incremento o decremento. De esta manera nos aseguramos la atomicidad del cálculo haciéndolo inmune a la concurrencia de procesos. En el caso particular en el que he tenido que realizar disgregar la operación en más de una operación, estas son independientes y tienen carácter absoluto. Es decir, son operaciones que almacenan datos intermedios y que realizan operaciones que se tienen que

cumplir en todo momento. Por tanto, si se produce un solapamiento no afectará al valor final del indicador.

La mayoría de los indicadores son valores que son afectados por una condición. En estos casos he optado por gestionarlos mediante incrementos o decrementos. Si un valor cumple la condición y antes no la cumplía se incrementa el valor del indicador. Si un valor cumplía la condición y ahora deja de cumplirla se decrementa su valor. Estos incrementos y decrementos se totalizan antes de ser incorporados al indicador. Dado que propia naturaleza de los disparadores en SQL Server no admite el tratamiento secuencial. Al trabajarlos como incrementos o decrementos evitaremos cualquier problema con la concurrencia de procesos.

Veamos un poco cada uno de los indicadores y como se han resuelto:

Tabla 23. Definición de indicadores de DW

INDICADOR 1	En el momento de ejecutar la consulta, porcentaje de riesgos no corregidos o mitigados con un impacto menor a 4	
Temporización:	Inmediata	
Obtención final:	SELECT ValorReal AS Indicador1 FROM DW.ValorReal WHERE ID = 1	
Cálculo:	TRIGGER ERM.TR_Riesgo_AfterInsertUpdate ON ERM.Riesgo AFTER INSERT,UPDATE	
Se inicializa en la creación de la base de datos y se mantiene en cada actualización sobre la tabla.		
INDICADOR 2	Dado un año en concreto, número total de riesgos con un impacto de 1	
Temporización:	Inmediata	
Obtención final:	SELECT Valor AS Indicador2 FROM DW.Anual WHERE ID = 2 AND AÑO = @AÑO	
Cálculo:	TRIGGER ERM.TR_Riesgo_AfterInsertUpdate ON ERM.Riesgo AFTER INSERT,UPDATE	
Se inicializa en la creación de la base de datos el año actual. El mismo indicador crea el registro del año en curso en el caso de que no exista y se mantiene en cada actualización sobre la tabla.		
INDICADOR 3	En el último año finalizado, número de riesgos que actualmente aún están en estado de abierto	
Temporización:	Inmediata e inicialización de valores al inicio de año	
Obtención final:	SELECT Valor AS Indicador3 FROM DW.ValorNum (NOLOCK) WHERE ID = 3	
Cálculo:	PROCEDURE [DW].[CierreAnual] TRIGGER ERM.TR_Riesgo_AfterInsertUpdate ON ERM.Riesgo AFTER INSERT,UPDATE	
Se inicializa en la creación de la base de datos. Cada cierre de año se inicializa su valor con el número de riesgos del año finalizado que están en estado abierto y se mantiene en cada actualización sobre la tabla sobre los registros del año anterior.		
INDICADOR 4	Con todos los datos disponibles en la BD, departamento con un número mayor de riesgos detectados por auditorías externas	
Temporización:	Inmediata	
Obtención final:	SELECT Valor AS Indicador4 FROM DW.ValorNum (NOLOCK) WHERE ID = 4	
Cálculo:	TRIGGER ERM.TR_Deteccion_Insert ON ERM.Deteccion AFTER INSERT	
Se inicializa en la creación de la base de datos. Dado que se admite el borrado de una detección. Con cada ejecución se incorporan los posibles nuevos departamentos a la tabla de trabajo Auxiliar. Incremento los valores auxiliares si se cumple la condición. Y si alguno de los departamentos afectados tiene un valor superior al valor del indicador se reemplaza. Se guarda el Identificador del departamento.		
INDICADOR 5	En el año actual, número total de acciones en estado de en curso	
Temporización:	Inmediata e inicialización de valores al inicio de año	
Obtención final:	SELECT Valor AS Indicador5 FROM DW.ValorNum (NOLOCK) WHERE ID = 5	
Cálculo:	PROCEDURE [DW].[CierreAnual] TRIGGER ERM.TR_Accion_AfterInsertUpdate ON ERM.Accion AFTER INSERT,UPDATE	
Se inicializa en la creación de la base de datos. Cada cierre de año se inicializa a cero y se mantiene en cada actualización sobre la tabla.		
INDICADOR 6	Dado un año en concreto, diferencia entre el número de riesgos de impacto 1 detectados por auditoría interna y externa	
Temporización:	Inmediata	

