

Diseño e Implementación de un Sistema de Gestión de Cambios ITIL para Aplicaciones Informáticas

Ángel Martínez de las Heras
Grado de Ingeniería Informática
Bases de Datos

Jordi Ferrer Duran
Josep Cobarsí Morales

01/2024



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

FICHA DEL TRABAJO FINAL

Título del trabajo:	<i>Diseño e Implementación de un Sistema de Gestión de Cambios ITIL para Aplicaciones Informáticas</i>
Nombre del autor:	<i>Ángel Martínez de las Heras</i>
Nombre del consultor/a:	<i>Jordi Ferrer Duran</i>
Nombre del PRA:	<i>Josep Cobarsí Morales</i>
Fecha de entrega (mm/aaaa):	01/2024
Titulación:::	<i>Grado de Ingeniería Informática</i>
Área del Trabajo Final:	<i>Bases de Datos</i>
Idioma del trabajo:	<i>Español</i>
Palabras clave	<i>ITIL, gestión de cambios, base de datos Oracle.</i>
<p>Resumen del Trabajo (máximo 250 palabras): <i>Con la finalidad, contexto de aplicación, metodología, resultados i conclusiones del trabajo.</i></p>	
<p>El presente Trabajo de Fin de Grado (TFG) se centra en el diseño e implementación de un sistema de gestión de cambios ITIL para aplicaciones informáticas. En un contexto empresarial dinámico, se identificó la necesidad de una solución especializada para la gestión de cambios, superando las limitaciones de las ofertas generalistas actuales.</p> <p>Se adoptó un enfoque mixto de desarrollo de producto nuevo, combinando el modelo en cascada con iterativo, ideal para bases de datos. Se logró implementar una base de datos robusta, utilizando <i>Oracle Database</i>, que permitió la gestión eficiente de cambios con procedimientos almacenados y trabajos programados, sin interacción directa de los usuarios con las tablas.</p> <p>Las pruebas confirmaron la correcta funcionalidad del sistema. El trabajo concluye destacando el éxito en el logro de los objetivos específicos, aunque señala áreas para mejoras futuras, como la interfaz de usuario y la gestión de permisos.</p> <p>Adjuntos a esta memoria, se pueden encontrar los <i>scripts</i> de creación de toda la estructura y funcionalidades de la base de datos, otros scripts que contienen las pruebas de ejecución de todos los procedimientos y las consultas para la comprobación de los resultados. Además, para la ejecución de los ficheros SQL</p>	

sin *hardware* dedicado, en el anexo se detalla el despliegue de una base de datos autónoma en *Oracle Cloud Infrastructure*.

Abstract (in English, 250 words or less):

This Final Degree Project (TFG) concerns the design and implementation of an ITIL change management system for IT applications. In a dynamic business context, the need for a specialized change management solution was identified, overcoming the limitations of current generalist offerings.

A mixed approach to new product development was adopted, combining the waterfall model with iterative, ideal for databases. A robust database was implemented, using Oracle Database, which allowed efficient change management with stored procedures and scheduled jobs, without direct user interaction with the tables.

The tests confirmed the correct functionality of the system. The paper concludes by highlighting the success in achieving the specific objectives, although it points out areas for future improvements, such as the user interface and permissions management.

Attached to this report are scripts for the creation of all the database structure and functionalities, other scripts containing the execution tests of all the procedures and the queries for the verification of the results. In addition, for the execution of SQL files without dedicated hardware, the annex details the deployment of a standalone database on Oracle Cloud Infrastructure.

Índice

1. Introducción.....	1
1.1 Contexto y justificación del Trabajo.....	1
1.2 Objetivos del Trabajo.....	1
1.2.1 Objetivo general	1
1.2.2 Objetivos específicos	2
1.3 Enfoque y método seguido.....	2
1.4 Planificación del Trabajo	3
1.4.1 Definición y temporalización de las tareas	3
1.4.2 Análisis de riesgos	7
1.5. Aspectos vinculados a ética, sostenibilidad, responsabilidad social y/o derechos humanos y diversidad.....	9
1.6 Breve resumen de productos obtenidos	10
1.7 Breve descripción de los otros capítulos de la memoria.....	10
1.8 Seguimiento de la planificación	11
1.8.1 PEC1	11
1.8.2 PEC2.....	11
1.8.3 PEC3.....	11
1.8.4 PEC4.....	12
2. Análisis de requisitos.....	13
3. Diseño de la Base de Datos	15
3.1 Diseño conceptual.....	15
3.1.1 Diagrama general.....	15
3.1.2 Diagrama de repositorio estadístico y logs de acciones.....	17
3.2 Diseño lógico.....	18
3.3 Diseño físico.....	20
3.3.1 Recursos.....	20
3.3.2 Tablespaces.....	21
3.3.3 Tablas	21
3.3.4 Índices.....	22
3.3.5 Disparadores.....	22
3.3.6 Procedimientos almacenados	23
3.3.7 Trabajos programados	30
4. Implementación	32
4.1 <i>Tablespaces</i>	32
4.2 Tablas.....	32
4.3 Índices	32
4.4 Disparadores	33
4.5 Procedimientos almacenados	33
4.6 Trabajos programados	33
5. Pruebas	34
5.1 Adición de aprobadores.....	34
5.2 Adición de aplicaciones	35
5.3 Asignación de aprobadores a aplicaciones	36
5.4 Adición de cambios	38
5.5 Aprobación de cambio.....	39

5.6 Rechazo de cambio.....	41
5.7 Ejecución de cambio	42
5.8 Cambio replanificado para modificación de App	43
5.9 Nuevos cambios con diferentes situaciones.....	44
5.10 Baja de aplicación	46
5.11 Eliminado de categoría de cambio	46
5.12 Auditorías	47
5.13 Comprobación de la tabla de logs y del repositorio estadístico.....	48
6. Conclusiones.....	51
7. Glosario	53
8. Bibliografía	55
9. Anexos	56

Lista de figuras

Il·lustració 1 - Diagrama temporal	3
Il·lustració 2 - Planificació de temps	5
Il·lustració 3 - Listado de tareas	6
Il·lustració 4 - Diagrama de Gantt.....	7
Il·lustració 5 - Diagrama UML general.....	16
Il·lustració 6 - Diagrama UML RE y logs	18
Il·lustració 7 - Modelo físico.....	22
Il·lustració 8 - Listado de aprobadores	35
Il·lustració 9 - Listado de aplicaciones.....	36
Il·lustració 10 - Listado de aprobadores de aplicació 1	37
Il·lustració 11- Listado de aprobadores de aplicació 2	38
Il·lustració 12 - Listado de cambios punto 5.4	39
Il·lustració 13 - Cambios en proceso de aprobación punto 5.4	39
Il·lustració 14 - Listado de votos de cambio 1	40
Il·lustració 15 - Listado de ejecución pendiente de cambio 1	40
Il·lustració 16 - Cambios en proceso de aprobación punto 5.5	40
Il·lustració 17 - Cambios aprobados en el año en curso punto 5.5.....	41
Il·lustració 18 - Técnico con más cambios en curso punto 5.5.....	41
Il·lustració 19 - Listado de votos cambio ID 2.....	41
Il·lustració 20 - Listado de cambios punto 5.6	42
Il·lustració 21 - Cambios en proceso de aprobación punto 5.6	42
Il·lustració 22 - Cambios no aprobados punto 5.6	42
Il·lustració 23 - Listado de ejecuciones punto 5.7.....	42
Il·lustració 24 - Técnico con más cambios ejecutados de forma correcta punto 5.7	43
Il·lustració 25 - Cambio con ID 5	43
Il·lustració 26 - Votos de cambio con ID 5.....	43
Il·lustració 27 - Plan de acción creado al reprogramar ejecución.....	44
Il·lustració 28 - Acción del plan de acción con ID 1	44
Il·lustració 29 - Listado de ejecuciones del cambio con ID 5.....	44
Il·lustració 30 - Porcentaje de cambios replanificados en zona Europa	44
Il·lustració 31 - Votos de cambio con ID 6.....	45
Il·lustració 32 - Cambio con ID 6	45
Il·lustració 33 - Ejecución con incidentes de cambio con ID 6.....	45
Il·lustració 34 - ActionPlan generado tras la ejecución con incidentes	45
Il·lustració 35 - Actions vinculadas al ActionPlan con ID 2	45
Il·lustració 36 - Porcentaje de Actions cerradas a tiempo punto 5.9.....	46
Il·lustració 37 - Número de ActionPlan abiertos punto 5.9	46
Il·lustració 38 - Cambios pendientes de ejecución punto 5.9	46
Il·lustració 39 - Detalle de la baja de la aplicación con ID 2	46
Il·lustració 40 - Eliminado de categoría de cambio.....	47
Il·lustració 41 - Número de incumplimientos en los últimos 6 meses antes de añadir datos.....	47
Il·lustració 42 - Incumplimientos detectados en auditoría.....	48
Il·lustració 43 - Número de incumplimientos en los últimos 6 meses el día siguiente de añadir datos	48

Il·lustració 44 - Listado de auditorías y número de incumplimientos	48
Il·lustració 45 - Listado de logs.....	49
Il·lustració 46 - Listado de errores en tabla de logs.....	49
Il·lustració 47 - Listado de ejecuciones del procedimiento "addComment"	49
Il·lustració 48 - Log de ejecución de trabajos programados	50
Il·lustració 49 - RE_Geographic	50
Il·lustració 50 - RE_LastSixMonths	50
Il·lustració 51 - RE_Year	50

1. Introducción

1.1 Contexto y justificación del Trabajo

En un entorno empresarial dinámico y en constante evolución como el del desarrollo de *software*, las organizaciones enfrentan la necesidad de adaptar y mejorar continuamente sus aplicaciones informáticas.

Los procesos de gestión de cambios dentro del marco de buenas prácticas de la *Information Technology Infrastructure Library* (ITIL), son adoptados por muchas empresas para manejar estas tareas de manera estructurada y controlada, minimizando los riesgos y asegurando la continuidad del negocio.

Sin embargo, el mercado actual parece carecer de soluciones que aborden específicamente las necesidades de gestión de cambios en aplicaciones informáticas, ya que las ofertas existentes son generalmente percibidas como demasiado generalistas.

El departamento comercial ha reconocido una oportunidad en el desarrollo de una solución dedicada, enfocada exclusivamente en este aspecto de la ITIL, que podría tener una gran acogida entre las empresas del sector, satisfaciendo una necesidad no cubierta por las soluciones actuales.

Para cumplir este objetivo, en primer lugar, una base de datos bien estructurada es fundamental para el desarrollo de esta aplicación especializada. Su estructura debe ser lo suficientemente robusta y flexible para soportar la variedad y complejidad del proceso definido de gestión de cambios, permitiendo a su vez la realización de análisis detallados y la generación de informes estadísticos de manera eficiente.

Además, la gestión efectiva de grandes volúmenes de datos es necesaria, dado el requisito de que la aplicación debe funcionar con cualquier volumen de datos. Esto implica la necesidad de implementar técnicas de "*Data Warehouse*", para asegurar respuestas rápidas y precisas a las consultas, incluso en entornos de alta demanda.

1.2 Objetivos del Trabajo

1.2.1 Objetivo general

- Implementar un sistema de Base de Datos (BD) para gestionar uno de los procesos ITIL más importantes: la gestión de cambios en aplicaciones informáticas.

1.2.2 Objetivos específicos

- Aplicar de forma práctica los conocimientos adquiridos sobre bases de datos en las tres asignaturas relativas, así como otros conocimientos adquiridos en otras asignaturas como “gestión de proyectos” o “fundamentos de sistemas de información”.
- Analizar y entender profundamente las necesidades fundamentales del proceso de gestión de cambios propuesto por la empresa.
- Diseñar una estructura de base de datos que se alinee con los requerimientos y restricciones especificados, asegurando eficiencia y escalabilidad.
- Utilizar eficientemente el lenguaje SQL para estructurar y consultar la base de datos diseñada para este propósito.
- Implementar procedimientos de base de datos que encapsulen y estandaricen las operaciones de acceso a la información, garantizando la integridad y seguridad de los datos.
- Identificar y proponer funcionalidades adicionales que puedan aportar un valor diferencial a la aplicación y a la gestión del proceso.

1.3 Enfoque y método seguido

Tras analizar las estrategias potenciales para llevar a cabo el proyecto de desarrollo del sistema de base de datos para la gestión de cambios en aplicaciones informáticas siguiendo los procesos ITIL, la estrategia escogida es el desarrollo de producto nuevo.

Como primera opción, se ha contemplado la adaptación de una solución existente. Sin embargo, se descarta por la percepción de que las soluciones actuales son demasiado generalistas.

Otra opción es la integración de múltiples sistemas para crear una solución más completa. En este caso, se descarta por el peligro de que resulte en problemas de integración y de complejidad en la administración.

Por último, la opción de desarrollo de nuevo producto enfocada en los objetivos generales y específicos del sistema da mayor especificidad a las necesidades transmitidas. Además, permite tener el control total del diseño y desarrollo para adaptarse a los procesos específicos de gestión de cambios.

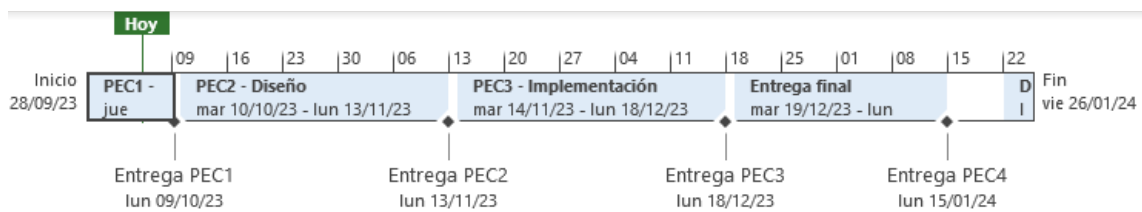
Por otro lado, se adoptará un enfoque de desarrollo de software que combina el modelo en cascada con un enfoque iterativo. Este método híbrido es ideal para el diseño e implementación de bases de datos en ingeniería de software, permitiendo tanto una planificación estructurada como la flexibilidad para realizar ajustes a lo largo del desarrollo.

En este caso, cada fase del proyecto en cascada se abordará de manera iterativa, lo que permitirá probar nuevas funcionalidades a medida que estén preparadas y corregir errores antes de que sea demasiado tarde. También, al final de cada fase, se realizará una revisión que destaque posibles ajustes y mejoras.

Este método garantiza que el proyecto avance de manera estructurada pero adaptable, permitiendo responder eficazmente a los desafíos y cambios, además de asegurar la entrega de una base de datos robusta en tiempo y forma.

1.4 Planificación del Trabajo

En primer lugar, como disponemos de cuatro entregas cerradas en fecha antes del debate virtual, se comienza añadiendo en un diagrama de Gantt cada entrega como tarea de resumen. Además, se establecen las fechas de entrega de cada PEC como hitos. Para realizar este diagrama se utilizará el *software Microsoft Project* con licencia obtenida a través de *Azure for Education* ¹.



II-lustració 1 - Diagrama temporal

1.4.1 Definición y temporalización de las tareas

Posteriormente al análisis del enunciado, se descompone el trabajo en tareas siguiendo una técnica “de mayor a menor”. Es decir, dividir el proyecto en varias partes, para después descomponer esas partes en tareas concretas a las que poder aplicar un seguimiento efectivo.

