Universitat Oberta de Catalunya

uoc.edu

Diseño e implementación de la base de datos para una aplicación de control de procesos de seguridad informática

Joan Marc Garcia Morales
Grado de Ingeniería Informática
Bases de Datos

Jordi Ferrer Duran
Xavier Baró Solé

10 de junio de 2022

FICHA DEL TRABAJO FINAL

Título del trabajo:	*Diseño e implementación de la base de datos para una aplicación de control de procesos de seguridad informática*
Nombre del autor:	*Joan Marc Garcia Morales*
Nombre del consultor/a:	*Jordi Ferrer Duran*
Nombre del PRA:	*Xavier Baró Solé*
Fecha de entrega (mm/aaaa):	06/2022
Titulación:	*Grado de Ingeniería Informática*
Área del Trabajo Final:	*Bases de Datos*
Idioma del trabajo:	*Castellano*
Palabras clave	*Base de datos, Ciberseguridad, Oracle*

Resumen del Trabajo (máximo 250 palabras): *Con la finalidad, contexto de aplicación, metodología, resultados i conclusiones del trabajo.*

El presente trabajo tiene como finalidad el diseño de una base de datos para una aplicación de control de procesos de seguridad informática. La aplicación se encargará de monitorizar las vulnerabilidades, junto con las acciones para mitigarlas, así como las políticas y sesiones formativas sobre ciberseguridad que aplique la empresa.

Se aplica una metodología iterativa, que nos permite tener desde la primera iteración un producto funcional que nos permite analizar si cumple con los requisitos o se deben implementar cambios al diseño realizado.

Para el repositorio estadístico se ha diseñado una versión básica utilizando las técnicas para el almacén de datos. En éste, tenemos una serie de tablas de temática especializada, que nos permitirán dar respuesta a las consultas que se nos plantean, en tiempo constante 1.

El resultado ha sido una base de datos en Oracle 18c Express, que nos permite almacenar tanto datos relacionados con la seguridad informática de la empresa como procesarlos e introducirlos en un repositorio estadístico que permite monitorizar en todo momento el estado en que se encuentra. Además, se incluyen todos los procedimientos ABM (alta + baja + modificación) y otros necesarios para su función realizados mediante el lenguaje PL/SQL.

Abstract (in English, 250 words or less):

The purpose of this work is the design of a database for a computer security process control application. The application will monitor the vulnerabilities, along with the actions to mitigate them, as well as the policies and training sessions on cybersecurity applied by the company.

An iterative methodology is applied, which allows us to have a functional product from the first iteration that allows us to analyze whether it meets the requirements or changes to the design must be implemented.

For the statistical repository, a basic version has been designed using the techniques for the data warehouse. In it, we have a series of tables on specialized topics, which will allow us to respond to the queries that arise, in constant time 1.

The result has been a database in Oracle 18c Express, which allows us to store both data related to the company's computer security and process it and enter it in a statistical repository that allows us to monitor its status at all times. In addition, all the CRUD procedures (create + read (not this one) + update + drop) and others necessary for their function, carried out using the PL/SQL language, are included.

Índice

Lista de figuras

1. Introducción

1.1 Contexto y justificación del Trabajo

Actualmente las pérdidas más importantes para las empresas pueden venir de ataques informáticos y son de vital importancia los procesos de seguridad informática. Todos los procesos críticos de las grandes empresas están gestionados informáticamente y es clave tener controlada cualquier vulnerabilidad a la cual la empresa pueda estar expuesta.

Las empresas tienen que invertir en diferentes procesos de revisión y análisis de todos sus procesos informáticos para detectar cualquier posible vulnerabilidad, entendida como una exposición a un posible ataque informático. En el momento que se han detectado todas las posibles vulnerabilidades se deben definir e implementar las posibles acciones de mitigación proactiva para asegurar los intereses de la compañía.

El trabajo por realizar viene de la necesidad de una gran empresa del sector automovilístico, que quiere implantar una nueva aplicación para controlar todos sus procesos de seguridad informática. La aplicación debe permitirle controlar, de manera rápida, la situación en términos de seguridad informática. Se conocerá en todo momento la situación y acciones en curso para mitigar cualquier vulnerabilidad abierta.

En concreto, el trabajo a realizar consistirá en analizar la problemática planteada y definir una posible estructura de base de datos que de soporte a la futura aplicación de gestión de procesos de seguridad informática. Ésta nueva aplicación deberá permitir el control de los diferentes aspectos que se detallarán, la ejecución de las consultas que se consideren necesarias para la correcta gestión y análisis de los datos almacenados en el sistema, y también una visualización rápida del estado de los indicadores principales de monitorización del estado de seguridad de la empresa que se definan en cada momento. El desarrollo o implantación de la futura aplicación queda fuera del alcance del trabajo.

1.2 Objetivos del Trabajo

El objetivo de este TFG es el diseño e implementación de la base de datos para una aplicación de control de procesos de seguridad informática. El enunciado nos marca los siguientes objetivos en el desarrollo del trabajo:

- Poner en práctica los conocimientos adquiridos en asignaturas de Bases de Datos
- Utilizar el lenguaje SQL

- Ampliar conocimientos utilizando nuevas herramientas
- Detectar cuales son las necesidades básicas de un determinado sistema a analizar
- Detectar posibles funcionalidades adicionales de valor añadido
- Proponer un diseño que se ajuste a los requerimientos expuestos
- Implementar un sistema que encapsule las funciones de acceso de datos

El diseño de la base de datos tendrá los siguientes objetivos:

- Analizar la problemática planteada y definir una posible estructura de Base de Datos
- Uso de las técnicas que se aplican a grandes volúmenes de información (Data Warehouse)
- Analizar posibles líneas de evolución que podrá seguir la base de datos

1.3 Enfoque y método seguido

La empresa automovilística para la que realizaremos la base de datos tiene como objetivo la implantación de una aplicación, siendo nuestro trabajo una parte de ésta. Debido a esto y ya que no se nos menciona que exista una versión anterior, se ha escogido desarrollar un producto nuevo, que se adapte completamente a los requerimientos del trabajo.

El hecho de realizar un producto nuevo no restringe la comunicación con productos que puedan existir, como por ejemplo la base de datos del personal, si fuera necesario.

Para el desarrollo de la base de datos se ha optado por seguir la metodología de ciclo de vida iterativo e incremental. Se desarrollará mediante una serie de iteraciones que nos marcarán las entregas de la asignatura, en que cada una de ellas es un mini proyecto autocontenido que amplía el resultado final[1].

De esta manera se acelera la retroalimentación, lo que nos permite actuar ante posibles errores que se hayan podido producir en el diseño desde el principio, no al final del proyecto.

[1] [1] J.Pradel Miguel y J. Raya Martos, Módulo 1: Introducción a Ingeniería de Software

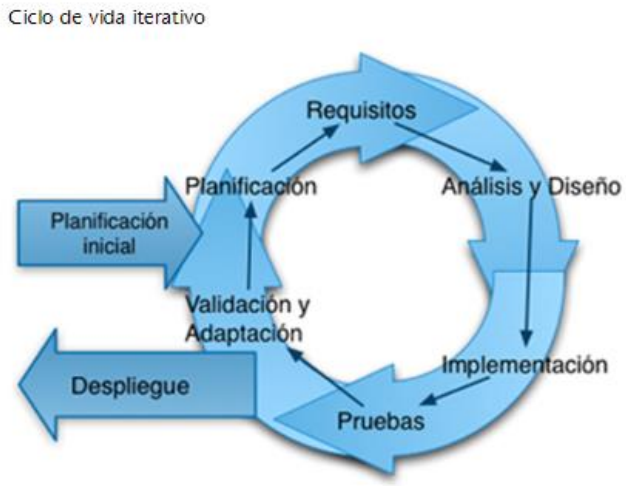


Ilustración 1. Ciclo de vida iterativo

1.4 Planificación del Trabajo

1.4.1 Hitos principales

Los hitos principales del trabajo vienen marcados por las fechas de las diferentes entregas, ya que no admiten desviaciones temporales de ningún tipo:

Plan de trabajo	07/03/2022
Segunda entrega	11/04/2022
Tercera entrega	12/05/2022
Entrega final	10/06/2022
Tribunal	17/06/2022

1.4.2 Detalle de las tareas

El detalle de las tareas que compondrán el trabajo es el siguiente:

Tarea	**Descripción**
Primera entrega - Plan de trabajo	
Lectura del tema propuesto	Lectura de la propuesta del trabajo
Plan de trabajo	Planificación del trabajo
Preparación del entorno de trabajo	Elección e instalación del software necesario para el desarrollo del trabajo
Segunda entrega	
Revisión del trabajo realizado	Modificaciones al trabajo ya realizado debido a indicaciones del consultor o errores que puedan surgir
Diseño de la base de datos	Primera iteración del diseño. Se realizará la primera versión del análisis de requisitos, diseños
Análisis de requisitos	

Diseño conceptual	e implementación que darán lugar a una primitiva versión de la base de datos. Se detallará en los capítulos posteriores.
Diseño lógico	
Diseño físico	
Implementación	
Diseño Data Warehouse y repositorio estadístico	Autodescriptiva, se detallará en capítulos posteriores
Pruebas	Diseño e implementación de pruebas para esta iteración
Memoria TFG	Recopilación en la memoria de todo el trabajo realizado
Tercera entrega	
Revisión del trabajo realizado	Modificaciones al trabajo ya realizado debido a indicaciones del consultor o errores que puedan surgir
Diseño de la base de datos	Segunda iteración del diseño. Se revisarán diseños y requisitos presentados en la anterior entrega. Se implementará el repositorio estadístico acorde al Data Warehouse. Se diseñará e implementará aquellos elementos necesarios para el correcto funcionamiento de la base de datos. Se detallará en capítulos posteriores
Actualización diseño y requisitos anteriores	
Diseño e implementación de elementos adicionales	
Pruebas	Diseño e implementación de pruebas para esta iteración
Memoria TFG	Recopilación en la memoria de todo el trabajo realizado
Entrega final	
Revisión del trabajo realizado	Modificaciones al trabajo ya realizado debido a indicaciones del consultor o errores que puedan surgir
Pruebas	Diseño e implementación de un juego más exhaustivo de pruebas
Corrección de errores	Tarea dedicada a subsanar posibles errores encontrados
Memoria TFG	Terminar la memoria del TFG
Video y presentación	Video y presentación del trabajo
Informe de autoevaluación	Cumplimentar el documento de autoevaluación
Tribunal	
Defensa del trabajo	Defensa del trabajo, responder a las preguntas que se puedan realizar sobre éste.

1.4.3 Cronograma

Las tareas se ejecutarán de acuerdo con el siguiente cronograma. Para el cómputo de días se ha tenido en cuenta una relación de un día de trabajo equivale a 4 horas.

Nombre de tarea	Duración	Comienzo	Fin
TFG Base de datos	**348 h**	**17/02/22**	**17/06/22**
Primera entrega - Plan de trabajo	**52 h**	**17/02/22**	**07/03/22**
Lectura del tema propuesto	4 h	17/02/22	17/02/22
Plan de trabajo	48 h	18/02/22	07/03/22
Preparación del entorno de trabajo	4 h	07/03/22	07/03/22
Segunda entrega	**100 h**	**08/03/22**	**11/04/22**
Revisión del trabajo realizado	8 h	08/03/22	09/03/22
Diseño de la base de datos I	**60 h**	**10/03/22**	**30/03/22**
Análisis de requisitos	20 h	10/03/22	16/03/22
Diseño conceptual	20 h	17/03/22	23/03/22
Diseño lógico	4 h	24/03/22	24/03/22
Diseño físico	4 h	25/03/22	25/03/22
Implementación	12 h	28/03/22	30/03/22
Diseño de Data Warehouse y repositorio estadístico	12 h	31/03/22	04/04/22
Pruebas	8h	05/04/22	06/04/22
Memoria TFG	12 h	07/04/22	11/04/22
Tercera entrega	**92 h**	**12/04/22**	**12/05/22**
Revision del trabajo realizado	8 h	12/04/22	13/04/22
Diseño de la base de datos II	**64 h**	**14/04/22**	**05/05/22**
Actualización diseños y requisitos	12 h	14/04/22	18/04/22
Diseño e implementación elementos adicionales	52 h	19/04/22	05/05/22
Pruebas	8 h	06/05/22	09/05/22
Memoria TFG	12 h	10/05/22	12/05/22
Entrega final	**84 h**	**13/05/22**	**10/06/22**
Revisión del trabajo realizado	8 h	13/05/22	16/05/22
Pruebas	44 h	17/05/22	31/05/22
Corrección de errores	12 h	01/06/22	03/06/22
Memoria TFG	10 h	06/06/22	10/06/22
Video y presentación	5 h	06/06/22	10/06/22
Informe de autoevaluación	5 h	06/06/22	10/06/22
Tribunal	5 días	13/06/22	17/06/22

1.4.4 Diagrama de Gantt

Id	Modo de tarea	Nombre de tarea	Duración	Comienzo	Fin	Predecesoras
1		**TFG Base de datos**	**87 días**	**jue 17/02/22**	**vie 17/06/22**	
2		Inicio de proyecto	0 días	jue 17/02/22	jue 17/02/22	
3		Plan de trabajo	0 días	lun 07/03/22	lun 07/03/22	
4		Segunda entrega	0 días	lun 11/04/22	lun 11/04/22	
5		Tercera entrega	0 días	jue 12/05/22	jue 12/05/22	
6		Entrega final	0 días	vie 10/06/22	vie 10/06/22	
7		Tribunal	0 días	vie 17/06/22	vie 17/06/22	
8		**Primera entrega - Plan de trabajo**	**13 días**	**jue 17/02/22**	**lun 07/03/22**	
9		Lectura del tema propuesto	1 día	jue 17/02/22	jue 17/02/22	
10		Plan de trabajo	12 días	vie 18/02/22	lun 07/03/22	9
11		Preparación del entorno de trabajo	1 día	lun 07/03/22	lun 07/03/22	
12		**Segunda entrega**	**25 días**	**mar 08/03/22**	**lun 11/04/22**	**8**
13		Revisión del trabajo realizado	2 días	mar 08/03/22	mié 09/03/22	
14		**Diseño de la base de datos I**	**15 días**	**jue 10/03/22**	**mié 30/03/22**	**13**
15		Análisis de requisitos	5 días	jue 10/03/22	mié 16/03/22	
16		Diseño conceptual	5 días	jue 17/03/22	mié 23/03/22	15
17		Diseño lógico	1 día	jue 24/03/22	jue 24/03/22	16
18		Diseño físico	1 día	vie 25/03/22	vie 25/03/22	17
19		Implementación	3 días	lun 28/03/22	mié 30/03/22	18
20		Diseño de Data Warehouse	3 días	jue 31/03/22	lun 04/04/22	14
21		Pruebas	2 días	mar 05/04/22	mié 06/04/22	20
22		Memoria TFG	3 días	jue 07/04/22	lun 11/04/22	21
23		**Tercera entrega**	**23 días**	**mar 12/04/22**	**jue 12/05/22**	**12**
24		Revision del trabajo realizado	2 días	mar 12/04/22	mié 13/04/22	
25		**Diseño de la base de datos II**	**16 días**	**jue 14/04/22**	**jue 05/05/22**	**24**
26		Actualización diseños y requisitos	3 días	jue 14/04/22	lun 18/04/22	
27		Diseño e implementación de elementos adicionales	13 días	mar 19/04/22	jue 05/05/22	26
28		Pruebas	2 días	vie 06/05/22	lun 09/05/22	25
29		Memoria TFG	3 días	mar 10/05/22	jue 12/05/22	28
30		**Entrega final**	**21 días**	**vie 13/05/22**	**vie 10/06/22**	**23**
31		Revisión del trabajo realizado	2 días	vie 13/05/22	lun 16/05/22	
32		Pruebas	11 días	mar 17/05/22	mar 31/05/22	31
33		Corrección de errores	3 días	mié 01/06/22	vie 03/06/22	32
34		Memoria TFG	5 días	lun 06/06/22	vie 10/06/22	33
35		Video y presentación	5 días	lun 06/06/22	vie 10/06/22	33
36		Informe de autoevaluación	5 días	lun 06/06/22	vie 10/06/22	33
37		Tribunal	5 días	lun 13/06/22	vie 17/06/22	30

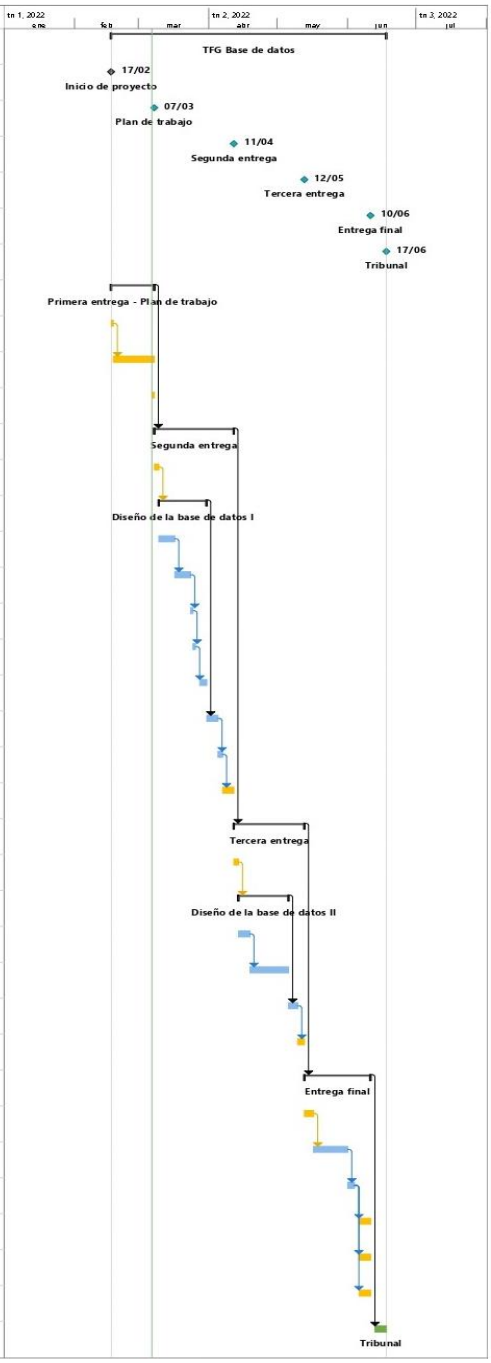


Ilustración 2. Diagrama de Gantt

1.4.5 Análisis de riesgos

A continuación, se detallan los riesgos que se han identificado para el desarrollo del proyecto, así como las acciones correctivas que se deberán realizar.