Obtención final:	<code>SELECT Valor AS Indicador6 FROM DW.Anual WHERE ID = 6 AND AÑO = @AÑO</code>
Cálculo:	<code>TRIGGER ERM.TR_Riesgo_AfterInsertUpdate ON ERM.Riesgo AFTER INSERT,UPDATE</code>
Se inicializa en la creación de la base de datos el año actual. Cada nueva ejecución agrega el año en curso con valor 0 si es necesario. Cada actualización realiza las modificaciones oportunas en el año correspondiente. Se mantienen en los valores auxiliares las cantidades separadas para auditorías internas y externas	
INDICADOR 7	En el momento de ejecutar la consulta, persona de la empresa que tiene más acciones abiertas (en estado de en curso o de definida) asignadas
Temporización:	Inmediata
Obtención final:	<code>SELECT ValorString AS Indicador7 FROM DW.ValorString (NOLOCK) WHERE ID = 7</code>
Cálculo:	<code>TRIGGER ERM.TR_Accion_AfterInsertUpdate ON ERM.Accion AFTER INSERT,UPDATE</code>
Se inicializa en la creación de la BD. Cada nueva ejecución agrega las personas que faltan en la tabla auxiliar. Mediante incrementos o decrementos mantengo los valores de la tabla auxiliar. Y dado que existe la posibilidad de un decremento actualizo al final el indicador con el valor más elevado de la tabla de auxiliar.	
INDICADOR 8	Número medio de muestreos realizados por año sin considerar el año actual
Temporización:	Con cada cierre
Obtención final:	<code>SELECT ValorReal AS Indicador8 FROM DW.ValorReal (NOLOCK) WHERE ID = 8</code>
Cálculo:	<code>PROCEDURE [DW].[CierreAnual]</code>
Se inicializa en la creación de la BD. Cada cierre de año se calcula su valor y se actualiza el indicador	
INDICADOR 9	Teniendo en cuenta el año en curso y el anterior, número de riesgos de Ciberseguridad detectados por autoevaluación de los departamentos
Temporización:	Inmediata e inicialización de valores al inicio de año
Obtención final:	<code>SELECT Valor AS Indicador9 FROM DW.ValorNum (NOLOCK) WHERE ID = 9</code>
Cálculo:	<code>PROCEDURE [DW].[CierreAnual]</code> <code>TRIGGER ERM.TR_Riesgo_AfterInsertUpdate ON ERM.Riesgo AFTER INSERT,UPDATE</code>
Se inicializa en la creación de la BD. Cada cierre de año se inicializa su valor con el cálculo del año anterior y se mantiene en cada actualización sobre la tabla sobre los valores del año anterior o el año en curso.	
INDICADOR 10	En el último año finalizado, coste medio de todas las auditorías externas realizadas
Temporización:	Con cada cierre
Obtención final:	<code>SELECT ValorReal AS Indicador10 FROM DW.ValorReal (NOLOCK) WHERE ID = 10</code>
Cálculo:	<code>PROCEDURE [DW].[CierreAnual]</code>
Se inicializa en la creación de la BD. Cada cierre de año se calcula su valor y se actualiza el indicador.	
INDICADOR 11	Teniendo en cuenta todos los datos disponibles, año con un número mayor de acciones que al final del año en cuestión estaban en estado de implementada con el riesgo mitigado
Temporización:	Con cada cierre
Obtención final:	<code>SELECT Valor AS Indicador11 FROM DW.Anual WHERE ID = 11 AND auxiliar = 1</code>
Cálculo:	<code>PROCEDURE [DW].[CierreAnual]</code>
Se inicializa en la creación de la BD. Cada cierre de año se calcula el valor existente para el año de cierre y se anota en la tabla de años. Y se anota el valor del auxiliar para indicar el año que tiene mayor valor.	
INDICADOR 12	Teniendo en cuenta sólo el último año finalizado, Top3 de acciones definidas teniendo en cuenta el tiempo que estuvieron abiertas
Temporización:	Con cada cierre
Obtención final:	<code>SELECT numero1 AS Id12_Top1, numero2 AS Id12_Top2, numero3 AS Id12_Top3 FROM DW.Top3 (NOLOCK) WHERE ID = 12</code>
Cálculo:	<code>PROCEDURE [DW].[CierreAnual]</code>
Se inicializa en la creación de la BD. Cada cierre de año se calcula en todas las acciones con fecha de cierre el tiempo que estuvieron abiertas. Se anota el valor de las tres que estuvieron más tiempo abiertas en la tabla. Se retornan los indicadores	
INDICADOR 13	En el momento de ejecutar la consulta, número de riesgos de categoría 1 (en cualquier estado)
Temporización:	Inmediata
Obtención final:	<code>SELECT valor AS Indicador13 FROM DW.ValorNum (NOLOCK) WHERE ID = 13</code>
Cálculo:	<code>TRIGGER ERM.TR_Riesgo_AfterInsertUpdate ON ERM.Riesgo AFTER INSERT,UPDATE</code>
Se inicializa en la creación de la base de datos y se mantiene en cada actualización sobre la tabla.	
INDICADOR 14	Dado un año en concreto, porcentaje de acciones descartadas
Temporización:	Inmediata

Obtención final:	<code>SELECT valor AS Indicador14 FROM DW.Anual WHERE ID = 14 AND AÑO = @AÑO</code>
Cálculo:	<code>TRIGGER ERM.TR_Accion_AfterInsertUpdate ON ERM.Accion AFTER INSERT,UPDATE</code>
Se inicializa en la creación de la base de datos el año actual. Cada nueva ejecución agrega el año en curso con valor 0 si es necesario. Cada actualización realiza las modificaciones oportunas en el año correspondiente.	