Por otro lado, se calcula la disponibilidad mínima para dedicar al proyecto en 14 horas semanales divididas en 5 horas en fin de semana y 2 horas otros 2 días laborales. Se tratará de ampliar esta dedicación en lo posible, pero se decide ser conservador en un primer cálculo. Al tener en cuenta este último dato y

¹ [1] Microsoft Azure. (12/11/2023). https://portal.azure.com/?Microsoft_Azure_Education_correlationId=b0a3a712-9b12-4148-a348-57fab1323a47#view/Microsoft_Azure_Education/EducationMenuBlade/~/software

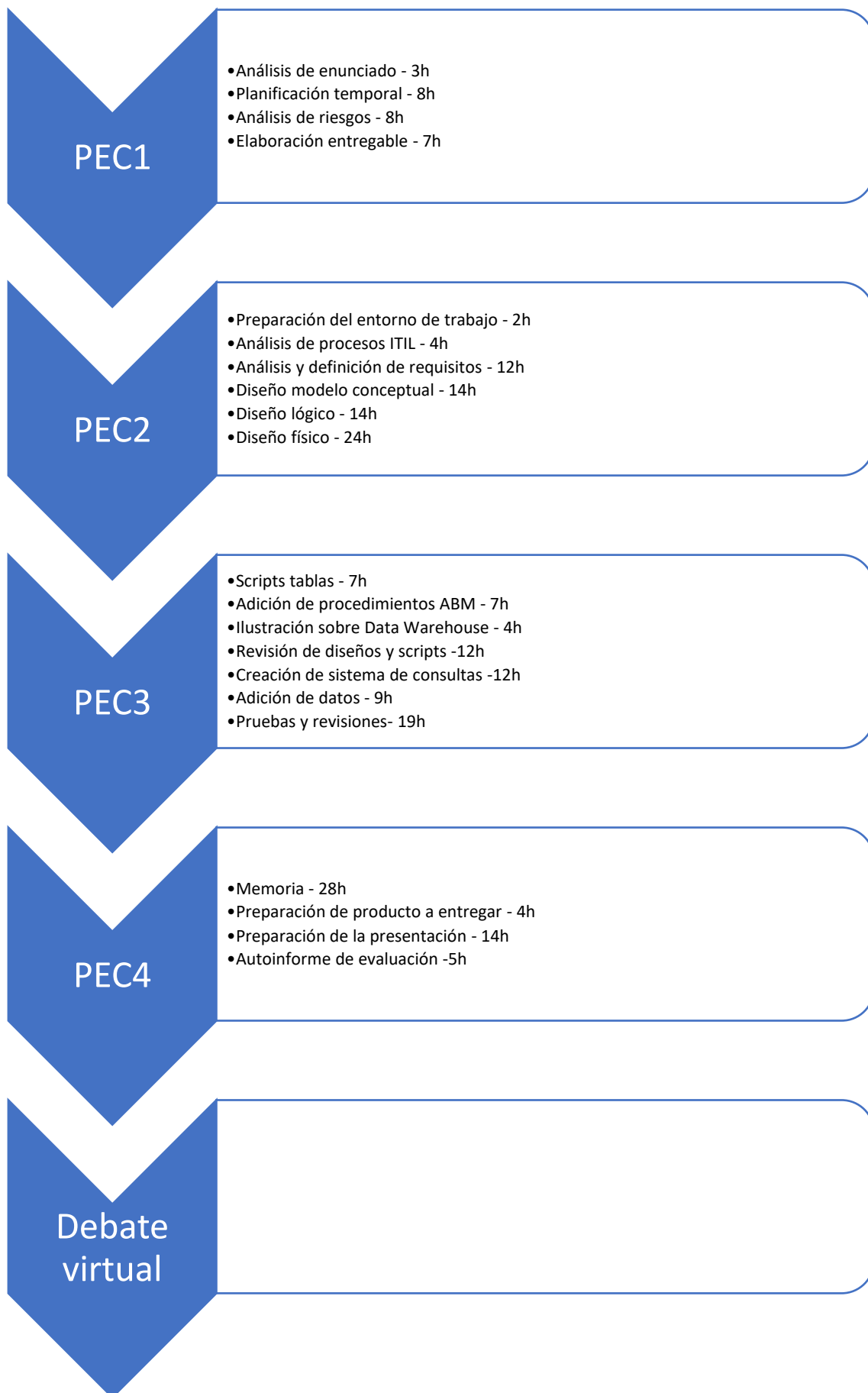
fijándonos en la fecha de las entregas, se deberá dedicar el siguiente tiempo en cada entrega:

Entrega	Fecha de inicio	Fecha de fin	Horas
<i>PEC1</i>	28/09/2023	09/10/2023	26
<i>PEC2</i>	10/10/2023	13/11/2023	70
<i>PEC3</i>	14/11/2023	18/12/2023	70
<i>PEC4</i>	19/12/2023	15/01/2024	56

En este momento, se comienza a planificar el tiempo de las diferentes tareas, teniendo en cuenta que se ha seguido un modelo de reparto de tareas ajustando cada entrega a una fase del proyecto. En el caso de la primera entrega viene claramente definida en la presentación del caso. Sin embargo, la segunda y tercera entrega no están definidas y se decide que se correspondan a la fase de diseño y la fase de implementación. Así, se dispondrá de tiempo de reacción suficiente hasta la entrega final en caso de correcciones².
































En este paso, se vuelve a recurrir al diagrama de Gantt para comprobar que todas las tareas planificadas bajo una entrega pueden llevarse a cabo con el tiempo suficiente. El resultado de la planificación de tiempo para cada tarea es la siguiente.

² [2] Rodríguez, & Mariné Jové, P. (2018). *Gestión de proyectos – módulo 4 / José Ramón Rodríguez (coordinador); Pere Mariné Jové* (Tercera edición). Universitat Oberta de Catalunya.



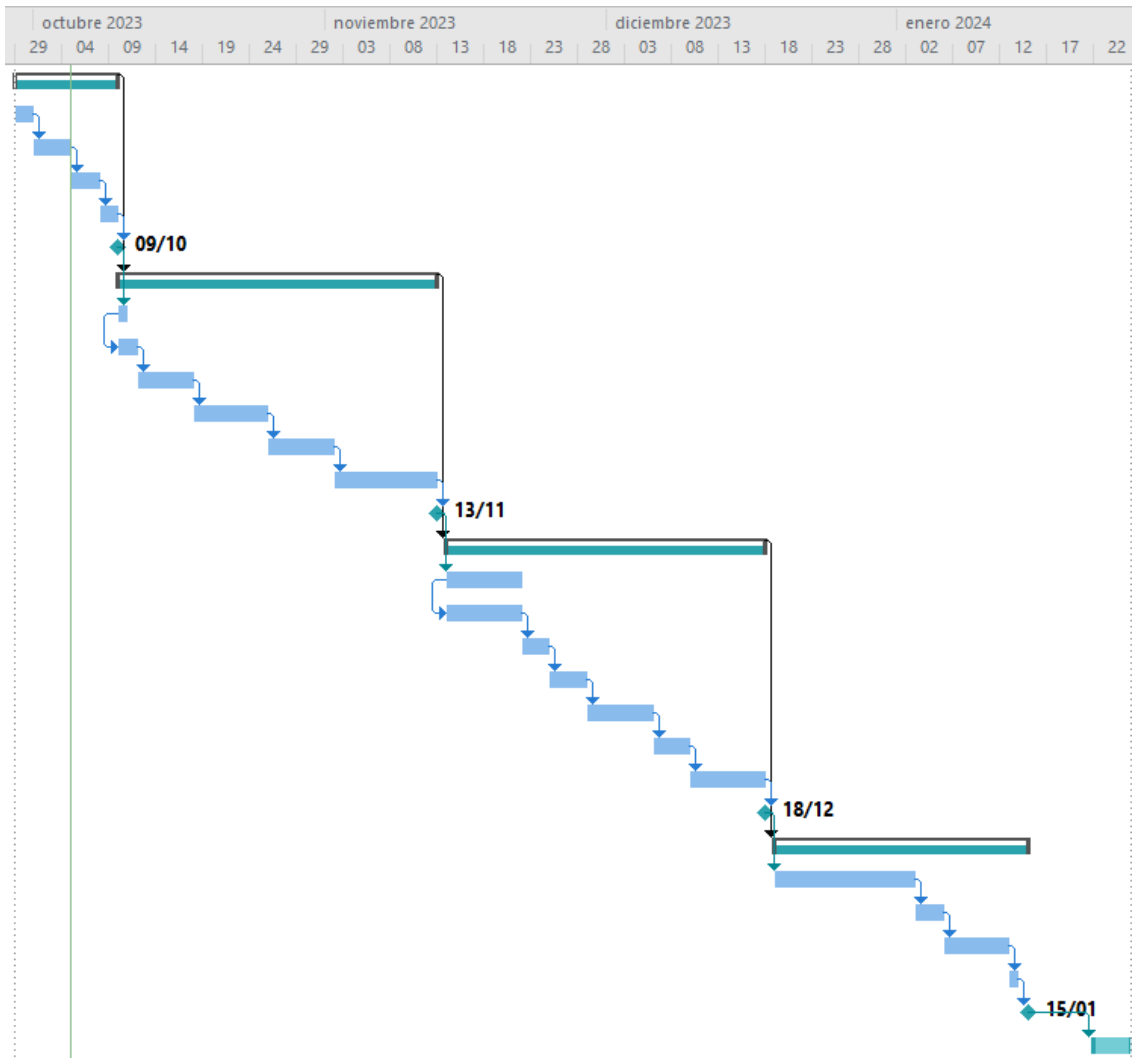
II-lustració 2 - Planificació de temps

A continuación, se listan las diferentes tareas distribuidas en fechas para cumplir la planificación:

		Modo de	Nombre de tarea	Duración	Comienzo	Fin	Predecesoras
1			PEC1 - Definición del plan de trabajo	11 días	jue 28/09/23	lun 09/10/23	
2			Análisis de enunciado	2 días	jue 28/09/23	sáb 30/09/23	
3			Planificación temporal	4 días	sáb 30/09/23	mié 04/10/23	2
4			Análisis de riesgos	3 días	mié 04/10/23	sáb 07/10/23	3
5			Elaboración entregable	2 días	sáb 07/10/23	lun 09/10/23	4
6			Entrega PEC1	0 días	lun 09/10/23	lun 09/10/23	5
7			PEC2 - Diseño	34 días?	mar 10/10/23	lun 13/11/23	1
8			Preparación del entorno de trabajo	1 día	mar 10/10/23	mié 11/10/23	6
9			Análisis de procesos ITIL	2 días	mar 10/10/23	jue 12/10/23	8CC
10			Análisis y definición de requisitos	6 días	jue 12/10/23	mié 18/10/23	9
11			Diseño modelo conceptual	8 días	mié 18/10/23	jue 26/10/23	10
12			Diseño lógico	7 días	jue 26/10/23	jue 02/11/23	11
13			Diseño físico	11 días	jue 02/11/23	lun 13/11/23	12
14			Entrega PEC2	0 días	lun 13/11/23	lun 13/11/23	13
15			PEC3 - Implementación	34 días?	mar 14/11/23	lun 18/12/23	7
16			Scripts tablas	8 días	mar 14/11/23	mié 22/11/23	14
17			Adición de procedimientos ABM	8 días	mar 14/11/23	mié 22/11/23	16CC
18			Ilustración sobre Data Warehouse	3 días	mié 22/11/23	sáb 25/11/23	17
19			Revisión de diseños y scripts	4 días	sáb 25/11/23	mié 29/11/23	18
20			Creación de sistema de consultas	7 días	mié 29/11/23	mié 06/12/23	19
21			Adición de datos	4 días	mié 06/12/23	dom 10/12/23	20
22			Pruebas y revisiones	8 días	dom 10/12/23	lun 18/12/23	21
23			Entrega PEC3	0 días	lun 18/12/23	lun 18/12/23	22
24			Entrega final	27 días?	mar 19/12/23	lun 15/01/24	15
25			Memoria	15 días	mar 19/12/23	mié 03/01/24	23
26			Preparación de producto a entregar	3 días	mié 03/01/24	sáb 06/01/24	25
27			Preparación de la presentación	7 días	sáb 06/01/24	sáb 13/01/24	26
28			Autoinforme de evaluación	1 día	sáb 13/01/24	dom 14/01/24	27
29			Entrega PEC4	0 días	lun 15/01/24	lun 15/01/24	28
30			Debate virtual	4 días	lun 22/01/24	vie 26/01/24	29

Il-lustració 3 - Listado de tareas

Por último, el diagrama de Gantt resultante:



Il·lustració 4 - Diagrama de Gantt

1.4.2 Análisis de riesgos

Para cumplir la planificación anterior, es necesario analizar y reducir en la medida de lo posible cualquier situación que impida el desarrollo del proyecto. Esto significa que el mayor riesgo es el incumplimiento de plazos de cada entrega, pero en especial la entrega final y el debate virtual. Sin embargo, este riesgo principal puede descomponerse en riesgos específicos que se listan a continuación.

	Impacto	Probabilidad	Mitigación
<i>Avería de red</i>	Medio	Baja	Se dispone de conexión 4g que se podría compartir hacia el dispositivo habitual de trabajo. También es posible la compartición de conexión a Internet por parte de vecinos.

<p><i>Avería eléctrica</i></p>	<p>Alto</p>	<p>Baja</p>	<p>Si bien se puede trabajar con los equipos con baterías y realizar entregas mediante conexión 4g, es durante un periodo breve.</p> <p>Por un lado, se poseen personalmente conocimientos en electricidad y herramienta para una reparación de urgencia. También un seguro de hogar con servicio de urgencia.</p> <p>Por otro lado, no existen antecedentes necesarios como para considerar este riesgo muy probable.</p>
<p><i>Avería de hardware</i></p>	<p>Medio</p>	<p>Media</p>	<p>Junto al dispositivo de trabajo habitual, se dispone de un ordenador portátil en caso de avería.</p> <p>En cuanto a los ficheros de trabajo, se trabaja con carpetas sincronizadas con un sistema de almacenamiento en red.</p> <p>Estas carpetas del proyecto también se sincronizan en el equipo portátil y en <i>Microsoft OneDrive</i>.</p>
<p><i>Situaciones personales imprevistas graves</i></p>	<p>Alto</p>	<p>Baja</p>	<p>Ya sean situaciones personales o familiares de salud o ineludibles, situaciones laborales que requieran dedicación extra o deberes de carácter inexcusable.</p> <p>Se planifica el tiempo de dedicación de forma conservadora, esperando poder dedicar más tiempo del planificado y mitigar este tipo de situaciones si ocurren.</p> <p>En caso de ser necesario, se comentaría la situación con el consultor o tutor del TFG en busca de soporte y consejo.</p>
<p><i>Conceptos olvidados o nuevos</i></p>	<p>Medio</p>	<p>Media</p>	<p>A lo largo del desarrollo del proyecto pueden existir conceptos</p>

		<p>o procedimientos no practicados o incluso nuevos.</p> <p>Esto puede llevar a un retraso en la ejecución. Para reducirlo, se lleva recordando conceptos sobre bases de datos relacionales todo el periodo estival.</p> <p>Por otro lado, se dispone de material suficiente (p.e.: biblioteca de UOC, recursos en Internet) para resolver dudas.</p> <p>También se planea escribir al consultor de la asignatura en caso de tener dudas o problemas específicos.</p>
--	--	---

Después de presentar estos riesgos y su acción mitigadora o correctora correspondiente, se concluye que el riesgo menos evitable o corregible y con mayor impacto es una situación personal grave imprevista. Aunque no se prevén ni es común que existan, la propia definición del riesgo indica que, por su naturaleza, no se puede prever.

1.5. Aspectos vinculados a ética, sostenibilidad, responsabilidad social y/o derechos humanos y diversidad

En el contexto de un proyecto de implementación de un sistema de Base de Datos, es esencial trabajar en aspectos vinculados a ética, sostenibilidad, responsabilidad social, derechos humanos y diversidad. A continuación, se detalla cómo se abordarán:

En primer lugar, garantizando la confidencialidad de los datos personales e información almacenados en la BD, asegurando que solo las personas autorizadas tengan acceso a la información que necesitan, y evitando por diseño que existan manipulaciones ocultas.

En el sentido de los datos personales, se evitará guardar características que puedan inducir al racismo, exclusión o cualquier tipo de discriminación. Por ejemplo, datos relativos a religión, género, raza o incluso preferencias sobre gustos, ya que no son necesarios para este proyecto.

Por otro lado, este proyecto contribuirá a la sostenibilidad ambiental diseñando e implementando el sistema de BD de una manera que consuma la menor cantidad de recursos posible, lo que reducirá el consumo eléctrico y la “chatarra electrónica”.

Además, en el momento de alojamiento de la BD, se aconsejará encarecidamente que sea en la nube y se tendrá preferencia por soluciones sostenibles medioambientalmente como un centro de datos autoabastecido en el mayor porcentaje de consumo eléctrico posible mediante energía renovable.