R-01	**Desviaciones en las tareas**
Descripción	Error al estimar los tiempos en el cronograma
Impacto	Alto
Probabilidad	Media
Acción de mitigación	A01 – Aprovechar los fines de semana para recuperar tiempo A02 – Aumentar la estimación de horas por día del cronograma A03 – Modificación del cronograma

R-02	**Problemas de salud**
Descripción	Retrasos debido a posible enfermedad
Impacto	Medio
Probabilidad	Media
Acción de mitigación	A01 – Aprovechar los fines de semana para recuperar tiempo A02 – Aumentar la estimación de horas por día del cronograma

R-03	**Problemas de hardware**
Descripción	Daños en el equipo que puedan suponer pérdida de información y/o la manera de continuar el trabajo
Impacto	Medio
Probabilidad	Baja
Acción de mitigación	A04 – Tener copias de seguridad en diferentes lugares, tanto en formato físico como la nube A05 – Contar con herramientas alternativas para poder continuar con el trabajo

R-04	**Problemas de software**
Descripción	Las herramientas seleccionadas no funcionan como se espera.
Impacto	Medio
Probabilidad	Baja
Acción de mitigación	A06 – Reinstalar o probar con otra versión A07 – Buscar otras herramientas que puedan cumplir con la función.

R-05	**Problemas de conexión**
Descripción	No se puede realizar la entrega o no se pueden realizar consultas para continuar con el trabajo
Impacto	Bajo
Probabilidad	Baja
Acción de mitigación	A08 – Uso de vías alternativas para recuperar conexión (uso del móvil como router, ir a casa de algún familiar…) A09 – Ponerse en contacto con el consultor

1.5 Breve sumario de productos obtenidos

- Scripts desarrollados para la creación de la base de datos que se pide para el desarrollo de la aplicación para la que nos ha contratado la empresa automovilística.
- Pruebas unitarias y juego de datos para la simulación de un caso para el testeo de todos los procedimientos, tanto ABM como los del repositorio estadístico, así como comprobar las consultas mínimas que se deben poder responder con el repositorio.
- La presente memoria.

1.6 Breve descripción de los otros capítulos de la memoria

Capítulo 2 Diseño de la base de datos. Este capítulo contiene las cinco fases que se detallan en el módulo de introducción al diseño de base de datos de J. Casas Romas. Estas son[2]:

- Recogida y análisis de requisitos. Conocer y analizar las expectativas, necesidades y los objetivos de los futuros usuarios de la base de datos.
- Diseño conceptual. Crear un esquema conceptual de alto nivel e independiente de la tecnología a partir de los requisitos, las especificaciones y las restricciones que se han recogido en la fase anterior.
- Diseño lógico. La transformación del modelo conceptual al modelo lógico dependiente del tipo de SGBD en el que se quiere implementar la base de datos.
- Diseño físico. Fase en que se adapta el esquema lógico al SGBD concreto que utilizará el sistema de información.
- Implementación y optimización. Realizar la carga de datos y posteriormente ajustar algunos parámetros relacionados con el modelo físico de la base de datos para optimizar el rendimiento.

Capítulo 3. Pruebas. En este capítulo se han recogido las diferentes pruebas, tanto unitarias como la del caso teórico, para comprobar el correcto funcionamiento de la base de datos.

Capítulo 4. Conclusiones. En este capítulo, como indica su nombre, se incluyen las conclusiones obtenidas de la realización del trabajo.

El resto de los capítulos corresponden al glosario, bibliografía y anexos.

[2] [2] J. Casas Romas, Módulo 1: Introducción al diseño de bases de datos

1.7 Seguimiento de la planificación

Primera entrega 07/03/22

Las tareas de la primera entrega se realizaron todas en plazo y sin incidentes.

Primera entrega - Plan de trabajo	**Estado**
Lectura del tema propuesto	Finalizado en plazo
Plan de trabajo	Finalizado en plazo
Preparación del entorno de trabajo	Finalizado en plazo

Segunda entrega 11/04/22

Durante la realización de la segunda entrega y pese a los planes de contingencia, debido a problemas de salud, no se pudo acabar todo lo planificado, dejando el diseño del repositorio estadístico sin finalizar.

Segunda entrega	**Estado**
Revisión del trabajo realizado	Finalizado en plazo
Diseño de la base de datos	Finalizado en plazo
Diseño Data Warehouse y repositorio estadístico	Inacabada
Pruebas	Finalizado en plazo
Memoria TFG	Finalizado en plazo

Tercera entrega 12/05/22

Los problemas de salud han continuado durante la entrega, se ha finalizado la tarea que quedó pendiente de la entrega anterior y el resto de las tareas que pertenecían a esta entrega.

Tercera entrega	**Estado**
Diseño Data Warehouse y repositorio estadístico	Finalizado
Revisión del trabajo realizado	Finalizado en plazo
Diseño de la base de datos	Finalizado en plazo
Pruebas	Finalizado en plazo
Memoria TFG	Inacabada

Entrega final 10/06/22

Para la entrega final no se ha tenido ningún problema y se ha podido finalizar todo a tiempo.

Entrega final	**Estado**
Memoria TFG – tercera entrega	Finalizado
Revisión del trabajo realizado	Finalizado en plazo

Pruebas	Finalizado en plazo
Corrección de errores	Finalizado en plazo
Memoria TFG – entrega final	Finalizado en plazo
Video y presentación	Finalizado en plazo
Informe de autoevaluación	Finalizado en plazo

2. Diseño de la base de datos

2.1 Recogida y análisis de requisitos

En esta primera fase del diseño de la base de datos nos centraremos en obtener los requisitos y restricciones de los datos del problema. En concreto nos centraremos en los requisitos que hacen referencia a la funcionalidad que debe proporcionar el sistema y los datos que tiene que conocer y guardar (requisitos funcionales) y aquellos que implican calidades esperadas del sistema, como la usabilidad, fiabilidad, rendimiento o mantenibilidad (requisitos no funcionales)[3]

2.1.1 Requisitos funcionales

Código	Requisito
RF01	Registrar los procesos de gestión interna de la empresa
RF02	Registrar y controlar las vulnerabilidades asociadas a los procesos de gestión de la empresa. La lista se irá ampliando en el tiempo.
RF03	Las vulnerabilidades deberán tener uno de los estados siguientes: identificada, no mitigada, parcialmente mitigada o totalmente mitigada. Además, deberán indicar si son críticas.
RF04	Registrar y controlar las acciones de mitigación asociadas a cada vulnerabilidad.
RF05	Las acciones de mitigación contarán como mínimo con un responsable, fecha límite de implantación y uno de los estados siguientes: definida, en proceso, acabada o en revisión
RF06	El estado de la lista de acciones de mitigación será la que controlará el estado de la vulnerabilidad: - 0 acciones acabadas = no mitigada - 1 acción acabada = parcialmente mitigada - Todas las acciones acabadas = totalmente mitigada
RF07	Registrar las políticas de seguridad de obligado cumplimiento para las personas y departamentos de la empresa
RF08	Registrar a los trabajadores de la empresa
RF09	Registrar a los departamentos de la empresa
RF10	Registrar y controlar los incumplimientos de las políticas de seguridad de la empresa

[3] [2] Módulo 1: Introducción a la ingeniería de requisitos

RF11	Registrar el número de incumplimientos por departamento y política de seguridad de la empresa
RF12	Llevar un registro exhaustivo de todas las sesiones de formación, tanto telemáticas como presenciales que se realicen en la empresa referentes a temas de seguridad. Se registrarán tanto las sesiones como los usuarios que participan
RF13	Registrar y controlar diferentes vías para detectar incumplimientos, como por ejemplo simulaciones de ataque
RF14	Referente al RF13, gestionar el control de las diferentes auditorias de seguridad definidas por la empresa. Éstas estarán ligadas a las diferentes políticas definidas y aprobadas por la empresa. Pueden ser realizadas por equipos internos o externos y analizarán los diferentes procesos de gestión para detectar incumplimientos.
RF15	Se deben poder guardar todos los muestreos que se realicen durante las auditorias.
RF16	El repositorio estadístico como mínimo deberá responder a las siguientes consultas: - Departamento que, en un año concreto, tiene un número mayor de incumplimientos de seguridad registrados en la BD - Proceso de gestión interno que, teniendo en cuenta toda la información de la que se dispone en la BD, ha tenido un mayor número de vulnerabilidades detectadas - Top5 de usuarios por número de incumplimientos asociados directamente a ellos, o a su departamento, durante el año en curso - Porcentaje de vulnerabilidades que, en el momento de hacer la consulta, están totalmente mitigadas - Número total de acciones de mitigación que, en el momento de hacer la consulta, no están totalmente acabadas - Política de seguridad que, en el momento de hacer la consulta, ha tenido más incumplimientos (teniendo en cuenta todos los departamentos de la empresa) - Dado un determinado departamento de la empresa, y teniendo en cuenta el momento de ejecutar la consulta, porcentaje de usuarios del departamento que no han acabado todas las formaciones de seguridad asignadas - Porcentaje de usuarios de la empresa que, en el año en curso, no tienen ningún incumplimiento asignado - Teniendo en cuenta todas las auditorías externas realizadas, año en el cual se han detectado más incumplimientos (teniendo en cuenta únicamente los detectados durante la

	auditoría) - Porcentaje de vulnerabilidades críticas que, en el momento de hacer la consulta, tienen alguna acción de mitigación abierta (que no esté en estado “acabada”) - Teniendo en cuenta el último año (el anterior al año en curso, que es cuando se ejecuta la consulta), título de la sesión formativa telemática que ha tenido un porcentaje menor de participantes en total. - Número de vulnerabilidades críticas detectadas internamente teniendo en cuenta todos los datos de que se dispone. Se consideran detectadas internamente si se detectaron en posterioridad al análisis realizado inicialmente por la consultora externa - En el momento de hacer la consulta, porcentaje de acciones de mitigación en el sistema que están en los estados “en proceso” o “en revisión” - Teniendo en cuenta todas las acciones de mitigación en estado “en proceso”, persona responsable con más acciones asignadas
RF17	En la tabla de log se deberán almacenar todas las llamadas a procedimientos que se hagan, almacenando el procedimiento ejecutado y los parámetros de entrada y salida.

2.1.2 Requisitos no funcionales

Código	Requisito
RNF01	La base de datos debe ser relacional
RNF02	El SGBD que se utilizará será Oracle Database
RNF03	Los incumplimientos de las políticas de seguridad deberán tener en cuenta todas las restricciones de confidencialidad que deban seguir estas acciones[4]
RNF04	Toda la gestión y acceso a la información se realizará mediante procedimientos de BD, siendo ésta la única manera de acceder.
RNF05	A nivel de procedimientos, será necesario implementar y describir en detalle los procedimientos ABM (Alta, baja, modificación) de todas las entidades que se consideren relevantes

[4] [3] Ley Orgánica 3/2018, de 5 de diciembre, de Protección de Datos Personales y garantía de los derechos digitales.

RNF06	La aplicación debe servir para cualquier volumen de datos, por lo que la gestión de los datos almacenados se debe hacer siguiendo técnicas que se aplican a grandes volúmenes de datos (Data Warehouse)
RNF07	El repositorio estadístico tendrá que ofrecer los diferentes resultados en tiempo constante 1
RNF08	La base de datos deberá ser escalable
RNF09	Para facilitar el mantenimiento del sistema, se deberá disponer de mecanismos que permitan resolver potenciales problemas de integración con el resto del sistema informático de la empresa: un log de las acciones realizadas con la BD, mecanismos para el testeo de la funcionalidad de la BD…
RNF10	Para estandarizar el log, los procedimientos almacenados deben cumplir las siguientes condiciones: - Como mínimo dispondrán de un parámetro de salida llamado RSP, de tipo String, que indicará si la ejecución ha finalizado con éxito (valor 'OK') o si ha fracasado (valor 'ERROR+TIPO DE ERROR'). - Dispondrán de tratamiento de excepciones

2.2 Diseño conceptual

Una vez se ha realizado la recogida y análisis de requisitos, se pasa a la fase del diseño conceptual. Ésta tiene como objetivo crear un esquema conceptual de alto nivel e independiente de la tecnología a partir de los requisitos, las especificaciones y las restricciones que se han escogido.

Un esquema conceptual es una descripción concisa de los requisitos de datos por parte de los usuarios e incluye descripciones detalladas de las entidades que están involucradas, las relaciones entre estas entidades y las restricciones de integridad que tienen.

Se ha realizado el diseño del esquema mediante el modelo ER utilizando el lenguaje UML. Se ha seguido una metodología centralizada en la que aparecen todas las entidades, aunque se deben diferenciar 4 módulos para las entidades:

- Datos de la empresa. Contendrán los datos que pertenecen a la empresa automovilística. En un caso diferente, en que la empresa existiera y tuviera una base de datos, se conectarían ambas, ya se importando de manera temporizada la información actualizada o que ambas estén conectadas a tiempo real. Aparecen en color verde en el diseño conceptual.

- Entidades de la aplicación. Consisten en aquellas entidades necesarias para poder desarrollar la aplicación del proyecto. Aparecen en color amarillo en el diseño conceptual.

- Log. Tabla reservada para la auditoría de los procedimientos. Aparece en color azul en el diseño conceptual.

- Repositorio estadístico. Contendrán los datos que se solicitan en las consultas que se mencionan en el RF16. Aparece en color morado en el diseño conceptual.

2.2.1 Diseño conceptual de los datos de la empresa y las entidades propias de la aplicación.

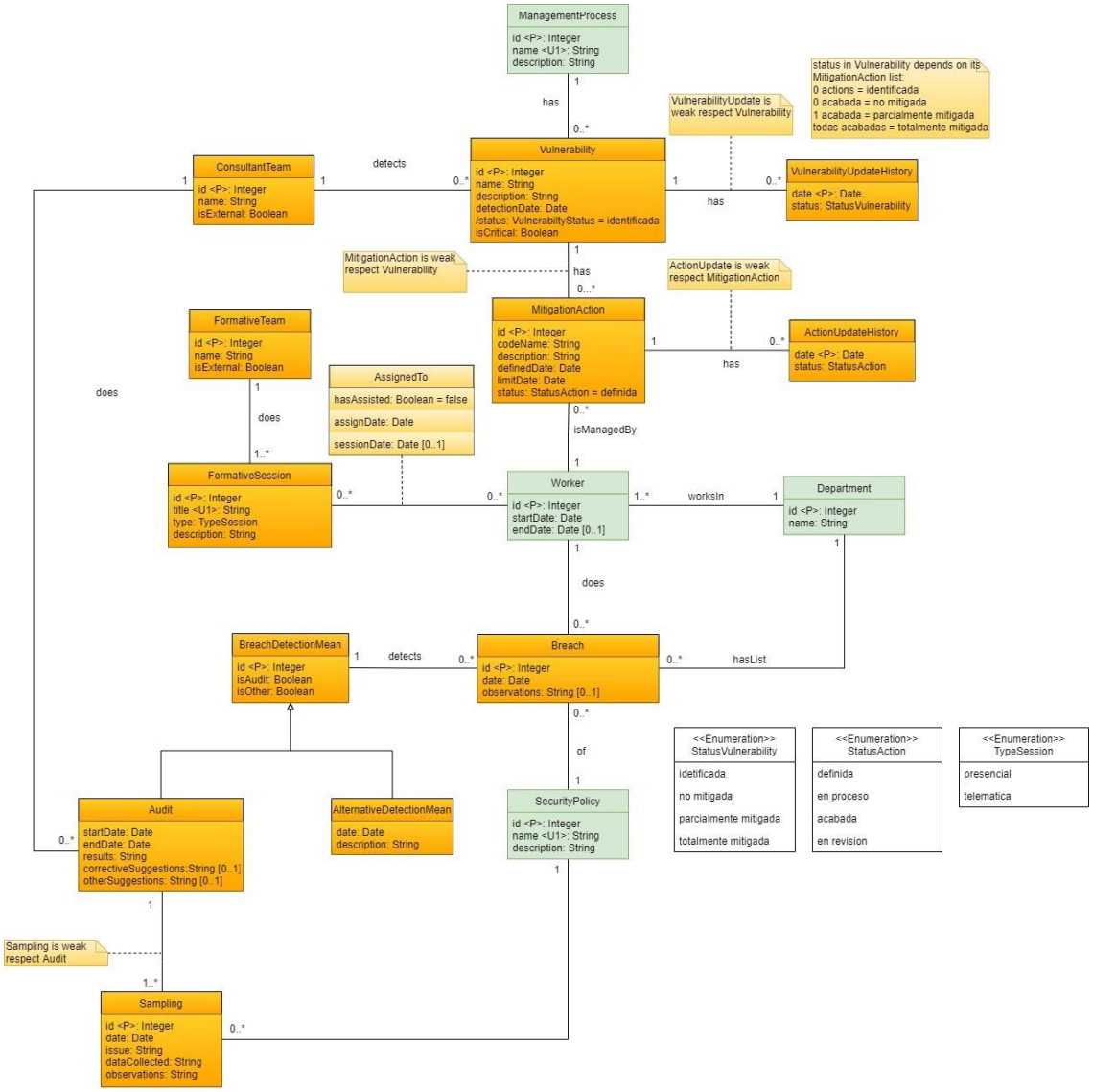


Ilustración 3. Diseño conceptual. Datos empresa y aplicación

Decisiones de diseño

A continuación, se explicará el diseño conceptual realizado tabla por tabla. Debido a que tanto las tablas que contendrán los datos que debe proporcionar la empresa como las tablas que contendrán los datos necesarios para el funcionamiento de la aplicación están relacionadas, se han mostrado en la misma imagen. En el caso del log de procedimientos y de las tablas para el repositorio estadístico, se mostrarán por separado para su posterior explicación.