Tanto el script de creación de los disparadores como el procedimiento que realizará el cierre del año se incluirán en la entrega final. Sin embargo, también se encuentran disponibles a través de los siguientes enlaces:

- [02 Disparadores.sql](#)
- [\[DW\].\[CierreAnual\].sql](#)

6. Validación

En esta fase se debe de validar la solución implementada. En un proyecto normal lo habitual sería que esta operación fuera realizada de manera conjunta con el cliente final, con la intención verificar lo desarrollado y de obtener una aceptación del cliente. Sin embargo, dada la naturaleza de este proyecto esta verificación deberá de ser reemplazada por una validación más de carácter formal. Para ello implementaremos una batería de prueba sobre cada uno de los procedimientos desarrollados en la etapa anterior para asegurar que se comportan del modo esperado.

Aunque simulando la iteración con el cliente, y por ser correctos, la primera validación que he realizado consiste en revisar si lo desarrollado cumple los requerimientos. Omitiré incluir la lista de verificación (*checklist*) porque no creo que aporte mucho. Todos requisitos funcionales son de fácil comprobación, dado que la mayoría se refieren a que valores debe de almacenar nuestro sistema y como se debe de comportar. Por lo que ya se tuvieron en cuenta en la fase de diseño. En cuanto a los requisitos no funcionales, es decir las calidades esperadas del sistema, la mayoría también se cumplen. El único que es un poco más subjetivo es el requisito RNF003 Diseñar la aplicación para cualquier volumen de datos. Para poder comprobarlo necesitaríamos hacer pruebas de estrés sobre el sistema. Que por la capacidad existente ya se decidieron dejarlas fuera del alcance del proyecto durante la fase de planificación. El último grupo de requisitos fijados, los de proceso, se refieren al propio proceso de desarrollo de la solución y es fácil comprobar su cumplimiento.

He diferenciado la validación de la parte de ERM y la referida al DW. Veamos:

6.1 ERM

Tal como he indicado en el apartado anterior para poder validar esta parte he creado una serie de pruebas. Con su ejecución se comprobará que todos los procedimientos desarrollados se comportan según la definición indicada. Para ello se desarrollarán una serie de pruebas que comprobarán que, si no se cumple alguna de las precondiciones la operación no se realiza. Además, se desarrollan una serie de pruebas finales para determinar que si se cumplen las condiciones la operación esta se realiza de manera correcta. Debido a lo ajustado de los tiempos, y a la cantidad ingente de procedimientos he tenido que modelar un sistema que permitiera ejecutar de manera ágil este conjunto de pruebas. Dicho

sistema se basa en una tabla de gestión LOG.PruebasAutomatizadas. Esta tabla nos permite definir para cada prueba el procedimiento a testear, una serie de parámetros de entrada, si el resultado será correcto o incorrecto y analizar si se cumple comprobación final. Además una vez se complete la ejecución se guardará una serie de información adicional de control que facilitará cualquier análisis posterior: RSP, identificador del log generado, fecha hora y duración en segundos. La definición de la tabla ya ha sido referida en el apartado [4.3.2](#) LOG del apartado de diseño físico.

Para la correcta ejecución de estas pruebas se ha desarrollado un conjunto de datos iniciales de carácter reducido. Además se ha implementado un script que dada las pruebas anotadas en esta tabla realizará su ejecución de los procedimientos definidos y anotará los resultados en la misma tabla. Para facilitar la realización de las pruebas y evitar modificar el conjunto de datos inicial dicho script de partida ejecuta todos los procedimientos de prueba de manera transaccional, muestra los resultados obtenidos y deshace la operación. Si se desea se podrá modificar el script para que los cambios sean permanentes reemplazando la instrucción final `ROLLBACK TRANSACTION` por `COMMIT TRANSACTION`.