Por último, como es previsible que el sistema se implementará para cubrir múltiples áreas geográficas, se asegurará el cumplimiento de la normativa aplicable a todas las dichas áreas, especialmente en materia de derechos humanos, laborales y protección de datos.

1.6 Breve resumen de productos obtenidos

A la finalización de este proyecto, se obtendrán diferentes productos que serán referenciados a lo largo de esta memoria, y son los siguientes:

- **Memoria:** documento que detalla la planificación y desarrollo del proyecto.
- **Presentación:** vídeo explicativo del proyecto para ser presentado ante el tribunal.
- **Producto:** scripts en formato SQL demostrativos del proyecto. Incluye en diferentes ficheros todas las sentencias necesarias para la creación de la base de datos, pruebas de todos los procedimientos, consultas comunes de comprobación y fichero para eliminar todo lo creado.
- **Autoinforme:** informe de autoevaluación a presentar junto al proyecto.
- **Anexo:** despliegue de base de datos autónoma en *Oracle Cloud Infrastructure*.

1.7 Breve descripción de los otros capítulos de la memoria

- **Análisis de requisitos:** definición de requisitos del proyecto, incluyendo los requisitos para el repositorio estadístico y un log de acciones.
- **Diseño de la Base de Datos:** proceso de diseño de la base de datos, dividido en tres secciones secuenciales: diseño conceptual, diseño lógico y diseño físico.
- **Implementación:** realización práctica de creación del sistema, implementando los elementos del diseño lógico y físico en el sistema gestor de bases de datos.
- **Pruebas:** detalle de pruebas realizadas para verificar la funcionalidad y robustez del sistema.

1.8 Seguimiento de la planificación

1.8.1 PEC1

La primera entrega, tras una consulta sobre el contenido del documento a entregar con el profesor del trabajo se realiza la planificación y se cumple con todas las tareas definidas en este primer periodo.

Sin embargo, tras la revisión por parte del profesor, se detecta la falta de los aspectos vinculados a ética, sostenibilidad, responsabilidad social y/o derechos humanos y diversidad. Hecho que se corrige en la siguiente entrega añadiéndolos al documento y teniendo en cuenta su cumplimiento en todo el proyecto.

1.8.2 PEC2

El periodo hasta la segunda entrega transcurre con normalidad, aunque se ha necesitado más tiempo del planificado para el diseño lógico debido a la falta de bagaje con esta fase.

Por otro lado, en el transcurso del diseño físico se aprovecha para crear simultáneamente los diferentes ficheros de creación de tablas, ya que se prevé que la implementación de los procedimientos ABM supondrán un esfuerzo considerable por su cantidad, hecho detectado durante la fase de diseño de los procedimientos. Fase que no se ha terminado en el momento de la entrega.

Pese a la ligera demora al inicio, se puede llegar a la entrega con un ritmo de trabajo muy aceptable. En todo caso, se ha avanzado con una tarea de la fase posterior y se debe terminar una tarea de la fase actual, por lo que la planificación ha variado ligeramente.

1.8.3 PEC3

En la tercera entrega se cumplen todas las fases y tareas de la planificación original, si bien es cierto que es necesario simultanear ciertas tareas, como en el final de la entrega anterior, para aumentar la eficiencia del desarrollo. Sin embargo, no se trata de cambios que alteren el proyecto o la fecha de las fases.

Por ejemplo, en el transcurso de la adición de los procedimientos ABM, también se realizaban pruebas de adición de datos para garantizar la funcionalidad. Si en este punto se detectaba un hecho que afectaba a otra fase y era necesaria una revisión, se realizaba en ese momento sin esperar a que

En cuanto a imprevistos, aproximadamente en la mitad del periodo de elaboración de esta fase, aparece una avería en la máquina virtual que aloja la

base de datos que la vuelve inservible. A pesar de este contratiempo, se mitiga rápidamente creando una cuenta gratuita en *Oracle Cloud Infrastructure* con el objetivo de seguir trabajando de forma inmediata.

Al poseer desde el inicio todos los scripts de creación de la base de datos en varios lugares, es posible replicarla en muy poco tiempo mientras se crea una nueva máquina virtual y se realizan las instalaciones de *software* necesarias. Esta base de datos de respaldo no se elimina cuando se vuelve a tener lista la máquina virtual, sino que se aprovecha para comprobar la ejecución de los trabajos programados.

1.8.4 PEC4

En esta entrega final el enfoque principal fue asegurar la preparación del producto final y todos los documentos relacionados, cumpliéndose satisfactoriamente. Se destinó tiempo suficiente para completar la memoria, incluyendo capturas de pantalla de las pruebas y la redacción de secciones pendientes. Además, se elaboró un anexo útil para las pruebas.

Ä pesar de ser la entrega con menos trabajo técnico, se invirtió un esfuerzo considerable en la creación de la presentación, enfocada en comunicar el trabajo realizado y su relevancia.

En síntesis, la temporalización para este periodo no ha variado en cuanto a la planificación inicial, ha sido suficiente para completar el alcance del proyecto y realizar las entregas según lo previsto.

2. Análisis de requisitos

Tras el análisis del enunciado y de los requisitos establecidos para el proyecto, se procede a definirlos³:

- Gestión de cambios al variar cualquier funcionalidad técnica de una aplicación, ya sea en su software o hardware (pág. 2).
- Se debe tener un inventario de todas las aplicaciones en producción (pág. 3).
- Cada cambio debe seguir un proceso de aprobación que involucre a varios aprobadores, como el responsable técnico, responsable del área de negocio, gestor de cambios de la empresa y GCAB (pág. 3).
- Existen GCAB sin derecho a voto que pueden estar informadas o aportar su opinión sobre el cambio (pág. 3).
- Se debe poder gestionar personas substitutas para actuar en ausencia de los aprobadores (pág. 4).
- La BD debe registrar todas las aprobaciones realizadas, indicando quién la aprobó, con qué rol cuándo, junto con los comentarios realizados (pág. 4).
- Se deben poder definir diferentes categorías de cambio basadas en su importancia (pág. 4).
- Se debe considerar el impacto geográfico de los cambios para definir la categoría del cambio, teniendo en cuenta el número de usuarios afectados y el impacto económico (pág. 4).
- Se debe monitorizar la ejecución de los cambios después de su aprobación. Si la ejecución no se clasifica como correcta, se almacenarán las acciones para corregirla (pág. 4-5).
- La aplicación debe permitir la realización de auditorías para validar la correcta ejecución del proceso de gestión de cambios y almacenar los incumplimientos (pág. 5).
- Se deben implementar procedimientos de Alta, Baja y Modificación de todas las entidades relevantes (pág. 6).
- La gestión de los datos almacenados debe seguir técnicas aplicadas a grandes volúmenes de datos (pág. 6).
- Se deben poder realizar diversas consultas estadísticas sobre los datos (pág. 6).
- Se debe implementar un log de acciones realizadas y herramientas para testear la funcionalidad de la BD (pág. 7).
- Los procedimientos almacenados deben tener un parámetro de salida llamado RSP, que indica si la ejecución ha sido correcta o ha fracasado y tratamiento de excepciones (pág. 7).

³ [3] Casas Roma. (2019). Introducción al diseño de bases de datos / Jordi Casas Roma. Universitat Oberta de Catalunya.

En cuanto al repositorio estadístico al que realizar consultas, se definen los siguientes requisitos:

- El repositorio estadístico debe ofrecer resultados en tiempo constante 1, es decir, haciendo solo una SELECT sobre un registro de una tabla sin usar vistas calculadas o materializadas, ni funciones de agregado con Group by.
- Como mínimo, el repositorio deberá permitir las siguientes consultas:
 - Número de cambios registrados en proceso de aprobación.
 - Número de cambios aprobados en el último año con ejecución incorrecta.
 - Responsable técnico con más cambios ejecutados correctamente.
 - Porcentaje por año de acciones definidas para ejecuciones no correctas cerradas en el tiempo inicialmente definido.
 - Número total de cambios aprobados en la GCAB en el año en curso.
 - Número de cambios no aprobados en un año concreto.
 - Porcentaje de cambios replanificados en una región geográfica concreta.
 - Número de incumplimientos detectados en auditorías en los últimos 6 meses.
 - Número total de planes de acción sobre ejecuciones no correctas que están abiertos.
 - Responsable técnico con más cambios en curso.
 - Persona con rol de aprobador que ha sido sustituida más veces por indisponibilidad no planificada.

3. Diseño de la Base de Datos

3.1 Diseño conceptual

La primera fase de diseño tras el análisis de requisitos es el diseño conceptual de la BD. Para realizarlo, se sigue una metodología basada en el modelo Entidad-Relación (ER). Como se comprobó al realizar el análisis de requisitos, el enunciado explica una situación donde se proporciona una visión general del sistema, pero también se han introducido detalles específicos funcionales y no funcionales que deben incorporarse en el diseño, por lo que se sigue una estrategia mixta⁴.

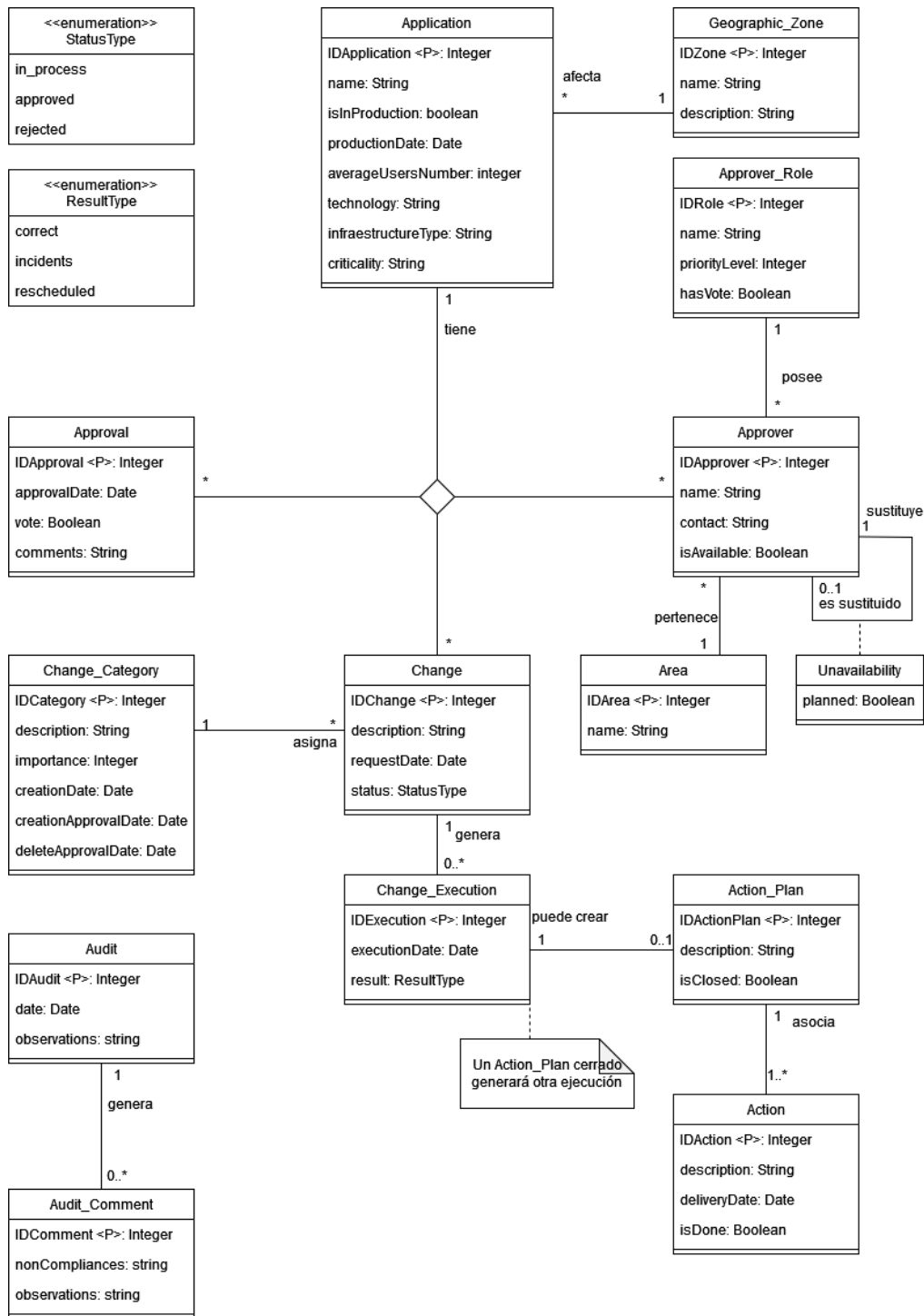
En primer lugar, se sigue un enfoque descendente identificando las principales entidades explicadas en el documento y sus relaciones a alto nivel. En este punto, también se identifican atributos de cada entidad. Posteriormente, se realiza un desglose de dichas entidades principales para seguir cumpliendo requisitos.

En segundo lugar, se cambia a un enfoque ascendente para identificar atributos y relaciones que no fueron evidentes al realizar el primer paso, e incluso añadir nuevas entidades secundarias.

A continuación, se presenta los diferentes diagramas UML resultantes del diseño.

3.1.1 Diagrama general

⁴ [4] Casas Roma. (2020). Diseño conceptual de bases de datos / Jordi Casas Roma, Josep Cuartero Olivera (Quinta edición). Universitat Oberta de Catalunya.



II-lustració 5 - Diagrama UML general

A partir del caso de uso típico identificado, una aplicación es dada de alta y asociada con los responsables (*approver*) necesarios. Dicha alta genera un cambio, que es la puesta en producción, con su consecuente aprobación, si procede, y su ejecución. Aunque parezca que tenemos un dato redundante como si la aplicación está en producción y una fecha, es posible que se planifique cierta fecha de puesta en producción pero que no se lleve a cabo.

Durante el proceso de aprobación y para seguir el flujo correcto, cada responsable cuenta con una prioridad de voto, así que primero votará el que

menor valor tiene en prioridad y así sucesivamente. Que existan responsables con el mismo nivel de prioridad no será un problema, ya que la siguiente prioridad votará cuando no existan más responsables con esa prioridad.

También pueden existir responsables asociados sin derecho a voto, que podrán realizar comentarios. Todo esto dentro de la libertad de poder añadir nuevos roles de responsables con derecho a voto o no y con prioridad variable, o un colaborador que observe todo el proceso y pueda comentar como si, por ejemplo, no tiene nivel de prioridad, que se entenderá como que tampoco tiene voto.

Posteriormente, se pueden generar los cambios que se consideren oportunos, para que vuelvan a ser aprobados o no de igual forma.

Si dicha ejecución de un cambio no es correcta, genera un plan de acción el cual puede tener las acciones que sean necesarias. Cuando el plan de acción es cerrado, se generará una nueva ejecución.

En cuanto a los aprobadores sin derecho a voto, se ha tenido en cuenta categorizándolos con otra entidad. Además, esto permitirá añadir más colaboradores a cada aplicación, tanto con derecho a voto como solo para proporcionar observaciones o tener información sobre el cambio.

Por otro lado, se ha tenido en cuenta el sistema de sustitución de aprobadores al considerar si están disponibles y creando una relación entre ellos. Esta relación es opcional ya que no pueden existir sustitutos infinitos, por lo que se deberá controlar en la implementación que todos los aprobadores con derecho a voto tengan sustituto.