2.2.1.1 Datos de empresa

Como se ha comentado, estas tablas contienen toda la información que se importará de la base de datos de la empresa. Ya que no se tiene información real de qué tipo de datos contendrán, se supondrá que existen al menos los definidos y del tipo indicado.

ManagementProcess

Entidad que contendrá los diferentes procesos de gestión interna que utiliza la empresa y que se han analizado en busca de vulnerabilidades.

Atributo	Tipo	Descripción
id	Integer	Clave primaria de la tabla. Número que identifica inequívocamente el registro en la tabla.
name	String	Nombre del proceso de gestión interna. Se ha supuesto que no habrá duplicidad de nombres.
description	String	Breve texto adicional descriptivo del proceso de gestión interna.

Se relaciona con las siguientes tablas:

- Vulnerability para indicar el listado de vulnerabilidades asociadas al proceso de gestión interna. Multiplicidad: 1-0..*
- Sampling para indicar el listado de muestreos que se han realizado sobre el proceso de gestión interna en diferentes auditorías. Multiplicidad: 1-0..*

Worker

Entidad que contendrá los diferentes trabajadores de la empresa

Atributo	Tipo	Descripción
id	Integer	Clave primaria de la tabla. Número que identifica inequívocamente el registro en la tabla. Idealmente coincidirá con el número o código de empleado. En este caso se ha supuesto que número.
startDate	Date	Fecha en la que comenzó a trabajar en la empresa.
endDate	Date	Opcional. Atributo que marca la fecha en que el acabó de trabajar en la empresa.

Se relaciona con las siguientes tablas:

- Department para indicar el departamento en el que trabaja. Multiplicidad: 1-1..*
- Breach para indicar el listado de incumplimientos del trabajador. Multiplicidad: 1-0..*
- AssignedTo para indicar el listado de sesiones formativas asignadas al trabajador. Multiplicidad: 1-0..*
- MitigationAction para indicar el listado de acciones de mitigación de las

cuales el trabajador es responsable. Multiplicidad: 1-0..*

Department

Entidad que contendrá los diferentes departamentos existentes en la empresa.

Atributo	Tipo	Descripción
id	Integer	Clave primaria de la tabla. Idealmente coincidirá con el número o código del departamento. En este caso se ha supuesto que número.
name	String	Nombre del departamento.

Se relaciona con las siguientes tablas:

- Worker para indicar el listado de trabajadores que pertenecen al departamento. Multiplicidad: 1-1..*
- Breach para indicar todos los incumplimientos del departamento. Multiplicidad: 1-0..*

SecurityPolicy

Entidad que contendrá las diferentes políticas de seguridad que ha establecido la empresa.

Atributo	Tipo	Descripción
id	Integer	Clave primaria de la tabla. Número que identifica inequívocamente el registro en la tabla.
name	String	Nombre de la política de seguridad. Se ha supuesto que no habrá duplicidad de nombres y por eso se ha marcado como única.
description	String	Breve texto adicional descriptivo de la política de seguridad.

Se relaciona con las siguientes tablas:

- Sampling para indicar los muestreos que se han realizado sobre cada política de seguridad. Multiplicidad: 1-0..*
- Breach para indicar el listado de incumplimientos sobre cada política de seguridad. Multiplicidad: 1-0..*

2.2.1.2 Entidades de la aplicación

Tablas que contendrán las entidades necesarias para el correcto funcionamiento de la aplicación. Son las siguientes:

ConsultantTeam

Esta entidad contendrá las diferentes consultorías o equipos internos que se encargarán de realizar tanto los análisis de seguridad para la detección de vulnerabilidades como las auditorías sobre las diferentes políticas de seguridad.

Respecto a la consultora que realiza la búsqueda de vulnerabilidades, a pesar de que se nos indica que únicamente habrá una externa y se realizarán mediante equipos internos, se incluye relación de vulnerabilidad con esta tabla, de cara a que, si quieren realizar otros análisis externos, no supondría un gran trabajo para incluirlas.

Atributo	Tipo	Descripción
id	Integer	Clave primaria de la tabla. Número que identifica inequívocamente el registro en la tabla.
name	String	Nombre del equipo o empresa al que corresponde la entrada
isExternal	Boolean	Atributo de verdadero/falso para indicar si la empresa o equipo indicado es externo o interno.

Se relaciona con las siguientes tablas:

- Audit para indicar el listado de auditorías sobre las políticas de seguridad realizadas. Multiplicidad: 1-0..*
- Vulnerability para las vulnerabilidades encontradas. Multiplicidad: 1-0..*

Vulnerability

Esta entidad contendrá aquellas vulnerabilidades que se encuentren en los diferentes análisis (tanto externos como internos) que se realicen sobre los procesos de gestión interna de la empresa.

Atributo	Tipo	Descripción
id	Integer	Clave primaria de la tabla. Número que identifica inequívocamente el registro en la tabla.
name	String	Nombre o código dado a la vulnerabilidad
description	String	Descripción de la vulnerabilidad
detectionDate	Date	Fecha en que se ha detectado la vulnerabilidad
status	Enumerado	El estado de las vulnerabilidades, tal como

		se nos indica en el enunciado corresponderá a una de las siguientes: - Identificada. Estado por defecto, sin acciones de mitigación asignadas. - No mitigada. Si tiene acciones de mitigación, pero ninguna está acabada. - Parcialmente mitigada. Si tiene al menos 1 pero no todas acciones de mitigación acabadas. - Totalmente mitigada. Si todas sus acciones de mitigación están acabadas. Debido a que el estado de la vulnerabilidad depende de sus acciones de mitigación, el estado por defecto de ésta cuando se añade a la base de datos será "identificada" y volverá a este estado si por algún motivo deja de tener acciones de mitigación.
isCritical	Boolean	Atributo de verdadero/falso para indicar si la vulnerabilidad es crítica.

Se relaciona con las siguientes tablas:

- ConsultantTeam para indicar el equipo/empresa que la ha detectado Multiplicidad: 0..*-1
- ManagementProcess para indicar a qué proceso pertenece Multiplicidad: 0..*-1
- VulnerabilityUpdateHistory listado de cambios de status Multiplicidad: 1-0..*
- MitigationAction listado de acciones de mitigación a realizar Multiplicidad: 1-0..*

VulnerabilityUpdateHistory

Esta entidad mantendrá un historial de cada cambio de estado que se realice sobre las vulnerabilidades. Por ejemplo, si pasa de no mitigada a parcialmente mitigada, se registrará éste en la entidad.

Atributo	Tipo	Descripción
date	Date	Clave primaria de la tabla junto con el id de la vulnerabilidad. Fecha y tiempo en que se realiza el cambio en vulnerabilidad.
status	Enumerado	Debido a que registra los cambios de estado de la vulnerabilidad, este atributo, contendrá el nuevo estado al que cambiará la vulnerabilidad.

Se relaciona con las siguientes tablas:

- Vulnerability indica el listado de cambios que ha habido para cada

vulnerabilidad. Multiplicidad: 0..*-1

FormativeTeam

Esta entidad registrará todos los equipos y empresas que realizarán las sesiones formativas sobre seguridad que realizará la empresa.

Atributo	Tipo	Descripción
id	Integer	Clave primaria de la tabla. Número que identifica inequívocamente el registro en la tabla.
name	String	Nombre del equipo o empresa al que corresponde la entrada
isExternal	Boolean	Atributo de verdadero/falso para indicar si la empresa o equipo indicado es externo o interno.

Se relaciona con las siguientes tablas:

- FormativeSession para indicar el equipo/empresa que realiza la sesión formativa. Multiplicidad: 1-1..*

FormativeSession

Esta entidad registra todas las sesiones formativas que realizará la empresa para sus empleados.

Atributo	Tipo	Descripción
id	Integer	Clave primaria de la tabla. Número que identifica inequívocamente el registro en la tabla.
title	String	Clave alternativa para identificar a la tabla. Título de la sesión formativa. Las asignaciones a las sesiones formativas se registrarán en otra entidad, debido a esto, se guardará cada sesión formativa como única, aunque se pueda realizar en varias ocasiones o incluso que consistan en sesiones en diferido o cursos con tests al final que no tengan una fecha específica de realización.
type	Enumerado	Indica el tipo de sesión. Tal como se nos indica, las sesiones podrán ser: - Telematica - Presencial
Description	String	Breve descripción acerca de la sesión

Se relaciona con las siguientes tablas:

- FormativeSession para indicar el equipo/empresa que realiza la sesión formativa. Multiplicidad: 1-1..*

AssignedTo

Esta entidad registrará todas las asignaciones de los trabajadores a las diferentes sesiones de formación que se realizarán.
No se ha contemplado el caso de que un trabajador tenga que renovar la formación o que se asigne de nuevo una misma sesión formativa al mismo trabajador.

Atributo	Tipo	Descripción
hasAssisted	Boolean	Indica si el trabajador ha asistido a la sesión de formación, ya sea telemática o presencial, o, si fuera el caso, si ha cumplido con todas las condiciones en otros tipos de curso o sesiones formativas.
assignedDate	Date	Fecha en que la empresa ha asignado al trabajador la sesión formativa.
sessionDate	Date	Opcional. Para sesiones formativas presenciales o telemáticas (no en diferido), fecha en que se realizará la sesión que se le ha asignado al trabajador.

Se relaciona con las siguientes tablas:

- Worker y FormativeSession. La entidad en sí se considera un listado que relaciona sesiones formativas con los empleados. Con ambas se relaciona con multiplicidad 0..*-1.

MitigationAction

Esta entidad registrará todas las acciones de mitigación que se realizarán sobre las vulnerabilidades encontradas.

Atributo	Tipo	Descripción
id	Integer	Clave primaria de la tabla junto con el id de la vulnerabilidad. Número que identifica inequívocamente el registro en la tabla.
codeName	String	Código o nombre que se le dará a la acción de mitigación.
description	String	Descripción de la acción de mitigación
definedDate	Date	Fecha en que se ha definido la acción de mitigación.
limitDate	Date	Fecha límite para la ejecución de la acción de mitigación.
status	Enumerado	Las acciones de mitigación pueden tener uno de los siguientes estados: - Definida

		- En proceso - Acabada - En revisión Por defecto, una acción de mitigación comenzará con el estado “definida” a menos que se le indique otro.

Se relaciona con las siguientes tablas:

- Vulnerability. Como se ha comentado, se relaciona indicando a qué vulnerabilidad pertenece cada acción de mitigación. Multiplicidad: 0..*-1
- Worker. Cada acción de mitigación tendrá algún empleado de la empresa como responsable. Multiplicidad: 0..*-1
- ActionUpdateHistory. La relación tiene la misma función que en VulnerabilityUpdateHistory en que mantiene un historial de cambios de estado de las acciones de mitigación. Multiplicidad: 1-0..*

ActionUpdateHistory

Esta entidad registrará los cambios que se realicen sobre el estado de las acciones de mitigación. Ésta permite conocer el momento y estado por los que ha pasado una acción de mitigación.

Atributo	Tipo	Descripción
date	Date	Clave primaria de la tabla junto con el id de la vulnerabilidad y el id de la acción de mitigación. Fecha y tiempo en que se realiza el cambio en la acción de mitigación
status	Enumerado	Debido a que registra los cambios de estado de la acción de mitigación, este atributo, contendrá el nuevo estado al que cambiará la acción de mitigación

Se relaciona con las siguientes tablas:

- MitigationAction. Como se ha indicado anteriormente, esta relación permite establecer un listado de los cambios que ha sufrido una acción de mitigación. Multiplicidad: 0..*-1

Breach

En esta entidad se registrarán los diferentes incumplimientos que realicen los empleados de las diferentes políticas de seguridad que establecerá la empresa.

Atributo	Tipo	Descripción
id	Integer	Clave primaria de la tabla. Número que identifica inequívocamente el registro en la tabla.
date	Date	Fecha en que se produjo el incumplimiento.
observations	String	Pequeño espacio opcional donde incluir

		observaciones o una breve descripción.

Se relaciona con las siguientes tablas:

- SecurityPolicy. Los incumplimientos son sobre las políticas de seguridad definidas por la empresa, la relación sirve para identificar cuál. Multiplicidad: 0..*-1
- Department y Worker. Ambos se relacionan con Breach para indicar qué empleado y qué departamento son responsables del incumplimiento. Multiplicidad de ambos: 0..*-1
- BreachDetectionMean. La relación sirve para indicar el método que se ha utilizado para descubrir el incumplimiento, ya sea una auditoria, un simulacro de ataque o de casualidad. Multiplicidad 0..*-1

BreachDetectionMean

Esta entidad sirve de generalización de los métodos de detección de incumplimientos que se han utilizado. Posteriormente se especializa en otras dos entidades: Audit y AlternativeDetectionMean, que se detallarán en breve.

Atributo	Tipo	Descripción
id	Integer	Clave primaria de la tabla. Número que identifica inequívocamente el registro en la tabla.
isAudit	Boolean	Indican si el método es una auditoria u otro. Es excluyente con isOther, no pueden tener ambos el mismo valor al mismo tiempo y al menos uno de ellos deberá ser cierto. Pequeño espacio opcional donde incluir observaciones o una breve descripción.
isOther	Boolean	

Se relaciona con las siguientes tablas:

- Breach. Como se ha comentado, se relaciona con los incumplimientos para indicar la manera en que se han detectado. Multiplicidad 1-0..*

AlternativeDetectionMean

En esta entidad se registrarán todos los diferentes métodos que no sean una auditoría por el que se han detectado incumplimientos. Por ejemplo, se incluirían las simulaciones de ataque que se mencionan en el enunciado.

Atributo	Tipo	Descripción
date	Date	Fecha en que se produjo el incumplimiento.
description	String	Espacio donde indicar todos los detalles que se necesiten sobre el método de detección.

No se relaciona directamente con ninguna tabla

Audit

En esta entidad se registrarán todas las auditorias (tanto internas como externas) que realice la empresa sobre sus políticas de seguridad.

Atributo	Tipo	Descripción
startDate	Date	Fecha de inicio de la auditoría
endDate	Date	Fecha en que finaliza la auditoría
results	String	Espacio donde incluir los resultados obtenidos de la auditoría
correctiveSuggestions	String	Opcional. Espacio donde incluir posibles sugerencias para corregir problemas encontrados.
otherSuggestions	String	Opcional. Espacio donde incluir sugerencias no correctivas

Se relaciona con las siguientes tablas:

- ConsultantTeam. La relación sirve para indicar el equipo que se encargó de la auditoria (ya sea interno o externo). Multiplicidad 0..*-1
- Sampling. La relación sirve para indicar la lista de muestreos que se realizan durante la auditoría. Multiplicidad 1-1..*

Sampling

En esta entidad se registrarán los muestreos que se realicen durante una auditoría de las diferentes políticas de seguridad.

Atributo	Tipo	Descripción
id	Integer	Clave primaria de la tabla. Número que identifica inequívocamente el registro en la tabla.
date	Date	Fecha en que se realiza el muestreo
issue	String	Tema que estudia el muestro
dataCollected	String	Espacio donde incluir la información relevante obtenida durante el muestreo.
observations	String	Espacio donde incluir alguna observación

Se relaciona con las siguientes tablas:

- SecurityPolicy. La relación permite identificar la política de seguridad sobre la que se realiza el muestreo. Multiplicidad 0..*-1
- Audit. Como se ha indicado, indica el listado de muestreos que pertenecen a la auditoría con la que se relaciona. Multiplicidad 1..*-1

2.2.2 Diseño conceptual log

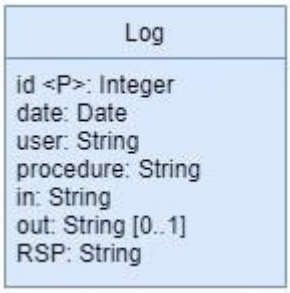


Ilustración 4. Diseño conceptual log

Log		
Como se indica en el enunciado y en se recoge como requisito RF17. Las diferentes llamadas a los procedimientos que se implementen tendrán que registrarse en esta tabla, indicando al menos, el procedimiento, los parámetros de entrada y salida y, dentro de los parámetros de salida, como mínimo uno llamado RSP.		
Atributo	**Tipo**	**Descripción**
id	Integer	Clave primaria, identificador de la entrada al log.
date	Date	Fecha en que se realiza la llamada al procedimiento
user	String	Identificador del usuario que ha realizado la llamada al procedimiento
procedure	String	Identificador del procedimiento que se ha llamado.
in	String	Parámetros de entrada del procedimiento
out	String	Opcional. Parámetros de salida del procedimiento, si los tuviera.
RSP	String	Parámetro de salida mínima obligatorio que tendrán los procedimientos. Para estandarizarlos, como mínimo tendrá los siguientes requisitos, detallados en el RNF10: - Como mínimo dispondrán de un parámetro de salida llamado RSP, de tipo String, que indicará si la ejecución ha finalizado con éxito (valor 'OK') o si ha fracasado (valor 'ERROR+TIPO DE ERROR').
No se relaciona directamente con ninguna tabla		

2.2.3 Diseño conceptual DW

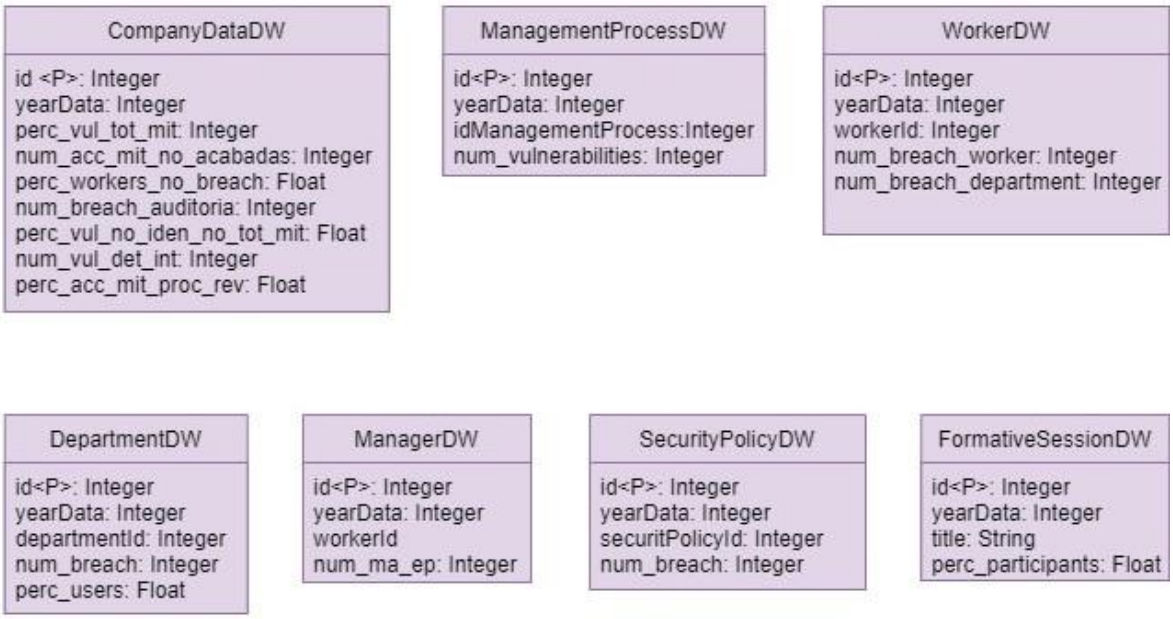


Ilustración 5. Diseño conceptual repositorio estadístico

Tal como se nos indica en el enunciado y se recoge en los RNF06 y RNF07, la aplicación debe servir para cualquier volumen de datos, por lo que la gestión de los datos almacenados se debe hacer siguiendo técnicas que se aplican a grandes volúmenes de datos (Data Warehouse) grandes, además, las consultas que se realicen sobre el repositorio estadístico deben poder realizarse en tiempo constante 1.