Dichos script serán suministrados en la entrega final. Pero son accesibles a través de los enlaces:

- [P00 Datos iniciales.sql](#)
- [P01 Insertar pruebas.sql](#)
- [P02 Lanzar pruebas.sql](#)

Como resultado de la ejecución de las 208 pruebas automatizadas obtendremos:

Figura 18. Resultado de ejecución de batería de prueba de procedimientos

TIPO	resultado	Cantidad
1 AUTOMATIZADA TOTAL	CORRECTA	208

TIPO	id	prueba	resultado	RSP	idLog	retomo	FechaHora	DuracionSeg	inform
1 AUTOMATIZADA	1	SP001.TS001	CORRECTA	KO ERROR SP <ERM.uspDepartamentoAlta>: Debe de ind...	417	-1	2023-05-23 17:52:45.543	0,017	
2 AUTOMATIZADA	2	SP001.TS002	CORRECTA	KO ERROR SP <ERM.uspDepartamentoAlta>: Ya se encu...	418	-1	2023-05-23 17:52:45.560	0,01	
3 AUTOMATIZADA	3	SP001.TS003	CORRECTA	KOERROR SP <ERM.uspDepartamentoAlta>: Usuario invál...	419	-1	2023-05-23 17:52:45.570	0,006	
4 AUTOMATIZADA	4	SP001.TS004	CORRECTA	OK	420	6	2023-05-23 17:52:45.577	0,007	
5 AUTOMATIZADA	5	SP002.TS005	CORRECTA	KOERROR SP <ERM.uspDepartamentoBaja>: Departamen...	421	-1	2023-05-23 17:52:45.583	0,017	
6 AUTOMATIZADA	6	SP002.TS006	CORRECTA	KOERROR SP <ERM.uspDepartamentoBaja>: Existen usu...	422	-1	2023-05-23 17:52:45.600	0,006	
7 AUTOMATIZADA	7	SP002.TS007	CORRECTA	KOERROR SP <ERM.uspDepartamentoBaja>: Usuario invá...	423	-1	2023-05-23 17:52:45.607	0,004	
8 AUTOMATIZADA	8	SP002.TS008	CORRECTA	OK	424	2	2023-05-23 17:52:45.610	0,006	
9 AUTOMATIZADA	9	SP003.TS009	CORRECTA	KOERROR SP <ERM.uspDepartamentoModificacion>: Dep...	425	-1	2023-05-23 17:52:45.617	0,014	
10 AUTOMATIZADA	1..	SP003.TS010	CORRECTA	KO ERROR SP <ERM.uspDepartamentoModificacion>: De...	426	-1	2023-05-23 17:52:45.630	0,003	

Retomo	MensajeResultado	Total duración
1 0	Operación completada con éxito	3.087 seg

Consulta ejecutada correctamente.

Una primera tabla total donde se contabilizan las operaciones según el resultado obtenido. Una tabla con el detalle de cada ejecución. Y un mensaje final que indicará si se ha producido cualquier tipo de error durante la ejecución.

6.2 DW

En cuanto a la validación de cada uno de los indicadores requeridos. Se suministra a continuación el cálculo necesario para su comprobación que se deberá de cotejar con el existente anotado en la tabla. Debido a la escasez de tiempo en la fase de implementación y pruebas no he podido realizar un juego de pruebas más específico para esta parte. Por lo que solo me ha dado tiempo a incluir estas comprobaciones al final del script de pruebas automatizadas. De manera que se realice una validación al final de las pruebas cotejando el valor registrado en el indicador con el cálculo suministrado. Dado que en todo momento los valores de los indicadores deben de ser correctos. He omitido los indicadores 8, 10, 11 y 12 dado que sus valores solo se calculan en el cierre y para poder probarlos sería necesario generar un conjunto de pruebas más específicas.