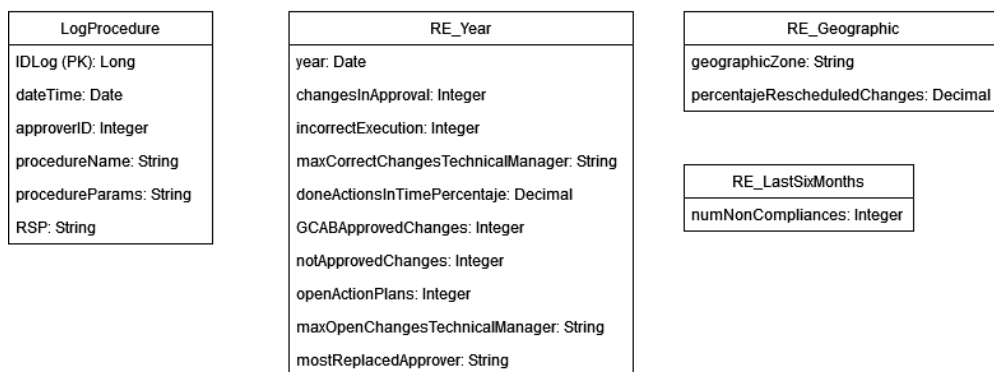
La categoría de un cambio también está contemplada con todos sus atributos necesarios. En cuanto a la asignación, aunque se planea en un principio que se asigne manualmente, existen los atributos necesarios tanto en la entidad aplicación como en la entidad de categoría para realizar la implementación de un sistema de prioridades en importancia e impacto de cambios.

Por supuesto, es posible crear nuevas categorías de cambio con una importancia igual, menor o superior a otra, hecho que combinado con el sistema explicado en el párrafo anterior podrá utilizarse para priorizar un cambio frente a otro, aunque, por ejemplo, tenga la misma importancia, pero el impacto en número de usuarios sea mayor.

Finalmente, la entidad auditoría no está relacionada con el resto de entidades principales, pero permite añadir tantos incumplimientos como sea necesario en cada auditoría.

3.1.2 Diagrama de repositorio estadístico y logs de acciones

Por otro lado, se presenta el diseño del repositorio estadístico y de la tabla para almacenar los logs de acciones realizadas.



II-Il·lustració 6 - Diagrama UML RE y logs

En este caso, no existen relaciones debido a que los datos contenidos en estas tablas serán añadidos mediante procedimientos para una consulta posterior en tiempo constante 1.

En el caso de la tabla de logs, la clave primaria se define de tipo *long* ya que se prevé que contendrá una gran cantidad de registros.

En el caso de las tablas del repositorio, se ha creado una sola entidad para las consultas del año actual y del repositorio de años para añadir consultas estadísticas sobre dichas combinaciones. En el caso de las consultas que requieren que el dato sea de varios años, se planifica que cada año contenga datos de todo el histórico hasta ese año.

Estas entidades se actualizarán conforme varíen los datos relacionados y se añadirá una nueva fila cada año. Para la métrica sobre incumplimientos de los últimos 6 meses, se planifica almacenar un procedimiento que realice el cálculo diariamente.

3.2 Diseño lógico

Tras poseer el modelo conceptual y todavía sin decidir el sistema gestor de bases de datos (SGBD), siguiendo las reglas de transformación contempladas en el recurso “Diseño lógico de bases de datos”⁵, se obtendrá el esquema relacional.

En resumen, las reglas seguidas y notación consisten en:

- **Relaciones a muchos:** la entidad en el lado "muchos" (*) debe incluir una columna adicional que será la clave primaria de la entidad en el otro lado. Esta nueva columna actúa como una clave foránea.
- **Relaciones muchos a muchos:** esta relación se convierte en una nueva tabla. Esta nueva tabla tendrá claves foráneas que son las claves primarias de las dos entidades que participan en la relación.

⁵ [5] Burgués i Illa. (2020). Diseño lógico de bases de datos / Xavier Burgués Illa, Josep Cuartero Olivera (Quinta edición). Universitat Oberta de Catalunya.

- **Relaciones recursivas:** en este tipo de relación se crea añadiendo una clave foránea en la propia entidad que corresponde a la propia clave primaria de la entidad.
- **Atributos simples:** Se convierten directamente en columnas de la tabla correspondiente.
- **Atributos multivaluados:** Se crean tablas separadas para estos atributos, con una relación de uno a muchos con la entidad original.
- **Claves primarias:** subrayado con línea continua.
- **Claves foráneas:** notación textual tabulada.
- **Atributos NOT NULL:** letra negrita.

Siguiendo estas reglas e iterando con el modelo conceptual, se obtiene el siguiente diseño lógico de la BD:

Application (IDApplication, **name**, **isInProduction**, productionDate, averageUsersNumber, **technology**, **infraestructureType**, **criticality**, **geographicZoneID**, **technicalContact**, **businessContact**)
 {geographicZoneID} is foreign key to GeographicZone (IDZone)
 {technicalContact} is foreign key to Approver (IDApprover)
 {businessContact} is foreign key to Approver (IDApprover)

GeographicZone (IDZone, **name**, description)

ApproverRole (IDRole, **name**, priorityLevel, **hasVote**)

Area (IDArea, **name**)

Approver (IDApprover, **name**, **contact**, **isAvailable**, **approverRoleID**, **areaID**, **substitutelD**)
 {approverRoleID} is foreign key to ApproverRole (IDRole)
 {areaID} is foreign key to Area (IDArea)
 {substitutelD} is foreign key to Approver (IDApprover)

Unavailability (IDUnavailability, **approverID**, **planned**)
 {approverID} is foreign key to Approver (IDApprover)

Approval (approverID, changeID, **approvalDate**, vote, comments)
 {approverID} is foreign key to Approver (IDApprover)
 {changeID} is foreign key to Change (IDChange)

ApplicationApprovers (applicationID, approverID)
 {applicationID} is foreign key to Application (IDApplication)
 {approverID} is foreign key to Approver (IDApprover)

ChangeCategory (IDCategory, **description**, **importance**, **creationDate**, **creationApprovalDate**, deleteApprovalDate)

Change (IDChange, **description**, **requestDate**, **status**, **changeCategoryID**, **applicationID**)

{changeCategoryID} is foreign key to ChangeCategory (IDCategory)

{applicationID} is foreign key to Application (IDApplication)

ChangeExecution (IDExecution, **executionDate**, **result**, **changeID**)

{changeID} is foreign key to Change (IDChange)

ActionPlan (IDActionPlan, **description**, **isClosed**, **changeExecutionID**)

{changeExecutionID} is foreign key to ChangeExecution (IDExecution)

Action (IDAction, **description**, **deliveryDate**, **doneDate**, **actionPlanID**)

{actionPlanID} is foreign key to ActionPlan (IDActionPlan)

Audit (IDAudit, **auditDate**, observations)

AuditComment (IDComment, **nonCompliances**, observations, **auditID**)

{auditID} is foreign key to Audit (IDAudit)

LogProcedure (IDLog, **dateTime**, **user**, **procedureName**, procedureParams, **RSP**)

RE_Year (year, changesInApproval, incorrectExecution, maxCorrectChangesTechnicalManager, doneActionsInTimePorcentaje, GCABApprovedChanges, notApprovedChanges, openActionPlans, maxOpenChangesTechnicalManager, mostReplacedApprover)

RE_Geographic (geographicZone, porcentajeRescheduledChanges)

RE_LastSixMonths (lastDate, numNonCompliances)

3.3 Diseño físico

Tras la fase anterior, el siguiente paso es transformar el diseño lógico en estructuras físicas que indiquen cómo se almacenarán los datos y cómo se accederá a ellos, tratando de optimizar el rendimiento y eficiencia de consulta y almacenamiento de los datos.

3.3.1 Recursos

Para la obtención del producto resultado de este proyecto se utilizará *Oracle Database 21c Express Edition*⁶ en una máquina virtual *Windows*. En cuanto al

⁶ [6] *Oracle Database Express Edition* / Oracle España. (12/11/2023).

<https://www.oracle.com/es/database/technologies/appdev/xe.html>

SGBD a utilizar, este será Oracle SQL Developer 23.1⁷, ambos obtenidos a través del sitio web de Oracle.

3.3.2 Tablespaces

En Oracle Database, los *tablespaces* son contenedores lógicos para almacenar los datos. Por ello, y teniendo en cuenta el requisito relacionado con *Data-Warehouse* y escalabilidad, se decide la creación de dos *tablespaces* en adición a los cinco que existen por defecto en la base de datos. Estos contenedores servirán para almacenar, por un lado, los índices que se crearán posteriormente y, por otro lado, el repositorio estadístico.

3.3.3 Tablas

El siguiente paso es la creación de tablas. Sin embargo, antes de ello se debe decidir el tipo de dato a utilizar en el diseño físico a partir de lo analizado en anteriores fases.

Mientras que para la mayoría de IDs se utilizará el tipo Integer, para la tabla de logs se utilizará el tipo Number ya que previsiblemente contendrá gran cantidad de registros y este tipo permite almacenar hasta 38 dígitos. En el caso de las cadenas de texto, en su gran mayoría se implementarán como Varchar2 y el caso de las fechas, serán de tipo Date. En el caso de los atributos binarios, se escoge un tipo Number de un dígito con restricción de valor a 0 o a 1.

En este proceso, y debido a la elección realizada de BD, se cambian algunos nombres de columnas ya que son palabras reservadas. En concreto, “year” por “logYear”, “user” por “logUser” y el nombre de la tabla “Audit” por “ProcessAudit”, A continuación, se muestra el modelo físico obtenido tras la generación:

⁷ [7] Oracle SQL Developer Downloads. (15/11/2023). <https://www.oracle.com/database/sqldeveloper/technologies/download/>

Después, la creación de disparadores será vital para el manejo de las claves primarias de las tablas. Se creará un disparador por tabla que tenga un ID único. En el caso de las tablas del repositorio estadístico, no se crearán estos disparadores ya que la clave primaria se manejará mediante procedimientos almacenados.

3.3.6 Procedimientos almacenados

En este punto, será el momento de crear los diferentes procedimientos para interactuar con la base de datos, ya que los usuarios no podrán realizar interacciones directas con las tablas. Por lo tanto, se decide crear un procedimiento por cada acción a realizar por los usuarios.

Para garantizar la trazabilidad de los datos, no se crearán procedimientos que eliminen filas. En caso de que alguna acción desencadene el eliminado de alguna fila, se utilizará el borrado lógico indicando en un campo que ese registro está eliminado.

Tampoco se permitirá la introducción, edición o eliminado de datos en el repositorio estadístico ni en la tabla de *logs*. Todos sus datos serán alimentados por otros procedimientos que tengan impacto en sus métricas. Este repositorio solo podrá ser consultado.

También, todos los procedimientos tendrán un parámetro de salida RSP que será “OK” si el procedimiento se ha ejecutado correctamente o “ERROR” – Tipo de error si se ha encontrado algún error. Esta salida se registrará en la tabla de logs por si es necesaria su consulta.

Además de los procedimientos relacionados con cada acción que se pueda realizar en la aplicación, se crean otros para mantener la correcta funcionalidad del repositorio estadístico. Sin embargo, estos procedimientos serán ejecutados por el sistema de forma periódica tal y como se explicará en el siguiente apartado.

A continuación, se detallan los procedimientos almacenados a crear junto con su descripción y funcionamiento:

addApplication	
<i>Descripción</i>	Este procedimiento se utiliza para añadir un nuevo registro en la tabla 'Application'
<i>Parámetros</i>	name, averageUsersNumber, technology, infraestructureType, criticality, geographicZoneID, technicalContact, businessContact
<i>Funcionamiento</i>	<ol style="list-style-type: none"> 1. Inserta el nuevo registro en la tabla 'Application' con 'isInProduction' en 0 y 'productionDate' en NULL. 2. Registra la operación en la tabla 'LogProcedure'.

3. Maneja cualquier excepción que ocurra durante la ejecución.

launchApplication

<i>Descripción</i>	Actualiza un registro en 'Application', estableciendo 'isInProduction' en 1 y 'productionDate' en la fecha actual.
<i>Parámetros</i>	IDApplication
<i>Funcionamiento</i>	<ol style="list-style-type: none"> 1. Verifica 'P_applicationID'. 2. Actualiza 'isInProduction' a 1 y 'productionDate' con fecha actual. 3. Registra la operación en 'LogProcedure'. 4. Maneja excepciones.

updateApplication

<i>Descripción</i>	Este procedimiento se utiliza para actualizar un registro existente en la tabla 'Application'. Permite modificar los campos averageUsersNumber, technology, infraestructureType, criticality, geographicZoneID, technicalContact y businessContact.
<i>Parámetros</i>	IDApplication, averageUsersNumber, technology, infraestructureType, criticality, geographicZoneID, technicalContact, businessContact (excepto IDApplication, el resto de parámetros son opcionales).
<i>Funcionamiento</i>	<ol style="list-style-type: none"> 1. Actualiza los campos especificados de la aplicación en la tabla 'Application'. 2. Registra la operación en 'LogProcedure'. 3. Maneja excepciones.

endApplicationLifeCycle

<i>Descripción</i>	Este procedimiento se utiliza para marcar una aplicación como fuera de producción, indicando el fin de su ciclo de vida en la tabla 'Application'.
<i>Parámetros</i>	IDApplication
<i>Funcionamiento</i>	<ol style="list-style-type: none"> 1. Actualiza el campo 'isInProduction' a 0 para la aplicación especificada. 2. Registra la operación en 'LogProcedure'. 3. Maneja excepciones.

addGeographicZone

<i>Descripción</i>	Este procedimiento se utiliza para añadir un nuevo registro en la tabla 'GeographicZone' y un registro correspondiente en 'RE_Geographic' con un porcentaje de cambios reprogramados inicial en 0.
<i>Parámetros</i>	name, description
<i>Funcionamiento</i>	<ol style="list-style-type: none"> 1. Inserta el nuevo registro en la tabla 'GeographicZone'.

2. Inserta un registro correspondiente en 'RE_Geographic' con 'porcentajeRescheduledChanges' en 0.
3. Registra la operación en 'LogProcedure'.
4. Maneja excepciones.

addApproverRole

<i>Descripción</i>	Este procedimiento se utiliza para añadir un nuevo registro en la tabla 'ApproverRole'.
<i>Parámetros</i>	name, priorityLevel, hasVote
<i>Funcionamiento</i>	<ol style="list-style-type: none"> 1. Inserta un nuevo registro en la tabla 'ApproverRole'. 2. Registra la operación en 'LogProcedure'. 3. Maneja excepciones.

addArea

<i>Descripción</i>	Este procedimiento se utiliza para añadir un nuevo registro en la tabla 'Area'.
<i>Parámetros</i>	name
<i>Funcionamiento</i>	<ol style="list-style-type: none"> 1. Inserta un nuevo registro en la tabla 'Area'. 2. Registra la operación en 'LogProcedure'. 3. Maneja excepciones.

addApprover

<i>Descripción</i>	Este procedimiento se utiliza para añadir un nuevo registro en la tabla 'Approver'.
<i>Parámetros</i>	name, contact, approverRoleID, areaID
<i>Funcionamiento</i>	<ol style="list-style-type: none"> 1. Inserta el nuevo registro en la tabla 'Approver' con 'isAvailable' en 1 y 'substituteID' en NULL. 2. Registra la operación en 'LogProcedure'. 3. Maneja excepciones.

assignSubstituteToApprover

<i>Descripción</i>	Asigna un aprobador sustituto a un aprobador oficial en la tabla 'Approver'. Actualiza el campo 'substituteID' del aprobador oficial con el ID del sustituto.
<i>Parámetros</i>	IDApprover (oficial), IDApprover (sustituto)
<i>Funcionamiento</i>	<ol style="list-style-type: none"> 1. Verifica la igualdad de roles entre el aprobador oficial y el sustituto. 2. Actualiza 'substituteID' en 'Approver'. 3. Registra la operación en 'LogProcedure'. 4. Maneja excepciones.

addUnavailability

<i>Descripción</i>	Este procedimiento registra la indisponibilidad de un approver y asigna autorización a su sustituto.
--------------------	--

<i>Parámetros</i>	approverID, planned
<i>Funcionamiento</i>	<ol style="list-style-type: none"> 1. Inserta un registro en 'Unavailability' y actualiza 'isAvailable' a 0 en 'Approver'. 2. Añade autorización para el sustituto en 'ApplicationApprovers'. 3. Actualiza el approver más sustituido en 'RE_Year' si es no planificado. 4. Registra la operación en 'LogProcedure'. 5. Maneja excepciones.