Un data warehouse extrae datos de fuentes existentes (la base de datos), especifica un conjunto de reglas para transformarlos y los carga en un repositorio central para que se pueda acceder a ellos y controlarlos rápidamente. [5]

Generalmente, los data warehouse van acompañados de otras herramientas, como servidores en la nube, servidores OLAP y otras herramientas y APIs que el cliente pueda necesitar para procesar los datos.

Debido a que, en nuestro caso, no se tienen ni servidores en la nube, ni servidores OLAP ni herramientas o APIs externas que nos puedan ayudar con el estudio estadístico, únicamente se considerará la versión básica. En que tendremos un repositorio estadístico, en que diferenciamos los datos por temas específicos y los tratamos por separado.

En concreto, se han generado 7 tablas que contendrán los datos procesados de la base de datos. Estas tablas contendrán la información necesaria para

poder dar respuesta a las consultas que se indican en el RF16. Las tablas se rellenarán mediante un procedimiento específico para cada tabla.

A pesar de que se aconseja nunca modificar los datos de las tablas, se han concebido con la intención de poder manipular los datos hasta el fin del año en curso, es decir, la información que contendrán será correcta para el año en curso, pero si se intenta modificar la de años anteriores, los datos pueden no reflejar la realidad en ese momento.
Una manera de asegurar que los datos están al día sería mediante el uso de temporizadores, ya sea desde el mismo SGBD o desde la aplicación que se tiene que implementar. Los periodos quedarían a elección de la empresa.

Tablas de hechos del repositorio estadístico

CompanyDataDW		
Esta entidad contendrá aquellos datos que no pertenezcan a información global de la empresa, como por ejemplo el número de acciones de mitigación que no están acabadas. Es decir, aquella información numérica que no depende de otra entidad.		
Atributo	**Tipo**	**Descripción**
id	Integer	Identificador
yearData	Integer	Año al que pertenecen los datos
perc_vul_tot_mit	Float	Este atributo almacena el porcentaje de vulnerabilidades totalmente mitigadas.
num_acc_mit_no_acabadas	Integer	Este atributo almacena el número de acciones de mitigación no acabadas
perc_workers_no_breach	Float	Este atributo almacena el porcentaje de trabajadores que en el año en que se introdujeron los datos no tenían ningún incumplimiento asignado.
num_breach_auditoria	Integer	Este atributo almacena el número de incumplimientos que se han detectado mediante una auditoría externa.
perc_vul_no_iden_no_tot_mit	Float	Este atributo almacena el porcentaje de vulnerabilidades que tienen alguna acción de mitigación abierta, es decir,

[5] [6] https://insightsoftware.com/es/blog/database-vs-data-warehouse-whats-the-difference/

		aquellas vulnerabilidades que no están en los estados: identificada o totalmente mitigada.
num_vul_det_int	Integer	Este atributo almacena el número de vulnerabilidades detectadas internamente.
perc_acc_mit_proc_rev	Float	Este atributo almacena el porcentaje de acciones de mitigación que se encuentran en los estados: en proceso o en revisión.
No se relaciona directamente con ninguna tabla		

ManagementProcessDW

Esta entidad contendrá los datos estadísticos que corresponden a un proceso de gestión interna específico. En este caso se incluye un único proceso de gestión interna por año (modificable hasta antes de que cambie el año). Se mira la información acumulada, es decir, aquel proceso de gestión interna que desde se creó la base de datos ha tenido más vulnerabilidades detectadas.

Atributo	Tipo	Descripción
id	Integer	Clave primaria, identificador de la entrada al log.
yearData	Integer	Año al que pertenecen los datos
idManagementProcess	Integer	Número que identifica al proceso de gestión interna
num_vulnerabilities	Integer	Este atributo almacena el número de vulnerabilidades detectadas para el proceso de gestión interna
No se relaciona directamente con ninguna tabla		

WorkerDW

Esta entidad contendrá los datos estadísticos que corresponden a todos los empleados. Cada año se incluyen todos los empleados, junto con su número de incumplimientos (personal y de departamento), para poder obtener el top5 de usuarios.

Atributo	Tipo	Descripción
id	Integer	Clave primaria, identificador de la entrada al log.
yearData	Integer	Año de los datos
workerId	Integer	Identificador del trabajador

num_breach_worker	Integer	Número de incumplimientos realizados por el propio trabajador.
num_breach_department	Integer	Número de incumplimientos del departamento para el que trabaja
No se relaciona directamente con ninguna tabla		

DepartmentDW

Esta entidad contendrá los datos estadísticos específicos de cada departamento, en concreto, el número de incumplimientos, así como el porcentaje de trabajadores de éste que no han acabado todas las formaciones que tienen asignadas.

Atributo	Tipo	Descripción
id	Integer	Clave primaria, identificador de la entrada al log.
yearData	Integer	Año de los datos
departmentId	Integer	Identificador del departamento
num_breach	Integer	Número de incumplimientos realizados por el departamento.
perc_users	Integer	Porcentaje de trabajadores del departamento que para un año en concreto no han asistido a todas sus formaciones.
No se relaciona directamente con ninguna tabla		

ManagerDW

A pesar de que los responsables de las acciones de mitigación son trabajadores, se ha preferido especializar la tabla, para incluir únicamente aquellos que son responsables de acciones de mitigación en proceso. En concreto, en esta entidad, de momento, se almacena el responsable por año con más acciones de mitigación en proceso.

Atributo	Tipo	Descripción
id	Integer	Clave primaria, identificador de la entrada al log.
yearData	Integer	Año de los datos
workerId	Integer	Identificador del trabajador
num_ma_ep	Integer	Número de acciones de mitigación de las que es responsable y están en proceso
No se relaciona directamente con ninguna tabla		

SecurityPolicyDW		
Esta tabla de hechos está especializada para almacenar datos que correspondan a una política de seguridad. En este caso, se almacena cada año la política de seguridad con más incumplimientos, teniendo en cuenta toda la información contenida en la base de datos.		
Atributo	**Tipo**	**Descripción**
id	Integer	Identificador para la tabla
yearData	Integer	Año de los datos
securityPolicyId	Integer	Identificador de la política de seguridad
num_breach	Integer	Número total de incumplimientos que se tienen registrados en la base de datos de la política de seguridad.
No se relaciona directamente con ninguna tabla		

FormativeSessionDW		
Esta tabla de hechos está especializada para almacenar datos que correspondan a las diferentes sesiones formativas. En este caso únicamente almacenamos todas las sesiones formativas telemáticas junto con el porcentaje de participantes a las sesiones. Debido a como se han contemplado las sesiones, que pueden consistir también en cursos y sesiones en diferido, se almacenan todas las sesiones por año y se mira el porcentaje de asistencia de ese año. Se ha puesto por defecto que el porcentaje es 100% para aquellas que no tienen ningún trabajador asignado ese año. Esto es debido a que únicamente se consulta aquella sesión con el menor número de participantes.		
Atributo	**Tipo**	**Descripción**
id	Integer	Identificador para la tabla
yearData	Integer	Año de los datos
title	String	Título de la sesión formativa
perc_participants	Float	Porcentaje de usuarios que han asistido a la sesión formativa.
No se relaciona directamente con ninguna tabla		

2.3 Diseño lógico

En la fase de diseño lógico se transforma el modelo conceptual, independientemente del tipo de tecnología, en un modelo lógico dependiente del tipo de SGBD en el que se quiere implementar la base de datos.
Seguiremos la notación siguiente[6]:

- Denotaremos las relaciones a partir del nombre, seguido de la lista de atributos entre paréntesis y separados por comas.
- Denotaremos las claves primarias subrayando con una línea continua los atributos que la forman.
- Denotaremos las claves alternativas subrayando con una línea discontinua los atributos que las forman.
- Utilizaremos el tipo de letra negrita en los nombres de atributo que queremos declarar NOT NULL.
- Para indicar las claves foráneas, lo indicaremos debajo de cada relación.

El diseño lógico será el siguiente:

ManagementProcess (id, **name**, **description**)

ConsultantTeam (id, **name**, **isExternal**)

Vulnerability (id, **idManagementProcess**, **name**, **description**, **detectionDate**, **status**, **isCritical**, **idConsultantTeam**)
idManagementProcess es clave foránea de ManagementProcess(id)
idConsultantTeam es clave foránea de ConsultantTeam(id)

VulnerabilityUpdateHistory (vulnerabilityId, updateDate, **status**)
vulnerabilityId es clave foránea de Vulnerability(id)

Department (id, **name**)

Worker (id, **startDate**, endDate, **departmentId**)
departmentId es clave foránea de Department(id)

MitigationAction (id, vulnerabilityId, **codeName**, **description**, **definedDate**, **limitDate**, **status**, **managerId**)
vulnerabilityId es clave foránea de Vulnerability(id)
managerId es clave foránea de Worker(id)
ActionUpdateHistory (actionId, updateDate, **status**)

actionId es clave foránea de MitigationAction(id)

BreachDetectionMean (id, **isAudit, isOther**)

Audit (id, **startDate, endDate, results,**
correctiveSuggestions, otherSuggestions,
consultantTeamId)
id es clave foránea de BreachDetectionMean(id)
consultantTeamId es clave foránea de ConsultantTeam

Sampling (id, auditId, **samplingDate, issue, dataCollected, observations**)
auditId es clave foránea de Audit

AlternativeDetectionMean (id, **date, description**)
id es clave foránea de BreachDetectionMean(id)

SecurityPolicy (id, name, **description**)

Breach (id, **breachDate, observations, workerId,**
departmentId, policyId, breachDetectionMeanId)
workerId es clave foránea de Worker(id)
departmentId es clave foránea de Department(id)
policyId es clave foránea de SecurityPolicy(id)
breachDetectionMeanId es clave foránea de
BreachDetectionMean(id)

FormativeTeam (id, **name, isExternal**)

FormativeSession (id, title, **type, description,**
formativeTeamId)
formativeTeamId es clave foránea de FormativeTeam

AssignedTo (workerId, formativeSessionId, **hasAssisted,**
assignDate, sessionDate)
workerId es clave foránea de Worker(id)
formativeSessionId es clave foránea de
FormativeSession(id)

Para el Log sería:
Log (id, **date, user, procedure, in,** out, **RSP**)

[6] [5] J. Casas Romas y J. Cuartero Olivera, Módulo 2: Diseño conceptual de bases de datos

Para el DW:

CompanyDataDW (id, **yearData**, perc_vul_tot_mit, num_acc_mit_no_acabadas, perc_workers_no_breach, num_breach_auditoria, perc_vul_no_iden_no_tot_mit, num_vul_det_int, perc_acc_mit_proc_rev)

ManagementProcessDW (id, **yearData**, idManagementProcess, num_vulnerabilities)

WorkerDW (id, **yearData**, workerId, num_breach_worker, num_breach_department)

DepartmentDW (id, **yearData**, departmentId, num_breach, perc_users)

ManagerDW (id, **yearData**, workerId, num_ma_ep)

SecurityPolicyDW (id, **yearData**, securityPolicyDW, num_breach)

ManagementProcessDW (id, **yearData**, title, perc_participants)

2.4 Diseño físico

Una vez se han realizado el diseño conceptual y lógico, se debe adaptar al diseño escogido, que dependerá del SGBD escogido, en este caso será Oracle 18c express. Para poder trabajar con el SGBD, se utilizará sqlDeveloper 21.4.3.

Se han creado las tablas en Oracle utilizando los siguientes tipos de datos:

Tipo de datos	Descripción
VARCHAR2	String de tamaño variable al que se le especifica un tamaño máximo de bytes o caracteres que contendrá.
NUMBER	Número al que se le puede indicar precisión y escala, que puede contener hasta 22 bytes. Para almacenar porcentajes se ha optimizado el espacio marcando (5,2), permitiendo tres números de precisión y dos de escala.
DATE	Tipo de datos utilizado para guardar las fechas
CHAR	Se ha utilizado para guardar aquellos strings de tamaño pequeño, como los booleanos, que no existen en Oracle y comprueban si el valor es ‘Y’ o ‘N’
TIMESTAMP	Marca de tiempo, utilizado para los registros de cambio de vulnerability y mitigation_action, para no colapsar los cambios si entra un gran volumen de datos.

Además, se han indicado las claves primarias, claves alternativas, claves foráneas y las restricciones para evitar atributos nulos o atributos que no se ajustan a los enumerados o booleanos, que no existen en Oracle y se han considerado como que tienen valor ‘Y’ o ‘N’.

El diseño físico de las tablas de la base de datos es el siguiente:

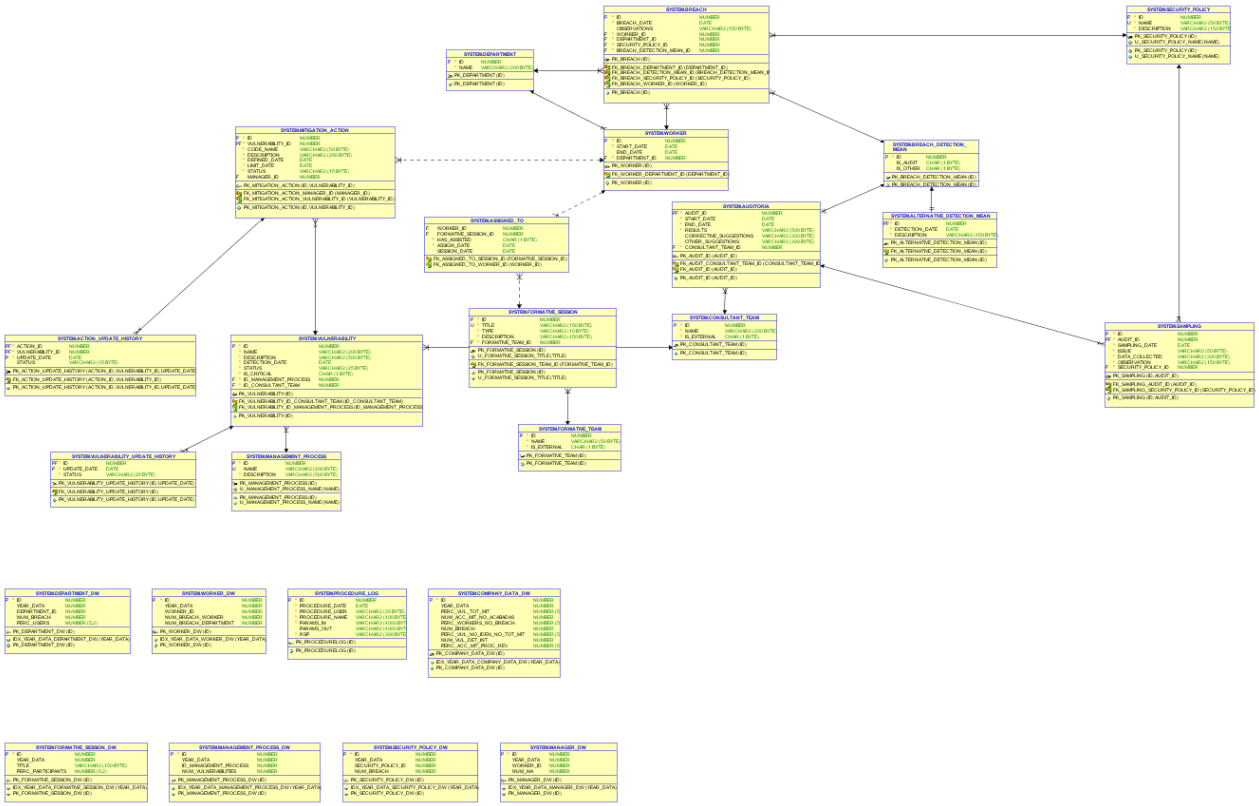


Ilustración 6. Data modeler

2.5 Implementación y optimización

2.5.1 Tablespaces

Una base de datos se divide en una o más unidades de almacenaje lógicas llamadas tablespaces. Por defecto en Oracle se crea junto con la base de datos el espacio SYSTEM.

Se han creado tres espacios de tabla:

- TFG. Espacio para las tablas principal del presente trabajo.
- TFG_TMP. Espacio para tablas temporal.
- TFG_DW. Espacio para las tablas del repositorio estadístico. Debido a su naturaleza que solo se amplía a lo largo del tiempo, se ha creado un espacio exclusivo para ésta.