Tabla 24. Validación de indicadores de DW

INDICADOR 1	En el momento de ejecutar la consulta, porcentaje de riesgos no corregidos o mitigados con un impacto menor a 4
Indicador:	SELECT ValorReal AS Indicador1 FROM DW.ValorReal WHERE ID = 1
Cálculo:	SELECT CAST(SUM(CASE WHEN ((Riesgo.activo = 1) AND (Riesgo.estado In ('abierto', 'mitigado')) AND (Riesgo.impacto < 4)) THEN 1 ELSE 0 END)/CAST(COUNT(*) AS decimal(9,3)) AS decimal(9,3)) AS CalculoIndicador1 FROM ERM.Riesgo (NOLOCK)
INDICADOR 2	Dado un año en concreto, número total de riesgos con un impacto de 1
Indicador:	SELECT Valor AS Indicador2 FROM DW.Anual WHERE ID = 2 AND AÑO = @AÑO
Cálculo:	SELECT YEAR(Riesgo.fechaCreacion) AS año, SUM(CASE WHEN ((Riesgo.activo = 1) AND (Riesgo.impacto = 1)) THEN 1 ELSE 0 END) AS CalculoIndicador2 FROM ERM.Riesgo (NOLOCK) GROUP BY YEAR(Riesgo.fechaCreacion)
INDICADOR 3	En el último año finalizado, número de riesgos que actualmente aún están en estado de abierto
Indicador:	SELECT Valor AS Indicador3 FROM DW.ValorNum (NOLOCK) WHERE ID = 3
Cálculo:	SELECT ISNULL(SUM(CASE WHEN ((Riesgo.activo = 1) AND (Riesgo.estado = 'abierto')) THEN 1 ELSE 0 END),0) AS CalculoIndicador3 FROM ERM.Riesgo (NOLOCK) WHERE YEAR(Riesgo.fechaCreacion) = YEAR(GETDATE())-1
INDICADOR 4	Con todos los datos disponibles en la BD, departamento con un número mayor de riesgos detectados por auditorías externas
Indicador:	SELECT Valor AS Indicador4 FROM DW.ValorNum (NOLOCK) WHERE ID = 4
Cálculo:	;WITH Departamentos AS (SELECT Departamento.id AS CalculoIndicador4, Count(*) AS Cant FROM ERM.Deteccion (NOLOCK) INNER JOIN ERM.AuditoriaExterna (NOLOCK) ON Deteccion.idCampaña = AuditoriaExterna.id INNER JOIN ERM.Departamento (NOLOCK) ON Deteccion.idDepartamento = Departamento.id GROUP BY Departamento.id) SELECT Departamentos.CalculoIndicador4, Departamentos.Cant FROM Departamentos INNER JOIN (SELECT MAX(Cant) As Maximo FROM Departamentos) AS RM ON Departamentos.Cant = RM.Maximo
INDICADOR 5	En el año actual, número total de acciones en estado de en curso
Indicador:	SELECT Valor AS Indicador5 FROM DW.ValorNum (NOLOCK) WHERE ID = 5
Cálculo:	SELECT ISNULL((SELECT Count(*) FROM ERM.Accion (NOLOCK) WHERE ((Accion.activo = 1) AND (Accion.estado = 'enCurso') AND (YEAR(Accion.fechaCreacion) = (YEAR(GETDATE()))))), 0) AS CalculoIndicador5
INDICADOR 6	Dado un año en concreto, diferencia entre el número de riesgos de impacto 1 detectados por auditoría interna y externa
Indicador:	SELECT Valor AS Indicador6 FROM DW.Anual WHERE ID = 6 AND AÑO = @AÑO
Cálculo:	SELECT YEAR(Riesgo.fechaCreacion) AS año, SUM (CASE WHEN (Riesgo.origen = 'auditoriaInterna') AND (Riesgo.activo = 1) AND (Riesgo.impacto = 1) THEN 1 ELSE 0 END - CASE WHEN (Riesgo.origen = 'auditoriaExterna') AND (Riesgo.activo = 1) AND (Riesgo.impacto = 1) THEN 1 ELSE 0 END) AS CalculoIndicador6 FROM ERM.Riesgo (NOLOCK) GROUP BY YEAR(Riesgo.fechaCreacion)
INDICADOR 7	En el momento de ejecutar la consulta, persona de la empresa que tiene más acciones abiertas (en estado de en curso o de definida) asignadas
Indicador:	SELECT ValorString AS Indicador7 FROM DW.ValorString (NOLOCK) WHERE ID = 7

Cálculo:	<pre>;WITH Responsables AS (SELECT idResponsable AS CalculoIndicador7, SUM (CASE WHEN ((Accion.activo = 1) AND (Accion.estado In ('enCurso', 'definida'))) THEN 1 ELSE 0 END) AS Cant FROM ERM.Accion (NOLOCK) GROUP BY idResponsable) SELECT Responsables.CalculoIndicador7 FROM Responsables INNER JOIN (SELECT MAX(Cant) AS Maximo FROM Responsables) AS RM ON Responsables.Cant = RM.Maximo</pre>
INDICADOR 8	Número medio de muestreos realizados por año sin considerar el año actual
Indicador:	<pre>SELECT ValorReal AS Indicador8 FROM DW.ValorReal (NOLOCK) WHERE ID = 8</pre>
Cálculo:	<pre>SELECT ISNULL(CAST(SUM(MuestreosAño.Cantidad) AS float)/COUNT(*),0) AS CalculoIndicador8 FROM (SELECT YEAR(fechaCreacion) AS Año, COUNT(*) AS Cantidad FROM ERM.Muestreo (NOLOCK) WHERE YEAR(fechaCreacion) <> YEAR(GETDATE()) GROUP BY YEAR(fechaCreacion)) AS MuestreosAño</pre>
INDICADOR 9	Teniendo en cuenta el año en curso y el anterior, número de riesgos de Ciberseguridad detectados por autoevaluación de los departamentos
Indicador:	<pre>SELECT Valor AS Indicador9 FROM DW.ValorNum (NOLOCK) WHERE ID = 9</pre>
Cálculo:	<pre>SELECT COUNT(*) AS CalculoIndicador9 FROM ERM.Riesgo (NOLOCK) INNER JOIN ERM.Categoria (NOLOCK) ON Riesgo.idCategoria = Categoria.id Where ((Riesgo.activo = 1) AND (Categoria.Descripcion = 'Ciberseguridad') AND (Riesgo.origen = 'autoevaluación') AND (YEAR(Riesgo.fechaCreacion) In (YEAR(GETDATE())-1, YEAR(GETDATE()))))</pre>
INDICADOR 10	En el último año finalizado, coste medio de todas las auditorías externas realizadas
Indicador:	<pre>SELECT ValorReal AS Indicador10 FROM DW.ValorReal (NOLOCK) WHERE ID = 10</pre>
Cálculo:	<pre>Select ISNULL(CAST(SUM(AuditoriaExterna.coste) AS float)/COUNT(*),0) AS CalculoIndicador10 FROM ERM.AuditoriaExterna (NOLOCK) INNER JOIN ERM.Campaña (NOLOCK) ON AuditoriaExterna.id = Campaña.id AND ((Campaña.activo = 1) AND (YEAR(Campaña.fechaFin) = YEAR(GETDATE())-1))</pre>
INDICADOR 11	Teniendo en cuenta todos los datos disponibles, año con un número mayor de acciones que al final del año en cuestión estaban en estado de implementada con el riesgo mitigado
Indicador:	<pre>SELECT Valor AS Indicador11 FROM DW.Anual WHERE ID = 11 AND auxiliar = 1</pre>
Cálculo:	<pre>-- Su valor se basa en datos históricos solo se puede resolver sobre los datos guardados sobre la tabla DW.Anual SELECT año AS CalculoIndicador11 FROM DW.Anual (NOLOCK) INNER JOIN (SELECT MAX(valor) AS Cantidad FROM DW.Anual (NOLOCK) WHERE ID = 11 AND valor <> 0) AS Mayor ON Anual.valor = Mayor.Cantidad WHERE ID = 11</pre>
INDICADOR 12	Teniendo en cuenta sólo el último año finalizado, Top3 de acciones definidas teniendo en cuenta el tiempo que estuvieron abiertas
Indicador:	<pre>SELECT numero1 AS Id12_Top1, numero2 AS Id12_Top2, numero3 AS Id12_Top3 FROM DW.Top3 (NOLOCK) WHERE ID = 12</pre>
Cálculo:	<pre>SELECT TOP 3 numero As idAccion, DATEDIFF(day, fechaCreacion, fechaEjecucion) As Valor FROM ERM.Accion (NOLOCK) WHERE NOT fechaEjecucion IS NULL ORDER BY DATEDIFF(day, fechaCreacion, fechaEjecucion) DESC</pre>
INDICADOR 13	En el momento de ejecutar la consulta, número de riesgos de categoría 1 (en cualquier estado)
Indicador:	<pre>SELECT valor AS Indicador13 FROM DW.ValorNum (NOLOCK) WHERE ID = 13</pre>
Cálculo:	<pre>SELECT SUM(CASE WHEN ((Riesgo.activo = 1) AND (Riesgo.idCategoria = 1)) THEN 1 ELSE 0 END) AS CalculoIndicador13 FROM ERM.Riesgo (NOLOCK)</pre>
INDICADOR 14	Dado un año en concreto, porcentaje de acciones descartadas
Indicador:	<pre>SELECT valor AS Indicador14 FROM DW.Anual WHERE ID = 14 AND AÑO = @AÑO</pre>
Cálculo:	<pre>SELECT YEAR(Accion.fechaCreacion) AS año, CAST(SUM(CASE WHEN ((Accion.activo = 1) AND (Accion.estado = 'descartada')) THEN 1 ELSE 0 END)/CAST(COUNT(*) AS float) AS float) AS CalculoIndicador14 From ERM.Accion (NOLOCK) Group By YEAR(Accion.fechaCreacion)</pre>

Como resultado de la ejecución de pruebas automáticas obtenemos:

Figura 19. Resultado de ejecución de validación de indicadores

Resultados			Mensajes		
	TIPO	indicador	resultado		
1	INDICADOR	1	CORRECTA		
2	INDICADOR	2	CORRECTA		
3	INDICADOR	3	CORRECTA		
4	INDICADOR	4	CORRECTA		
5	INDICADOR	5	CORRECTA		
6	INDICADOR	6	CORRECTA		
7	INDICADOR	7	CORRECTA		
8	INDICADOR	9	CORRECTA		
9	INDICADOR	9	CORRECTA		
10	INDICADOR	13	CORRECTA		
11	INDICADOR	14	CORRECTA		

Retomo	Mensaje/Resultado	Total duración
1	0 Operación completada con éxito	2.757 seg

✓ Consulta ejecutada correctamente.

Una tabla con cada indicador comprobado y el resultado de la comprobación. Y un mensaje final que indicará si se ha producido cualquier tipo de error durante la ejecución.