setApproverAvailability

<i>Descripción</i>	Actualiza el estado de disponibilidad de un approver en 'Approver' a disponible (isAvailable = 1).
<i>Parámetros</i>	approverID
<i>Funcionamiento</i>	<ol style="list-style-type: none"> 1. Actualiza 'isAvailable' a 1 en 'Approver'. 2. Elimina las autorizaciones del sustituto en 'ApplicationApprovers'. 3. Registra la operación en 'LogProcedure'. 4. Maneja excepciones.

addApplicationApprover

<i>Descripción</i>	Este procedimiento se utiliza para añadir un nuevo registro en la tabla 'ApplicationApprovers'.
<i>Parámetros</i>	applicationID, approverID
<i>Funcionamiento</i>	<ol style="list-style-type: none"> 1. Inserta el nuevo registro en la tabla 'ApplicationApprovers'. 2. Registra la operación en 'LogProcedure'. 3. Maneja excepciones.

addChangeCategory

<i>Descripción</i>	Este procedimiento se utiliza para añadir un nuevo registro en la tabla 'ChangeCategory'.
<i>Parámetros</i>	description, importance, creationApprovalDate
<i>Funcionamiento</i>	<ol style="list-style-type: none"> 1. Inserta el nuevo registro en la tabla 'ChangeCategory' con 'creationDate' actual, 'creationApprovalDate' proporcionado y 'deleteApprovalDate' en NULL. 2. Registra la operación en 'LogProcedure'. 3. Maneja excepciones.

deleteChangeCategory

<i>Descripción</i>	Este procedimiento se utiliza para marcar una categoría de cambio como eliminada lógicamente en la tabla 'ChangeCategory'.
<i>Parámetros</i>	IDCategory

<i>Funcionamiento</i>	<ol style="list-style-type: none"> 1. Actualiza 'deleteApprovalDate' en la tabla 'ChangeCategory' para la categoría dada, marcándola como eliminada lógicamente. 2. Registra la operación en 'LogProcedure'. 3. Maneja excepciones.
-----------------------	--

addChange

<i>Descripción</i>	Este procedimiento se utiliza para añadir un nuevo registro en la tabla 'Change'.
<i>Parámetros</i>	description, changeCategoryID, applicationID
<i>Funcionamiento</i>	<ol style="list-style-type: none"> 1. Inserta el nuevo registro en la tabla 'Change' con 'requestDate' actual y 'status' en 'in_process'. 2. Incrementa 'changesInApproval' en 'RE_Year' para el año actual. 3. Registra la operación en 'LogProcedure'. 4. Maneja excepciones.

rejectChange

<i>Descripción</i>	Actualiza el estado de un registro en 'Change' a 'rejected' solo si el registro existe y su estado es 'in_process'.
<i>Parámetros</i>	IDChange
<i>Funcionamiento</i>	<ol style="list-style-type: none"> 1. Verifica la existencia del cambio y su estado. 2. Actualiza 'status' en 'Change' a 'rejected'. 3. Actualiza 'RE_Year' restando 1 a 'changesInApproval' y sumando 1 a 'notApprovedChanges' para el año actual. 4. Registra la operación en 'LogProcedure'. 5. Maneja excepciones.

approveChange

<i>Descripción</i>	Actualiza el estado de un registro en 'Change' a 'approved' solo si el registro existe y su estado es 'in_process'.
<i>Parámetros</i>	IDChange
<i>Funcionamiento</i>	<ol style="list-style-type: none"> 1. Verifica la existencia del cambio y su estado. 2. Actualiza 'status' en 'Change' a 'approved'. 3. Añade un registro en 'ChangeExecution' con valores nulos para 'executionDate' y 'result'. 4. Calcula y actualiza el técnico con más 'ChangeExecution' sin resultado en 'RE_Year'. 5. Actualiza 'RE_Year' restando 1 a 'changesInApproval' y sumando 1 a 'GCABApprovedChanges' para el año actual. 6. Registra la operación en 'LogProcedure'. 7. Maneja excepciones.

vote

<i>Descripción</i>	Añade un registro en la tabla 'Approval' para registrar el voto.
<i>Parámetros</i>	approverID, changelID, vote, comments
<i>Funcionamiento</i>	<ol style="list-style-type: none">1. Verifica la disponibilidad, autorización y capacidad de voto del approverID.2. Inserta el registro en 'Approval'.3. Registra la operación en 'LogProcedure'.4. Maneja excepciones.

addComment

<i>Descripción</i>	Añade un registro en la tabla 'Approval' enfocado en los comentarios, sin registrar un voto.
<i>Parámetros</i>	approverID, changelID, comments
<i>Funcionamiento</i>	<ol style="list-style-type: none">1. Verifica si el approverID está disponible y habilitado para la aplicación.2. Inserta el registro en 'Approval' con la fecha actual y 'vote' en NULL.3. Registra la operación en 'LogProcedure'.4. Maneja excepciones.

executeChange

<i>Descripción</i>	Este procedimiento actualiza un registro en la tabla 'ChangeExecution' y, si es necesario, crea un nuevo registro en 'ActionPlan'. Dependiendo del resultado de la ejecución del cambio, realiza varias actualizaciones estadísticas.
<i>Parámetros</i>	IDExecution, executionDate, result
<i>Funcionamiento</i>	<ol style="list-style-type: none">1. Actualiza el registro de 'ChangeExecution' con la fecha y resultado proporcionados.2. Si el resultado es 'incidents' o 'rescheduled', crea un nuevo registro en 'ActionPlan' y aumenta 'incorrectExecution' en 'RE_Year' para el año actual.3. Cada vez que se crea un 'ActionPlan', aumenta en 1 'openActionPlans' en 'RE_Year' para el año actual.4. Si el resultado es 'correct', calcula el técnico con más cambios correctos ejecutados de cualquier aplicación.5. Si el resultado es 'rescheduled', calcula el porcentaje de cambios replanificados por zona geográfica.6. Independientemente del resultado, calcula el técnico con más cambios en curso y actualiza 'maxOpenChangesTechnicalManager' en 'RE_Year' para el año actual.7. Registra la operación en 'LogProcedure'.8. Maneja cualquier excepción que ocurra durante la ejecución.

addAction

<i>Descripción</i>	Este procedimiento se utiliza para añadir un nuevo registro en la tabla 'Action'.
<i>Parámetros</i>	description, deliveryDate, actionPlanID
<i>Funcionamiento</i>	<ol style="list-style-type: none">1. Inserta el nuevo registro en la tabla 'Action' con 'doneDate' en NULL.2. Registra la operación en 'LogProcedure'.3. Maneja cualquier excepción que ocurra durante la ejecución.

closeAction

<i>Descripción</i>	Este procedimiento actualiza la fecha de finalización (doneDate) de una acción específica en la tabla 'Action' y recalcula el porcentaje de acciones cerradas a tiempo en 'RE_Year'.
<i>Parámetros</i>	IDAction, doneDate
<i>Funcionamiento</i>	<ol style="list-style-type: none">1. Actualiza 'doneDate' de la acción especificada y recalcula el porcentaje de acciones completadas a tiempo.2. Registra la operación en 'LogProcedure'.3. Maneja cualquier excepción que ocurra durante la ejecución.

closeActionPlan

<i>Descripción</i>	Este procedimiento se utiliza para marcar como cerrado un 'ActionPlan' específico y generar un nuevo registro en 'ChangeExecution' con valores nulos para 'executionDate' y 'result'.
<i>Parámetros</i>	IDActionPlan, description
<i>Funcionamiento</i>	<ol style="list-style-type: none">1. Actualiza el estado de 'isClosed' a 1 en 'ActionPlan' y, opcionalmente, su descripción.2. Crea un nuevo registro en 'ChangeExecution' con 'executionDate' y 'result' como NULL.3. Calcula el técnico con más 'ChangeExecution' sin result y actualiza 'maxOpenChangesTechnicalManager' en 'RE_Year'.4. Actualiza 'RE_Year', restando 1 de 'openActionPlans'.5. Registra la operación en 'LogProcedure'.6. Maneja cualquier excepción que ocurra durante la ejecución.

addProcessAudit

<i>Descripción</i>	Este procedimiento se utiliza para añadir un nuevo registro en la tabla 'ProcessAudit' con la fecha actual.
<i>Parámetros</i>	observations

<i>Funcionamiento</i>	<ol style="list-style-type: none"> 1. Inserta el nuevo registro en la tabla 'ProcessAudit' con la fecha actual. 2. Registra la operación en 'LogProcedure'. 3. Maneja cualquier excepción que ocurra durante la ejecución.
-----------------------	---

addAuditComment

<i>Descripción</i>	Este procedimiento se utiliza para añadir un nuevo registro en la tabla 'AuditComment'.
<i>Parámetros</i>	nonCompliances, observations, auditID
<i>Funcionamiento</i>	<ol style="list-style-type: none"> 1. Inserta el nuevo registro en la tabla 'AuditComment'. 2. Registra la operación en 'LogProcedure'. 3. Maneja cualquier excepción que ocurra durante la ejecución.

updateLastSixMonthsAuditCount

<i>Descripción</i>	Este procedimiento cuenta los registros AuditComment generados en los últimos 6 meses y los inserta en la tabla RE_LastSixMonths.
<i>Parámetros</i>	
<i>Funcionamiento</i>	<ol style="list-style-type: none"> 1. Cuenta los registros AuditComment de los últimos 6 meses en función de la fecha de ejecución. 2. Inserta el conteo en la tabla RE_LastSixMonths. 3. Registra la operación en 'LogProcedure'. 4. Maneja cualquier excepción que ocurra durante la ejecución.

checkAndCreateCurrentYearREYear

<i>Descripción</i>	Este procedimiento verifica si existe un registro en la tabla 'RE_Year' para el año actual. Si no existe, crea un nuevo registro con el año actual y copia ciertos valores del año anterior.
<i>Parámetros</i>	
<i>Funcionamiento</i>	<ol style="list-style-type: none"> 1. Verifica la existencia de un registro para el año actual en la tabla 'RE_Year'. 2. Si no existe, crea un nuevo registro con valores del año anterior para campos específicos. 3. Registra la operación en 'LogProcedure'. 4. Maneja cualquier excepción que ocurra durante la ejecución.

3.3.7 Trabajos programados

Una vez comentados todos los procedimientos necesarios, se puede observar que existen dos de ellos sin parámetros que se planifica que sean ejecutados periódicamente por el sistema.

Uno de ellos se encargará de contar el número de incumplimientos en las auditorías en los últimos 6 meses. Por lo tanto, este dato del repositorio deberá ser actualizado diariamente para ser coherente y se creará un trabajo programado que se encargue de ello con nombre 'dailyAuditCountUpdate'.

El siguiente trabajo programado necesario se encargará de la actualización de la tabla 'RE_Year'. Como existen valores que se desea que sean históricos y otros valores anuales, en el momento de que comience un nuevo año se deberá ejecutar el procedimiento 'checkAndCreateCurrentYearREYear'.

Este procedimiento añadirá una fila con el nuevo año como clave primaria y copiará los valores históricos deseados en sus correspondientes columnas. De esta forma, se mantiene la funcionalidad requerida para este repositorio, pero se añade la posibilidad de consultar también estos datos de forma anual.

4. Implementación

Según lo planificado, se sigue con la implementación creando los ficheros necesarios para que la base de datos pueda ser replicada en las instancias necesarias. Mientras se realiza esta creación, también se añaden las sentencias necesarias a un fichero de eliminación de todo lo creado hasta ese momento con nombre "00_erase.sql".

4.1 Tablespaces

Además de facilitar la administración y la gestión de accesos al repositorio estadístico, estos dos contenedores contribuirán a la mejora del rendimiento del sistema, especialmente si se almacenan en diferentes dispositivos de almacenamiento. El detalle de creación de los *tablespaces* se encuentra en el fichero "01_tablespaces.sql".

4.2 Tablas

Además de crear cada tabla en su *tablespace* correspondiente, en el mismo paso, se generarán las claves primarias de cada tabla. Posteriormente y para evitar problemas en el orden de creación, se editan las tablas para añadir las claves foráneas. Puede consultarse y ejecutar el fichero de creación de tablas llamado "02_tables.sql".

En Oracle BD, las secuencias son objetos independientes y no se vinculan a una tabla o columna específica. En lugar de eso, se deben utilizar dentro de las sentencias SQL "Insert" para generar los valores únicos necesarios. Por tanto, deberán ser utilizadas en los procedimientos ABM que se crearán para manipulación de datos. Este proceso de creación de secuencias se puede encontrar en el fichero "03_sequences.sql".

4.3 Índices

En puntos anteriores se creó un *tablespace* especialmente dedicado a almacenar los índices creados, por lo que se indicará en cada sentencia. Como se ha comentado, se creará uno por clave foránea en cada tabla que disponga de al menos una. El detalle se puede consultar y ejecutar en el fichero "04_indexes.sql".

4.4 Disparadores

Como se comenta en el punto anterior, el uso de las secuencias creadas será mediante disparadores para que el ID de dichas tablas sea generado automáticamente. El fichero de creación de estos disparadores es "05_triggers.sql".

El nombre de cada disparador será "trg_%nombreTabla%_id y serán todos similares. Deberán garantizar que cada vez que se inserte una nueva fila en una tabla, la columna "ID%tabla%" de esa fila reciba automáticamente el siguiente valor disponible de la secuencia correspondiente a su tabla.

4.5 Procedimientos almacenados

El siguiente paso es la implementación de los procedimientos necesarios para la interacción con la base de datos según la descripción y funcionamiento ya planificado. El detalle de la creación se puede consultar en el fichero "06_procedures.sql".

Además del código listo para ejecutar de la creación de todos los procedimientos explicados, se encuentra cada uno de ellos ampliamente documentado en cuanto a descripción, parámetros a recibir junto con los valores y tipos esperados, además de su funcionamiento y comentarios en partes clave.

4.6 Trabajos programados

Con los procedimientos creados, podemos hacer que se ejecuten periódicamente programando un trabajo. Como se ha comentado, 'dailyAuditCountUpdate' se ejecutará cada día a medianoche mientras que 'CHECK_CREATE_YEARLY_RE_YEAR_JOB' solo se ejecutará cada día 1 de enero anualmente.

En el fichero "07_jobs.sql" puede realizarse la consulta o programación de estos trabajos. También se encuentra documentación explicativa.

5. Pruebas

En primer lugar, se desea destacar que, con fines de ahorro energético, pero también por razones de gestión de riesgos, se crea una segunda base de datos paralela en *Oracle Cloud Infrastructure (OCI)*⁹.

Esto permitirá que los trabajos programados a medianoche o en el inicio de año puedan ejecutarse sin mantener un equipo doméstico encendido y simulando un entorno de producción, sin perjuicio de que en el equipo físico se realicen las pruebas necesarias para los trabajos temporales manipulando la fecha del sistema.

Por otro lado, ya que, en este punto, la base de datos está recién creada y sin datos, se desea destacar la importancia de inicializar la tabla RE_Year del repositorio estadístico. Esta acción deberá ser realizada por la administración de la base de datos, aunque en nuestro caso está incluida en el fichero de pruebas llamado "08_addData.sql".

A partir de la inicialización de dicha tabla, se realizan varias simulaciones de los casos de uso principales haciendo uso de todos los procedimientos programados. También se incluyen ejecuciones que deben arrojar error para probar la correcta funcionalidad del procedimiento, así como su correcto almacenamiento en la tabla de *logs*.

Para la comprobación del correcto funcionamiento, además de pedir al SGBD que muestre la salida RSP por consola en el momento de la ejecución, se diseñan diversas consultas SELECT con relaciones habituales para observar los cambios en cada ejecución. Estas consultas pueden encontrarse en el fichero "09_queries.sql". También en este fichero se pueden encontrar las consultas necesarias para comprobar los resultados del repositorio estadístico definidas en tiempo constante 1.

5.1 Adición de aprobadores

Tras la ejecución de la primera sentencia para iniciar el repositorio estadístico, se ejecuta el primer bloque de adiciones que crea en primer lugar los roles de inicio. Posteriormente, se crean las diferentes áreas de la organización.

En tercer lugar, se añaden las personas que van a ejercer el rol de aprobador, cada una con su rol correspondiente. También se añaden aprobadores sin derecho a voto. Por último, se ejecuta el procedimiento para asignar sustitutos a los aprobadores.