2.5.2 Usuarios y roles de la base de datos

Debido a que la base de datos únicamente es una parte de la aplicación que se desarrollará, la seguridad y privilegios se han establecido de manera muy básica, diseñando cuatro roles diferentes y creando un usuario para cada rol a modo de demostración.

A pesar de ser una mala práctica, todo el trabajo se ha realizado mediante el usuario SYSTEM, que es el usuario por defecto de la base de datos y nos permite crear el resto de los roles y usuarios. Se ha decidido de esta manera ya que todo se realiza en un entorno simulado, que no supondría el daño, corrupción o pérdida de información real. En una situación real, se habría partido desde un usuario con rol TFG_Admin.

Los roles y usuarios descritos a continuación tienen como único privilegio el poder conectarse a la base de datos. Esto es debido, como se ha comentado al principio, de que el presente trabajo únicamente supondría una parte de la aplicación, sin conocer el tipo de seguridad que se vaya a utilizar.

Se han creado los siguientes roles:

- TFG_Admin. Rol dedicado para gestionar la base de datos completa.
- TFG_User. Rol dedicado a los usuarios que pueden interactuar con la base de datos.
- DW_Admin. Rol dedicado para los usuarios que gestionarán únicamente el almacén de datos.
- DW_User. Rol dedicado para los usuarios que podrán consultar el repositorio estadístico.

En base a éstos, se han creado los siguientes usuarios:

- TFG_Admin_1. Usuario con los permisos del rol TFG_Admin, usuario administrador de la base de datos.
- TFG_User_1. Usuario que únicamente puede ver los datos de la base de datos.
- DW_Admin_1. Usuario administrador de la data warehouse.
- DW_User_1. Usuario que únicamente puede ver los datos de la data warehouse.

2.5.3 Scripts de la base de datos

Para facilitar la creación, uso y modificación de todas las entidades necesarias para la BD, se han separado todas en sus correspondientes scripts, ya sean para crear, eliminar o realizar pruebas. En el producto entregado al principio nos encontramos con los siguientes:

- CREATE_DB. Es un script que únicamente se encarga de ejecutar el resto de scripts que existen en la carpeta CREATE, para poder crear las diferentes tablas y entidades necesarias para el funcionamiento de la BD.
- DROP_DB. Este script realiza el proceso contrario al anterior, llama a los scripts de la carpeta DROP, que elimina todas las tablas, vistas, roles, usuarios, índices, procedimientos… que se hayan podido crear.
- CONSULTAS_DW. Este script contiene las consultas mínimas que se nos piden en el RF16.
- Carpeta CREATE. Se desglosará a continuación.
- Carpeta DROP. Se desglosará a continuación.
- Carpeta PRUEBAS. Se desglosará a continuación.

<u>CREATE</u>

En esta carpeta nos encontramos con una carpeta PROCEDURES, junto con los scripts:

- CREATE_TABLESPACES. Script llamado por CREATE_DB. Genera los diferentes TABLESPACES mencionados anteriormente.
- CREATE_ROLES. Script llamado por CREATE_DB. Genera los diferentes roles que se han definido para la base de datos.
- CREATE_USERS.sql. Script llamado por CREATE_DB. Genera los diferentes usuarios que se han definido para la base de datos.
- CREATE_TABLES.sql. Script llamado por CREATE_DB. Genera todas las tablas que se han definido para la base de datos así como para el repositorio estadístico. La creación de tablas se realiza en su tablespace correspondiente TFG para las de la base de datos y TFG_DW para las del repositorio estadístico.

- CREATE_INDEX.sql. Script llamado por CREATE_DB. Genera los índices que se han definido para mejorar las consultas a la DW.
- CREATE_PROCEDURES.sql. Script llamado por CREATE_DB. Este script ejecuta los scripts que se encuentran en la carpeta PROCEDURES. Se ha realizado de esta manera para poder tener cada procedimiento localizable de manera rápida. Si se debe modificar o consultar un procedimiento resulta mucho más sencillo localizarlo que si se unificaran en un único script.
- CREATE_VIEWS.sql. Script llamado por CREATE_DB. El script genera dos vistas para el log, una en que podemos ver únicamente los errores y otra para ver los procedimientos que no han tenido errores.
- CREATE_TRIGGERS.sql. Script llamado por CREATE_DB. Genera los disparadores que añadirán a vulnerability_update_history y action_update_history los registros de cambio de estado de vulnerabilidad y acción de mitigación, respectivamente.

CREATE/PROCEDURES

En esta carpeta se encuentran los scripts para la creación de los procedimientos, tanto los ABM como los de la DW. En el caso de los ABM, incluyen el procedimiento de alta, baja y modificación de la tabla al que su nombre hace referencia. Para los de la DW, cuentan únicamente con un procedimiento que actualiza la tabla a la que hacen referencia.

Encontramos los siguientes:
- ABM_VULNERABILITY
- ABM_CONSULTANT_TEAM
- ABM_FORMATIVE_TEAM
- ABM_MITIGATION_ACTION
- ABM_FORMATIVE_SESSION
- ABM_AUDITORIA
- ABM_ALTERNATIVE_DETECTION_MEAN
- ABM_BREACH
- ABM_SAMPLING
- ABM_ASSIGNED_TO
- DW_COMPANY_DATA
- DW_MANAGEMENT_PROCESS
- DW_DEPARTMENT
- DW_WORKER
- DW_MANAGER
- DW_SECURITY_POLICY
- DW_FORMATIVE_SESSION
- DW_UPDATE_ALL. Este es el único diferente, ya que contiene dos procedimientos, uno que llama a todos los procedimientos indicando un año

para actualizar los datos de ese año y otro para actualizar con el año actual. El primero se ha diseñado para poder establecer un caso para las pruebas, para simular varios años de datos. El segundo es el que se utilizaría para programar actualizaciones para la DW o actualizarla en cualquier momento con la información del año en curso.

DROP

En esta carpeta encontramos los scripts para eliminar entidades. Todos son ejecutados por DROP_DB en el caso de querer eliminar todas las entidades creadas para la base de datos.

- DROP_INDEX. Elimina los índices.
- DROP_PROCEDURES. Elimina todos los procedimientos.
- DROP_ROLES_USERS. Elimina tanto los roles como los usuarios creados.
- DROP_TABLES. Elimina las tablas.
- DROP_TRIGGERS Elimina los disparadores.
- DROP_VIEWS. Elimina las vistas.

PRUEBAS

En esta carpeta podemos encontrar:

- Pruebas unitarias. Contiene los scripts utilizados para cargar datos y las pruebas unitarias realizadas sobre los diferentes procedimientos.
- Caso. Contiene los datos utilizados para la simulación de caso real. Una carpeta con los datos en excel, otra con los scripts para introducirlos (alternativa a importar los archivos de excel) y por último otra con los resultados del repositorio en excel.
- CARGA_DATOS_CASO. Script que llama a todos los scripts necesarios para montar el caso de prueba.

2.5.4 Índices

Los índices son unos elementos del diseño físico de la base de datos que tienen como finalidad mejorar el rendimiento de las aplicaciones cuando acceden a las tablas[7]. En concreto se han creado los siguientes índices:

- IDX_year_data_company_data_dw
- IDX_year_data_management_process_DW
- IDX_year_data_department_DW
- IDX_year_data_worker_DW

[7] [9] Módulo 4. Diseño físico de bases de datos

- IDX_year_data_manager_DW
- IDX_year_data_security_policy_DW
- IDX_year_data_formative_session_DW

Todos estos añaden como índice el atributo year_data de todas las tablas del repositorio estadístico. Esto mejorará las consultas que se deben realizar y vienen indicados en el RF16.

Además, por defecto, el SGBD asigna como índice de las tablas la clave primaria.

2.5.5 Procedimientos

Se ha detallado en el requisito RNF05 que se deben crear procedimientos ABM (Alta+Baja+Modificación) para todas las entidades que se consideren relevantes.

Funcionamiento de los procedimientos ABM:

Los nombres de los parámetros de entrada corresponden a los de las columnas de las tablas a las que afectan.

Debido a que siguen un proceso muy similar, en lugar de explicar el funcionamiento del procedimiento uno por uno, se agruparán en la siguiente explicación:

create:

- Se comprueba que los parámetros de entrada (aquellos que no puedan ser nulos) no sean null.
    - Si ninguno es null ni se lanza otra excepción: Se da de alta y se añade una entrada al log.
    - Si alguno es null o se lanza otra excepción, se modifica RSP con los datos del error, se deshace el trabajo realizado y se añade una entrada al log.

delete:

- Se comprueba que los parámetros de entrada no sean null. Si alguno es null se lanza excepción
- Se comprueba que exista el objeto a borrar en la base de datos. Si no existe, se lanza excepción
    - Si ninguno es null ni se lanza otra excepción: Se da de baja y se añade una entrada al log.

- Si alguno es null o se lanza otra excepción, se modifica RSP con los datos del error, se deshace el trabajo realizado y se añade una entrada al log.

update:

- Se comprueba que los parámetros de entrada no sean null. Si alguno es null se lanza excepción.
- Se comprueba que exista el objeto a modificar en la base de datos. Si no existe, se lanza excepción
    - Si ninguno es null ni se lanza otra excepción: Se modifica y se añade una entrada al log.
    - Si alguno es null o se lanza otra excepción, se modifica RSP con los datos del error, se deshace el trabajo realizado y se añade una entrada al log.

Procedimiento	**create_vulnerability**
Descripción	Procedimiento para dar de alta una vulnerabilidad
Pre-condiciones	- El equipo consultor debe existir - El proceso de gestión interna debe existir - is_critical será 'Y' o 'N' - status tendrá como valor: 'identificada, 'no mitigada', 'parcialmente mitigada', 'totalmente mitigada' o null
Información adicional	- Status tendrá valor por defecto en caso de ser nulo "identificada"
Parámetros de entrada	name, description, detection_date, status, is_critical, id_management_process, id_consultant_team
Parámetros de salida	RSP

Procedimiento	**delete_vulnerability**
Descripción	Procedimiento para dar de baja una vulnerabilidad
Pre-condiciones	- La vulnerabilidad debe existir
Parámetros de entrada	id
Parámetros de salida	RSP

Procedimiento	**update_vulnerability**
Descripción	Procedimiento para modificar una vulnerabilidad
Pre-condiciones	- La vulnerabilidad debe existir - is_critical será 'Y' o 'N'

	- status tendrá como valor: 'identificada, 'no mitigada', 'parcialmente mitigada' o 'totalmente mitigada'
Parámetros de entrada	id, name, description, detection_date, status, is_critical, id_management_process, id_consultant_team
Parámetros de salida	RSP

Procedimiento	**create_consultant_team**
Descripción	Procedimiento para dar de alta un equipo consultor
Pre-condiciones	- is_external tendrá como valor: 'Y' o 'N'
Parámetros de entrada	name, is_external
Parámetros de salida	RSP

Procedimiento	**delete_consultant_team**
Descripción	Procedimiento para dar de baja un equipo consultor
Pre-condiciones	- El equipo consultor debe existir
Parámetros de entrada	id
Parámetros de salida	RSP

Procedimiento	**update_consultant_team**
Descripción	Procedimiento para modificar un equipo consultor
Pre-condiciones	- El equipo consultor debe existir - is_external tendrá como valor: 'Y' o 'N'
Parámetros de entrada	id, name, is_external
Parámetros de salida	RSP

Procedimiento	**create_formative_team**
Descripción	Procedimiento para dar de alta un equipo formador
Pre-condiciones	- is_external tendrá como valor: 'Y' o 'N'
Parámetros de entrada	name, is_external
Parámetros de salida	RSP

Procedimiento	**delete_formative_team**
Descripción	Procedimiento para dar de baja un equipo formador

Pre-condiciones	- El equipo formador debe existir
Parámetros de entrada	id
Parámetros de salida	RSP

Procedimiento	**update_formative_team**
Descripción	Procedimiento para modificar un equipo formador
Pre-condiciones	- El equipo formador debe existir - is_external tendrá como valor: 'Y' o 'N'
Parámetros de entrada	id, name, is_external
Parámetros de salida	RSP

Procedimiento	**create_mitigation_action**
Descripción	Procedimiento para dar de alta una acción de mitigación
Pre-condiciones	- La vulnerabilidad a la que pertenece debe existir - La persona que lo gestionará debe existir - La fecha de creación no puede ser menor que la fecha límite - status tendrá como valor: 'definida, 'en proceso', 'acabada', 'en revision' o null
Información adicional	- Status tendrá valor por defecto en caso de ser nulo "definida" La creación de una acción de mitigación podrá modificar el status de la vulnerabilidad a la que se asocia en uno de los siguientes casos: Para todos los casos se tendrá en cuenta el número de acciones de mitigación asociadas a la vulnerabilidad y el número de éstas que estén en estado 'acabada'. 1. Si la vulnerabilidad pasa de 0 a 1 acciones de mitigación, pasará a 'no mitigada' si la nueva no tiene estado 'acabada' o a 'totalmente mitigada' en el caso contrario. 2. Si la vulnerabilidad tiene acciones de mitigación y pasa de 0 acciones de mitigación acabadas a 1, cambiará a estado 'parcialmente mitigada'. En el resto de los casos no cambia de estado.

Parámetros de entrada	vulnerability_id, code_name, description, defined_date, limit_date status, manager_id
Parámetros de salida	RSP

Procedimiento	**delete_mitigation_action**
Descripción	Procedimiento para dar de baja una acción de mitigación
Pre-condiciones	- La acción de mitigación debe existir
Parámetros de entrada	id
Información adicional	Borrar una acción de mitigación puede suponer cambios en el estado de la vulnerabilidad en los casos siguientes: Para todos los casos se tendrá en cuenta el número de acciones de mitigación asociadas a la vulnerabilidad y el número de éstas que estén en estado 'acabada'. 1. Si la vulnerabilidad pasa de 1 a 0 acciones de mitigación asociadas, entonces la vulnerabilidad pasa a estado 'identificada'. 2. Si la vulnerabilidad tiene 2 o más acciones de mitigación y sólo 1 acabada y se borra ésta, pasará de 'parcialmente mitigada' a 'no mitigada'. 3. Si la vulnerabilidad tiene 2 o más acciones de mitigación y sólo 1 no acabada y se borra ésta, pasará de 'parcialmente mitigada' a 'totalmente mitigada. En el resto de los casos no cambia de estado.
Parámetros de salida	RSP

Procedimiento	**update_mitigation_action**
Descripción	Procedimiento para modificar una acción de mitigación
Pre-condiciones	- La acción de mitigación debe existir - La vulnerabilidad a la que va a pertenecer debe existir - La persona que lo gestionará debe existir - La fecha de creación no puede ser menor que la

	fecha límite - status tendrá como valor: ‘definida, ‘en proceso’, ‘acabada’, ‘en revision’
Información adicional	Modificar una acción de mitigación puede suponer cambios en el estado de la vulnerabilidad en los casos siguientes: Para todos los casos se tendrá en cuenta el número de acciones de mitigación asociadas a la vulnerabilidad y el número de éstas que estén en estado ‘acabada’. 1. Si la vulnerabilidad tiene una única acción de mitigación y ésta pasa a estado ‘acabada’, la vulnerabilidad pasa a ‘totalmente mitigada’. 2. Si la vulnerabilidad tiene una única acción de mitigación y pasa de estado ‘acabada’ a otro, entonces la vulnerabilidad cambia de ‘totalmente mitigada’ a ‘no mitigada’. 3. Si la vulnerabilidad tiene 2 o más acciones de mitigación y ninguna acabada, si se modifica una a ‘acabada’ entonces pasa de ‘no mitigada’ a ‘parcialmente mitigada’ 4. Si la vulnerabilidad tiene 2 o más acciones de mitigación todas en estado ‘acabada’ y se modifica el estado de una, entonces pasa de ‘totalmente mitigada’ a ‘parcialmente mitigada’ En el resto de los casos no cambia de estado.
Parámetros de entrada	id, vulnerability_id, code_name, description, defined_date, limit_date status, manager_id
Parámetros de salida	RSP

Procedimiento	**create_formative_session**
Descripción	Procedimiento para dar de alta una sesión de formación.
Pre-condiciones	- El equipo formador existe. - type tendrá como valor: ‘presencial’ o ‘telematica’ - No existe otra sesión con el mismo título
Parámetros de entrada	title, type, description, formative_team_id
Parámetros de	RSP

salida	

Procedimiento	**delete_formative_session**
Descripción	Procedimiento para dar de baja una sesión de formación
Pre-condiciones	- La sesión debe existir - La sesión no está asignada a ningún trabajador
Parámetros de entrada	id
Parámetros de salida	RSP

Procedimiento	**update_formative_session**
Descripción	Procedimiento para modificar una sesión de formación
Pre-condiciones	- La sesión debe existir - type tendrá como valor: 'presencial' o 'telematica' - No existe otra sesión con el mismo título que se le intenta modificar
Parámetros de entrada	id, title, type, description, formative_team_id
Parámetros de salida	RSP

Procedimiento	**create_auditoria**
Descripción	Procedimiento para dar de alta una auditoria
Pre-condiciones	- La fecha de inicio debe ser anterior o igual a la fecha de finalización. - El equipo consultor debe existir
Información adicional	Antes de crear la auditoria, se registra un BreachDetectionMean con el atributo isAudit = 'Y' y se usa el id de éste para dar de alta de la auditoria.
Parámetros de entrada	start_date, end_date, results, corrective_suggestions, other_suggestions, consultant_team_id
Parámetros de salida	RSP

Procedimiento	**delete_auditoria**
Descripción	Procedimiento para dar de baja una auditoria
Pre-condiciones	- La auditoría debe existir
Información adicional	Al eliminar la auditoria, también se elimina de BreachDetectionMean el registro con el id de la auditoría.