7. Conclusiones

Esta parte del documento tiene la intención de resumir recapitular un poco sobre el trabajo realizado. Uno de sus puntos necesarios a abordar en este capítulo es realizar un poco de retrospectiva con el fin de concretar qué lecciones y ha aprendido durante su realización. En mi caso particular he decir que valoró muy positivamente toda la parte del diseño conceptual y lógico. Es una parte que no puedo realizar en mí día a día, dado que trabajo siempre sobre un sistema de bases de datos fijo. Por tanto, siempre se suelen obviar este tipo de diseños e implementar el modelo físico final. Optando siempre por modelar una serie de patrones fijos para que el sistema sea lo más versátil. Sin embargo, la realización del trabajo me ha permitido revisar otras opciones de modelado, quizás más complejas, pero con sus ventajas. Por otro lado, tampoco tenía una gran experiencia sobre la implementación de los disparadores, dado que siempre he huido de su utilización. Pero después de su utilización en el proyecto, y en este caso, supongo que como todo, tienen cabida en su justa medida.

Otro de los puntos necesarios a concretar en este apartado es si hemos alcanzado los objetivos definidos. En este punto, como en todo el trabajo, tenemos la dualidad con respecto al trabajo fin de grado y al producto en sí mismo. Con respecto a los objetivos definidos en el documento, que se centraban en el trabajo fin de grado, considero que se han logrado. Estos se centraban en poner en práctica todas las lecciones aprendidas durante la realización del grado de informática. Pero si miramos los objetivos desde el punto de vista del producto final. Los cuales aunque no se han enumerado se pueden extrapolar de lo indicado durante la planificación y el análisis del producto, considero que no ha sido tanto así. La cantidad de trabajo para desarrollar el núcleo del producto en cada una de sus fases ha sido tan ingente que se ha comido el resto de las posibilidades. Esto no ha permitido finalizar el desarrollo o realizar muchas de las partes propias que deben de contener el producto, como seguridad y control de accesos, revisión de rendimiento o pruebas más allá de las unitarias.

También, deberemos hacer retrospectiva sobre la metodología utilizada durante el desarrollo del proyecto. En este caso considero que la metodología elegida ha sido la correcta. El hecho de tener fijados desde el inicio una serie de hitos me ha permitido desde fases muy tempranas acotar el trabajo que estaba realizando. A título personal he decir que, en mi caso, que cumplir la planificación ha sido en todo momento una constante carrera contrarreloj. Considero que el tiempo para elaborar un trabajo con cierta calidad no se corresponde con el tiempo planificado para la asignatura. Sin embargo, entiendo que es el que es, porque es el que tiene cabida con respecto a los cuatrimestres formativos existentes. En mi caso particular la calidad del producto final no ha sido fijada en gran medida por el calendario.

Por tanto, surgen una gran cantidad de temas pendientes que no se han podido abordar durante la realización de este. Algunos que son propios del producto, como el trabajo de la parte que no se ha podido completar. Otros consisten en

aspectos del producto que se deberían de concretar antes de su puesta en marcha: ampliar la verificación del producto más allá de las pruebas unitarias, definir la seguridad del sistema (encriptación, usuarios, visibilidad, permisos...), la implementación física de la solución (distribución de ficheros, rendimiento...), mantenimiento (historificación, copias de seguridad...), etc. Otros se refieren a más a lo funcional y se centran en la evolución del producto a futuro. Que son cambios más propios del ciclo de vida de un producto. En cuanto a ellos sería necesario refinar y concretar más de alguno de los aspectos modelados, como por ejemplo: concretar el tipo de información necesaria a almacenar para la realización de las auditorías y controles. También, se debería de explorar como realizar un tratamiento más proactivo del tratamiento de los riesgos, más allá de las modificaciones mínimas planteadas. Y si es posible ver como incorporar esta herramienta a la propia cultura de la empresa. Para que su uso en el día a día tuviera los beneficios esperados por “la empresa” del control de riesgos.

8. Glosario

En este documento se han hecho referencia a los siguientes términos:

BD (Base de datos). Colección de información que está organizada de manera que se pueda acceder, administrar y actualizar fácilmente.

Cifrado de datos transparente (TDE). Cifrado de Datos transparente es una tecnología empleada por Microsoft, IBM y Oracle para cifrar archivos de datos de SQL Server, Azure SQL Database y Azure Synapse Analytics, lo que se conoce como cifrado de datos en reposo.

Concurrencia (informática). En ciencias de la computación, concurrencia se refiere a la habilidad de distintas partes de un programa, algoritmo, o problema de ser ejecutado en desorden o en orden parcial, sin afectar el resultado final. Los cálculos (operaciones) pueden ser ejecutados en múltiples procesadores, o ejecutados en procesadores separados física o virtualmente en distintos hilos de ejecución.

Datawarehouse. Almacén de datos que aglutina información de diversas fuentes y que ayuda a la hora de la toma de decisiones.

Disparador (trigger en inglés). Es un objeto que se asocia con tablas y se almacena en la base de datos.

Gestión de los riesgos empresariales (ERM). Es el proceso empresarial encargado de identificar y analizar las posibles amenazas que puedan afectar a la gestión o los resultados de una compañía. Con la intención de minimizar su impacto en las operaciones de la empresa.

Mapa mental. Es un diagrama usado para representar palabras, ideas, tareas, lecturas, dibujos, u otros conceptos ligados y dispuestos radicalmente a través de una palabra clave o de una idea central. Los mapas mentales son un método muy eficaz para extraer y memorizar información.

Pipeline CI/CD (Tubería). Se refiere al conjunto de procesos para incorporar la automatización continua y el control permanente en todo el ciclo de vida de un software.

Procedimiento almacenado (stored procedure en inglés). Es un programa (o procedimiento) almacenado físicamente en una base de datos.

SDLC. Ciclo de vida de desarrollo de un sistema (SDLC)

SQL. Es un lenguaje de dominio específico, diseñado para administrar, y recuperar información de sistemas de gestión de bases de datos relacionales.

TDD o Test Driven Development. Práctica de programación en la que se antepone la realización de las pruebas antes del desarrollo.

9. Bibliografía

- Marco Simó, Josep María. Escaneando la informática. Barcelona: Editorial UOC, 2010. 261 p. ISBN 9788497889254.
- José Ramón Rodríguez. La gestión de proyectos. Conceptos básicos. Material de la asignatura Gestión de proyectos. UOC PID_00247944
- Pere Mariné Jové y José Ramón Rodríguez. Componentes de la gestión de proyectos: las áreas de conocimiento. Material de la asignatura Gestión de proyectos. UOC PID_00247940
- Pere Mariné Jové y José Ramón Rodríguez. Planificación del proyecto. Material de la asignatura Gestión de proyectos. UOC. PID_00247943
- Metodologías de desarrollo de software: ¿qué son? [en línea] [consulta: 18 de marzo de 2023]. Disponible en: <https://www.becas-santander.com/es/blog/metodologias-desarrollo-software.html>
- Jordi Casas Roma y Josep Cuartero Olivera. Diseño conceptual de bases de datos. Material de la asignatura diseño de Bases de datos. UOC PID_00270598
- Xavier Burgués Illa y Josep Cuartero Olivera. Diseño lógico de bases de datos. Material de la asignatura diseño de Bases de datos. UOC PID_00270596
- Blai Cabré i Segarra, Jordi Casas Roma, Dolors Costal Costa, Pere Juanola, Ivo Plana Vallvé, Àngels Rius Gavidia y Ramon Segret i Sala. Diseño físico de bases de datos. Material de la asignatura diseño de Bases de datos. UOC PID_00278244
- Jordi Pradel Miquel y Jose Raya Martos. Análisis UML. Material de la asignatura Ingeniería del software. UOC PID_00275615
- Jordi Pradel i Miquel y José Antonio Raya Martos. Catálogo de patrones. Material de la asignatura Análisis y diseño con patrones. UOC PID_00276109
- Jordi Pradel Miquel y Jose Raya Martos. Documentación de requisitos. Material de la asignatura Ingeniería de requisitos. UOC PID_00191264
- Josep Curto Díaz. Diseño de un data *warehouse*. UOC PID_00238513
- Jordi Pradel Miquel y Jose Raya Martos. Análisis UML. Material de la asignatura Ingeniería del software. UOC PID_00275615
- ¿Cuál es la metodología más adecuada para tu proyecto? [en línea]. Disponible en: <https://www2.deloitte.com/es/es/pages/technology/articles/waterfall-vs-agile.html>
- ¿QUÉ ES EL MODELO COSO? [en línea]. Disponible en: <https://www.globalsuitesolutions.com/es/que-es-modelo-coso/>
- Oracle Enterprise Resource Planning (ERP) [en línea]. Disponible en: <https://www.oracle.com/es/erp/risk-management/what-is-enterprise-risk-management/>
- SAP Risk Management [en línea]. Disponible en: <https://winterhawk.com/sap-grc/risk-management/>
- Pelican [en línea]. Disponible en: <https://www.vosesoftware.com/products/pelican/>

- Usando ENUM con SQL [en línea]. Disponible en: <https://sqltemuco.wordpress.com/2017/02/03/usando-enum-con-sql/>
- Archivos y grupos de archivos de base de datos [en línea]. Disponible en: <https://learn.microsoft.com/es-es/sql/relational-databases/databases/database-files-and-filegroups?view=sql-server-ver16>
- Tutorial: Crear y ejecutar una prueba unitaria de SQL Server [en línea]. Disponible en: <https://learn.microsoft.com/es-es/sql/ssdt/walkthrough-creating-and-running-a-sql-server-unit-test?view=sql-server-ver16>
- CREATE TRIGGER (Transact-SQL) [en línea]. Disponible en: <https://learn.microsoft.com/es-es/sql/t-sql/statements/create-trigger-transact-sql?view=sql-server-ver16>
- Tipos de datos (Transact-SQL) [en línea]. Disponible en: <https://learn.microsoft.com/es-es/sql/t-sql/data-types/data-types-transact-sql?view=sql-server-ver16>
- Wikipedia usada en las definiciones [en línea]. Disponible en: <https://es.wikipedia.org/>