⁹ [9] Descubre Oracle Cloud infrastructure. (14/12/2023). <https://www.oracle.com/es/cloud/>

Tras comprobar mediante la salida RSP que todas las adiciones han sido correctas, y que los procedimientos ejecutados con errores han sido correctamente registrados en la tabla de logs, se listan los aprobadores.

```
-- Listado de aprobadores
SELECT
  A.name AS ApproverName,
  AR.name AS RoleName,
  A.contact AS contact,
  AR2.name AS AreaName,
  A.isAvailable AS isAvailable,
  S.name AS SubstituteName
FROM Approver A
INNER JOIN ApproverRole AR ON A.approverRoleID = AR.IDRole
INNER JOIN Area AR2 ON A.areaID = AR2.IDArea
LEFT JOIN Approver S ON A.substituteID = S.IDApprover;
```

	APPROVERNAME	ROLENAME	CONTACT	AREANAME	ISAVAILABLE	SUBSTITUTENAME
1	John Doe	Technical Manager	john.doe@example.co	Development	1	Jane Brown
2	Jane Smith	Business area respons	jane.smith@example.c	Administration	1	Sarah Wilson
3	Alice Johnson	Change Manager	alice.johnson@examp	Administration	1	Michael Harris
4	Bob Brown	GCAB with vote	bob.brown@example.	Administration	1	Bob Martinez
5	Carlos Green	GCAB with vote	carlos.green@exampl	Development	1	Linda Martinez
6	Emma White	GCAB without vote	emma.white@exampl	Data	1	(nulo)
7	Michael Harris	Change Manager	michael.harris@exam	Administration	1	(nulo)
8	Sarah Wilson	Business area respons	sarah.wilson@exampl	Administration	1	(nulo)
9	Linda Martinez	GCAB with vote	linda.martinez@exam	Development	1	(nulo)
10	Jane Brown	Technical Manager	jane.brown@example.	Development	1	(nulo)
11	Bob Martinez	GCAB with vote	bob.martinez@examp	Administration	1	(nulo)
12	James Wilson	Technical Manager	james.wilson@exampl	Data	1	Charles Moore
13	Patricia Brown	Business area respons	patricia.brown@exam	Administration	1	Barbara Jackson
14	David Garcia	Change Manager	david.garcia@example	Data	1	William White
15	Susan Anderson	GCAB with vote	susan.anderson@exar	IoT	1	Charles Skies
16	Joseph Miller	GCAB with vote	joseph.miller@exampl	Development	1	Patricia Smith
17	Margaret Taylor	GCAB without vote	margaret.taylor@exar	Administration	1	(nulo)
18	Charles Moore	Technical Manager	charles.moore@exam	Data	1	(nulo)
19	Barbara Jackson	Business area respons	barbara.jackson@exar	Administration	1	(nulo)
20	William White	Change Manager	william.white@exampl	Data	1	(nulo)
21	Charles Skies	GCAB with vote	charles.skies@exampl	IoT	1	(nulo)
22	Patricia Smith	GCAB with vote	patricia.smith@examp	Development	1	(nulo)

II-lustració 8 - Listado de aprobadores

5.2 Adición de aplicaciones

En segundo lugar, se procede a añadir las aplicaciones. Sin embargo, antes se deben crear zonas geográficas para que puedan ser asignadas.

Se decide crear dos aplicaciones diferentes para el contexto de pruebas. Al ser creadas, ninguna de ellas estará todavía en producción.

```
-- Listado de aplicaciones
SELECT
  App.IDApplication,
  App.name AS ApplicationName,
  App.isInProduction,
  App.productionDate,
  App.averageUsersNumber,
  App.technology,
  App.infraestructureType,
  App.criticality,
  GZ.name AS GeographicZoneName,
  TC.name AS TechnicalContactName,
  TC.contact AS TechnicalContactContact,
  BC.name AS BusinessContactName,
  BC.contact AS BusinessContactContact
FROM Application App
INNER JOIN GeographicZone GZ ON App.geographicZoneID = GZ.IDZone
INNER JOIN Approver TC ON App.technicalContact = TC.IDApprover
INNER JOIN Approver BC ON App.businessContact = BC.IDApprover;
```

IDAPPI	APPLICATIONNAME	ISINPRODUCTION	PRODUCTIONDATE	AVERAGEUSERSNUM	TECHNOLOGY	INFRAESTRU	CRITICA	GEOGRAPHICZ	TECHNICALCOI	TECHNICALCONTAC	BUSINESSCONTACTI	BUSINESSCONTACTI
1	1 Finance App - Molex	0	(nulo)	500	Java	Cloud	Alta	North America	John Doe	john.doe@example.co	Jane Smith	jane.smith@example.c
2	2 Data Scramble	0	(nulo)	(nulo)	Python	On-Premise	Media	Europe	James Wilson	james.wilson@exampl	Patricia Brown	patricia.brown@exam

II-lustració 9 - Listado de aplicaciones

Además, comprobamos que se han añadido las zonas geográficas creadas como nuevas filas en la tabla “RE_Geographic”.

GEOGRAPHICZONE	
1	North America
2	Europe
3	Asia Pacific

5.3 Asignación de aprobadores a aplicaciones

Con la base de datos alimentada con aprobadores y aplicaciones, es el momento de asignar qué personas son los aprobadores de qué aplicación. En este bloque SQL se asignan, además de ejecutar el procedimiento para indicar que un aprobador no está disponible.

```
-- Listado de aprobadores de una aplicación
SELECT
  A.name AS ApproverName,
```

```

AR.name AS RoleName,
A.contact,
AR2.name AS AreaName,
A.isAvailable,
S.name AS SubstituteName
FROM ApplicationApprovers AA
INNER JOIN Approver A ON AA.approverID = A.IDApprover
INNER JOIN ApproverRole AR ON A.approverRoleID = AR.IDRole
INNER JOIN Area AR2 ON A.areaID = AR2.IDArea
LEFT JOIN Approver S ON A.substituteID = S.IDApprover
WHERE AA.applicationID = 1;

```

	APPROVERNAME	ROLENAME	CONTACT	AREANAME	ISAVAILABLE	SUBSTITUTENAME
1	Alice Johnson	Change Manager	alice.johnson@examp	Administration	1	Michael Harris
2	Jane Smith	Business area respons	jane.smith@example.c	Administration	1	Sarah Wilson
3	Carlos Green	GCAB with vote	carlos.green@exampl	Development	1	Linda Martinez
4	John Doe	Technical Manager	john.doe@example.co	Development	1	Jane Brown
5	Bob Brown	GCAB with vote	bob.brown@example.	Administration	0	Bob Martinez
6	Emma White	GCAB without vote	emma.white@exampli	Data	1	(nulo)
7	Bob Martinez	GCAB with vote	bob.martinez@examp	Administration	1	(nulo)

Il-lustració 10 - Listado de aprobadores de aplicación 1

Como se puede observar, el aprobador “Bob Martinez” aparece en este listado ya que el titular, “Bob Brown”, no está disponible.

```

-- Listado de aprobadores de una aplicación
SELECT
A.name AS ApproverName,
AR.name AS RoleName,
A.contact,
AR2.name AS AreaName,
A.isAvailable,
S.name AS SubstituteName
FROM ApplicationApprovers AA
INNER JOIN Approver A ON AA.approverID = A.IDApprover
INNER JOIN ApproverRole AR ON A.approverRoleID = AR.IDRole
INNER JOIN Area AR2 ON A.areaID = AR2.IDArea
LEFT JOIN Approver S ON A.substituteID = S.IDApprover
WHERE AA.applicationID = 2

```

	APPROVERNAME	ROLENAME	CONTACT	AREANAME	ISAVAILABLE	SUBSTITUTENAME
1	James Wilson	Technical Manager	james.wilson@exampl	Data	1	Charles Moore
2	Patricia Brown	Business area respons	patricia.brown@exam	Administration	1	Barbara Jackson
3	David Garcia	Change Manager	david.garcia@example	Data	0	William White
4	Susan Anderson	GCAB with vote	susan.anderson@exar	IoT	1	Charles Skies
5	Joseph Miller	GCAB with vote	joseph.miller@exampl	Development	1	Patricia Smith
6	Margaret Taylor	GCAB without vote	margaret.taylor@exar	Administration	1	(nulo)
7	William White	Change Manager	william.white@exampl	Data	1	(nulo)

II-Ilustració 11- Listado de aprobadores de aplicación 2

De igual forma que en la consulta anterior, se observa que “William White” tiene permiso para votar debido a que “David Garcia” no está disponible.

Por otro lado, como en el bloque ejecutado se ha establecido como no disponible en varias ocasiones a diferentes aprobadores, en ocasiones con indisponibilidad planificada y otras sin planificar, podemos comprobar que esta funcionalidad del repositorio estadístico funciona correctamente.

```
-- Aprobador más veces sustituido por indisponibilidad no planificada:
SELECT mostReplacedApprover
FROM RE_Year
WHERE yearData = EXTRACT(YEAR FROM SYSDATE);
```

	MOSTREPLACEDAPPROVER
1	David Garcia-david.garcia@example.com

5.4 Adición de cambios

Para añadir un cambio, al ser la primera vez que se utiliza la base de datos, es necesario añadir antes categorías, ya que el otro requisito está cumplido, que es la existencia de aplicaciones.

Una vez hecho esto, lanzamos las aplicaciones a producción y creamos varios cambios para seguir probando todas las funcionalidades de la base de datos.

```
-- Listado de cambios
SELECT
  C.IDChange,
  C.description AS ChangeDescription,
  C.requestDate,
  C.status,
  CC.description AS CategoryDescription,
  CC.importance AS CategoryImportance,
  A.IDApplication,
  A.name AS ApplicationName,
  A.criticality AS ApplicationCriticality,
```

```

A.averageUsersNumber
FROM Change C
INNER JOIN ChangeCategory CC ON C.changeCategoryID = CC.IDCategory
INNER JOIN Application A ON C.applicationID = A.IDApplication;

```

IDCHANGE	CHANGEDESCRIPC	REQUESTDATE	STATUS	CATEGORYDESCRIP1	CATEGORYIMPORTA	IDAPPLICATION	APPLICATIONNAME	APPLICATIONCRITIC	AVERAGEUSERSNUM
1	Security patch under I	28/12/2023, 17:28:4€	in_process	Critical Change	10	1	Finance App - Molex	Alta	500
2	Minor authentication	28/12/2023, 17:28:4€	in_process	Minor Change	30	2	Data Scramble	Media	(null)
3	Login update for Goo	28/12/2023, 17:28:4€	in_process	Standard Change	20	1	Finance App - Molex	Alta	500

Il-lustració 12 - Listado de cambios punto 5.4

Como se puede observar, por ahora todos los cambios están en proceso de aprobación. Por lo tanto, se aprovecha para realizar la consulta correspondiente al repositorio estadístico.

```

-- Número de cambios en proceso de aprobación actualmente
SELECT changesInApproval
FROM RE_Year
WHERE yearData = EXTRACT(YEAR FROM SYSDATE);

```

CHANGESINAPPROV	
1	3

Il-lustració 13 - Cambios en proceso de aprobación punto 5.4

5.5 Aprobación de cambio

La primera funcionalidad clave a probar es la aprobación de un cambio, para lo que se escoge el cambio con ID 1.

En primer lugar, se procede a los votos de los aprobadores. En este caso, se prueba a que el aprobador no disponible vote, lo que resulta en un error. En su lugar, es su sustituto quien puede votar. También se inserta un comentario por parte de un aprobador con rol sin derecho a voto.

Posteriormente, el cambio es aprobado, testeando que no pueda aprobarse dos veces un cambio. También se comprueba que se ha generado una ejecución sin resultado, avisando de que está pendiente.

```

-- Listado de votos y comentario de un cambio
SELECT
  A.approverID,
  AP.name AS ApproverName,
  AP.contact AS ApproverContact,
  A.vote,
  A.comments
FROM Approval A
INNER JOIN Approver AP ON A.approverID = AP.IDApprover

```

```
WHERE A.changeID = :ChangeID; -- Insertar ID del cambio
```

	APPROVERID	APPROVERNAME	APPROVERCONTACT	VOTE	COMMENTS
1	1	John Doe	john.doe@example.co	1	Aprobado para produ
2	2	Jane Smith	jane.smith@example.c	1	Entiendo la urgencia
3	11	Bob Martinez	bob.martinez@examp	1	Aprobado con observ
4	5	Carlos Green	carlos.green@exampl	1	(nulo)
5	3	Alice Johnson	alice.johnson@examp	1	Aprobado con observ
6	6	Emma White	emma.white@exampl	(nulo)	Empeorará la respues

Il·lustració 14 - Listado de votos de cambio 1

```
-- Listado de ejecuciones
SELECT
  CE.IDExecution,
  CE.executionDate,
  CE.result,
  C.IDChange,
  C.description AS ChangeDescription,
  C.status AS ChangeStatus,
  A.name AS ApplicationName
FROM ChangeExecution CE
INNER JOIN Change C ON CE.changeID = C.IDChange
INNER JOIN Application A ON C.applicationID = A.IDApplication;
```

	IDEXECUTION	EXECUTIONDATE	RESULT	IDCHANGE	CHANGEDESCRPTION	CHANGESTATUS	APPLICATIONNAME
1	1	(nulo)	(nulo)	1	Security patch under l	approved	Finance App - Molex

Il·lustració 15 - Listado de ejecución pendiente de cambio 1

Seguimos con la comprobación del funcionamiento del repositorio estadístico comprobando los cambios en curso, los cambios aprobados en el año en curso y el técnico con más cambios en curso (aunque solo exista un cambio en curso todavía).

CHANGESINAPPROVAL	
1	2

Il·lustració 16 - Cambios en proceso de aprobación punto 5.5

```
-- Número total de cambios aprobados por la GCAB en el año en curso:
SELECT GCABApprovedChanges
FROM RE_Year
WHERE yearData = EXTRACT(YEAR FROM SYSDATE);
```


GCABAPPROVEDCHANGES	
1	1

Il-lustració 17 - Cambios aprobados en el año en curso punto 5.5

```
-- Responsable técnico con más cambios en curso actualmente:
SELECT maxOpenChangesTechnicalManager
FROM RE_Year
WHERE yearData = EXTRACT(YEAR FROM SYSDATE);
```

MAXOPENCHANGESTECHNICALMAI	
1	John Doe-john.doe@example.com

Il-lustració 18 - Técnico con más cambios en curso punto 5.5

5.6 Rechazo de cambio

En este punto nos centraremos en el cambio con ID 2. En este caso, existirán votos en contra y será rechazado. Además de las pruebas habituales del procedimiento tratando de insertar votos inválidos, también se trata de rechazar dos veces el cambio y rechazar un cambio que no existe.

Tras las inserciones, comprobamos los votos del cambio y volvemos a listar los cambios existentes para observar que el cambio con ID 2 ha sido rechazado.

```
-- Listado de votos y comentario de un cambio
SELECT
  A.approverID,
  AP.name AS ApproverName,
  AP.contact AS ApproverContact,
  A.vote,
  A.comments
FROM Approval A
INNER JOIN Approver AP ON A.approverID = AP.IDApprover
WHERE A.changeID = 2;
```

	APPROVERID	APPROVERNAME	APPROVERCONTACT	VOTE	COMMENTS
1	15	James Wilson	james.wilson@exampl	1	Aprobado para producción
2	16	Patricia Brown	patricia.brown@exam	1	Es una necesidad transmitida por el cliente
3	23	William White	william.white@exampl	0	Rechazado por falta de documentación
4	20	Margaret Taylor	margaret.taylor@exar	(nulo)	No tenemos documentación que analizar

Il-lustració 19 - Listado de votos cambio ID 2

IDCHANGE	CHANGEDESCRIPC	REQUESTDATE	STATUS	CATEGORYDESCRIP1	CATEGORYIMPORTA	IDAPPLICATION	APPLICATIONNAME	APPLICATIONCRITIC	AVERAGEUSERSNUM
1	1 Security patch under	28/12/2023, 17:28:46	approved	Critical Change	10	1	Finance App - Molex	Alta	500
2	2 Minor authentication	28/12/2023, 17:28:46	rejected	Minor Change	30	2	Data Scramble	Media	(nulo)
3	3 Login update for Goo	28/12/2023, 17:28:46	in_process	Standard Change	20	1	Finance App - Molex	Alta	500

Il-lustració 20 - Listado de cambios punto 5.6

También seguimos con las comprobaciones del repositorio estadístico listando los cambios en proceso de aprobación (que ahora es uno) y los cambios rechazados en el año actual.