Parámetros de entrada	id
Parámetros de salida	RSP

Procedimiento	**update_auditoria**
Descripción	Procedimiento para modificar una auditoria
Pre-condiciones	- La auditoría existe - Si se modifica el equipo consultor, éste debe existir.
Parámetros de entrada	id, start_date, end_date, results, corrective_suggestions, other_suggestions, consultant_team_id
Parámetros de salida	RSP

Procedimiento	**create_alternative_detection_mean**
Descripción	Procedimiento para dar de alta un método de detección alternativo de detección de incumplimientos
Pre-condiciones	- Ninguna
Información adicional	Antes de crear el método alternativo, se registra un BreachDetectionMean con el atributo isOther = 'Y' y se usa el id de éste para dar de alta del método
Parámetros de entrada	detection_date, description
Parámetros de salida	RSP

Procedimiento	**delete_alternative_detection_mean**
Descripción	Procedimiento para dar de baja un método alternativo de detección de incumplimientos
Pre-condiciones	- El método debe existir
Información adicional	Al eliminar el método, también se elimina de BreachDetectionMean el registro con el id de la del método.
Parámetros de entrada	id
Parámetros de salida	RSP

Procedimiento	**update_alternative_detection_mean**
Descripción	Procedimiento para modificar una auditoria
Pre-condiciones	- El método debe existir

Parámetros de entrada	id, detection_date, description
Parámetros de salida	RSP

Procedimiento	**create_breach**
Descripción	Procedimiento para dar de alta un incumplimiento
Pre-condiciones	- La fecha del incumplimiento no puede ser anterior a la de inicio o finalización del empleado. - La fecha del incumplimiento no puede ser anterior a la del método que la ha detectado - El trabajador debe existir - El departamento debe existir - La política de seguridad debe existir - El método de detección del incumplimiento debe existir
Parámetros de entrada	breach_date, observations, worker_id, department_id, security_policy_id, breach_detection_mean_id
Parámetros de salida	RSP

Procedimiento	**delete_breach**
Descripción	Procedimiento para dar de baja un incumplimiento
Pre-condiciones	- El incumplimiento debe existir
Parámetros de entrada	id
Parámetros de salida	RSP

Procedimiento	**update_breach**
Descripción	Procedimiento para modificar una auditoria
Pre-condiciones	- La fecha del incumplimiento no puede ser anterior a la de inicio o finalización del empleado. - La fecha del incumplimiento no puede ser anterior a la del método que la ha detectado - El trabajador debe existir - El departamento debe existir - La política de seguridad debe existir - El método de detección del incumplimiento debe existir - El incumplimiento debe existir
Parámetros de entrada	id, breach_date, observations, worker_id, department_id, security_policy_id,

	breach_detection_mean_id
Parámetros de salida	RSP

Procedimiento	**create_sampling**
Descripción	Procedimiento para dar de alta un muestreo de una auditoría
Pre-condiciones	- La fecha del muestreo debe estar entre el inicio y fin de la auditoría. - La auditoría debe existir - La política de seguridad debe existir
Parámetros de entrada	audit_id, sampling_date, issue, data_collected, observation, security_policy_id
Parámetros de salida	RSP

Procedimiento	**delete_sampling**
Descripción	Procedimiento para dar de baja un muestreo de una auditoría
Pre-condiciones	- El muestreo y la auditoría deben existir
Parámetros de entrada	id, audit_id
Parámetros de salida	RSP

Procedimiento	**update_sampling**
Descripción	Procedimiento para modificar una auditoria
Pre-condiciones	- La fecha del muestreo debe estar entre el inicio y fin de la auditoría. - La auditoría debe existir - La política de seguridad debe existir - El muestreo debe existir
Parámetros de entrada	id, audit_id, sampling_date, issue, data_collected, observation, security_policy_id
Parámetros de salida	RSP

Procedimiento	**create_assigned_to**
Descripción	Procedimiento para dar de alta una asignación a una sesión de formación a un trabajador
Pre-condiciones	- El trabajador debe existir - La sesión de formación debe existir - has_assisted debe tener com valor ‘Y’ o ‘N’

	- assigned_date debe estar entre el inicio y la finalización del periodo de trabajo del trabajador
Parámetros de entrada	worker_id, formative_session_id, has_assisted, assign_date, session_date
Parámetros de salida	RSP

Procedimiento	**delete_assigned_to**
Descripción	Procedimiento para dar de baja una asignación a una sesión de formación a un trabajador
Pre-condiciones	- La asignación, identificada por la combinación del trabajador y la sesión, debe existir
Parámetros de entrada	worker_id, formative_session_id
Parámetros de salida	RSP

Procedimiento	**update_assigned_to**
Descripción	Procedimiento para modificar una una asignación a una sesión de formación a un trabajador
Pre-condiciones	- La asignación, identificada por la combinación del trabajador y la sesión, debe existir - has_assisted debe tener com valor 'Y' o 'N' - assigned_date debe estar entre el inicio y la finalización del periodo de trabajo del trabajador
Parámetros de entrada	worker_id, formative_session_id, has_assisted, assign_date, session_date
Parámetros de salida	RSP

2.5.6 Disparadores

Se han creado dos disparadores para poder llevar un historial de los cambios de estado de las vulnerabilidades y de las acciones de mitigación estos son:

- TRG_INSERT_AM_HISTORY. Salta cuando se produce un cambio en el estado de una acción de mitigación e introduce en action_update_history un nuevo registro para indicar el momento de cambio mediante una marca de tiempo.
- TRG_INSERT_VUL_HISTORY. La función es la misma que en el anterior, pero salta cuando se produce un cambio en el estado de una vulnerabilidad y añade el registro a la tabla vulnerability_update_history.

3. Pruebas

Las pruebas que se han realizado se pueden dividir en tres tipos:

- Pruebas unitarias de los procesos ABM.
- Pruebas disparadores
- Pruebas de los procesos del repositorio estadístico.

Pruebas unitarias

Se han diseñado pruebas unitarias para cada uno de los procedimientos ABM. Se pueden observar con mayor detalle en el anexo pruebas.

Pruebas triggers

Las pruebas de los triggers se realizan a la vez que las pruebas del cambio de estado de vulnerabilidad a través de los procedimientos ABM de las acciones de mitigación.
El principal motivo de juntar las tres pruebas es que el cambio de estado de las vulnerabilidades depende exclusivamente de sus acciones de mitigación, por lo tanto, se deben realizar al menos las pruebas del trigger de inserción en el historial de los cambios de estado de vulnerabilidades. Ya que también se realizan cambios de estado de acciones de mitigación, a pesar de que no se realizan todas posibles inserciones, si bastan para comprobar que funciona el disparador.

Tabla – vulnerability_update_history y action_update_history

Procedimiento	Pruebas realizadas
create_mitigation_action **update_mitigation_action** **delete_mitigation_action** **TRG_insert_vul_history** **TRG_insert_action_history**	- ✓ Crear acción acabada – vulnerabilidad totalmente mitigada - ✓ Crear acción no acabada – vulnerabilidad parcialmente mitigada - ✓ Eliminar acción acabada – vulnerabilidad no mitigada - ✓ Eliminar acción no acabada – vulnerabilidad identificada - ✓ Crear acción no acabada – vulnerabilidad no mitigada - ✓ Modificar acción a acabada – vulnerabilidad totalmente mitigada - ✓ Modificar acción a no acabada – vulnerabilidad no mitigada - ✓ Crear acción no acabada – no cambia - ✓ Modificar acción a acabada – vulnerabilidad parcialmente mitigada - ✓ Modificar acción no acabada – vulnerabilidad totalmente mitigada

Resultado log

	ID	...	PRO...	PROCEDURE_NAME	PARAMS...	PARAMS_OUT	RSP
1	1	...	SYSTEM	CREATE_MITIGATION_ACTION	Mitiga...	(null)	OK
2	2	...	SYSTEM	CREATE_MITIGATION_ACTION	Mitiga...	(null)	OK
3	3	...	SYSTEM	DELETE_MITIGATION_ACTION	Mitiga...	(null)	OK
4	4	...	SYSTEM	DELETE_MITIGATION_ACTION	Mitiga...	(null)	OK
5	5	...	SYSTEM	CREATE_MITIGATION_ACTION	Mitiga...	(null)	OK
6	6	...	SYSTEM	UPDATE_MITIGATION_ACTION	Mitiga...	(null)	OK
7	7	...	SYSTEM	UPDATE_MITIGATION_ACTION	Mitiga...	(null)	OK
8	8	...	SYSTEM	CREATE_MITIGATION_ACTION	Mitiga...	(null)	OK
9	9	...	SYSTEM	UPDATE_MITIGATION_ACTION	Mitiga...	(null)	OK
10	10	...	SYSTEM	UPDATE_MITIGATION_ACTION	Mitiga...	(null)	OK

Resultado tabla vulnerability_update_history

	VULNERABILITY_ID	UPDATE_TIMESTAMP	STATUS
1	1	09/06/22 21:15:20,281000000	totalmente mitigada
2	1	09/06/22 21:15:20,285000000	parcialmente mitigada
3	1	09/06/22 21:15:20,288000000	no mitigada
4	1	09/06/22 21:15:20,290000000	identificada
5	1	09/06/22 21:15:20,291000000	no mitigada
6	1	09/06/22 21:15:20,295000000	totalmente mitigada
7	1	09/06/22 21:15:20,315000000	no mitigada
8	1	09/06/22 21:15:20,337000000	parcialmente mitigada
9	1	09/06/22 21:15:20,338000000	totalmente mitigada

Resultado tabla action_update_history

	ACTION_ID	VULNERABILITY_ID	UPDATE_TIMESTAMP	STATUS
1	3	1	09/06/22 21:15:20,309000000	acabada
2	3	1	09/06/22 21:15:20,318000000	en proceso
3	3	1	09/06/22 21:15:20,337000000	acabada
4	4	1	09/06/22 21:15:20,338000000	acabada

Pruebas de los procesos del repositorio estadístico

Las tablas pertenecientes al repositorio estadístico, como mínimo, deben poder dar respuesta a las siguientes consultas:

- Departamento que, en un año concreto, tiene un número mayor de incumplimientos de seguridad registrados en la BD.

- Proceso de gestión interno que, teniendo en cuenta toda la información de la que se dispone en la BD, ha tenido un mayor número de vulnerabilidades detectadas.
- Top5 de usuarios por número de incumplimientos asociados directamente a ellos, o a su departamento, durante el año en curso.
- Porcentaje de vulnerabilidades que, en el momento de hacer la consulta, están totalmente mitigadas.
- Número total de acciones de mitigación que, en el momento de hacer la consulta, no están totalmente acabadas.
- Política de seguridad que, en el momento de hacer la consulta, ha tenido más incumplimientos (teniendo en cuenta todos los departamentos de la empresa).
- Dado un determinado departamento de la empresa, y teniendo en cuenta el momento de ejecutar la consulta, porcentaje de usuarios del departamento que no han acabado todas las formaciones de seguridad asignadas.
- Porcentaje de usuarios de la empresa que, en el año en curso, no tienen ningún incumplimiento asignado.
- Teniendo en cuenta todas las auditorías externas realizadas, año en el cual se han detectado más incumplimientos (teniendo en cuenta únicamente los detectados durante la auditoría).
- Porcentaje de vulnerabilidades críticas que, en el momento de hacer la consulta, tienen alguna acción de mitigación abierta (que no esté en estado “acabada”).
- Teniendo en cuenta el último año (el anterior al año en curso, que es cuando se ejecuta la consulta), título de la sesión formativa telemática que ha tenido un porcentaje menor de participantes en total.
- Número de vulnerabilidades críticas detectadas internamente teniendo en cuenta todos los datos de que se dispone. Se consideran detectadas internamente si se detectaron en posterioridad al análisis realizado inicialmente por la consultora externa.
- En el momento de hacer la consulta, porcentaje de acciones de mitigación en el sistema que están en los estados “en proceso” o “en revisión”.
- Teniendo en cuenta todas las acciones de mitigación en estado “en proceso”, persona responsable con más acciones asignadas.

Además, se nos impone otra condición: que tengan tiempo constante igual a 1.

Para poder realizar las pruebas sobre el repositorio, se ha creado un juego de datos que, si bien podría ser mayor, puede ser útil para poder comprobar que los resultados que se obtienen al aplicar los procedimientos son correctos.

Para comenzar, se han cargado los diferentes datos. Para asegurar que los datos se introducían todos de forma correcta, a pesar de que podría haberse realizado importando los datos directamente desde las hojas de Excel, se han importado como INSERT y modificado para llamar a los diferentes procedimientos para insertar los datos.

```
/*****************************************************
*   Script para importar los datos
*****************************************************/
@..\DROP_DB.sql
@..\CREATE_DB.sql
@.\Caso\DATOS_APP\CARGA_DATOS_CASO_EMPRESA.sql
@.\Caso\DATOS_APP\CONSULTANT_TEAM.sql
@.\Caso\DATOS_APP\VULNERABILITY.sql
@.\Caso\DATOS_APP\MITIGATION_ACTION.sql
@.\Caso\DATOS_APP\AUDITORIA.sql
@.\Caso\DATOS_APP\ALTERNATIVE_DETECTION_MEAN.sql
@.\Caso\DATOS_APP\BREACH.sql
@.\Caso\DATOS_APP\FORMATIVE_TEAM.sql
@.\Caso\DATOS_APP\FORMATIVE_SESSION.sql
@.\Caso\DATOS_APP\ASSIGNED_TO.sql
```

Ilustración 7. Script CARGA_DATOS_CASO

Desde el script CARGA_DATOS_CASO, ejecutamos todos los scripts preparados para importar los datos. Para asegurar que los datos son exclusivamente los que necesitamos, antes eliminamos y creamos la base de datos entera, para evitar que cualquier interferencia y que el caso sea replicable.

Salvo con los datos que “importamos” de la empresa, para el resto de datos podemos comprobar si ha habido algún problema mediante el log de procedimientos. En concreto mediante la vista creada para ver únicamente si han dado error.

```
17  select * from view_error_log;
```

Salida de Script | Resultado de la Consulta

SQL | Todas las Filas Recuperadas: 0 en 0,008 segundos

ID	PROCEDU...	PROCEDU...	PROCEDU...	PARAMS_IN	PARAMS_...	RSP

Ilustración 8. Comprobar log

Como se puede observar, no hay errores al realizar la carga de datos.

Una vez realizada la carga de los datos en la base de datos, se deben ejecutar los diferentes procedimientos para procesar los datos e introducirlos en las tablas del repositorio estadístico.

Esto lo podemos hacer mediante un bucle que se encuentra comentado en el script de las consultas:

```
149  -- Bucle para rellenar el repositorio estadistico con datos de 2018 a 2022
150
151  DECLARE
152      RSP VARCHAR2(500);
153  BEGIN
154      FOR var_year IN 2018..2022 LOOP
155          UPDATE_ALL_BY_YEAR_DW(var_year, RSP);
156      END LOOP;
157  END;
```

Ilustración 9. Bucle procesamiento de datos

De nuevo, podemos comprobar si ha habido cualquier problema mediante la vista del log.

```
149  -- Bucle para rellenar el repositorio estadistico con datos de 2018 a 2022
150  /*
151  DECLARE
152      RSP VARCHAR2(500);
153  BEGIN
154      FOR var_year IN 2018..2022 LOOP
155          UPDATE_ALL_BY_YEAR_DW(var_year, RSP);
156      END LOOP;
157  END;
158  */
159
160  select * from view_error_log;
```

Salida de Script × | Resultado de la Consulta ×

SQL | Todas las Filas Recuperadas: 0 en 0,002 segundos

ID	PROCEDU...	PROCEDU...	PROCEDU...	PARAMS_IN	PARAMS_...	RSP

Ilustración 10. Comprobación de log II

Una vez procesados los datos y rellenadas las tablas del repositorio, aunque no sea demasiado real, ya que debería ser únicamente con los del año en curso debido a que hay datos que pueden variar con el tiempo, como los estados de vulnerabilidades y acciones de mitigación. Con el objetivo de comprobar la ejecución de las consultas, se ha decidido aceptar esa situación “no real”.

Los datos procesados se encuentran también exportados en Excel para poder comprobar que el resultado de las consultas es correcto.

Si vamos ejecutando consulta por consulta (se mostrarán 3 ejemplos a continuación para ver que se obtienen correctamente los resultados esperados de las consultas, el resto se incluirán en el archivo anexo de pruebas):

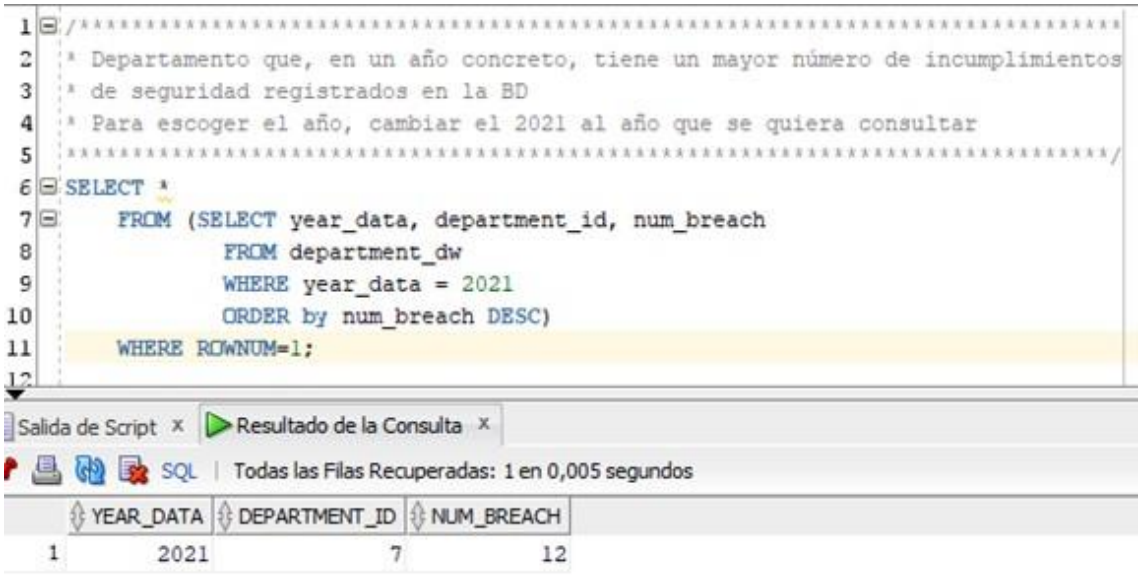


Ilustración 11. Consulta 1

Como se puede comprobar en el archivo Excel (o si se miran todos los datos de la tabla):

ID	YEAR_DATA	DEPARTMENT_ID	NUM_BRE
34	2021	1	2
35	2021	2	3
36	2021	3	2
37	2021	4	5
38	2021	5	4
39	2021	6	6
40	2021	7	12
41	2021	8	4
42	2021	9	0
43	2021	10	2
44	2021	11	6

Ilustración 12. Datos department_dw

El resultado que nos muestra la consulta es correcto. El departamento con más incumplimientos, en este caso de 2021, es el departamento 7.