CHANGESINAPPROV	
1	1

Il-lustració 21 - Cambios en proceso de aprobación punto 5.6

```
-- Cambios no aprobados y no ejecutados en un año específico:
SELECT notApprovedChanges
FROM RE_Year
WHERE yearData = :yearData; -- Insertar año
```

NOTAPPROVEDCHA	
1	1

Il-lustració 22 - Cambios no aprobados punto 5.6

5.7 Ejecución de cambio

En este bloque de código se procederá a ejecutar el cambio con ID 1 que fue aprobado. Esta ejecución será correcta y también se probará a tratar de insertar un estado no válido para observar el error. Una vez hecho, podemos volver a listar las ejecuciones como en el punto 5.5

IDEXECUTION	EXECUTIONDATE	RESULT	IDCHANGE	CHANGEDESCRIPC	CHANGESTATUS	APPLICATIONNAME
1	1 9/12/2023, 0:00:00	correct	1	Security patch under	approved	Finance App - Molex

Il-lustració 23 - Listado de ejecuciones punto 5.7

En este momento, ya existirán datos en el repositorio estadístico acerca del responsable técnico con más cambios de forma correcta.

```
-- Responsable técnico con más cambios ejecutados correctamente:
SELECT maxCorrectChangesTechnicalManager
FROM RE_Year
WHERE yearData = EXTRACT(YEAR FROM SYSDATE);
```

	MAXCORRECTCHANGESTECHNICALMANA
1	John Doe john.doe@example.com

II-luistració 24 - Técnico con más cambios ejecutados de forma correcta punto 5.7

5.8 Cambio replanificado para modificación de App

En este bloque de código de crea otro cambio con el objetivo de modificar la aplicación con ID 2, que además conllevará la necesidad de modificar el campo “infraestructura” de la aplicación.

En adición, la ejecución de este cambio será replanificada y, por lo tanto, creará un plan de acción con una acción que será ejecutada y cerrada. Al cerrar a su vez el plan de acción, se creará una nueva ejecución pendiente.

Por último, la aplicación será modificada y después se guardará el registro de la ejecución correcta.

A continuación, podemos ver capturas del nuevo cambio, de los votos que lo han aprobado y del plan de acción creado al reprogramar la ejecución.

5	Transfer to AWS clouc	29/12/2023, 11:18:21	approved	Standard Change	20	2	Data Scramble	Media	(nulo)
---	-----------------------	----------------------	----------	-----------------	----	---	---------------	-------	--------

II-luistració 25 - Cambio con ID 5

	APPROVERID	APPROVERNAME	APPROVERCONTAC	VOTE	COMMENTS
1	15	James Wilson	james.wilson@exampl	1	(nulo)
2	16	Patricia Brown	patricia.brown@exam	1	(nulo)
3	18	Susan Anderson	susan.anderson@exar	1	(nulo)
4	19	Joseph Miller	joseph.miller@exampl	1	(nulo)
5	23	William White	william.white@exampl	1	Aprobado en sustituc

II-luistració 26 - Votos de cambio con ID 5

```
-- Listado de ActionPlan
SELECT
  AP.IDActionPlan,
  AP.description AS ActionPlanDescription,
  AP.isClosed,
  C.IDChange,
  C.description AS ChangeDescription,
  CE.result AS ChangeExecutionResult
FROM ActionPlan AP
INNER JOIN ChangeExecution CE ON AP.changeExecutionID = CE.IDExecution
INNER JOIN Change C ON CE.changeID = C.IDChange;
```

	IDACTIONPLAN	ACTIONPLANDESCR	ISCLOSED	IDCHANGE	CHANGEDESCRPTIC	CHANGEEXECUTION
1	1	Reprogramado		1	5 Transfer to AWS cloud	rescheduled

Il-lustració 27 - Plan de acción creado al reprogramar ejecución

Por otro lado, la acción creada dentro del plan anterior, que ya ha sido ejecutada (pero un día más tarde de lo planificado), y las ejecuciones correspondientes a este cambio. Una replanificada y, posteriormente al plan de acción cerrado, una correcta.

```
-- Listado de Actions de un ActionPlan
SELECT
  IDAction,
  description,
  deliveryDate,
  doneDate
FROM Action
WHERE actionPlanID = 1;
```

	IDACTION	DESCRIPTION	DELIVERYDATE	DONEDATE
1	1	Subir el 2023-12-11 ya que se encuentra actualmente terminando un proceso	11/12/2023, 0:00:00	12/12/2023, 0:00:00

Il-lustració 28 - Acción del plan de acción con ID 1

	IDEXECUTION	EXECUTIONDATE	RESULT	IDCHANGE	CHANGEDESCRPTIC	CHANGESTATUS	APPLICATIONNAME
1	2	10/12/2023, 0:00:00	rescheduled	5	Transfer to AWS cloud	approved	Data Scramble
2	3	12/12/2023, 0:00:00	correct	5	Transfer to AWS cloud	approved	Data Scramble

Il-lustració 29 - Listado de ejecuciones del cambio con ID 5

Tras estas acciones, podemos observar un nuevo dato en el repositorio estadístico, que es el porcentaje de cambios replanificados por zona geográfica. Seleccionamos “Europe” en la consulta ya que la App 2 aplica a dicha zona.

```
-- Porcentaje de cambios replanificados en una región geográfica
específica:
SELECT porcentajeRescheduledChanges
FROM RE_Geographic
WHERE geographicZone = 'Europe';
```

	PERCENTAJERESCHEDULEDCHANGES
1	25

Il-lustració 30 - Porcentaje de cambios replanificados en zona Europa

5.9 Nuevos cambios con diferentes situaciones

Para seguir realizando pruebas con todas las posibilidades de ejecución y, además alimentar de forma suficiente la base de datos para demostrar la funcionalidad del repositorio estadístico, vamos a añadir nuevos cambios.

En primer lugar, se añade un nuevo cambio con sus correspondientes votos positivos que, por lo tanto, es aprobado. Esta ejecución resultará en “incidents”, que es la posibilidad no probada hasta este momento.

Posteriormente, se añaden varias acciones al plan de acción generado por este resultado, de las cuales solo cuatro se completan (unas a tiempo y otras no).

	APPROVERID	APPROVERNAME	APPROVERCONTACT	VOTE	COMMENTS
1	1	John Doe	john.doe@example.co	1	Se espera un significativo incremento del rendimiento
2	2	Jane Smith	jane.smith@example.c	1	Me parece correcto
3	3	Alice Johnson	alice.johnson@examp	1	Aprobado, asegurarse de monitorizar el rendimiento
4	5	Carlos Green	carlos.green@exampl	1	(nulo)
5	6	Emma White	emma.white@exampl	(nulo)	Necesarias mayores comprobaciones en pre-producción
6	11	Bob Martinez	bob.martinez@examp	1	(nulo)

Il-lustració 31 - Votos de cambio con ID 6

8 Actualización de edito 29/12/2023, 11:50:2€ approved Standard Change 20 1 Finance App - Molex Alta 500

Il-lustració 32 - Cambio con ID 6

	IDEXECUTION	EXECUTIONDATE	RESULT	IDCHANGE	CHANGEDESCRIPTIC	CHANGESTATUS	APPLICATIONNAME
1	4	10/12/2023, 0:00:00	incidents	6	Major performance er	approved	Finance App - Molex

Il-lustració 33 - Ejecución con incidentes de cambio con ID 6

	IDACTIONPLAN	ACTIONPLANDESCR	ISCLOSED	IDCHANGE	CHANGEDESCRIPTIC	CHANGEEXECUTION
1	2	Major performance er	0	6	Major performance er	incidents

Il-lustració 34 - ActionPlan generado tras la ejecución con incidentes

	IDACTION	DESCRIPTION	DELIVERYDATE	DONEDATE
1	2	Comprobar librería	12/12/2023, 0:00:00	12/12/2023, 0:00:00
2	3	Comprobar funciones	13/12/2023, 0:00:00	13/12/2023, 0:00:00
3	4	Comprobar retrocompatibilidad	15/12/2023, 0:00:00	16/12/2023, 0:00:00
4	5	Monitorizar 24h	16/12/2023, 0:00:00	17/12/2023, 0:00:00
5	6	Adaptar servidor	20/12/2023, 0:00:00	(nulo)
6	7	Monitorizar en nuevo entorno	25/12/2023, 0:00:00	(nulo)

Il-lustració 35 - Actions vinculadas al ActionPlan con ID 2

Como en este momento ya tenemos acciones cerradas a tiempo, se puede mostrar el dato correspondiente en el repositorio estadístico, además de los planes de acción abiertos actualmente.

```
-- Porcentaje de acciones cerradas a tiempo en un año específico:
SELECT doneActionsInTimePorcentaje
FROM RE_Year
WHERE yearData = 2023;
```

	DONEACTIONSINTIM
1	28.57

II-lustració 36 - Porcentaje de Actions cerradas a tiempo punto 5.9

```
-- Número total de planes de acción abiertos actualmente:
SELECT openActionPlans
FROM RE_Year
WHERE yearData = EXTRACT(YEAR FROM SYSDATE);
```

	OPENACTIONPLANS
1	1

II-lustració 37 - Número de ActionPlan abiertos punto 5.9

En segundo lugar, se añaden otros dos cambios que, como son votados de forma positiva son aprobados, pero quedan pendientes de ejecución.

	IDEXECUTION	EXECUTIONDATE	RESULT	IDCHANGE	CHANGEDESCRPTIC	CHANGESTATUS	APPLICATIONNAME
1	6	(nulo)	(nulo)	8	Actualización de edito	approved	Finance App - Molex
2	5	(nulo)	(nulo)	7	Actualización de GUI	approved	Finance App - Molex

II-lustració 38 - Cambios pendientes de ejecución punto 5.9

5.10 Baja de aplicación

Seguidamente, se prueba la baja de una aplicación, que tiene su propio procedimiento porque se trata de una modificación importante. Como se ha mencionado en puntos anteriores, se trata de un eliminado lógico, marcando el campo "isInProduction" con un 0.

Para seguir con la coherencia del sistema, se programa un cambio para ello, que será aprobado y ejecutado correctamente como se puede comprobar con el código correspondiente a este punto.

	IDAPPLICATION	APPLICATIONNAME	ISINPRODUCTION
1	2	Data Scramble	0

II-lustració 39 - Detalle de la baja de la aplicación con ID 2

5.11 Eliminada de categoría de cambio

Como también se ha contemplado que una categoría de cambio pueda eliminarse, se procede a probar el procedimiento correspondiente. Como en el caso anterior, se trata de un eliminado lógico, por lo que la fila no desaparecerá

de la base de datos. Simplemente se marcará una fecha en el campo correspondiente al eliminado para mantener esa información.

```
-- Listado de categorías de cambio
SELECT * FROM ChangeCategory;
```

	IDCATEGORY	DESCRIPTION	IMPORTANCE	CREATIONDATE	CREATIONAPPROVA	DELETEAPPROVALDATE
1	1	Critical Change	10	28/12/2023, 17:28:4€	1/11/2023, 0:00:00	(nulo)
2	2	Minor Change	30	28/12/2023, 17:28:4€	3/11/2023, 0:00:00	(nulo)
3	3	Standard Change	20	28/12/2023, 17:28:4€	3/11/2023, 0:00:00	(nulo)
4	4	Test Change	40	28/12/2023, 17:28:4€	7/11/2023, 0:00:00	29/12/2023, 16:20:34

II-lustració 40 - Eliminado de categoría de cambio

5.12 Auditorías

Por último, pero muy importante porque forma parte directa de los requisitos del proyecto, se prueba la gestión de las auditorías.

Antes de añadir datos sobre auditorías, se comprueba que ya existen datos sobre incumplimientos en los últimos seis meses en el repositorio estadístico, aunque la cuenta debe ser 0.

```
-- Número total de incumplimientos detectados en los últimos 6 meses:
SELECT numNonCompliances
FROM RE_LastSixMonths
WHERE lastDate = (SELECT MAX(lastDate) FROM RE_LastSixMonths);
```

	NUMNONCOMPLIANCES
1	0

II-lustració 41 - Número de incumplimientos en los últimos 6 meses antes de añadir datos

Posteriormente a esta comprobación, se añadirán los datos de una auditoría en el día actual (29/12/2023) con tres incumplimientos.

```
-- Listado de incumplimientos de una auditoría
SELECT
  IDComment,
  nonCompliances,
  observations
FROM AuditComment
WHERE auditID = 1;
```

	IDCOMMENT	NONCOMPLIANCES	OBSERVATIONS
1	4	Incumplimiento de normas de seguridad	Se requiere revisión inmediata de protocolos.
2	5	Falta de documentación adecuada	Es necesario actualizar la documentación de procesos.
3	6	Retrasos en la entrega de informes	Se observaron demoras significativas en la entrega de informes.

II-lustració 42 - Incumplimientos detectados en auditoría

Para comprobar que el repositorio estadístico funciona correctamente en cuanto al conteo de los incumplimientos, es necesario esperar hasta el día siguiente, ya que se recuerda que se calcula mediante un trabajo programado en el cambio de día.

	NUMNONCOMPLIANCES
1	3

II-lustració 43 - Número de incumplimientos en los últimos 6 meses el día siguiente de añadir datos

Con la consulta al repositorio presentada en este mismo punto, comprobamos como el dato se ha actualizado correctamente. Posteriormente se realiza una nueva adición de auditoría, esta vez sin incumplimientos.

```
-- Listado de auditorías y número de incumplimientos
SELECT
  PA.IDAudit,
  PA.auditDate,
  PA.observations,
  COUNT(AC.IDComment) AS NumberOfNonCompliances
FROM ProcessAudit PA
LEFT JOIN AuditComment AC ON PA.IDAudit = AC.auditID
GROUP BY PA.IDAudit, PA.auditDate, PA.observations;
```

	IDAUDIT	AUDITDATE	OBSERVATIONS	NUMBEROFNONCOMPLIANCES
1	1	29/12/2023, 16:32:05	Auditoría completa con varias inconformidades detect	3
2	2	30/12/2023, 10:48:55	Auditoría completa sin inconformidades.	0

II-lustració 44 - Listado de auditorías y número de incumplimientos

5.13 Comprobación de la tabla de logs y del repositorio estadístico

Por último, y ya con todos los datos de prueba añadidos y todos los procedimientos ejecutados para comprobar su funcionalidad, es comprobar la tabla de logs de acciones, para lo que podemos utilizar la siguiente consulta.

```
-- Ver Logs
SELECT * FROM LogProcedure;
```


	IDLOG	LOGDATETIME	LOGUSER	PROCEDURENAME	PROCEDUREPARAMS	RSP
1	191	2023-12-30T10:48:51	TESTER	addProcessAudit	AuditDate: 2023-12-30 - Observations: Auditori	OK
2	190	2023-12-30T00:00:00	TESTER	updateLastSixMonths	Execution Date: 2023-12-30	OK
3	189	2023-12-29T16:32:01	TESTER	addAuditComment	NonCompliances: Retrasos en la entrega de infor	OK
4	188	2023-12-29T16:32:01	TESTER	addAuditComment	NonCompliances: Falta de documentación adecu	OK
5	187	2023-12-29T16:32:01	TESTER	addAuditComment	NonCompliances: Incumplimiento de normas de	OK
6	186	2023-12-29T16:32:01	TESTER	addProcessAudit	AuditDate: 2023-12-29 - Observations: Auditori	OK
7	185	2023-12-29T16:31:51	TESTER	addAuditComment	NonCompliances: Retrasos en la entrega de infor	ERROR - -2291 - ORA-02291: restricción d
8	184	2023-12-29T16:31:51	TESTER	addAuditComment	NonCompliances: Falta de documentación adecu	ERROR - -2291 - ORA-02291: restricción d
9	183	2023-12-29T16:31:51	TESTER	addAuditComment	NonCompliances: Incumplimiento de normas de	ERROR - -2291 - ORA-02291: restricción d
10	182	2023-12-29T16:20:31	TESTER	deleteChangeCategor	CategoryID: 4	OK
11	181	2023-12-29T16:15:41	TESTER	executeChange	Execution ID: 7 - Execution Date: 16-DIC-23 - Re	OK
12	180	2023-12-29T16:15:41	TESTER	endApplicationLifeCyc	ApplicationID: 2	OK

II-lustració 45 - Listado de logs

Por supuesto, también se pueden consultar solo los errores de ejecución.

```
-- Ver errores
SELECT *
FROM LogProcedure
WHERE RSP LIKE 'ERROR%'
ORDER BY IDLog DESC;
```

	IDLOG	LOGDATETIME	LOGUSER	PROCEDURENAME	PROCEDUREPARAMS	RSP
1	185	2023-12-29T16:31:51	TESTER	addAuditComment	NonCompliances: Retrasos en la entrega de inform	ERROR - -2291 - ORA-02291: restricción de integridad
2	184	2023-12-29T16:31:51	TESTER	addAuditComment	NonCompliances: Falta de documentación adecuac	ERROR - -2291 - ORA-02291: restricción de integridad
3	183	2023-12-29T16:31:51	TESTER	addAuditComment	NonCompliances: Incumplimiento de normas de se	ERROR - -2291 - ORA-02291: restricción de integridad
4	126	2023-12-29T11:03:41	TESTER	executeChange	Execution ID: 1 - Execution Date: 09-DIC-23 - Resu	ERROR - -2290 - ORA-02290: restricción de control (T
5	124	2023-12-29T10:26:31	TESTER	rejectChange	Change ID: 1000	ERROR - Change not found or not in process
6	123	2023-12-29T10:26:31	TESTER	rejectChange	Change ID: 2	ERROR - Change not found or not in process
7	121	2023-12-29T10:26:31	TESTER	addComment	ApproverID: 20 - ChangeID: 1 - Comments: No te	ERROR - Approver not authorized or not available
8	115	2023-12-28T17:46:41	TESTER	approveChange	Change ID: 1000	ERROR - 100 - ORA-01403: No se ha encontrado ning
9	114	2023-12-28T17:46:41	TESTER	approveChange	Change ID: 1	ERROR - Change not found or not in process
10	111	2023-12-28T17:46:41	TESTER	vote	ApproverID: 1 - ChangeID: 10000 - Vote: 1 - Com	ERROR - Approver not authorized or has no voting rig
11	110	2023-12-28T17:46:41	TESTER	vote	ApproverID: 7 - ChangeID: 1 - Vote: 1 - Comment	ERROR - Approver not authorized or has no voting rig
12	109	2023-12-28T17:46:41	TESTER	vote	ApproverID: 6 - ChangeID: 1 - Vote: 1 - Comment	ERROR - Approver not authorized or has no votina rig

II-lustració 46 - Listado de errores en tabla de logs

Si deseamos ver las ejecuciones de cierto procedimiento, se puede utilizar la siguiente consulta cambiando el nombre del procedimiento por el deseado.

```
-- Ver ejecuciones de procedimiento
SELECT *
FROM LogProcedure
WHERE PROCEDURENAME = 'addComment'
ORDER BY IDLog DESC;
```

	IDLOG	LOGDATETIME	LOGUSER	PROCEDURENAME	PROCEDUREPARAM	RSP
1	146	2023-12-29T11:50:21	TESTER	addComment	ApproverID: 6 - Chan	OK
2	121	2023-12-29T10:26:31	TESTER	addComment	ApproverID: 20 - Cha	ERROR - Approver no
3	120	2023-12-29T10:26:31	TESTER	addComment	ApproverID: 20 - Cha	OK
4	112	2023-12-28T17:46:41	TESTER	addComment	ApproverID: 6 - Chan	OK

II-lustració 47 - Listado de ejecuciones del procedimiento "addComment"

Aunque ya haya sido probado en el entorno local en el que se ha desarrollado todo el proyecto, aprovechando el cambio de año, se comprueba su funcionamiento en un entorno de producción como OCI, resultando en una ejecución correcta, como se puede observar en la tabla de logs.

	IDLOG	LOGDATETIME	LOGUSER	PROCEDURENAME	PROCEDUREPARAM	RSP
1	194	2024-01-01T00:00:00	TESTER	updateLastSixMonthsAuditCount	Execution Date: 2024	OK
2	193	2024-01-01T00:00:00	TESTER	checkAndCreateCurrentYearREYear	Year: 2024	OK

II-lustració 48 - Log de ejecución de trabajos programados

Por otro lado, aunque hayamos mostrado a lo largo del capítulo todas las consultas específicas del repositorio estadístico, que se pueden encontrar en el fichero de consultas, a continuación, se muestra el estado final de todas las tablas.

	GEOGRAPHICZONE	PERCENTAJERESCHE
1	North America	0
2	Europe	25
3	Asia Pacific	0

II-lustració 49 - RE_Geographic

	LASTDATE	NUMNONCOMPLIA
1	1/1/2024, 0:00:00	3
2	31/12/2023, 0:00:00	3
3	30/12/2023, 0:00:00	3
4	29/12/2023, 0:00:00	0

II-lustració 50 - RE_LastSixMonths

	YEARDATA	CHANGESINAPPROVAL	INCORRECTEXECUTION	MAXCORRECTCHAN	DONEACTIONSINTI	GCABAPPROVEDCH	NOTAPPROVEDCHA	OPENACTIONPLAN	MAXOPENCHANGE	MOSTREPLACEDAPF
1	2023	1	2	James Wilson James.	28,57	6	1	1	John Doe-john.doe@ David Garcia-david.ga	
2	2024	1	0	James Wilson James.	0	0	0	1	John Doe-john.doe@ David Garcia-david.ga	

II-lustració 51 - RE_Year

Como se explicó en la fase de diseño, con la tabla RE_Year se pretende cumplir los requisitos establecidos para todos los puntos excepto el de porcentaje de cambios de cada región geográfica que su ejecución se planificó y los incumplimientos en los últimos 6 meses, que poseen tabla propia.

Al añadir una fila con el nuevo año y copiar los campos del año anterior para los resultados que se quiere un dato basado en toda la información de la base de datos. Sin embargo, también siempre será posible consultar los históricos de cada año de estos resultados, lo que se traduce en una mejora.

6. Conclusiones

En este proyecto, se ha valorado la importancia crucial de la metodología ITIL para gestionar cambios eficientemente en aplicaciones informáticas. El desarrollo de una base de datos robusta y adaptable ha sido fundamental para abordar esta necesidad. A través de este proceso, se han identificado y superado desafíos técnicos y de diseño, lo que ha enriquecido la comprensión de cómo manejar grandes volúmenes de datos y realizar una administración eficiente.

En cuanto a la gestión, este trabajo también ha demostrado la relevancia de una planificación detallada, pero con adaptabilidad frente a los imprevistos, resaltando la importancia de la flexibilidad en la gestión de proyectos. Por ejemplo, la creación de los procedimientos almacenados necesarios se subestimó en la planificación debido a una incorrecta evaluación sobre el contenido de la tarea, pero gracias a la previsión de cierta flexibilidad, no ha tenido impacto en el proyecto final.

Por otro lado, una parte importante de un proyecto es la temporalización. Como todo proyecto tiene una fecha límite de entrega, es esencial organizar y administrar el tiempo eficientemente para asegurar su entrega puntual y completa. Personalmente, no se ha tenido problema en este tipo de dinámica, gracias a experiencias previas en la entrega de proyectos, en los que se espera un avance significativo entre ellas, de forma parecida a este.

Al reflexionar sobre los objetivos iniciales, el proyecto alcanzó su objetivo principal de implementar un sistema de base de datos para la gestión de cambios en aplicaciones, siguiendo los procesos ITIL.

En cuanto a los objetivos específicos, como la aplicación práctica de conocimientos sobre bases de datos, el diseño de la estructura y la implementación de procedimientos de base de datos, también se han logrado con éxito.

Sin embargo, a pesar de que se han implementado algunas, se reconoce que la identificación de funcionalidades adicionales que aporten valor no fue tan exitosa. Por un lado, debido a la carencia de bagaje sobre la gestión de cambios en aplicaciones en un entorno de producción. Por el otro, debido a la carencia de *stakeholders* usuarios de quienes poder obtener retroalimentación directa.

En cuanto al seguimiento de la planificación y metodología, la planificación del proyecto inicial ha requerido ajustes, como es común en desarrollos de esta naturaleza. Se siguió un enfoque que combinaba el modelo en cascada con un enfoque iterativo, ideal para la implementación de bases de datos.

Así, este método permitió una planificación estructurada, pero aportando flexibilidad para realizar ajustes. Cada fase del proyecto se abordó de manera iterativa, permitiendo probar nuevas funcionalidades y corregir errores de manera eficiente. A pesar de algunos desafíos, como averías en la máquina

virtual, se logró mitigarlos rápidamente, demostrando la eficacia del enfoque y la metodología adoptados.

Finalmente, se identifican varias áreas de trabajo futuro que no se han podido explorar en este proyecto. Entre ellas, como es evidente incluso en el enunciado del proyecto que especifica que está fuera del alcance, el desarrollo de una aplicación con lenguaje de alto nivel que interactúe con la base de datos. Actualmente, pedir a usuarios que ejecuten procedimientos almacenados en una consola provocaría una resistencia al cambio muy fuerte en la gran mayoría.

Siguiendo con la usabilidad hacia los futuros usuarios, sería necesario crear un sistema de permisos, idealmente ya dentro de la aplicación gráfica, que controle qué usuarios pueden ejecutar qué procedimientos. Aunque se ha tenido en cuenta la seguridad del sistema, por ejemplo, no pudiendo eliminar registros mediante procedimientos almacenados, esta gestión de permisos sería necesaria para que cada usuario interactuara independientemente.

Otra área de trabajo muy interesante es un periodo de pruebas por parte de *stakeholders* del sistema, especialmente usuarios. Tras pocas iteraciones de pruebas y mejoras al obtener retroalimentación, se identificarían posibles problemas de manejo, mejoras y funcionalidades adicionales que con poco esfuerzo aportan mucho valor.

Por último, al monitorizar las métricas de la base de datos cuando sea utilizada por varias personas en un periodo de tiempo más largo, se identificarán mejoras significativas de rendimiento pasadas por alto en la fase de diseño. Por ejemplo, la modificación o creación de índices para las columnas más recuperadas o el estudio de fragmentaciones por temporalización cuando exista un gran volumen de datos.

7. Glosario

Base de datos (BD): es un sistema organizado para almacenar, gestionar y recuperar información. El propósito principal de una base de datos es almacenar y recuperar información relacionada de manera eficiente.

Chatarra electrónica: dispositivos electrónicos descartados, que han llegado al final de su vida útil o que ya no son útiles, ya sea por obsolescencia o porque ya no son funcionales.

Data-Warehouse (almacén de datos): sistema utilizado para almacenar y gestionar grandes cantidades de datos procedentes de múltiples fuentes. Su objetivo principal es facilitar el análisis de datos y el soporte a procesos de toma de decisiones en una organización.

Disparador: tipo de procedimiento almacenado que se ejecuta automáticamente en respuesta a ciertos eventos en una tabla o vista. Su objetivo principal es mantener la integridad de los datos, automatizar procesos y realizar tareas de gestión de la base de datos de forma eficiente y en tiempo real.

GROUP BY: cláusula SQL que se utiliza para agrupar filas que tienen los mismos valores en columnas específicas. Esto es especialmente útil cuando se realizan operaciones de agregación, como contar, sumar, calcular promedios, entre otros.

Modelo Entidad-Relación: modelo conceptual ampliamente utilizado en el campo de las bases de datos para representar de manera estructurada los datos y sus relaciones.

Procedimiento almacenado: conjunto de instrucciones SQL y código PL/SQL que se compila y almacena en la base de datos. Puede ser llamado y ejecutado por aplicaciones o por otros componentes de la base de datos.

SELECT: instrucción SQL que se utiliza para recuperar datos de una o más tablas.

Sistema de Gestión de Bases de Datos (SGBD): software que permite crear, mantener y utilizar una base de datos. Los SGBD proporcionan las herramientas necesarias para almacenar, modificar, extraer y buscar información de manera eficiente.

Stakeholder (parte interesada): cualquier individuo, grupo o entidad que tiene un interés o se ve afectado por el proyecto. Sus necesidades y opiniones pueden influir significativamente en el diseño y funcionalidad del proyecto, ya que habitualmente se deben cumplir con los requisitos y las expectativas de todas las partes involucradas.

Tablespace: unidad de almacenamiento lógica donde se guardan estructuras de datos, como tablas e índices. Su objetivo es agrupar estructuras relacionadas

para facilitar su administración y para proporcionar flexibilidad en la gestión del almacenamiento de la base de datos.

Trabajo programado: tarea o conjunto de operaciones que se configuran para ejecutarse automáticamente en un horario específico. Esta funcionalidad es una parte importante de la administración de bases de datos, ya que permite automatizar tareas repetitivas o esenciales, como respaldos, actualizaciones de datos, mantenimiento de índices, entre otros.

Palabra reservada: término o identificador que tiene un significado especial en el lenguaje de consulta de la base de datos, en este caso, SQL. Estos términos son fundamentales para la sintaxis del lenguaje y no pueden ser utilizados para otros propósitos, como nombres de variables, identificadores de columnas o nombres de tablas.

Secuencia: objeto de base de datos que se utiliza para generar números secuenciales, los cuales son únicos y se generan de manera consecutiva, y son comúnmente utilizadas para crear identificadores únicos.

Unified Modeling Language (UML): metodología de modelado y especificación que se utiliza ampliamente en el diseño y desarrollo de software. Proporciona una gama de diagramas y técnicas que pueden ser utilizadas para representar visualmente los aspectos complejos de sistemas de software, facilitando así su entendimiento, desarrollo y documentación.

8. Bibliografía

[1] *Microsoft Azure*. (12/11/2023).

[https://portal.azure.com/?Microsoft_Azure_Education_correlationId=b0a3a712-9b12-4148-a348-57fab1323a47#view/Microsoft_Azure_Education/EducationMenuBlade/~software](https://portal.azure.com/?Microsoft_Azure_Education_correlationId=b0a3a712-9b12-4148-a348-57fab1323a47#view/Microsoft_Azure_Education/EducationMenuBlade/~/software)

[2] Rodríguez, & Mariné Jové, P. (2018). *Gestión de proyectos – módulo 4 / José Ramón Rodríguez (coordinador); Pere Mariné Jové* (Tercera edición). Universitat Oberta de Catalunya.

[3] Casas Roma. (2019). *Introducción al diseño de bases de datos / Jordi Casas Roma*. Universitat Oberta de Catalunya.

[4] Casas Roma. (2020). *Diseño conceptual de bases de datos / Jordi Casas Roma, Josep Cuartero Olivera* (Quinta edición). Universitat Oberta de Catalunya.

[5] Burgués i Illa. (2020). *Diseño lógico de bases de datos / Xavier Burgués Illa, Josep Cuartero Olivera* (Quinta edición). Universitat Oberta de Catalunya.

[6] *Oracle Database Express Edition | Oracle España*. (12/11/2023).

<https://www.oracle.com/es/database/technologies/appdev/xe.html>

[7] *Oracle SQL Developer Downloads*. (15/11/2023).

<https://www.oracle.com/database/sqldeveloper/technologies/download/>

[8] Casany Guerrero, Rodríguez, M. E., & Conesa Caralt, J. (2016). *Bases de datos para Data Warehouse – módulo 4 / M. José Casany Guerrero* [i cinc més]; el encargo y la creación de este material docente han sido coordinados por los profesores: M. Elena Rodríguez González, Jordi Conesa Caralt (Primera edición). Universitat Oberta de Catalunya.

[9] Descubre Oracle Cloud infrastructure. (14/12/2023).
<https://www.oracle.com/es/cloud/>

9. Anexos

Anexo I - Despliegue de una base de datos autónoma en *Oracle Cloud Infrastructure*.