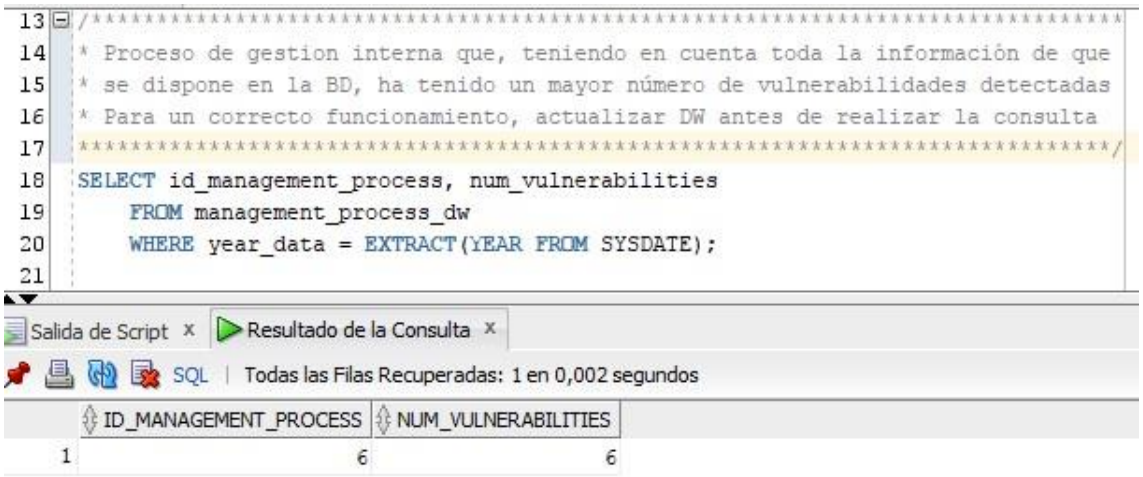


Ilustración 13. Consulta 2

Si se comprueban los datos en el archivo Excel, se puede observar si los datos con correctos:

	A	B	C	D	E	F
	ID	YEAR_DAT	ID_MANA	NUM_VULNERABILITIES		
	1	2018	8	4		
	2	2019	6	6		
	3	2020	6	6		
	4	2021	6	6		
	5	2022	6	6		

Ilustración 14. Datos management_process_dw

Nuevamente el resultado mostrado por la consulta es correcto, tal y como se puede comprobar posteriormente con los datos disponibles en el Excel.

```
/*******************************************************************************
* Top5 de usuarios por número de incumplimientos asociados directamente a ellos,
* o a su departamento, durante el año en curso
* Para un correcto funcionamiento, actualizar DW antes de realizar la consulta
*******************************************************************************/
SELECT *
    FROM (SELECT year_data, worker_id, num_breach_worker, num_breach_department
            FROM worker_dw
            WHERE year_data = EXTRACT(YEAR FROM SYSDATE)
            ORDER BY num_breach_department DESC, num_breach_worker DESC)
    WHERE ROWNUM <= 5;
```

Salida de Script | Resultado de la Consulta

SQL | Todas las Filas Recuperadas: 5 en 0,004 segundos

	YEAR_DATA	WORKER_ID	NUM_BREACH_WORKER	NUM_BREACH_DEPARTMENT
1	2022	17	2	4
2	2022	20	2	4
3	2022	18	0	4
4	2022	19	0	4
5	2022	6	2	3

Ilustración 15. Consulta 3

De nuevo, si se comprueban los datos exportados en el archivo Excel de las comprobaciones, se tiene:

ID	YEAR_DAT	WORKER_	NUM_BRE	NUM_BREACH_DEPARTMENT
130	2022	1	0	2
131	2022	2	0	2
132	2022	3	0	2
133	2022	4	2	2
134	2022	5	0	3
135	2022	6	2	3
136	2022	7	1	3
137	2022	8	2	2
138	2022	9	0	2
139	2022	10	0	2
140	2022	11	1	2
141	2022	12	1	2
142	2022	13	0	2
143	2022	14	0	2
144	2022	15	2	2
145	2022	16	0	2
146	2022	17	2	4
147	2022	18	0	4
148	2022	19	0	4
149	2022	20	2	4
150	2022	21	1	3
151	2022	22	0	3
152	2022	23	1	3
153	2022	24	1	3
154	2022	25	0	3
155	2022	26	0	3
156	2022	27	0	3
157	2022	28	1	1
158	2022	29	0	1
159	2022	30	0	0
160	2022	31	0	0
161	2022	32	0	0
162	2022	33	0	0
163	2022	34	0	0
164	2022	35	0	0
165	2022	36	0	0
166	2022	37	2	2
167	2022	38	0	2
168	2022	39	0	2

Ilustración 16. Datos worker_dw

Como se ha podido comprobar con las 3 consultas de ejemplo, el resultado tras ejecutar las mismas es correcto.

4. Conclusiones

La realización del plan de trabajo y su posterior seguimiento todo lo rigurosamente que se pueda es de vital importancia para que el proyecto no se vuelva contra la persona (o equipo). Lo he descubierto de mala manera debido a problemas de salud más o menos graves que complicaron la ejecución del proyecto durante el final de la segunda entrega y gran parte de la tercera entrega.

La ejecución del proyecto, aunque al principio parecía que sería una extensión de trabajos ya realizados en otras asignaturas, en concreto una fusión entre las asignaturas gestión de proyectos y diseño de bases de datos, al final ha resultado ser muy diferente. Debido a su naturaleza más práctica, se ha aprendido mucho sobre el SGBD escogido, así como buenas y malas prácticas. También se ha descubierto el lenguaje PL/SQL, que, aunque se basa en SQL, supone un nuevo conocimiento. Investigando para la aplicación del Data Warehouse, también se ha descubierto la existencia de estos métodos y herramientas tan útiles.

Debido a la metodología escogida, ya que se realizaban pruebas antes de cada entrega, a pesar de que únicamente se entregaba la memoria, se encontraban errores en el diseño que podrían repercutir en posteriores iteraciones, por lo que tanto el diseño inicial, como la metodología escogida, son vitales para el buen desarrollo del proyecto.

De los objetivos planteados:

- Poner en práctica los conocimientos adquiridos en asignaturas de Bases de Datos
- Utilizar el lenguaje SQL
- Ampliar conocimientos utilizando nuevas herramientas
- Detectar cuales son las necesidades básicas de un determinado sistema a analizar
- Detectar posibles funcionalidades adicionales de valor añadido
- Proponer un diseño que se ajuste a los requerimientos expuestos
- Implementar un sistema que encapsule las funciones de acceso de datos

El diseño de la base de datos tendrá los siguientes objetivos:

- Analizar la problemática planteada y definir una posible estructura de Base de Datos

- Uso de las técnicas que se aplican a grandes volúmenes de información (Data Warehouse)
- Analizar posibles líneas de evolución que podrá seguir la base de datos

Podemos decir que se han cumplido todos. Como se ha comentado, se han puesto en práctica los conocimientos adquiridos en asignaturas de Bases de Datos y se han ampliado. Se ha utilizado tanto SQL como PL/SQL. Se han utilizado herramientas y técnicas nuevas. Se han detectado e implementado las necesidades básicas del sistema y se han intentado añadir funcionalidades adicionales. Se ha propuesto un diseño ajustado a los requerimientos y se ha implementado.

Respecto a las posibles de líneas de evolución, son muchas, por ejemplo, la correcta implantación del Data Warehouse, con todas las herramientas adicionales que la hagan funcionar incluso mejor que ahora, que resulta muy básica. Otra podría ser también la inclusión de los problemas que puedan sufrir en sus productos, es decir, la aplicación se realizará para una empresa automovilística y actualmente todos los vehículos nuevos cuentan con ordenadores, que también tienen sus procesos y contar con vulnerabilidades a explotar, por ejemplo, hay funciones como encontrar el vehículo o incluso para el motor que podrían sufrir un mal uso si no se controlara la seguridad de éstos.

El seguimiento de la planificación no ha sido el esperado, como se ha comentado al inicio del capítulo como en el apartado de seguimiento, ha sido imposible seguir, incluso con posibles acciones de mitigación para los riesgos planteados y que al final han sucedido.

A nivel de pruebas, se podría haber mejorado mucho, las pruebas unitarias se han realizado de forma manual en lugar de automatizada, esto es debido a que sqlDeveloper no tiene desde hace mucho integrada la función para poder realizar tests unitarios. Existen otras herramientas que sí le permiten realizarlos, pero el desconocimiento de su uso ha resultado en preferir el método realizado. También, se podrían pulir los métodos del repositorio, para poder incluir datos mucho más reales para años anteriores, pero únicamente resultarían útiles para las pruebas, ya que, en el caso real, se podría programar una actualización del repositorio para el último día del año, lo que dejaría el repositorio con la información de ese año actualizada.

Ya que la base de datos es para una aplicación, se ha echado en falta un poco más de contexto, ya que hay ciertos aspectos de la aplicación que podrían llegar a afectar a la base de datos.

Por último, la metodología escogida, ha resultado muy útil. Si hubiera sido el proyecto completo, junto con aplicación para la empresa, la diferencia que

proporciona con respecto a otra como la metodología en cascada, hubiera sido incluso mayor, ya que ha permitido revisar todo el trabajo realizado y encontrar diferentes problemas que luego se han subsanado, como por ejemplo el incluir los cambios de estado de las vulnerabilidades en los procedimientos ABM de las acciones de mitigación, que en un principio se tenían diseñado disparadores para éstos, pero al iniciar la ronda de pruebas se vio que se producían errores de mutación en las tablas y posteriormente se subsanó el fallo.

5. Glosario

- ABM/CRUD: Son las siglas de Alta/Baja/Modificación en castellano y Create/Read/Update/Drop en inglés. En referencia a los procedimientos indican el tipo de operación que se espera de éstos.
- BD: Abreviación de Base de Datos.
- Data Warehouse: Almacén de datos. Define una base de datos que puede llegar a contener una gran cantidad de datos en su interior.
- OLAP. Son las siglas de procesamiento analítico en línea en inglés. Se basa en el modelo de datos multidimensionales (cubos).[8]
- PL-SQL: Es un lenguaje de procedimiento diseñado para abarcar sentencias SQL dentro de su sintaxis que viene integrado en los SGBD de Oracle. [7]
- SGBD: Sistema gestor de bases de datos (en inglés Database management system, DBMS)
- SQL: Siglas de Lenguaje de Consulta Estructurada.
- UML: Abreviación de Lenguaje Unificado de Modelado.

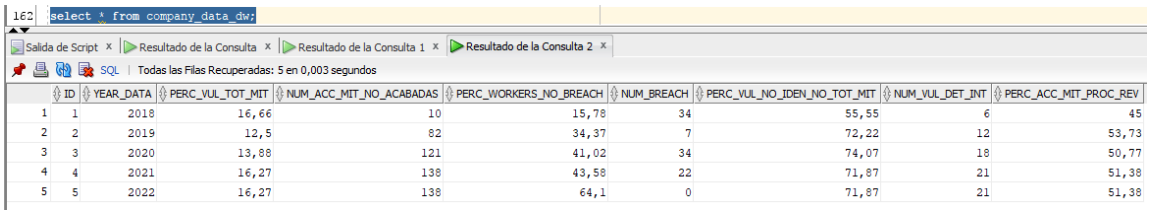
6. Bibliografía

- [1] J. Pradel Miquel y J. Raya Martos, Módulo 1: Introducción a Ingeniería de Software. UOC
- [2] J. Casas Romas, Módulo 1: Introducción al diseño de bases de datos. UOC
- [3] J. Pradel Miquel y J. Raya Martos, Módulo 1: Introducción a la ingeniería de requisitos. UOC
- [4] <<Ley Orgánica 3/2018, de 5 de diciembre, de Protección de Datos Personales y garantía de los derechos digitales>>. Visitada el 30/03/22. https://www.boe.es/eli/es/lo/2018/12/05/3
- [5] J. Casas Romas y J. Cuartero Olivera, Módulo 2: Diseño conceptual de bases de datos. UOC
- [6] <<database-vs-data-warehouse-whats-the-difference>> Visitada el 25/05/2022. https://insightsoftware.com/es/blog/database-vs-data-warehouse-whats-the-difference/
- [7] <<PL/SQL>> Visitado el 05/06/2022 https://www.oracle.com/es/database/technologies/appdev/plsql.html Visitada el 5/6/2022.
- [8]<<Cubos OLAP>>. Visitado el 05/06/2022 https://protecciondatos-lopd.com/empresas/data-warehouse/#Cubos_OLAP
- [9] B. Cabré i Segarra, J. Casas Roma, D. Costal Costa, P. Juanola Juanola, A. Rius Gavidia y R. Segret i Sala, Módulo 4: Diseño físico de bases de datos. UOC. PID_00223667.

7. Anexos

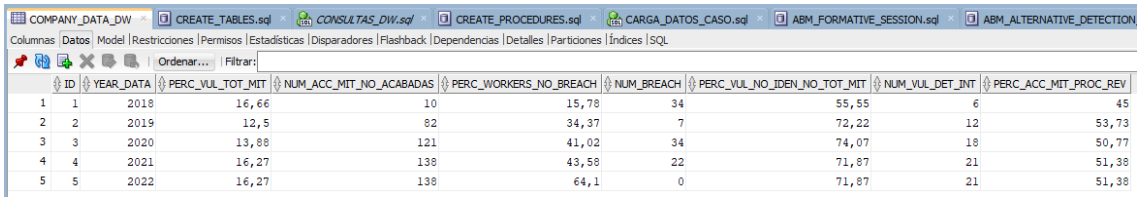
Anexo1: Exportar datos de una tabla a excel

Primer paso. Ejecutar una consulta que devuelva los datos o entrar en la pestaña datos de la tabla:



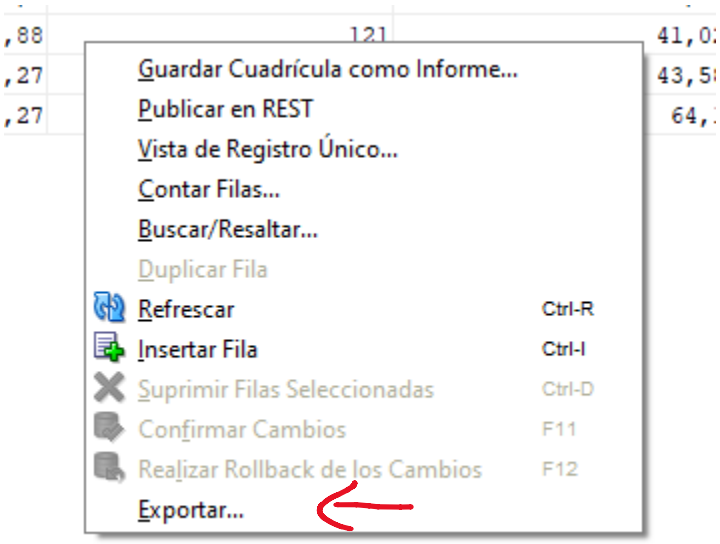
	ID	YEAR_DATA	PERC_VUL_TOT_MIT	NUM_ACC_MIT_NO_ACABADAS	PERC_WORKERS_NO_BREACH	NUM_BREACH	PERC_VUL_NO_IDEN_NO_TOT_MIT	NUM_VUL_DET_INT	PERC_ACC_MIT_PROC_REV
1	1	2018	16,66	10	15,78	34	55,55	6	45
2	2	2019	12,5	82	34,37	7	72,22	12	53,73
3	3	2020	13,88	121	41,02	34	74,07	18	50,77
4	4	2021	16,27	138	43,58	22	71,87	21	51,38
5	5	2022	16,27	138	64,1	0	71,87	21	51,38

O

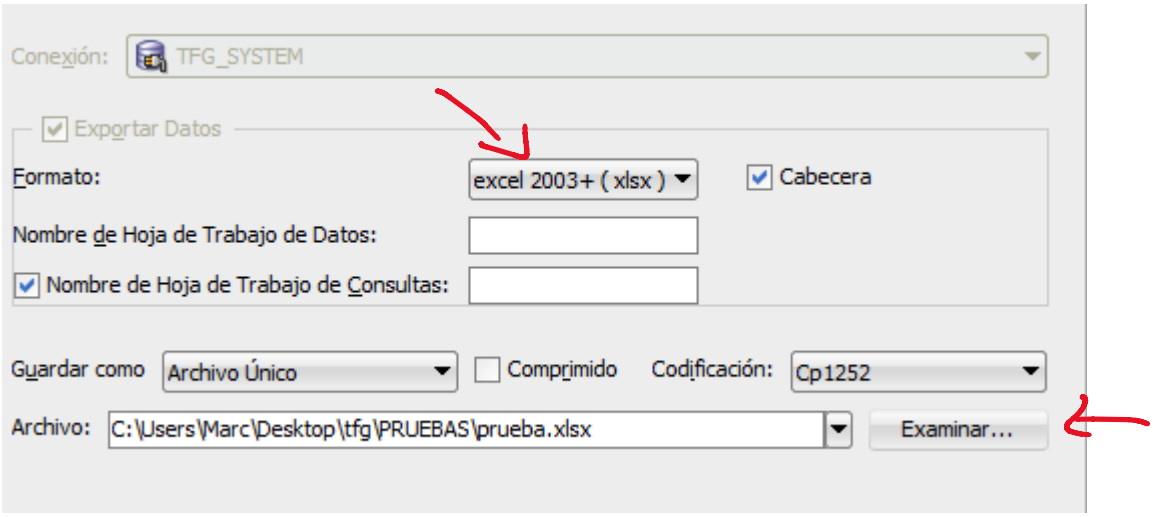


	ID	YEAR_DATA	PERC_VUL_TOT_MIT	NUM_ACC_MIT_NO_ACABADAS	PERC_WORKERS_NO_BREACH	NUM_BREACH	PERC_VUL_NO_IDEN_NO_TOT_MIT	NUM_VUL_DET_INT	PERC_ACC_MIT_PROC_REV
1	1	2018	16,66	10	15,78	34	55,55	6	45
2	2	2019	12,5	82	34,37	7	72,22	12	53,73
3	3	2020	13,88	121	41,02	34	74,07	18	50,77
4	4	2021	16,27	138	43,58	22	71,87	21	51,38
5	5	2022	16,27	138	64,1	0	71,87	21	51,38

Una vez se tienen los datos que se quieren exportar, realizar click derecho y seleccionar la opción de exportar..

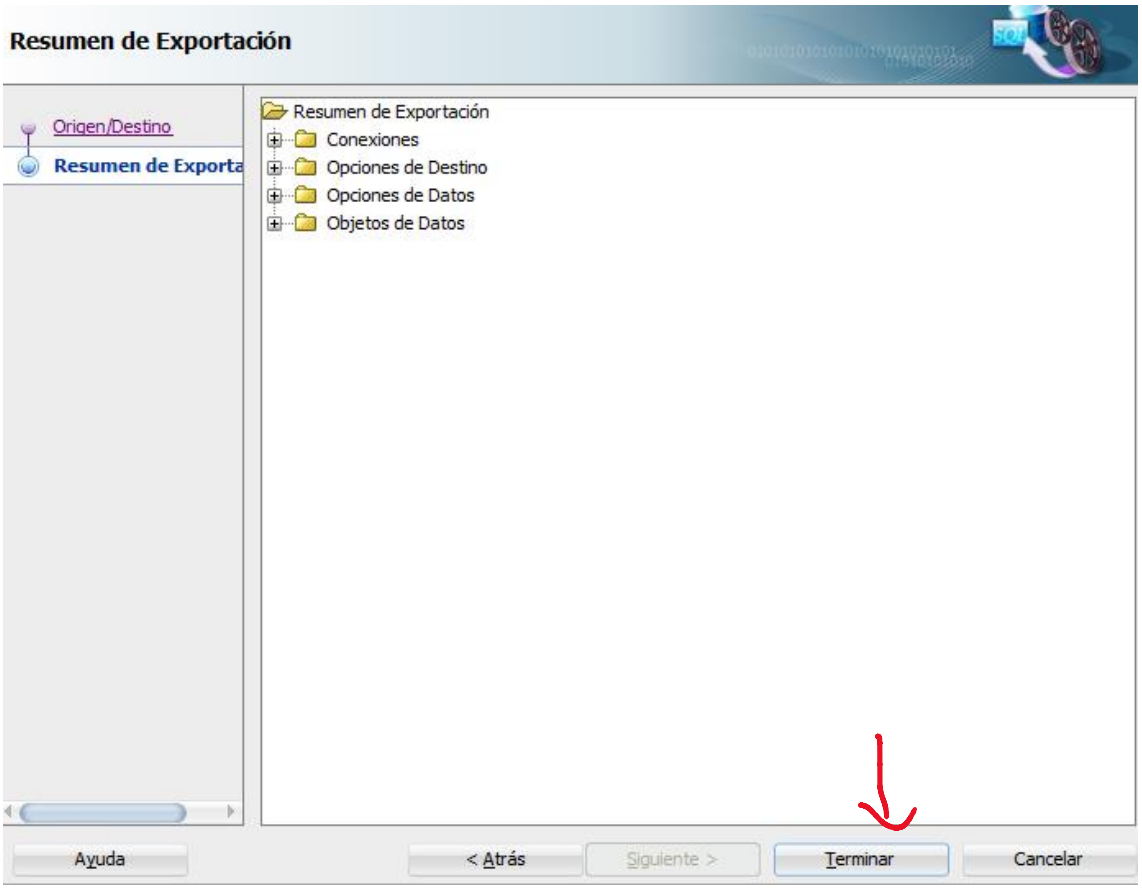


Se abrirá la siguiente ventana:



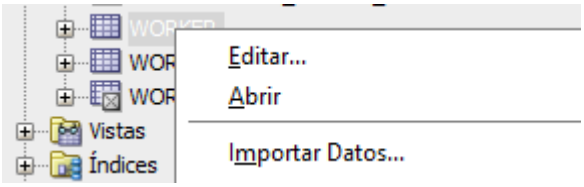
Aquí seleccionar el formato que se desea, en este caso excel 2003+ (xlsx) y seleccionar dónde se va a guardar y el nombre que se le va a dar. Continuar dándole a siguiente.

Se cambiará a un resumen del contenido que tendrá el archivo. Salir seleccionando Terminar.

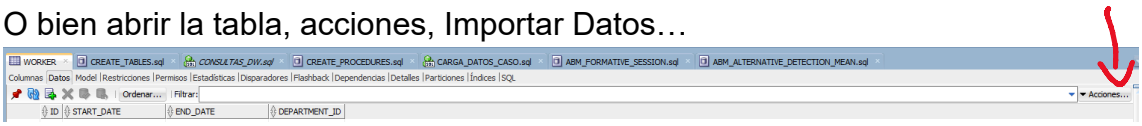


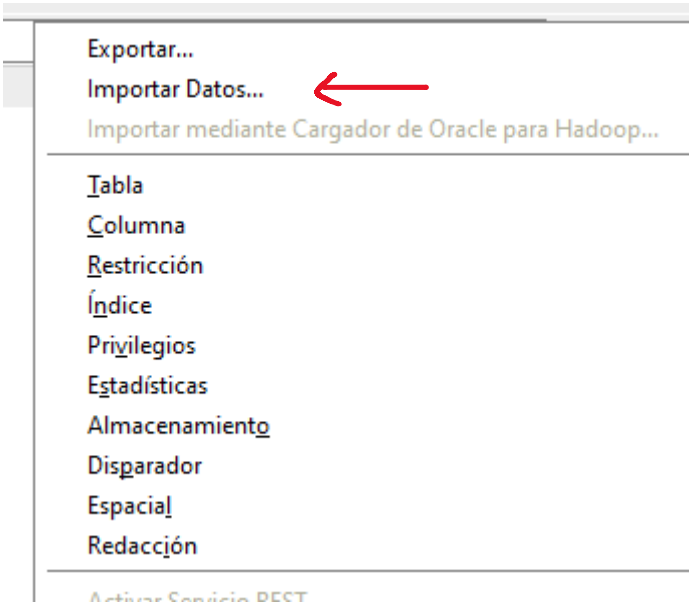
Anexo: Importar datos desde excel a una tabla.

Hacer click derecho sobre la tabla y seleccionar Importar Datos…

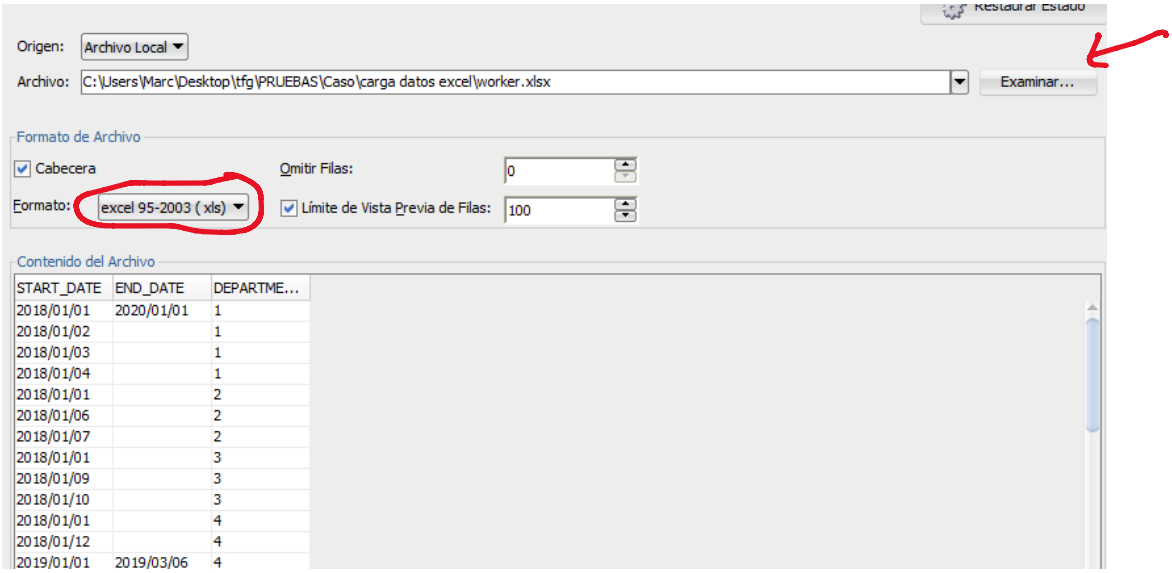


O bien abrir la tabla, acciones, Importar Datos…

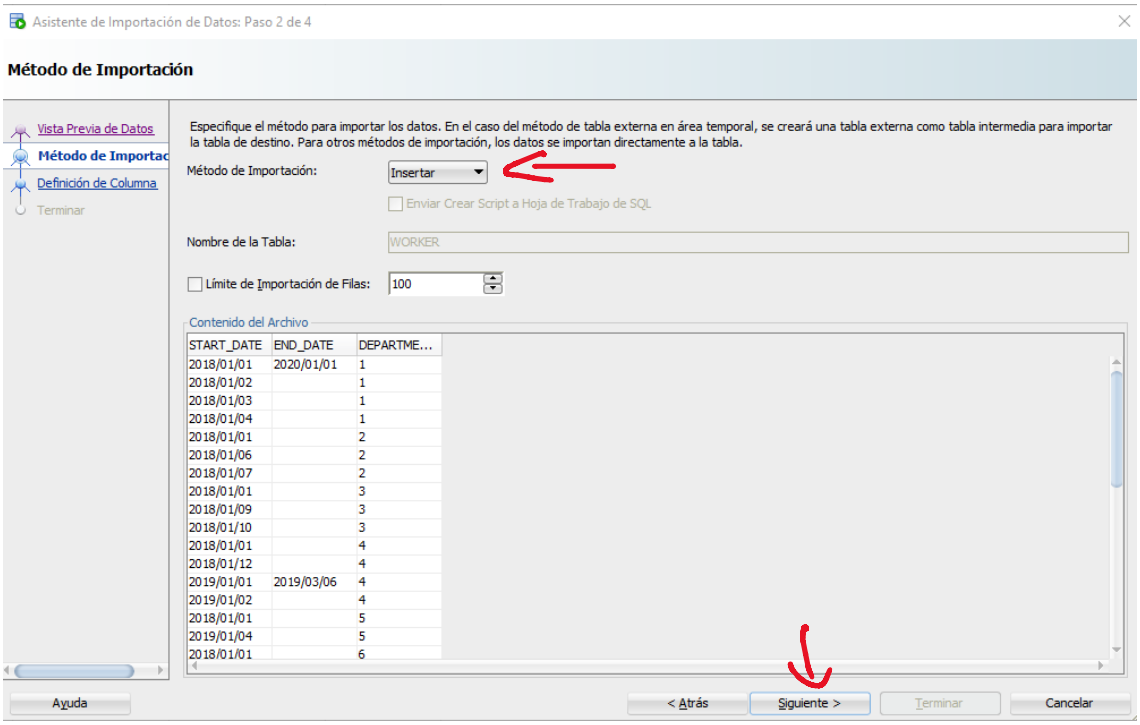




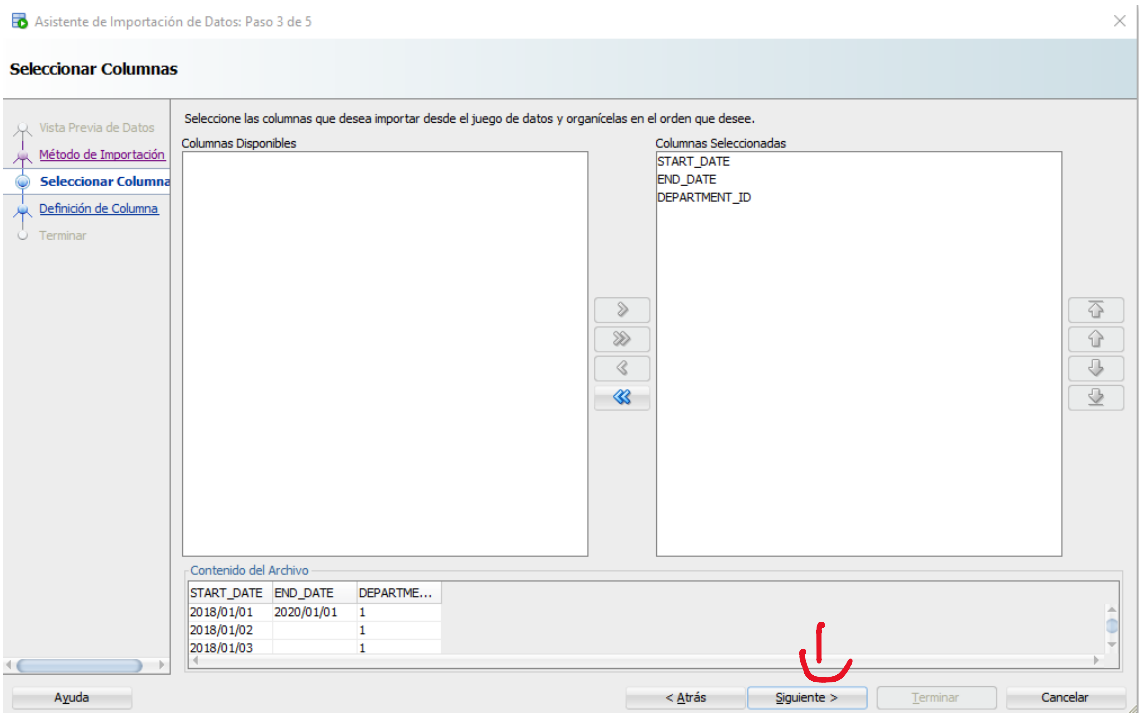
Se abrirá la siguiente ventana. Click en Examinar… para seleccionar el archivo que contiene los datos a importar. Aparecerá un pequeño resumen, así como el tipo de archivo que es.



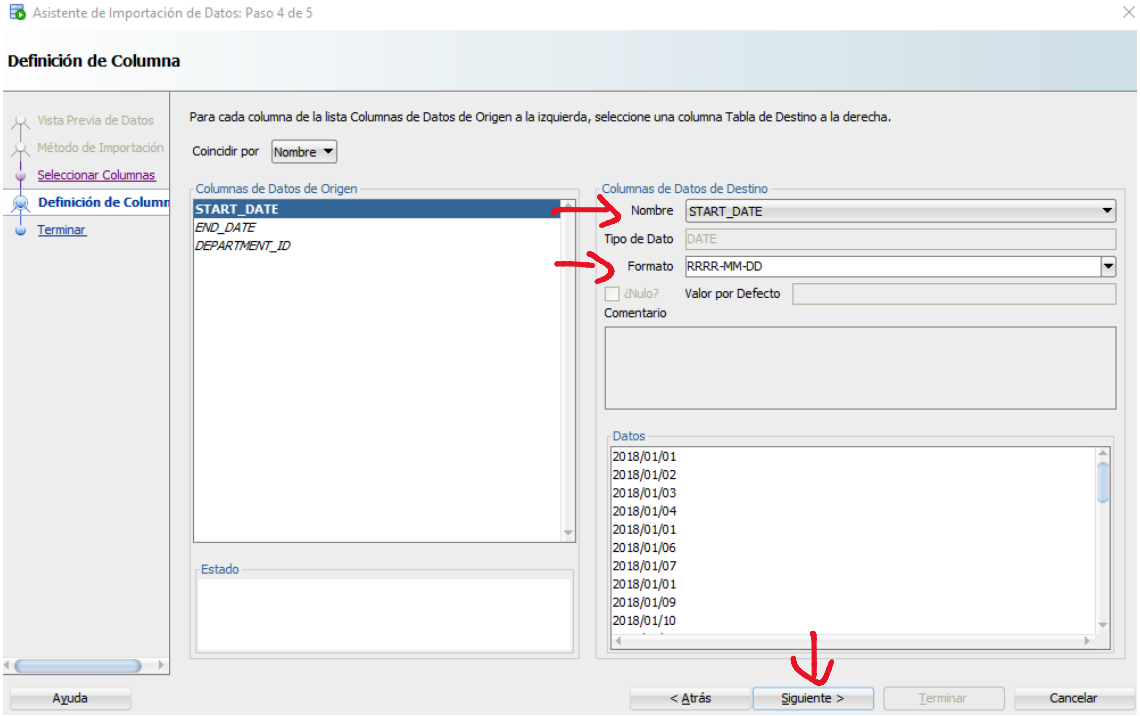
En la siguiente ventana, podemos escoger el método de importación. Insertar insertará directamente el contenido, mientras que insertar script nos creará una lista de métodos INSERT.



Si vamos a la siguiente, podemos escoger las columnas y ordenarlas.


Asistente de Importación de Datos: Paso 3 de 5
Seleccionar Columnas
Vista Previa de Datos
Método de Importación
Seleccionar Columna
Definición de Columna
Terminar
Seleccione las columnas que desea importar desde el juego de datos y organícelas en el orden que desee.
Columnas Disponibles
Columnas Seleccionadas
START_DATE
END_DATE
DEPARTMENT_ID
Contenido del Archivo
START_DATE
END_DATE
DEPARTME...
2018/01/01
2020/01/01
1
2018/01/02
1
2018/01/03
1
Ayuda
< Atrás
Siguiente >
Terminar
Cancelar

En el paso 4, podemos asociar qué datos corresponden a qué columnas de la tabla. También podemos escoger el formato del dato.



El último paso nos muestra un resumen de la importación. Simplemente darle a Terminar. Si como método hemos seleccionado Insertar, se insertarán, sino se abrirá un script como el siguiente: