

Diseño e implementación de la base de datos para una aplicación de control de prácticas en empresas de estudiantes universitarios

Miguel Ángel López Cobo
Grado de Ingeniería Informática
Bases de Datos

Jordi Ferrer Duran
Javier Jiménez Pelayo

Enero 2023



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

FICHA DEL TRABAJO FINAL

Título del trabajo:	<i>Diseño e implementación de la base de datos para una aplicación de control de prácticas en empresas de estudiantes universitarios</i>
Nombre del autor:	<i>Miguel Ángel López Cobo</i>
Nombre del consultor/a:	<i>Jordi Ferrer Duran</i>
Nombre del PRA:	<i>Javier Jiménez Pelayo</i>
Fecha de entrega (mm/aaaa):	01/2023
Titulación:	<i>Grado de Ingeniería Informática</i>
Área del Trabajo Final:	<i>Bases de Datos</i>
Idioma del trabajo:	<i>Castellano</i>
Palabras clave	<i>prácticas, universidad, empresa</i>
<p>Resumen del Trabajo (máximo 250 palabras): <i>Con la finalidad, contexto de aplicación, metodología, resultados i conclusiones del trabajo.</i></p>	
<p>La finalidad de este proyecto es el diseño de una base de datos que permita la centralización de toda la información y procesos necesarios para una futura aplicación para la gestión y control de las prácticas en empresas que los estudiantes universitarios realizan durante los últimos pasos de sus estudios.</p> <p>Dichas prácticas son una de las piezas más importantes para la consecución de una formación de calidad por parte de los estudiantes y, debido al gran número de actores que forman parte en dicho sistema, es normal que el Consell Interuniversitari de Catalunya quiera centralizar todo este proceso en una sola aplicación.</p> <p>Se decide hacer uso de la metodología en cascada para el desarrollo de este proyecto, dándole una gran importancia a las fases de definición de requisitos y de diseño conceptual con el objetivo de tomar las decisiones más relevantes al inicio del proyecto.</p> <p>A la finalización del proyecto se consiguen alcanzar todos los objetivos propuestos a la vez que queda patente que la mayor parte de las decisiones tomadas durante las etapas tempranas del desarrollo han sido un total acierto.</p>	
<p>Abstract (in English, 250 words or less):</p>	
<p>The purpose of this project is to design a database that allows the centralization of all the information and processes necessary for a future application for the management and control of internships in companies that university students perform during the last months of their studies.</p>	

These internships are one of the milestones for students to achieve a successful college career. Due to the large number of actors that are part of the process and its importance, It is normal for the *Consell Interuniversitari de Catalunya* to want to centralize this entire process in a single application.

The waterfall methodology is used for the development of this project, giving great importance to the definition of requirements and conceptual design phases. In this way, the most important decisions are made at the beginning of the project.

At the end of the project, all the proposed objectives are achieved. At the same time, it is concluded that most of the decisions made during the early stages of development have been a complete success.

Índice

1. INTRODUCCIÓN	1
1.1 CONTEXTO Y JUSTIFICACIÓN DEL TRABAJO	1
1.2 OBJETIVOS DEL TRABAJO	2
1.3 ENFOQUE Y MÉTODO SEGUIDO	2
1.4 PLANIFICACIÓN DEL TRABAJO	3
1.5 PLANES DE CONTINGENCIA	5
1.6 BREVE SUMARIO DE PRODUCTOS OBTENIDOS	6
1.7 BREVE DESCRIPCIÓN DE LOS OTROS CAPÍTULOS DE LA MEMORIA	6
1.8 SEGUIMIENTO DEL PROYECTO	7
2. FASE DE ANÁLISIS DE REQUISITOS	9
2.1 REQUISITOS FUNCIONALES	9
2.2 REQUISITOS NO FUNCIONALES	12
2.3 OTROS REQUISITOS A CONSIDERAR	14
3. FASE DE DISEÑO	15
3.1 DISEÑO CONCEPTUAL	15
3.2 DISEÑO LÓGICO	30
3.3 NORMALIZACIÓN	37
4. IMPLEMENTACIÓN FÍSICA	40
4.1 PROCEDIMIENTOS ALMACENADOS ABM	50
4.2 EL REPOSITORIO ESTADÍSTICO	62
5. FASE DE PRUEBAS	72
6. CONCLUSIONES	81
7. GLOSARIO	82
8. BIBLIOGRAFÍA	83

1. Introducción

1.1 Contexto y justificación del Trabajo

Son muchos los alumnos que, al final de sus estudios, realizan prácticas en empresas y organismos públicos con el objetivo de aprender como aquello que han estado estudiando se lleva a la práctica en un entorno laboral real.

La primera vez que estas prácticas fueron reguladas fue a través del Real Decreto 1497/1981, de 19 de junio, sobre Programas de Formación Educativa con el que se buscaba permitir a los estudiantes poner en práctica la teoría que habían estado aprendiendo durante su formación, del mismo modo que se estaba haciendo con éxito en otros países, especialmente aquellos que tenían una economía avanzada.

Han pasado más de cuarenta años desde aquella primera normativa que se ha ido actualizando hasta el actual Real Decreto 592/2014, de 11 de julio, por el que se regulan las prácticas académicas externas de los estudiantes universitarios y cuyo propósito es desarrollar, precisar y aclarar algunos aspectos como los objetivos de las prácticas, los requisitos o las tutorías.

Entrar en los detalles legales y jurídicos que hay detrás de las prácticas universitarias en empresas no es uno de los objetivos de este trabajo. Sin embargo, esto nos ayuda a comprender que existe un interés real y legítimo a la hora de querer regular y controlar dichas prácticas, pues tienen y tendrán un impacto relevante en la formación de los estudiantes universitarios.

Estas prácticas deben aportar un valor académico para el alumno, nunca sustituir un puesto de trabajo real y, por supuesto, tener un sistema por el cual se pueda medir los resultados obtenidos.

Gracias a todo esto, podemos entender fácilmente porque un organismo como el *Consell Interuniversitari de Catalunya* (CIC de ahora en adelante) ha planteado la necesidad de controlar de una manera más efectiva las prácticas de los estudiantes universitarios. Necesidad a la cual, a lo largo de este trabajo, daremos respuesta.

Con este claro objetivo en mente, a la finalización de este trabajo el resultado será una única base de datos (BD de ahora en adelante) que permita centralizar toda la información necesaria para el seguimiento de dichas prácticas, permitiendo al CIC poder mejorar la eficiencia con la que se ejecutan y controlan todas las etapas del proceso.

1.2 Objetivos del Trabajo

Como ya se ha adelantado en el anterior punto, este trabajo tiene un objetivo principal muy claro y fácil de identificar que consiste en la creación de una BD que permita centralizar toda la información necesaria y útil para poder hacer seguimiento de las prácticas que los alumnos universitarios realizan en la última etapa de sus estudios.

Se trata de un objetivo muy amplio y que puede desgranarse de forma sencilla en objetivos mucho más pequeños. Sin embargo, esto se verá en más detalle en los apartados relacionados con la planificación de trabajo y la descripción de los capítulos de la memoria.

Esta BD no solo servirá como un almacén centralizado de información, sino que también permitirá realizar una serie de consultas a un repositorio estadístico que funcionará de forma eficiente y en tiempo constante ¹ sin importar el volumen de datos tratado.

El producto entregado será escalable, ya que se prevé que se añadan nuevas funcionalidades de forma progresiva que aún no se han tenido en cuenta. Algunas de estas funcionalidades se mencionarán a lo largo de la memoria.

El último objetivo es la implementación de mecanismos que permitan la resolución de posibles problemas de integración que la BD pueda tener con el resto del sistema.

1.3 Enfoque y método seguido

No disponemos de ningún tipo de información sobre cómo se está gestionando hasta el momento y tampoco se nos habla de que exista ninguna tecnología heredada que deba mantenerse. Por lo tanto, para este proyecto se creará una nueva BD que resuelva todas las necesidades del CIC, utilizando *Oracle Database Express Edition* como sistema de gestión de bases de datos (SGBD de ahora en adelante).

Por lo tanto, el enfoque será realizar todas las etapas de diseño de una BD, desde la recogida y análisis de requisitos hasta la implementación definitiva. Este es el único modo de garantizar que se respetarán todos los requisitos indicados en el enunciado y que el producto final responde a todas las necesidades del CIC.

Para el desarrollo de la BD se utilizará la metodología en cascada como forma de trabajo. Los beneficios obtenidos de aplicar esta metodología para este proyecto en concreto superan con creces los que podrían obtenerse con una metodología ágil como *Scrum*.

En primer lugar, se trata de un proyecto con un objetivo muy claro y solo leyendo el enunciado ya se puede saber cómo será o como debería ser el resultado final entregado. Esto implica que, si se hace un buen trabajo en las fases de análisis de requisitos y en la implementación, se minimizará el riesgo de encontrar un error en las fases posteriores del proyecto que obligue a volver al principio.

Además, tal y como se estudió durante la asignatura de Diseño de Bases de Datos (75.585), el proceso para la creación de una BD tiene una serie de pasos que deben realizarse de forma secuencial, pues es necesario terminar uno para poder comenzar con el siguiente.

Esto se adapta muy bien a la filosofía que hay detrás de la metodología en cascada. Una nueva fase solo se comienza cuando ya se ha terminado la anterior y cuando la última fase se haya completado, se dispondrá de un producto entregable que cumple todos los requisitos.

La cascada de este proyecto tiene la siguiente forma:

1. Requisitos

- a. Análisis de los requisitos indicados en el enunciado.
- b. Requisitos funcionales no considerados en el enunciado.

2. Diseño

- a. Diseño Conceptual de la BD
- b. Diseño Lógico de la BD
 - i. Análisis de la Normalización

3. Implementación

- a. Diseño Físico de la BD
- b. Diseño del Repositorio Estadístico

4. Pruebas

- a. Creación e inyección de un conjunto de pruebas en la BD
- b. Diseño y ejecución de las pruebas de funcionalidad

Las fases de Verificación y Mantenimiento del producto quedan fuera del alcance de este proyecto.

1.4 Planificación del Trabajo

El desarrollo de este proyecto se compagina con otros 18 créditos ECTS y un trabajo a jornada completa con un horario a turnos que cambia cada semana, por lo que no será posible dedicar un tiempo fijo cada día.

En cualquier caso, se estima que se podrán trabajar aproximadamente 14 horas semanales, lo que da una media estimada de dos horas diarias.

Se ha previsto que durante la duración de la PEC 2 se realicen todos los puntos de la fase de Requisitos y el diseño conceptual de la BD de la fase de Diseño. Durante la PEC 3 se finalizará la fase de Diseño y la fase de Implementación. Esto deja la PEC 4 para la fase de Pruebas y para la finalización y corrección de la memoria.

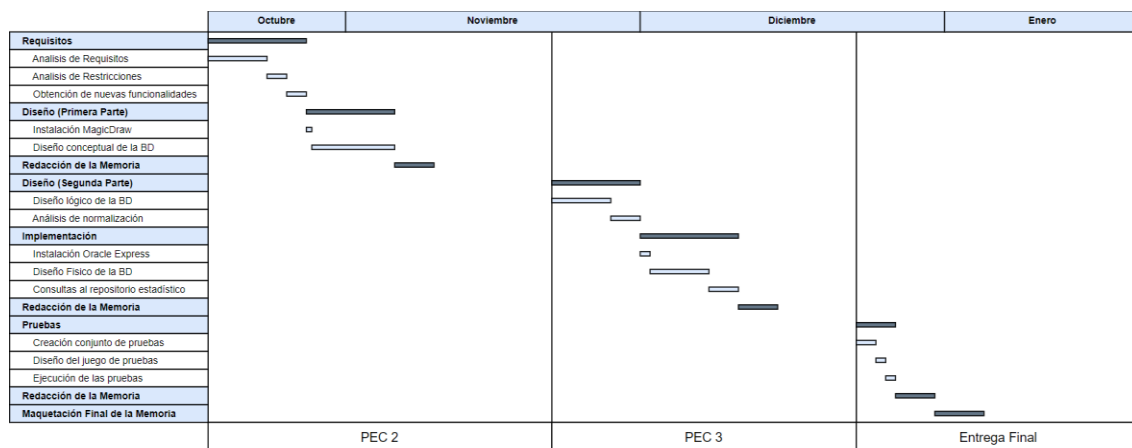
Con todo esto en mente, la planificación temporal de este proyecto queda de la siguiente forma:

	<i>Horas</i>	<i>Inicio</i>	<i>Fin</i>
<u>Duración PEC 2</u>		<u>18/10/22</u>	<u>21/11/22</u>
• Requisitos	20	18/10/22	27/10/22
○ Análisis de Requisitos	12	18/10/22	23/10/22
○ Análisis de las Restricciones	4	24/10/22	25/10/22
○ Obtención nuevas funcionalidades	4	26/10/22	27/10/22
• Diseño (Primera Parte)	17	28/10/22	05/11/22
○ Instalación <i>MagicDraw</i>	1	28/10/22	28/10/22
○ Diseño Conceptual de la BD	16	28/10/22	05/11/22
• Redacción de la memoria	8	06/11/22	09/11/22
<u>Duración PEC 3</u>		<u>22/11/22</u>	<u>22/12/22</u>
• Diseño (Segunda Parte)	18	22/11/22	30/11/22
○ Diseño Lógico de la BD	12	22/11/22	27/11/22
○ Análisis de Normalización	6	28/11/22	30/11/22
• Implementación	20	01/12/22	10/12/22
○ Instalación <i>Oracle Express</i>	2	01/12/22	01/12/22
○ Diseño Físico de la BD	12	02/12/22	07/12/22
○ Consultas al repositorio estadístico	6	08/12/22	10/12/22
• Redacción de la memoria	8	11/12/22	14/12/22
<u>Duración Entrega Final</u>		<u>23/12/22</u>	<u>20/01/23</u>
• Pruebas	8	23/12/22	26/12/22
○ Creación del conjunto de pruebas	4	23/12/22	24/12/22
○ Diseño del juego de pruebas	2	25/12/22	25/12/22
○ Ejecución de las pruebas	2	26/12/22	26/12/22
• Redacción de la memoria	8	27/12/22	30/12/22
• Maquetación final de la memoria	10	31/12/22	04/01/23

Como puede apreciarse, siempre queda algo de tiempo disponible entre la finalización de la última actividad prevista para cada PEC y la finalización de dicha

PEC. Esto es deliberado para poder contar con tiempo adicional para imprevistos o por si alguna actividad requiriera más tiempo del estimado.

En la siguiente figura puede apreciarse la temporización mediante un diagrama de Gantt:



En este diagrama puede apreciarse el carácter secuencial de este proyecto y también el tiempo reservado para posibles incidencias e imprevistos.

1.5 Planes de Contingencia

Durante el desarrollo de este proyecto existen diversos riesgos con diferentes consecuencias e impactos que deben ser previstos para tratar de paliar sus efectos, entre ellos se encuentran los siguientes:

Contratiempo	Alcance	Plan de Contingencia
1. Fallo en el equipo de trabajo.	<ul style="list-style-type: none"> Perdida de la capacidad de trabajar en el proyecto. Riesgo de pérdida de datos. 	<ul style="list-style-type: none"> Se dispone de dos equipos configurados con el mismo software y capacidades. El trabajo se realiza directamente sobre una carpeta de OneDrive.
2. Contratiempos diversos que puedan alterar la previsión de trabajo.	<ul style="list-style-type: none"> Perdida de la capacidad de trabajar en el proyecto. 	<ul style="list-style-type: none"> Para cada fase del proyecto se ha previsto tiempo extra. Ante un caso extremo y justificado se hablará con el tutor.
3. Pérdida de conexión a Internet.	<ul style="list-style-type: none"> Perdida de la capacidad de trabajar en el proyecto. Riesgo de pérdida de datos. Incapacidad de contactar con el tutor. 	<ul style="list-style-type: none"> Se dispone de un punto de acceso 4G en caso de perder la conexión a Internet en el punto de trabajo. Se guardan copias de seguridad del proyecto en un HDD externo por si no hubiera acceso a OneDrive.

1.6 Breve resumen de productos obtenidos

Una vez finalizado el proyecto, los productos obtenidos son los siguientes:

- La memoria del proyecto. Este documento incluye tanto los detalles del diseño conceptual, lógico y físico de la BD, así como también aquellas funcionalidades que se haya considerado que podrían incluirse en un futuro.
- El *script* SQL para *Oracle Express* que crea la BD que responde a todas las necesidades planteadas por el CIC, incluyendo el repositorio estadístico y aquellas medidas pensadas para resolver posibles problemas de integración, como un *log* para los procedimientos.
- Un conjunto de pruebas que contiene tanto los datos de prueba como un *script* SQL que permite validar el correcto funcionamiento de la BD.
- El conjunto de consultas SQL que permite validar el correcto funcionamiento de los cálculos que la BD realiza para calcular el repositorio estadístico.

1.7 Breve descripción de los otros capítulos de la memoria

Los siguientes capítulos se organizan siguiendo el mismo esquema de trabajo que ya se ha visto en apartados anteriores.

Capítulo 2 - Fase de análisis de requisitos

Este capítulo comienza con un análisis exhaustivo de los requisitos funcionales y no funcionales que se pueden extraer del enunciado. Luego continúa con la descripción de las restricciones que se aplicarán a la BD para, finalmente, hacer una reflexión sobre funcionalidades que podrían añadirse y que serían relevantes para el problema que se está tratando.

Capítulo 3 - Fase de diseño

El capítulo tres se centra en el diseño conceptual y lógico de la BD que finalmente se construirá. Comienza con un diagrama UML que muestra todas las entidades que se almacenarán en la BD, así como las relaciones entre ellas. En este apartado se abordará por primera vez como resolver la restricción sobre el tiempo de ejecución de las consultas realizadas al repositorio estadístico, así como las diferentes medidas que se implementarán para ayudar a la resolución de problemas de integración.

La segunda parte será el diseño lógico de la BD, junto a un análisis de la normalización cuyo objetivo es demostrar que, como mínimo, todas las relaciones consideradas se encuentran en Forma Normal de Boyce-Codd.

Capítulo 4 - Fase de implementación

Una vez finalizada la fase de diseño, en el penúltimo capítulo se realizará el diseño físico de la BD, que culminará en la colección de *scripts* SQL que crean la BD que da respuesta a todas las necesidades planteadas por el CIC. Se incluye también la colección de consultas SQL con un tiempo de ejecución constante 1 sobre el repositorio estadístico.

Capítulo 5 - Fase de pruebas

En el último capítulo se presentan las pruebas realizadas que permiten garantizar el correcto funcionamiento de la BD. Incluye tanto los datos de prueba introducidos, el juego de pruebas ejecutado y un informe que valora los resultados obtenidos en dichas pruebas.

1.8 Seguimiento del Proyecto

PEC 1

La conceptualización y diseño del plan de trabajo no ha presentado ninguna complejidad o problemática que haya provocado un retraso o una situación en la que peligrara la entrega. La parte más compleja fue tratar de ser lo más honesto posible con la capacidad semanal de trabajo y cuáles son los hitos que se pueden conseguir antes de cada entrega parcial.

PEC 2

La entrega parcial de la PEC 2 ha sido mucho más problemática de lo inicialmente previsto. Una gran serie de eventos personales y profesionales han afectado al tiempo de trabajo asignado y, por consecuencia, a la calidad de la entrega.

En cualquier caso, esto ha vuelto palpable el gran acierto que fue asignar una enorme cantidad de tiempo libre en cada fase del proyecto durante la elaboración del plan de contingencia, pues ese tiempo ha permitido que se pueda conseguir una entrega que cumpliera los requisitos mínimos esperados a pesar de todos los inconvenientes encontrados.

Sin embargo, esto ha provocado que se vaya a consumir algo del tiempo disponible para la PEC 3 en retocar y añadir algunos detalles a lo que ya debería haber quedado realizado en la PEC 2, lo que puede provocar un retraso en el avance de la próxima entrega parcial.

El diagrama UML y, en general, todo el diseño conceptual que se ha finalizado servirá como la base sobre la que se construirá todo lo que queda de proyecto, por lo que al finalizar esta PEC se ha conseguido finalizar los cimientos de todo el trabajo.

PEC 3

La entrega de la PEC 3 ha heredado problemas que ya aparecieron en la anterior. Por un lado, un cambio de paradigma laboral ha provocado una reducción drástica del tiempo disponible para el TFG y otras asignaturas.

A este hay que sumarle que a mediados de diciembre me contagié de COVID y que hubo tareas de la PEC 2 que debieron hacerse al principio de la PEC 3, como mejorar el modelo conceptual, por ejemplo.

Con la autorización del profesor consultor, se hace entrega de la PEC 3 tres días más tarde de la fecha oficial. La entrega se hace habiendo cumplido todos los objetivos propuestos. Junto a la memoria, se entrega el documento .SQL que ejecuta la implementación física de la Base de Datos. Quedando ya solo el juego de pruebas de cara a la entrega final.

Entrega Final

Esta etapa del proyecto ha estado marcada para la identificación inicial de un error en el diseño del repositorio estadístico que hacía que no cumpliera con el requisito de resolver en un tiempo constante 1.

Las primeras semanas se utilizan para corregir este defecto que, por la naturaleza del proyecto, se propagó desde la etapa de diseño hasta la etapa de implementación. En cualquier caso, se consigue completar a tiempo todos los objetivos gracias al tiempo adicional que se había guardado para los imprevistos.

Finalmente, gracias al diseño y la ejecución de las pruebas se demuestra que el diseño de la BD ha sido correcto. La entrega final se hace de forma satisfactoria antes de la finalización del tiempo límite.

2. Fase de análisis de requisitos

El propio enunciado narra las características, requisitos y restricciones que se espera que tenga nuestra BBDD. Por ello, en este capítulo se desgranará el enunciado para extraer de él toda la información necesaria para la culminación de este proyecto.

2.1 Requisitos funcionales

Los requisitos funcionales se definen como aquellos que hacen referencia a la funcionalidad que debe proporcionar el proyecto. En este caso, nuestra base de datos. Tras analizar el enunciado, estos son los requisitos funcionales detectados:

Código	Requisito
RF-01	<p>La base de datos debe permitir gestionar todas las etapas del proceso, desde la publicación de las ofertas hasta la finalización de las prácticas.</p> <ul style="list-style-type: none">• Pág. 2, párrafo 2: <i>“... lo que se pretende es el proceso en todas sus etapas, empezando desde las ofertas que las empresas presentan a las universidades, pasando por todos los detalles de las asignaciones, y acabando con la valoración de las prácticas.”</i>
RF-02	<p>Los datos elementales que constituyen la base de la BD serán las universidades, estudiantes, titulaciones, profesores responsables y empresas que ofrecen las posiciones.</p> <ul style="list-style-type: none">• Pág. 2, párrafo 3: <i>“Los primeros datos para incluir en el sistema de BD serán los básicos de los principales actores del proceso: universidades, estudiantes, titulaciones, profesores responsables y empresas que ofrecen posiciones.”</i>
RF-03	<p>Para cada empresa queremos almacenar los solicitantes internos que serán los responsables de los alumnos.</p> <ul style="list-style-type: none">• Pág. 3, párrafo 1: <i>“Un aspecto importante para considerar, dentro de los datos de las empresas, es el contacto de las personas clave en el proceso (básicamente la persona de RRHH y la persona que controlará al estudiante durante todo el periodo de prácticas)”</i>• Pág. 3, párrafo 3: <i>“... pero se puede considerar que cada empresa tiene un conjunto de solicitantes que serán los responsables de los estudiantes durante el periodo que duren las prácticas.”</i>
RF-04	<p>La BD guardará, para cada empresa, la lista de ofertas que publicará en cada universidad.</p> <ul style="list-style-type: none">• Pág. 3, párrafo 2: <i>“... y se le pide que mande una descripción de todas las plazas que quiere cubrir en cada una de las universidades que necesiten empresas donde asignar estudiantes.”</i>
RF-05	<p>La BD debe permitir guardar las candidaturas que los alumnos realizan a las ofertas publicadas por las empresas, almacenando también los requisitos establecidos que, para cada empresa, serán diferentes.</p> <ul style="list-style-type: none">• Pág. 3, párrafo 4: <i>“Estas ofertas están abiertas a los diferentes estudiantes</i>

del centro universitario con tal que los que han de cursar las prácticas las analicen y, entre todas las disponibles, se apunten a las que consideren interesantes. Una vez estén apuntados, los estudiantes deberán cumplir con los requisitos establecidos por la empresa para formalizar su candidatura. Los requisitos pueden ir desde tener que escribir una presentación individual o, como hacen muchas empresas, grabarse en un vídeo respondiendo a unas preguntas previamente preparadas por las empresas.”

RF-06 Para cada empresa queremos conocer el responsable de RRHH que será el encargado de recibir, analizar y evaluar las candidaturas emitidas. Cada empresa tendrá un único responsable que será también quien envíe las candidaturas validadas a los solicitantes internos.

- Pág. 3, párrafo 5: *“Estas candidaturas se deben enviar a la persona de contacto que las empresas definan que, habitualmente, suele ser una persona del departamento de RRHH. Esta persona se encargará de validar que las candidaturas de estudiantes se adapten a las ofertas enviadas y, una vez validadas, las enviará a los diferentes solicitantes internos.”*
- Pág. 3, párrafo 1: *“Un aspecto importante para considerar, dentro de los datos de las empresas, es el contacto de las personas clave en el proceso (básicamente la persona de RRHH y la persona que controlará al estudiante durante todo el periodo de prácticas)”*

RF-07 Para cada solicitud, debe haber un campo que permita indicar si la solicitud cumple los requisitos necesarios para ser considerada válida y, posteriormente, hacerla llegar a los solicitantes internos.

- Pág. 3, párrafo 6: *“Esta persona se encargará de validar que las candidaturas de estudiantes se adapten a las ofertas enviadas y, una vez validadas, las enviará a los diferentes solicitantes internos.”*

RF-08 Queremos poder almacenar la información relativa a las entrevistas virtuales que se realizarán a los candidatos que se consideren aptos, incluyendo los comentarios del entrevistador y la valoración final.

- Pág. 4, párrafo 1: *“...y nuestro sistema debe permitir guardar los comentarios efectuados por la persona que realiza la entrevista y la valoración final (que será apto o no apto).”*

RF-09 La BD debe almacenar también la información relativa a los contratos de prácticas efectuados entre la empresa y los alumnos en prácticas, incluyendo como mínimo la duración y la remuneración.

- Pág. 4, párrafo 2: *“El estudiante, con el asesoramiento de la universidad, deberá firmar un contrato de trabajo con la empresa como trabajador en prácticas. Este contrato no tiene una duración predeterminada y suele durar entre 6 meses y 1 año de colaboración. El contrato especifica claramente que es una colaboración en prácticas y no laboral, aunque el estudiante recibe un sueldo que varía mucho dependiendo de la empresa donde realice las prácticas.”*

RF-10 Para cada estudiante en prácticas, queremos guardar a la persona que será su responsable y su programa formativo, incluyendo la lista de temas relativos a las funciones que desempeñará.

	<ul style="list-style-type: none"> • Pág. 4, párrafo 3: <i>“Cuando el estudiante se incorpora a la empresa, su formación pasa a ser responsabilidad de la persona solicitante dentro de la empresa. Esta persona se encarga de preparar un programa formativo que permita al estudiante aprender diferentes temas relativos a la funcionalidad de la empresa.”</i>
RF-11	<p>La BD debe almacenar la información relativa a los informes periódicos que el responsable emitirá a las universidades en relación con el desempeño de cada alumno en prácticas.</p> <ul style="list-style-type: none"> • Pág. 4, párrafo 3: <i>“El estudiante deberá estar monitorizado constantemente con tal de asegurar la correcta evolución de su formación. Periódicamente, la persona responsable de la empresa deberá pasar informes a la universidad indicando cómo el estudiante está desarrollando sus prácticas. La periodicidad de estos informes dependerá de la empresa y de la universidad implicadas.”</i>
RF-12	<p>La BD debe almacenar también el informe final de prácticas, incluyendo la valoración subjetiva final y cualquier comentario que las personas implicadas en el proceso quieran aportar.</p> <ul style="list-style-type: none"> • Pág. 4, párrafo 4: <i>“la persona responsable de la empresa preparará el informe final de prácticas incluyendo su valoración subjetiva y si los considera aptos o no. Nuestro sistema de BD debe permitir almacenar estos informes y cualquier comentario adicional que las personas implicadas en el proceso puedan añadir.”</i>
RF-13	<p>La BD debe almacenar la información relativa a las inspecciones periódicas que serán realizadas por parte del gobierno con el objetivo de revisar los contratos y las tareas realizadas por los alumnos en prácticas.</p> <ul style="list-style-type: none"> • Pág. 4, párrafo 5: <i>“... debe permitir gestionar es el control de las diferentes inspecciones de trabajo que el correspondiente departamento del Gobierno pueda decidir efectuar. Dentro de estas inspecciones se analizarán los diferentes contratos y, principalmente, las tareas que el estudiante realiza en la empresa con tal de asegurar que no se trate de un puesto de trabajo encubierto.”</i>
RF-14	<p>Se creará un repositorio estadístico cuyo objetivo será aportar información sobre diferentes indicadores a los que se quiere acceder de forma rápida. Los indicadores mínimos que deben facilitarse son los siguientes y se encuentran en la página 6 del enunciado:</p> <ul style="list-style-type: none"> • <i>Universidad que, analizando todos los datos de que disponemos, tiene un porcentaje más alto de estudiantes que superan las prácticas en empresas. Se consideran superadas cuando la persona responsable del estudiante en la empresa lo califica como apto.</i> • <i>Top5 de empresas que han ofrecido más plazas para estudiantes en prácticas al CIC.</i> • <i>Empresa que, entre todos los datos de que disponemos, ha recibido más inspecciones de trabajo (sin considerar el resultado de la inspección).</i> • <i>Dado un año universitario en concreto, porcentaje de estudiantes, de cualquier universidad, que no superan las prácticas en empresas.</i>

- *En el año universitario en curso, sueldo medio que reciben los estudiantes durante sus prácticas.*
- *Número total de estudiantes en prácticas en el momento de ejecutar la consulta. Se considerarán los estudiantes que han iniciado las prácticas y no las han acabado.*
- *Durante el último año universitario, porcentaje de universidades controladas por el CIC que no tienen estudiantes en prácticas.*
- *Titulación universitaria con un mayor número de estudiantes que han acabado satisfactoriamente las prácticas en empresas. Se tendrán en cuenta todos los datos de que se dispone en la BD.*
- *Número medio de estudiantes que se presentan a cada oferta recibida por las diferentes universidades durante el año universitario en curso.*
- *Teniendo en cuenta todos los datos de que se dispone, duración media de los contratos de estudiantes con las empresas.*
- *Empresa que, en un año en concreto, ha realizado el mayor número de entrevistas.*
- *En el año universitario en curso, profesor que lleva un mayor número de estudiantes que están realizando prácticas en empresas.*
- *En el momento de ejecutar la consulta, número total de informes de prácticas pendientes de ser realizados por las personas responsables de las empresas.*
- *Teniendo en cuenta el año universitario en curso y el anterior, porcentaje de empresas que ofrecieron plazas el año anterior y no las han ofrecido en el actual.*

RF-15	<p>La BD deberá almacenar en una tabla de LOGs los parámetros de salida mencionados en RF-15, para poder controlar el correcto funcionamiento de la BD. La información para almacenar será el nombre del procedimiento, los parámetros de entrada y los parámetros de salida.</p> <ul style="list-style-type: none"> • Pág. 7, párrafo 5: <i>“Para el tema del log de las acciones realizadas, se recomienda almacenar todas las llamadas a procedimientos que se hagan en una tabla de log, almacenando el nombre del procedimiento ejecutado y los parámetros de entrada y de salida.”</i>
--------------	---

2.2 Requisitos no funcionales

Los requisitos no funcionales son aquellos que no describen la funcionalidad del proyecto, lo que definen son cualidades que se espera que nuestra BD. Del enunciado pueden desprenderse los siguientes requisitos no funcionales.

Código	Requisito
RNF-01	La base será construida con el motor de base de datos ORACLE Database Express Edition.
RNF-02	<p>La BD debe poder dar soporte a una futura aplicación que permitirá el control de todas las fases del proceso de prácticas universitarias en empresa.</p> <ul style="list-style-type: none"> • Pág. 5, párrafo 2: <i>“Esta nueva aplicación debería permitir, a grandes</i>

	<p><i>rasgos, el control de los diferentes aspectos detallados en los puntos anteriores, la ejecución de las consultas que se consideren necesarias para la correcta gestión y análisis de los datos almacenados en el sistema, y también una visualización rápida del estado de los indicadores principales de monitorización del estado de seguridad de la empresa que se definan en cada momento.”</i></p>
RNF-03	<p>La gestión de la BD se realizará mediante procedimientos y, de manera especial, se deberán implementar y describir los procedimientos de ABM de todas las entidades relevantes.</p> <ul style="list-style-type: none"> • Pág. 3, párrafo 4: <i>“Toda la gestión y acceso a la información se hará mediante procedimientos de BD, siendo ésta la única manera de acceder.”</i> • Pág. 3, párrafo 5: <i>“A nivel de procedimientos, se deberá implementar y describir con detalle los procedimientos de ABM (Alta + Baja + Modificación) de todas las entidades (o clases) que se consideren relevantes.”</i>
RNF-04	<p>La BD debe poder funcionar para cualquier volumen de datos.</p> <ul style="list-style-type: none"> • Pág. 5, párrafo 6: <i>“Se deberá tener en cuenta que la aplicación ha de funcionar para cualquier volumen de datos...”</i>
RNF-05	<p>El repositorio estadístico mencionado en RF-14 deberá ofrecer sus resultados en un tiempo constante 1.</p> <ul style="list-style-type: none"> • Pág. 6, párrafo 2: <i>“La única restricción que ha de tener este repositorio estadístico es que debe ofrecer los diferentes resultados que se definan en tiempo constante 1...”</i>
RNF-06	<p>La BD resultante debe ser escalable.</p> <ul style="list-style-type: none"> • Pág. 7, párrafo 2: <i>“La BD deberá de ser escalable para poder ir incorporando progresivamente todas aquellas necesidades que surjan durante su vigencia.”</i>
RNF-07	<p>La BD debe disponer de mecanismos que faciliten el mantenimiento del sistema y la resolución de potenciales errores que puedan surgir, como el almacenamiento de los LOGs mencionados en RF-15 y mecanismos de testeo.</p> <ul style="list-style-type: none"> • Pág. 7, párrafo 4: <i>“Finalmente, y con tal de facilitar el mantenimiento del sistema, se valorará muy positivamente el disponer de mecanismos que permitan resolver potenciales problemas de integración con el resto del sistema: un log de las acciones realizadas con la BD, mecanismos para testear la funcionalidad de la BD...”</i>
RNF-08	<p>Todos los procedimientos almacenados deberán disponer de un parámetro de salida de tipo <i>string</i> que índice si su ejecución ha sido exitosa o no, así como la capacidad para manejar excepciones.</p> <ul style="list-style-type: none"> • Pág. 7, párrafo 5: <i>“Para estandarizar el sistema que se debe desarrollar, se pide explícitamente que los procedimientos almacenados cumplan con las condiciones siguientes:</i> <ul style="list-style-type: none"> ○ <i>Como mínimo dispondrán de un parámetro de salida llamado RSP, de tipo string, que indicará si la ejecución ha finalizado correctamente (valor ‘OK’) o si ha fracasado (valor ‘ERROR+TIPO DE ERROR’).</i> ○ <i>Dispondrán de tratamiento de excepciones.”</i>

2.3 Otros requisitos a considerar

En el enunciado se menciona que cuando una empresa quiere tener estudiantes en prácticas debe comunicarlo con el CIC, el cual decidirá si considera apta o no a una empresa.

Sin embargo, en ninguna parte se menciona la posibilidad de que el CIC deje de considerar válida a una empresa y no vuelva a aceptar ofertas por su parte. Sin embargo, sería lógico pensar que pueden ocurrir situaciones que provoquen que el CIC no quiera volver a trabajar con una determinada empresa.

En el enunciado se menciona muy rápidamente que las empresas tendrán que superar diversas inspecciones de trabajo para garantizar que no se están usando las prácticas para cubrir puestos de trabajo reales. En ese sentido, creo que es normal pensar que el hecho de que una empresa sea sancionada una o más veces por la administración competente debe terminar provocando que dicha empresa no pueda volver a publicar ofertas en el CIC, aunque fuera de forma temporal.

Del mismo modo, si una empresa tuviera un índice de suspenso en las prácticas muy superior a la media, podría estar indicando que pueden no estar haciendo un buen trabajo y sería correcto estudiar por qué se produce esa discrepancia entre una determinada empresa y el resto.

Con esto en mente, se sugiere la inclusión de los siguientes indicadores en el repositorio estadístico:

- El Ranking de las 10 empresas con una tasa de superación de las prácticas más bajas.
- Listado de empresas que al menos el 10% de sus inspecciones de trabajo se han considerado fraudulentas.

El primera serviría para saber que empresas tienen un porcentaje de superación más bajo, permitiendo luego hacer un estudio más exhaustivo para tratar de descubrir si simplemente ha sido casualidad o si existe algún problema con esta empresa que provoque que se acaben considerando no-aptos a tantos candidatos.

El segundo indicador serviría para identificar empresas con las que el CIC podría no estar interesado en trabajar. El 10% es un porcentaje elegido de forma arbitraria y lo normal sería que fuera escogido por el CIC. En cualquier caso, la idea detrás de este identificador es que debe haber un punto en el que, ante cierto número de sanciones por parte de la administración tras una inspección, el CIC quiera cortar su relación con la empresa sancionada.

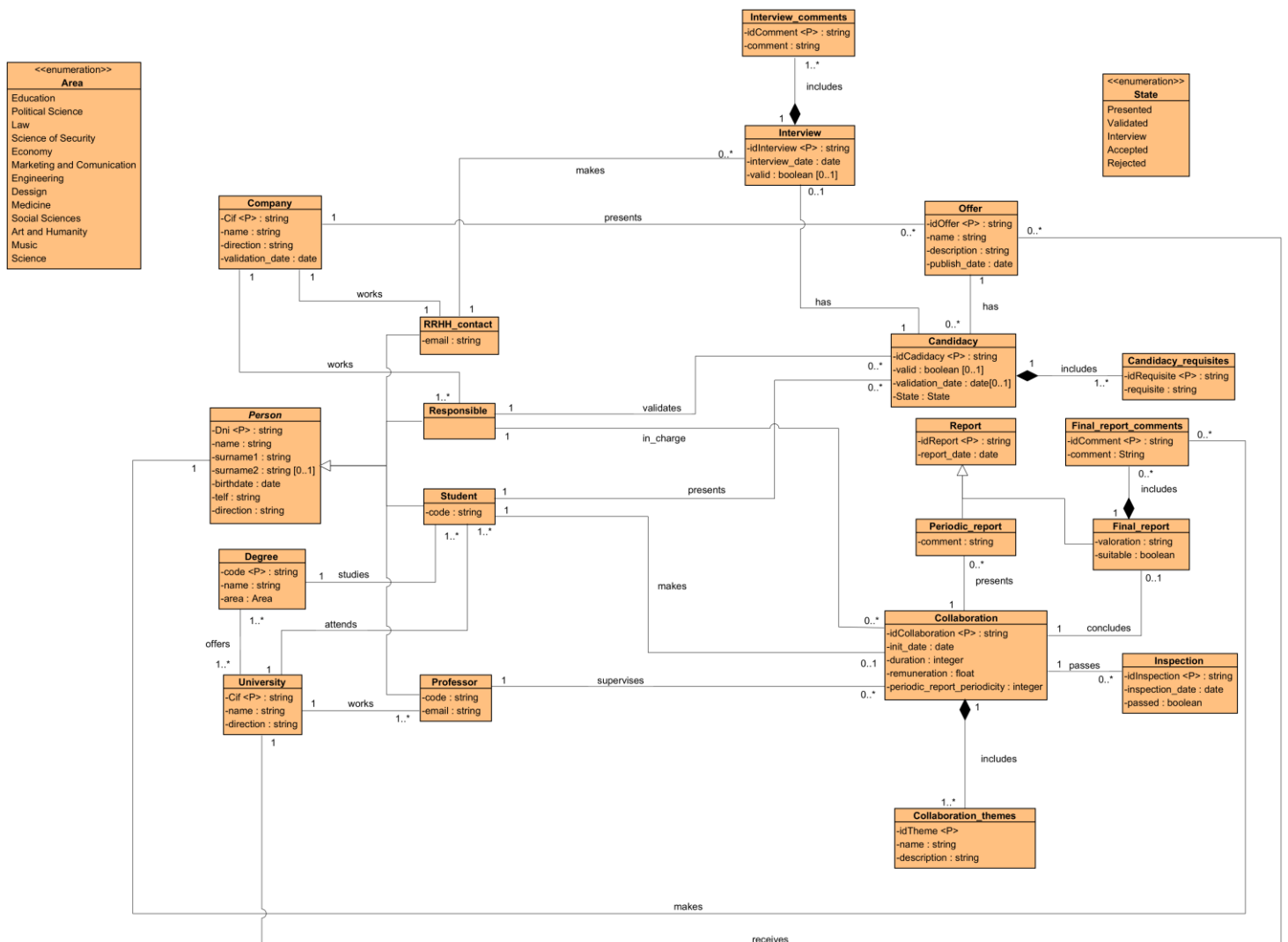
3. Fase de diseño

3.1 Diseño conceptual

Tras el análisis de los requisitos, el siguiente paso para construir una base de datos es la creación de un diseño conceptual de alto nivel que permita visualizar fácilmente las entidades que conformarán el modelo, junto con sus atributos y sus relaciones.

Para esta parte he decidido realizar un diagrama utilizando el lenguaje unificado de modelación (UML).

El primer diseño realizado es el del componente central de la BD, que incluye las entidades principales del modelo toda la información necesaria para satisfacer las necesidades del CIC. El diagrama de clases resultante es el siguiente:



De este diagrama se extraen las siguientes entidades:

Person		
Esta entidad representa al conjunto de personas que se encuentran involucradas en el proceso de realización de prácticas universitarias en empresas.		
Se trata de una superclase abstracta, por lo que todas las personas que haya en la BBDD deben pertenecer también a una de las subclases que heredan de <i>Person</i> .		
Realizar este proceso de generalización y crear la entidad <i>Person</i> es necesario ya que en el enunciado se menciona que cualquier involucrado en el proceso puede realizar comentarios en el informe final de prácticas, por lo que es necesario poder disponer de la entidad <i>Person</i> para poder referenciarla.		
Atributos:		
Atributo	Tipo	Descripción
DNI <P>	String	Se trata del identificador único del que todos los españoles o residentes disponen.
Name	String	Nombre de la persona.
Surname1	String	El primer apellido de la persona.
Surname2 [0..1]	String	El segundo apellido de la persona. Acepta el valor NULL ya que no todas las personas tienen segundo apellido.
Birthdate	Date	La fecha de nacimiento de la persona.
Telf	String	El número de teléfono de contacto de la persona.
Direction	String	La dirección de correspondencia postal de la persona.
Relaciones:		
E1 <Relación> E2	Descripción	Cardinalidad
Person makes Final_report_comment	Cada comentario realizado en el informe final de prácticas es realizado por una persona involucrada en el proceso.	1 : 0..*

Person > RRHH_contact		
Cada empresa define un contacto de recursos humanos que actúa como contacto entre las universidades y las empresas.		
Entre sus funciones se encuentra enviar a las universidades las ofertas realizadas por la empresa, validar que las candidaturas cumplan los requisitos, enviar estas candidaturas a los responsables y, una vez aceptadas, concertar la entrevista virtual con los estudiantes.		
Atributos:		
Atributo	Tipo	Descripción
Email	String	La dirección e-mail de contacto del responsable de recursos humanos.
Relaciones:		
E1 <Relación> E2	Descripción	Cardinalidad
RRHH_contact works Company	Una empresa tiene un contacto de recursos humanos.	1 : 1
RRHH_contact makes Interview	Cada entrevista es concertada por un contacto de recursos humanos.	1 : 0..*

Person > Responsible

Esta entidad representa el subconjunto de personas cuya función es ser un responsable dentro de cada empresa.

Estas personas son las que, una vez iniciada la colaboración entre estudiante y empresa, serán los encargados de hacer el seguimiento y formación de los alumnos en prácticas.

Entre sus funciones también se encuentra la de validar las candidaturas de cada estudiante y decidir si hay que realizarles o no una entrevista virtual.

Relaciones:

<i>E1 <Relación> E2</i>	<i>Descripción</i>	<i>Cardinalidad</i>
Responsible works Company	Cada responsable trabaja para una empresa.	1..* : 1
Responsible validates Candidacy	Cada candidatura es validada por un responsable.	1 : 0..*
Responsible supervises Collaboration	Cada colaboración es supervisada por un responsable.	1 : 0..*

Person > Student

Esta entidad representa al subconjunto de personas que son estudiantes universitarios. Cada uno de ellos pertenece a una universidad y estudia una carrera determinada.

También puede presentar diferentes candidaturas y, finalmente, participar en una colaboración con una empresa mediante un contrato de prácticas.

Atributos:

<i>Atributo</i>	<i>Tipo</i>	<i>Descripción</i>
Code	String	El código interno identificativo que cada estudiante tiene dentro de su universidad.

Relaciones:

<i>E1 <Relación> E2</i>	<i>Descripción</i>	<i>Cardinalidad</i>
Student studies Degree	Cada estudiante está estudiando una carrera.	1..* : 1
Student attends University	Cada estudiante está matriculado en una universidad.	1..* : 1
Student presents Candidacy	Cada candidatura está presentada por un estudiante.	1 : 0..*
Student makes Collaboration	Cada colaboración de prácticas en empresas está realizada por un estudiante.	1 : 0..1

Person > Professor

Esta entidad representa al subconjunto de personas que son profesores universitarios. Dentro del modelo, su principal función es ser el tutor de los alumnos que se encuentran realizando prácticas universitarias.

Atributos:

<i>Atributo</i>	<i>Tipo</i>	<i>Descripción</i>
Code	String	El código interno identificativo que los profesores tienen dentro de la universidad.
Email	String	La dirección de correo electrónico de contacto donde recibirán los informes emitidos durante las prácticas.

Relaciones:		
E1 <Relación> E2	Descripción	Cardinalidad
Professor works University	Cada profesor trabaja en una universidad.	1 : 1..*
Professor supervises Collaboration	Cada alumno en prácticas está siendo tutorizado por un profesor.	1 : 0..*

Company

Esta entidad representa al conjunto de empresas que participan en este proceso. Las empresas son las encargadas de emitir las ofertas a cada una de las universidades y, a través de sus personas responsables, gestionar todo el proceso de validación de candidaturas y las colaboraciones.

Atributos:

Atributo	Tipo	Descripción
CIF <P>	String	El código de identificación fiscal único del que disponen todas las personas jurídicas.
Name	String	El nombre de la empresa.
Direction	String	La dirección fiscal de la empresa.
Validation_date	Date	La fecha en la que la empresa fue considerada como valida por parte del CIC.

Relaciones:

E1 <Relación> E2	Descripción	Cardinalidad
Responsible works Company	Cada responsable trabaja para una empresa.	1..* : 1
RRHH_contact works Company	Una empresa tiene un contacto de recursos humanos.	1 : 1
Company presents Offer	Cada oferta está presentada por una empresa.	1 : 0..*

University

Esta entidad representa al conjunto de universidades catalanas que participan en el proceso de realización de prácticas en empresas. Las universidades reciben las ofertas de las empresas y son las encargadas de publicarlas a los alumnos.

Atributos:

Atributo	Tipo	Descripción
CIF <P>	String	El código de identificación fiscal único del que disponen todas las personas jurídicas.
Name	String	El nombre de la universidad.
Direction	String	La dirección fiscal de la empresa.

Relaciones:

E1 <Relación> E2	Descripción	Cardinalidad
University offers Degree	Cada universidad ofrece a sus estudiantes una serie de titulaciones universitarias.	1..* : 1..*
Student attends University	Cada estudiante está matriculado en una universidad.	1..* : 1
Professor works University	Cada profesor trabaja en una universidad.	1 : 1..*
University receives Offers	Cada oferta está emitida para una universidad.	1 : 0..*

Degree

Esta entidad representa al conjunto de universidades catalanas que participan en el proceso de realización de prácticas en empresas. Las universidades reciben las ofertas de las empresas y son las encargadas de publicarlas a los alumnos.

Atributos:

Atributo	Tipo	Descripción
Code <P>	String	Código de identificación único con el que se guarda la titulación dentro del sistema.
Name	String	El nombre de la titulación.
Area	Area	El nombre del área de estudios que engloba esta titulación.

Relaciones:

E1 <Relación> E2	Descripción	Cardinalidad
University offers Degree	Cada universidad ofrece a sus estudiantes una serie de titulaciones universitarias.	1..* : 1..*
Student studies Degree	Cada estudiante está estudiando una carrera.	1..* : 1

Offer

Esta entidad representa el conjunto de ofertas que una empresa emite. Cada empresa emite diferentes ofertas a cada una de las universidades.

Estas ofertas son revisadas por los alumnos que, posteriormente, decidirán o no presentar una candidatura.

Atributos:

Atributo	Tipo	Descripción
idOffer <P>	String	La identificación única que cada oferta tiene dentro del sistema.
Name	String	El nombre de la oferta.
Description	String	La descripción de la oferta.
Publish_date	Date	La fecha en que la oferta es publicada.

Relaciones:

E1 <Relación> E2	Descripción	Cardinalidad
Company presents Offer	Cada oferta está presentada por una empresa.	1 : 0..*
University receives Offers	Cada oferta está emitida para una universidad.	1 : 0..*
Offer has Candidacy	Cada candidatura está emitida para una oferta determinada.	1 : 0..*

Candidacy

Esta entidad representa el conjunto de candidaturas que son presentadas por los alumnos. Estas candidaturas pasan por varios estados diferentes. Inicialmente son publicadas por los estudiantes, revisadas por los contactos de RRHH, validadas por los responsables y, finalmente, pasan a una entrevista virtual si llegan al último paso.

Atributos:

Atributo	Tipo	Descripción
idCandidacy <P>	String	La identificación única que la candidatura tiene en el sistema.
Valid	Boolean [0..1]	Indica si ha sido aceptada o no por el responsable.

		Puede ser NULL ya que inicialmente este campo no tiene valor.
Validation_date	Date [0..1]	La fecha en que la candidatura es validada. Este campo puede ser NULL ya que no se decide en primera instancia.
State	State	Esta clase enumerada representa el estado actual de una candidatura, lo que permitirá conocer en qué estado se encuentra
Relaciones:		
E1 <Relación> E2	Descripción	Cardinalidad
Student presents Candidacy	Cada candidatura está presentada por un estudiante.	1 : 0..*
Responsible validates Candidacy	Cada candidatura es validada por un responsable.	1 : 0..*
Offer has Candidacy	Cada candidatura está emitida para una oferta determinada.	1 : 0..*
Candidacy includes Candidacy_requisites	Cada candidatura incluye una serie de requisitos formales para presentarse.	1 : 1..*
Candidacy has Interview	Cada candidatura puede llegar a tener una entrevista virtual.	1 : 0..1

Candidacy_requisites

Esta entidad representa el conjunto de requisitos que incluye cada candidatura. Se ha modelado como una entidad a parte ya que la lista de requisitos es diferente para cada oferta. Además, cada candidatura puede tener más de un requisito, por lo que es necesario gestionarle como una nueva entidad para respetar la atomicidad de la base de datos.

Atributos:

Atributo	Tipo	Descripción
idRequisite <P>	String	El identificador interno que los requisitos tienen dentro de la base de datos.
Requisite	String	Este campo puede incluir diferentes tipos de cadenas de caracteres, como una pequeña presentación del estudiante o el enlace a un video grabado y colgado en Internet.

Relaciones:

E1 <Relación> E2	Descripción	Cardinalidad
Candidacy includes Candidacy_requisites	Cada candidatura incluye una serie de requisitos formales para presentarse.	1 : 1..*

Interview

Esta entidad representa a las entrevistas virtuales que se realizan al final del proceso de validación de candidaturas. Estas entrevistas son gestionadas por la persona de RRHH y son el último paso antes de formalizar el contrato de prácticas.

Atributos:

Atributo	Tipo	Descripción
idInterview <P>	String	El identificar interno que cada entrevista tiene dentro de la BD.
Interview_date	Date	La fecha en la que se realizará la entrevista.

Valid	Boolean [0..1]	Se indica si finalmente se realizará la colaboración. Puede ser NULL ya que no se decide hasta que la entrevista tiene lugar.
Relaciones:		
E1 <Relación> E2	Descripción	Cardinalidad
RRHH_contact makes Interview	Cada entrevista es concertada por un contacto de recursos humanos.	1 : 0..*
Candidacy has Interview	Cada candidatura puede llegar a tener una entrevista virtual.	1 : 0..1
Interview includes Interview_comments	Cada entrevista dispone de una serie de comentarios realizados por el entrevistador.	1 : 1..*

Interview_comments		
Esta entidad representa a los comentarios que los responsables de RRHH efectúan sobre las entrevistas. Se modela como una entidad a parte ya que, para cada entrevista, pueden realizarse más de un comentario.		
Atributos:		
Atributo	Tipo	Descripción
idComment <P>	String	El identificador interno que cada comentario tiene dentro de la BD.
Comment	String	El comentario realizado por el contacto de RRHH encargado de la entrevista.
Relaciones:		
E1 <Relación> E2	Descripción	Cardinalidad
Interview includes Interview_comments	Cada entrevista dispone de una serie de comentarios realizados por el entrevistador.	1 : 1..*

Collaboration		
Esta entidad representa al conjunto de colaboraciones que se realizan entre una estudiante y una empresa. Cada una de estas colaboraciones son un contrato no laboral con una duración y remuneración variable.		
Para cada una de estas colaboraciones se les especifica un programa formativo que es supervisado por un responsable de la empresa, que tendrá que emitir diferentes informes a la universidad. Estos informes son los que se realizan de forma periódica y el informe final.		
Además, cada colaboración puede recibir inspecciones de trabajo para garantizar que no se trata de un puesto de trabajo encubierto.		
Atributos:		
Atributo	Tipo	Descripción
idCollaboration <P>	String	El código interno que cada colaboración tiene dentro de la BD.
Init_date	Date	La fecha de inicio de la colaboración.
Duration	Integer	El número de días que durará la colaboración.
Remuneration	Float	La remuneración, si tiene, que recibirá el estudiante.
Periodic_report_periodicity	Integer	El número de días que pasan entre la emisión de cada informe periódico. Se trata de un valor diferente para cada empresa y universidad, por eso se modela como un atributo a parte.

Relaciones:		
E1 <Relación> E2	Descripción	Cardinalidad
Student makes Collaboration	Cada colaboración de prácticas en empresas está realizada por un estudiante.	1 : 0..1
Responsible supervises Collaboration	Cada colaboración es supervisada por un responsable.	1 : 0..*
Professor supervises Collaboration	Cada alumno en prácticas está siendo tutorizado por un profesor.	1 : 0..*
Collaboration includes Collaboration_themes	Cada colaboración tiene una lista de temas que forman el programa formativo.	1 : 1..*
Collaboration presents Periodic_report	Cada colaboración debe presentar a la universidad una serie de informes periódicos.	1 : 0..*
Collaboration concludes Final_report	Cada colaboración concluye en un informe final.	1 : 0..1
Collaboration passes Inspection	Cada colaboración puede pasar diferentes inspecciones de trabajo.	1 : 0..*

Collaboration_themes

Esta entidad representa el conjunto de temas que forman parte de los diferentes programas formativos de cada colaboración. El aprendizaje y cumplimiento de estos temas son lo que permite al estudiante superar las prácticas.

Atributos:

Atributo	Tipo	Descripción
idTheme <P>	String	El código interno que cada tema tiene dentro de la BD.
Name	String	El nombre de cada tema.
Description	String	La descripción de cada tema, indicando que aprenderá el alumno y que se espera de él.

Relaciones:

E1 <Relación> E2	Descripción	Cardinalidad
Collaboration includes Collaboration_themes	Cada colaboración tiene una lista de temas que forman el programa formativo.	1 : 1..*

Report

Esta entidad representa el conjunto de informes que las empresas emitirán sobre los estudiantes.

Se trata de una entidad abstracta, pues cada informe deberá pertenecer a una de las subclases que heredan de él.

Atributos:

Atributo	Tipo	Descripción
idReport <P>	String	El código interno que cada informe tiene dentro de la BD.
Report_date	Date	La fecha de emisión del informe.

Report > Periodic_report

Esta entidad representa el subconjunto de informes que se emiten de forma periódica por parte de los responsables y cuyo objetivo es ir informando del desempeño de los alumnos.

La periodicidad de estos informes varía para cada empresa y universidad, por lo que se indica como un

atributo dentro de la entidad colaboración.		
Atributos:		
Atributo	Tipo	Descripción
Comment	String	El comentario que se incluye en el informe periódico.
Relaciones:		
E1 <Relación> E2	Descripción	Cardinalidad
Collaboration presents Periodic_report	Cada colaboración debe presentar a la universidad una serie de informes periódicos.	1 : 0..*

Report > Final_report		
Esta entidad representa el subconjunto de informes que son el informe final que se emite a la finalización de una colaboración. Este informe debe incluir la valoración subjetiva del responsable y también si se le considera apto o no.		
Atributos:		
Atributo	Tipo	Descripción
Valoration	String	El comentario final que se incluye en el informe.
Suitable	Boolean	Indica si el estudiante es considerado apto por parte del responsable que lo ha supervisado.
Relaciones:		
E1 <Relación> E2	Descripción	Cardinalidad
Collaboration concludes Final_report	Cada colaboración concluye en un informe final.	1 : 0..1
Final_report includes Final_report_comments	Cada informe final puede tener una serie de comentarios realizados por los implicados en el proceso.	1 : 0..*

Final_report_comments		
Esta entidad representa el conjunto de comentarios que diferentes personas pueden dejar sobre cada informe final, para plasmar así sus opiniones sobre todo el proceso. La existencia de esta entidad es la que provoca que sea necesario modelar a <i>Person</i> como una superclase de todas las personas involucradas, pues todas ellas pueden dejar comentarios.		
Atributos:		
Atributo	Tipo	Descripción
IdComment <P>	String	El código interno y único que cada comentario tiene dentro de la BD.
Comment	String	El texto del comentario.
Relaciones:		
E1 <Relación> E2	Descripción	Cardinalidad
Final_report includes Final_report_comments	Cada informe final puede tener una serie de comentarios realizados por los implicados en el proceso.	1 : 0..*
Person makes Final_report_comment	Cada comentario realizado en el informe final de prácticas es realizado por una persona involucrada en el proceso.	1 : 0..*

Inspection		
Esta entidad representa el conjunto de inspecciones que se realizarán sobre cada colaboración para		

determinar si se trata de un puesto de trabajo encubierto.		
Atributos:		
Atributo	Tipo	Descripción
idInspection <P>	String	El código interno que cada inspección tiene dentro de la BD.
Inspection_date	Date	La fecha en la que se ha realizado la inspección.
Passed	Boolean	En caso de ser False, se consideraría que la empresa no ha pasado la inspección sobre este contrato de prácticas en concreto.
Relaciones:		
E1 <Relación> E2	Descripción	Cardinalidad
Collaboration passes Inspection	Cada colaboración puede pasar diferentes inspecciones de trabajo.	1 : 0..*

Una vez descritas todas las clases en detalle, creo que es interesante hacer un breve resumen sobre las decisiones de diseño tomadas durante esta fase del proyecto y como tendrán un mayor sentido en las próximas etapas:

- He decidido que la clase Collaboration sea prácticamente el núcleo de la BD ya que sirve como punto de unión perfecto entre todas las partes. Creo que su existencia reduce el número de relaciones que deberían existir entre el resto de las clases.
- He decidido crear una superclase *Person* para RRHH_contact, Responsable, Student y Professor únicamente porque se menciona que en el reporte final tras las prácticas cualquier involucrado puede aportar comentarios sobre el proceso. Esto implica que, sin la existencia de *Person*, la clase Final_report_comments debería estar unida a las cuatro clases que representan a los involucrados.
- Se ha decidido añadir el atributo State a Candidacy ya que el enunciado se hace mucho hincapié en todas las etapas por las que pasa una candidatura. Esto permitirá poder obtener estadísticas sobre estas diferentes etapas, permitiendo analizar, por ejemplo, cual suele ser el principal motivo por el cual una candidatura no prospera. Concretamente, el estado *presented* representa el momento en que el alumno la ha presentado, el estado *validated* es cuando el personal de RRHH ha visto que la candidatura reúne los requisitos especificados, Interview es cuando el responsable de la empresa marca la candidatura para recibir una entrevista y, por último, *accepted* y *rejected* son los pasos finales que una candidatura puede tener.

Repositorio Estadístico

Lo siguiente que queremos modelar en nuestra base de datos será el repositorio estadístico. En el enunciado se menciona que queremos que tenga un tiempo constante de consulta 1.

Para ello, el enfoque será crear una serie de tablas donde almacenar el resultado final de las consultas que responderán a este repositorio. La idea detrás de esto es que, ante cada operación de inserción, modificación y eliminación, se produzcan disparadores que actualicen los valores del repositorio.

Sin embargo, el repositorio tiene una condición es que todo su tratamiento debe hacerse en tiempo constante ¹ sin utilizar funciones de agregación y agrupamiento. Para implementarlo, lo haré mediante contadores y operaciones simples que se irán actualizando ante cualquier inserción de datos.

Para poder llevar un control de estos contadores y valores intermedios será necesario también crear una serie de tablas auxiliares donde se irán almacenando. Por ejemplo, una tabla que, entre otras cosas, almacena un contador que se incrementa cada vez que agregamos una universidad para poder mantener un control del número de universidades sin tener que utilizar la función de agregación *count()*.

En resumen, existirán las tablas que almacenan los resultados finales del repositorio estadístico y las tablas que almacenarán los valores con los que se calculan los indicadores del repositorio.

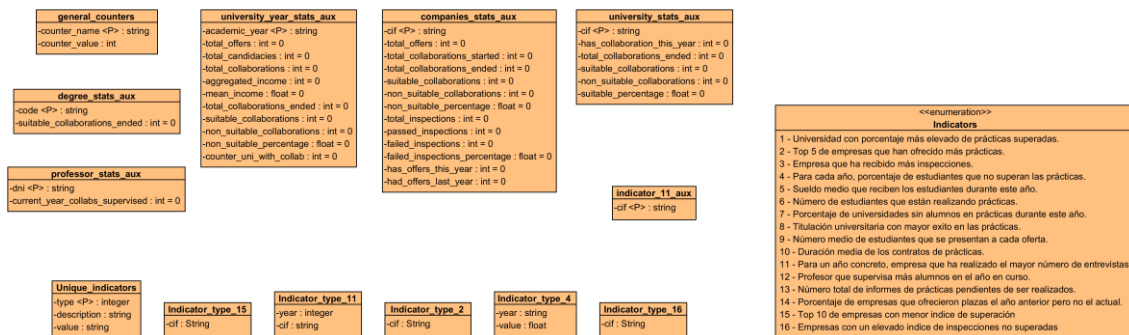
Respecto a las tablas que almacenan los indicadores, se ha observado que existen diferentes tipos de indicadores y, por lo tanto, deberán guardarse en diferentes tipos de tablas. Por ejemplo, hay indicadores que son simplemente un número y otros indicadores que son una lista de empresas.

Con esta idea en mente, podemos identificar tres tipos de estadísticas distintas, a saber:

- Estadísticos únicos que solamente son un valor, como por ejemplo un porcentaje, el DNI de un profesor o el CIF de una empresa. Todos estos estadísticos pueden almacenarse en una única tabla a razón de una fila por estadístico.
- Estadísticos que deben guardar información temporal. Para estos la solución escogida es crear una tabla individual por estadístico donde en una columna se guardará la información del año académico y en la otra el valor del estadístico. Esto permitirá hacer la consulta para cualquier año.
- Estadísticos que deben devolver una lista, como por ejemplo las empresas con más estudiantes en prácticas. Para estos se utilizará también una tabla para cada uno de ellos en el que cada fila será un miembro de la lista que se debe devolver.

Antes de pasar a mostrar el diseño conceptual del repositorio estadístico querría comentar que, finalmente, se ha decidido incluir los dos indicadores estadísticos que se han propuesto de manera adicional en el punto 2.3 de esta memoria.

Finalmente, el diseño conceptual del Repositorio Estadístico queda de la siguiente manera:



Las entidades que de aquí se desprenden son las siguientes:

general_counters			
En esta tabla se guardarán contadores generales sobre el sistema. Por ejemplo, el número total de universidades que hay registradas. Los contadores se insertan como filas.			
Atributos:			
Atributo	Tipo	Descripción	
counter_name <P>	string	El nombre del contador.	
counter_value	int	El valor del contador. Siempre inicializado a 0.	

university_year_stats_aux			
En esta tabla se guardarán los contadores y valores relacionados con los años académicos. Por ejemplo, el número de ofertas publicadas en un año determinado. Cada fila es un año y los contadores y valores las columnas.			
Atributos:			
Atributo	Tipo	Descripción	
academic_year <P>	string	El año académico en cuestión.	
total_offers	int	El número total de ofertas.	
total_candidacies	int	El número total de candidaturas.	
total_collaborations	int	El número total de colaboraciones.	
aggregated_income	int	La suma de los salarios.	
mean_income	int	La media de los salarios.	
total_collaborations_ended	int	El total de colaboraciones finalizadas.	
suitable_collaborations	int	Colaboraciones exitosas finalizadas.	
non_suitable_collaborations	int	Colaboraciones suspendidas finalizadas.	
non_suitable_percentage	int	El porcentaje de colaboraciones suspendidas.	
counter_uni_with_collab	int	El total de universidades con colaboración este año.	

companies_stats_aux

En esta tabla se guardarán los contadores y valores relacionados con las compañías. Por ejemplo, el total de ofertas publicadas por una compañía.

Atributos:

Atributo	Tipo	Descripción
cif <P>	string	Cif de la empresa.
total_offers	int	Total de ofertas presentadas.
total_collaborations_started	int	Total colaboraciones iniciadas.
total_collaborations_ended	int	Total colaboraciones finalizadas.
suitable_collaborations	int	Del total, solo las aprobadas.
non_suitable_collaborations	int	Del total, solo las suspendidas.
non_suitable_percentage	int	El porcentaje de colaboraciones suspendidas.
total_inspections	int	El total de inspecciones recibidas.
passed_inspections	int	Las inspecciones pasadas.
failed_inspections	int	Las inspecciones fallidas.
failed_inspections_percentage	int	El porcentaje de inspecciones fallidas.
has_offers_this_year	int	Indica si ha publicado ofertas este año.
had_offers_last_year	int	Indica si publicó ofertas el año pasado.

university_stats_aux

En esta tabla se guardarán los contadores y valores relacionados con las universidades. Por ejemplo, las colaboraciones que ha tenido.

Atributos:

Atributo	Tipo	Descripción
cif <P>	string	Cif de la universidad.
has_collaboration_this_year	int	Indica si este año tiene al menos una colaboración.
total_collaborations_ended	int	El total de colaboraciones que han finalizado.
suitable_collaborations	int	Las colaboraciones aprobadas.
non_suitable_collaborations	int	Las colaboraciones suspendidas.
suitable_percentage	int	El porcentaje de colaboraciones aprobadas.

degree_stats_aux

En esta tabla se guardarán los contadores y valores relacionados con los grados universitarios.

Atributos:

Atributo	Tipo	Descripción
code <P>	string	El código de los estudios.
suitable_collaborations_ended	int	El número de colaboraciones aprobadas.

professor_stats_aux

En esta tabla se guardarán los contadores y valores relacionados con los profesores. Por ejemplo, el número de colaboraciones que supervisa.

Atributos:

Atributo	Tipo	Descripción
dni <P>	string	Dni del profesor.
current_year_collabs_supervised	int	Total de colaboraciones supervisadas.

indicator_11_aux

Esta tabla es exclusiva para dar soporte al indicador 11 ya que en ella debemos guardar información para cada compañía y para cada año académico. La idea es que cada fila represente una empresa y cada columna un año académico. En su declaración, esta tabla solo tiene una sola columna que es la del CIF de la compañía. De forma dinámica y cada vez que se inserte un nuevo año académico en el sistema, se añadirá una nueva columna representado ese año. Lo que se guardará en esta tabla es el número total de entrevistas realizadas por cada empresa en cada año universitario.

Atributos:

<i>Atributo</i>	<i>Tipo</i>	<i>Descripción</i>
cif <P>	string	Cif de la compañía.

Unique_indicators

Esta entidad representa el conjunto de indicadores estadísticos que son únicos y que pueden almacenarse juntos en una única tabla. Cuando se quiera hacer una consulta sobre ellos, simplemente habrá que indicar en WHERE cuál es el indicador que queremos obtener.

Atributos:

<i>Atributo</i>	<i>Tipo</i>	<i>Descripción</i>
Type <P>	Integer	El número del indicador al que pertenece esa fila. Puede verse la relación en el tipo enumerado Indicators.
Description	String	Se trata del texto que explica el indicador, puede verse la relación en el tipo enumerado Indicators.
Value	String	El resultado que esperamos obtener. Es tipo String ya el resultado siempre se almacenará como este tipo, cuando se trate de un entero o un real, simplemente se le cambiará el tipo para poder guardarlo.

Indicator_type_2

Esta entidad representa el TOP 5 de empresas que han ofrecido más prácticas al CIC. Los disparadores se encargarán de que siempre haya un máximo de cinco filas y que las filas siempre estén ordenadas en orden descendiente. Por lo que no habrá que usar ni LIMIT ni WHERE. Solo SELECT * FROM Indicator_type_2.

Atributos:

<i>Atributo</i>	<i>Tipo</i>	<i>Descripción</i>
Name	String	El nombre de la empresa.

Indicator_type_4

En esta entidad se almacenarán los porcentajes de estudiantes que no superan las prácticas para cada año universitario. Por lo que cada fila representará un año universitario en concreto. Un simple SELECT usando WHERE para determinar el año universitario devolverá el valor esperado.

Atributos:

<i>Atributo</i>	<i>Tipo</i>	<i>Descripción</i>
Year	String	El año académico en concreto. Tiene el formato YYYY-YYYY. Por ejemplo, 2022-2023.
Value	Float	El porcentaje de estudiantes que no ha superado las prácticas ese año.

Indicator_type_11

En esta entidad se almacenarán el nombre de la empresa que realiza el mayor número de entrevistas para cada año. Cada fila representará un año y en ella se guardará el nombre de la empresa que hizo más entrevistas. De nuevo, se usará WHERE para seleccionar el dato que queramos.

Atributos:

<i>Atributo</i>	<i>Tipo</i>	<i>Descripción</i>
Year	String	El año académico en concreto. Tiene el formato YYYY-YYYY. Por ejemplo, 2022-2023.
Name	String	El nombre de la empresa con más entrevistas realizadas.

Indicator_type_15

Esta relación mostrará el Top 10 de empresas cuyo índice de superación de las prácticas es el más bajo. Esto servirá al CIC para poder ver si hay empresas que tienen este índice muy por debajo de la media, pues puede estar indicando que hay más motivo más allá del desempeño de los estudiantes.

Atributos:

<i>Atributo</i>	<i>Tipo</i>	<i>Descripción</i>
Name	String	El nombre de la empresa.

Indicator_type_16

Esta entidad mostrará a todas aquellas empresas cuyo porcentaje de no superación de las inspecciones de trabajo sea superior o igual al 10%, representando así a aquellas empresas que están haciendo un uso fraudulento de las prácticas y que, por lo tanto, deberían dejar de trabajar con el CIC durante al menos el próximo año.

Atributos:

<i>Atributo</i>	<i>Tipo</i>	<i>Descripción</i>
Name	String	El nombre de la empresa.

Almacenamiento de LOGs

Por último, en nuestra BD también almacenaremos los LOGs resultantes de cada una de las acciones realizadas sobre la BD, como consultas, inserciones y todo tipo de procedimientos.

LOG_Stored
-idLog <P> : string
-log_date : date
-process : string
-entry : String
-return : string

En este caso, se trata de una sola entidad en la que se almacenará toda la información relativa a los LOGs. De este modo, siempre se podrá consultar cual es el resultado de los procedimientos ejecutados sobre la BD y poder identificar así posibles problemas de funcionalidad.

LOG_Stored		
Esta entidad representa el conjunto de LOGs almacenados en la base de datos que se han generado como resultado de la ejecución de los diferentes procesos que la manejan.		
Atributos:		
Atributo	Tipo	Descripción
idLog <P>	String	El identificador único de cada log. Se generará de forma automática y secuencial para simplificar.
Log_date	Date	La fecha y hora en la que se ha generado el log.
Procces	String	El nombre del proceso que ha generado este log, ya que así servirá para identificar que procesos pueden no funcionar como lo esperado.
Entry	String	Los parámetros de entrada que ha recibido el proceso.
Return	String	El parámetro de salida devuelto por el proceso, en el que se indica si ha sido exitoso o si ha habido un error.

3.2 Diseño lógico

Ahora que ya hemos terminado de construir los cimientos sobre los que se edificará el resto del proyecto, toca dar el siguiente paso y convertir nuestro diseño conceptual en un diseño lógico y, al finalizar, dispondremos de un modelo lógico relacional con el que ya podremos implementar nuestra BD.

Esta etapa intermedia es un punto crucial en el diseño de bases de datos ya que, aunque el diagrama UML que hemos obtenido es una gran herramienta para representar una parte del mundo real, no permite una conversión directa y sencilla hacia una implementación física a través de un SGBD.

Este proceso nos servirá para identificar trampas y errores de diseño en los que hayamos caído a la hora de crear el esquema conceptual a partir del enunciado del proyecto. Además, nos servirá para definir como implementaremos en nuestra BD relacional las diferentes relaciones que existen entre las clases que hemos definido, así como algunas de las restricciones del sistema.

Un punto importante para tener en cuenta es que a partir de ahora y mientras estemos en el ámbito del diseño lógico, llamaremos relación a la unidad básica del modelo relacional. Hasta ahora una relación o tipo de relación era aquello que unía dos clases diferentes, por ejemplo: “*Student attends University*”.

A partir de ahora una relación es un conjunto de atributos que servirán para representar los objetos del mundo real que ya hemos modelado en el diagrama UML y la forma en la que se relacionan entre ellos. Nuestro conjunto de relaciones formará el modelo relacional resultante de esta etapa.

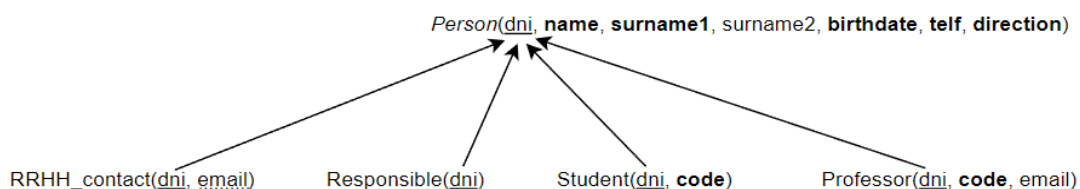
No está de más decir que, al final, una Base de Datos Relacional está compuesta por datos almacenados y relacionados entre sí, por lo que sin duda describir de forma correcta como estos datos se relacionan es un punto muy importante en las primeras etapas de diseño.

Para que sea más sencillo de explicar y entender, primero mostraré fragmentos del modelo relacional y como se han obtenido para, finalmente, mostrar el esquema relacional final.

La superclase *Person* y sus subclases

Hay tres formas de representar en el modelo relacional una generalización o especialización. Cada una tiene sus ventajas e inconvenientes y es importante elegir de forma correcta aquella que será la mejor para nuestros intereses. O aquella que simplemente nos dará menos problemas.

En el caso de *Person* y las subclases RRHH_contact, Responsable, Student y Professor he elegido la siguiente conversión:



He elegido esta forma de representar esta generalización debido a que, como ya se ha mencionado anteriormente, es importante que exista la tabla *Person* en la BD ya que cualquier persona, sin importar su subclase, podrá hacer comentarios en el reporte final de una colaboración.

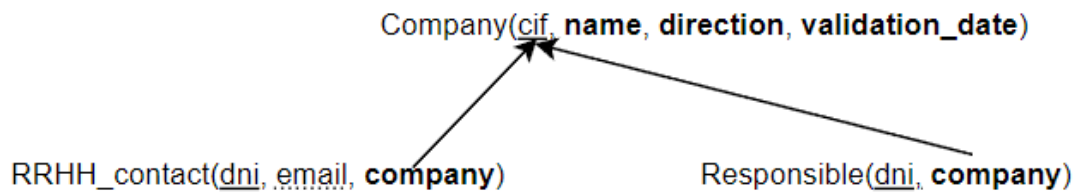
Los aspectos más relevantes en esta representación es que el DNI de cada persona es su Clave Primaria y, en el caso de las subclases, también es la Clave Foranea que lo relaciona con *Person*.

Los atributos que están en negrita son aquellos que no pueden aceptar valores nulos, por eso el segundo apellido es el unico atributo que puede aceptar valores nulos. En el caso de las Claves Candidatas, no hace falta ponerlas en negrita pues se entiende que no son nulos. Aunque existe SGBD que permiten la existencia de valores nulos en columnas con restricción *Unique*, no será el caso durante este proyecto.

En ese sentido, he tenido en cuenta que el email es una clave candidata pues no pueden repetirse. Sin embargo, el código interno que cada universidad le asigna a

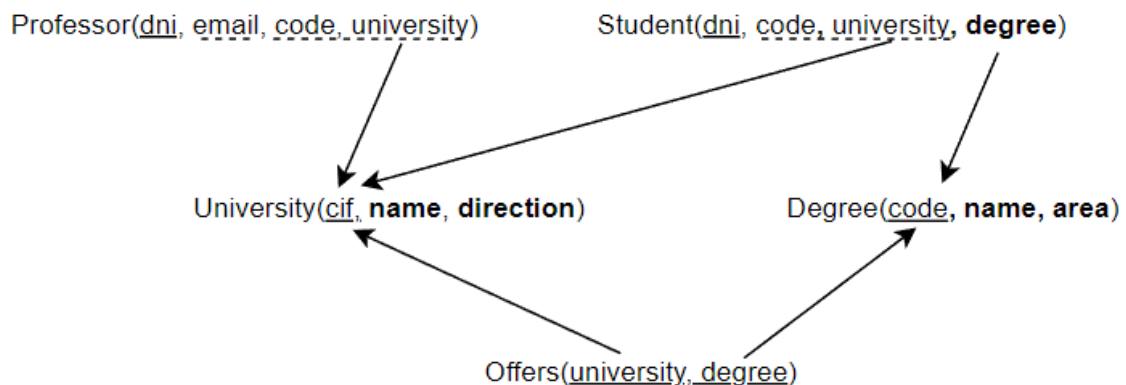
sus estudiantes y profesores si que podría llegar a repetirse entre diferentes universidades, por lo que no puede ser clave candidata de la relación.

La empresa y sus trabajadores



Aquí podemos ver como a las relaciones RRHH_contact y Responsable les hemos añadido un nuevo atributo llamado Company que es clave foránea de las empresas en las que trabaja.

La universidad y su entorno

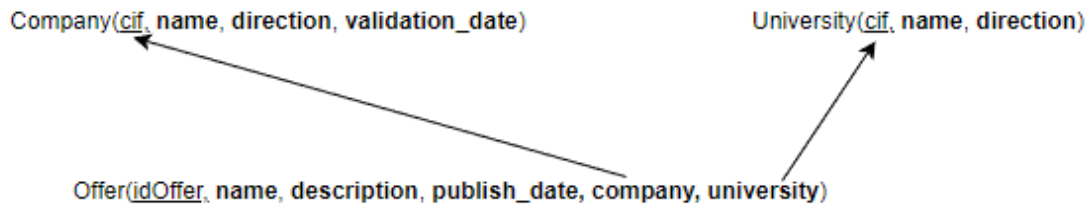


En este caso vemos las claves foráneas que hemos añadido a Profesor y Student para representar sus relaciones con University y Degree. Otro aspecto muy relevante es la nueva relación que hemos creado llamada Offers y cuyo objetivo es representar la relación *many-to-many* que existe entre Degree y University.

La combinación de la universidad y el grado es la Clave Primaria de Offers y, cada atributo por separado es la clave foránea que sirve para identificar la universidad y el grado universitario.

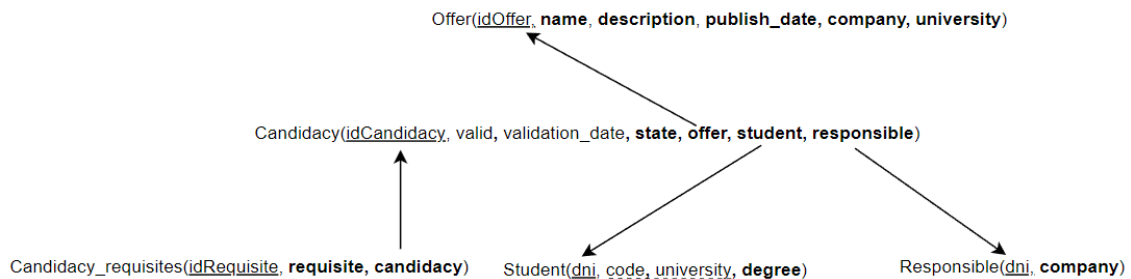
Un aspecto interesante es como ahora, al haber incluido la información de la Universidad, ese atributo *code* pasa a convertirse en Clave Candidata en el momento en que la juntamos con *university*, pues una Universidad no asignaría el mismo código a dos estudiantes o profesores diferentes.

Publicando ofertas en las universidades



La forma más sencilla de representar las ofertas es darse cuenta de que cada oferta tiene únicamente una empresa y únicamente una universidad, por lo que añadimos *company* y *university* como atributos a la relación *Offer* y luego las convertimos en claves foráneas.

Un estudiante presenta su candidatura



En este caso disponemos de muchas relaciones *one-to-many* en *Candidacy*, por lo que lo mejor es ir añadiendo atributos a esta relación para convertirlos en claves foráneas que se vinculen a los estudiantes, responsables y ofertas. La única excepción son los requisitos, que al ser una composición la clave foránea debe estar en la relación *Candidacy_requisites*.

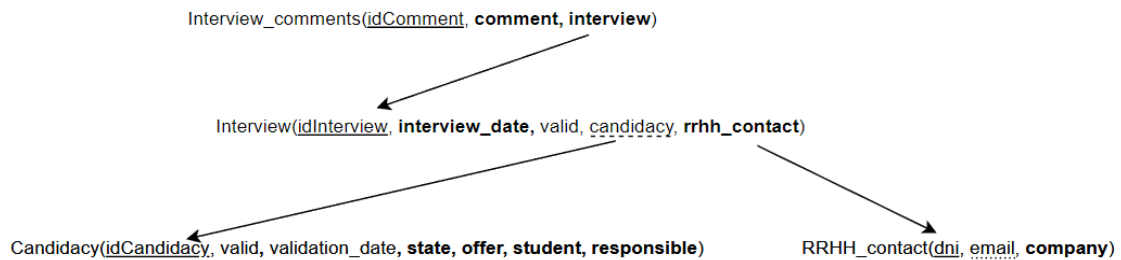
Se podría pensar que la combinación de *offer*, *student* y *responsible* podría ser una clave candidata de la relación. Sin embargo, he considerado que un mismo estudiante podría presentar más de una candidatura a una misma oferta si, por ejemplo, no cumpliera inicialmente los requisitos y si lo hiciera de forma posterior.

Aquí vemos que los atributos *valid* y *validation_date* pueden aceptar valores nulos, pues en el momento en que se presenta la candidatura todavía no se ha decidido si es válida o no. Estos atributos pueden parecer que no son interesantes al disponer del atributo *state*.

Sin embargo, juegan un papel importante a la hora de saber que ha pasado con la candidatura. Por ejemplo, si el estado de la candidatura es Rechazado pero el valor de *valid* es False, sabemos que la candidatura se rechazó al no cumplir los

requisitos. Por el contrario, si *valid* fuera True, sabremos que la candidatura se rechazó por parte del responsable o durante la entrevista con RRHH.

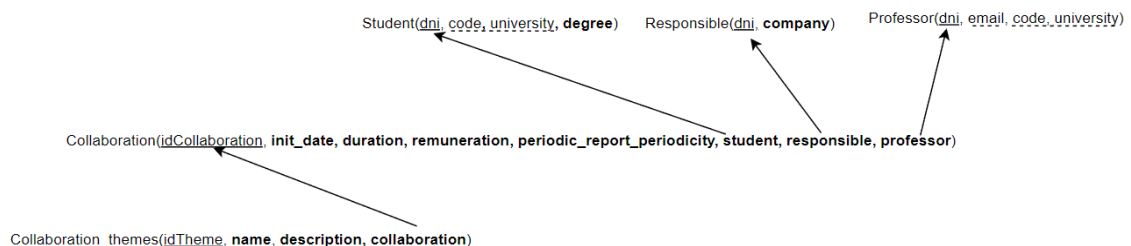
La candidatura deriva en una entrevista



De nuevo, aprovechamos que en las relaciones una de las multiplicidades es 1 para definir cual es el mejor sitio para establecer la clave foránea. En el caso de Interview y Candidacy, he elegido poner la clave foránea en Interview ya que una entrevista siempre tendrá una candidatura, mientras que una candidatura puede no tener una entrevista en caso de no haber pasado las etapas previas. De haber elegido que es la candidatura la que alojaría la clave foránea, entonces tendríamos valores nulos que es mejor evitar.

Ya que una entrevista solo puede tener una candidatura y una candidatura solo puede tener una entrevista, la Clave Foranea *candidacy* es, también, una Clave Candidata.

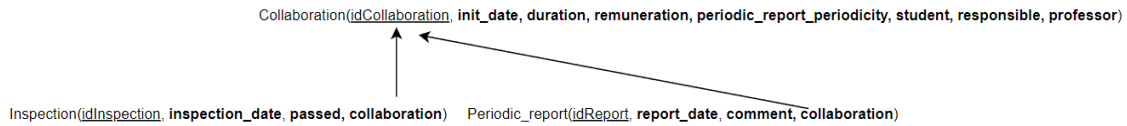
Se formaliza la colaboración



Una vez más, nos aprovechamos de que cada colaboración tiene únicamente un estudiante, un responsable y un profesor para definir las claves foráneas que servirán para relacionar todos estos conceptos.

Por otro lado, Collaboration_themes utiliza una clave foránea para representar la relación de composición que tiene con Collaboration.

La colaboración recibe seguimientos periódicos

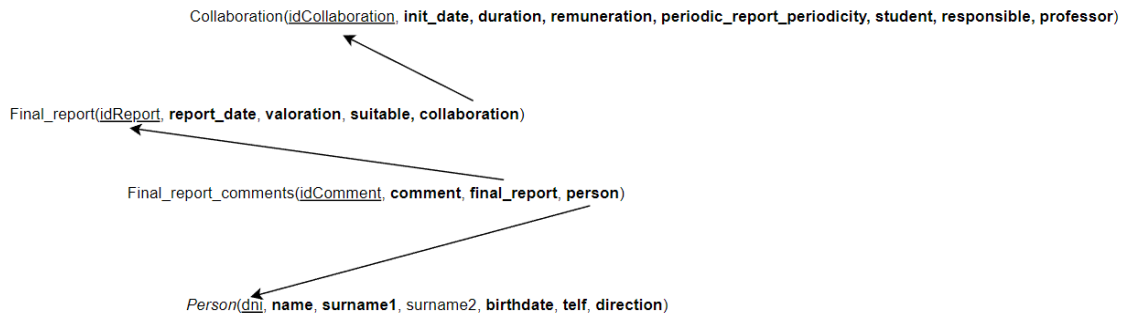


Cada colaboración recibirá inspecciones periódicas y emitirá reportes periódicos para hacer seguimiento. Cada inspección y cada reporte o informe tendrá una sola colaboración, por lo que la elección de la clave foránea es sencilla.

Un aspecto importante a notar aquí es que, en este caso, para la generalización que existe entre Report, Periodic_report y Final_report he optado por la opción de crear una relación para cada subclase e incluir en cada una de ellas todos los atributos de la superclase.

Debido a que la BD, a diferencia de la generalización de *Person*, no necesita interactuar de ninguna forma con la superclase *Report*, está será la mejor forma de representarlo.

La colaboración termina



Finalmente, aquí podemos ver como termina una colaboración. Esta dispone de un Final_report que contiene una clave foránea hacia Collaboration. Así mismo, podemos ver como cada comentario del informe final tiene dos claves forneas, la que apunta hacia el informe final y la que apunta hacia la persona que ha hecho dicho comentario.

Esquema Logico Relacional resultante

A continuación y antes de pasar al siguiente paso, uniremos todos los fragmentos del modelo para presentarlos, finalmente, en un unico esquema lógico relacional que representa a todo nuestro proyecto:

Debido al elevado número de claves foráneas, el modelo final puede ser algo confuso y, sin duda, es mejor revisarlo a partir de los esquemas parciales que se han ido mostrando en las páginas anteriores.

Se ha decidido no incluir en el diseño lógico al repositorio estadístico y los *logs* ya que en el diseño conceptual no disponen de ninguna relación entre sus clases y no aportaría nueva información ni nos permitiría descubrir trampas de diseño

Por lo tanto, con todo esto ya hemos finalizado nuestro Modelo Lógico Relacional y ya estamos listos para llevarlo a nuestro SGBD de Oracle con el que construiremos finalmente la BD.

3.3 Normalización

Para acabar con la fase de diseño, realizaremos un análisis sobre cada una de las relaciones que hemos definido en el punto anterior para identificar en que forma normal se encuentran y, por lo tanto, identificar posibles fallos o trampas de diseño que hayamos podido cometer.

Una correcta normalización sirve para acabar con la redundancia de datos y para garantizar la separación de conceptos con el objetivo de evitar futuras anomalías de actualización de los datos.

Recordemos que el simple hecho de poder construir una BD en un SGBD como Oracle ya garantiza que todas las relaciones están, al menos, en primera forma normal, pues es la que garantiza que la información representada en cada atributo de la relación es atómico o, lo que es lo mismo, que no son una relación, descomponibles ni con multiplicidad de valores.

- *Person*(dni, name, surname1, surname2, birthdate, telf, direction)

Esta relación se encuentra en Forma Normal de Boyce-Codd (FNBC). En este caso ha sido un análisis muy sencillo debido a que, en realidad, solo hay una clave candidata y solo hay una dependencia que es la siguiente:

{dni} → {name, surname1, surname2, birthdate, telf, direction}

Ningún otro atributo o grupo de atributos es Clave Candidata ya que, por raro que pudiera ser, podría haber dos personas que tuvieran todos los atributos iguales, siendo el DNI el único que es realmente único y no puede repetirse. Se podría considerar al teléfono como una clave candidata, pero teniendo en cuenta que el

teléfono puede cambiar con el tiempo o dos o más personas pueden compartir un mismo teléfono fijo, es mejor seguir usando el DNI como identificador único.

Por lo tanto, todos los atributos que no forman parte de una clave candidata dependen completamente de la única clave candidata que existe. Ninguno de los atributos que no forma parte de una clave candidata depende de un conjunto de atributos que contiene alguno que no forma parte de una clave candidata y, por último, los determinantes de todas las dependencias de la relación son claves candidatas de esta.

Esta situación se repite con la mayoría de las relaciones que hemos establecido, por lo que en lugar de analizar las 20 relaciones lo que haré a continuación será mencionar aquellas que tienen alguna característica o diferencias dignas de mencionar.

- Student(dni, code, university, **degree**)

Como ya dijimos antes, el código de un estudiante por si solo no tiene porque determinar una clave candidata. Sin embargo, el código y la universidad juntos si que debe ser único dentro de la BD. Por lo tanto, en este caso tenemos dos claves candidatas.

Sin embargo y a pesar de esta situación, seguimos estando ante una relación que se encuentra en FNBC. El atributo *degree* depende completamente de todas las claves candidatas, pues se puede obtener el grado universitario que estudia el alumno a partir de su DNI o de la combinación de la universidad en la que estudia y su código interno.

El atributo *degree* no depende de ningún conjunto de atributos que contenga alguno que no forme parte de una clave candidata y, finalmente, todos los determinantes de las dependencias de la relación son claves candidatas.

- Professor(dni, code, university, email)

Aquí estamos en otro caso en el que directamente todos los atributos son o forman parte de una clave candidata. Una vez más estamos ante una relación en FNBC.

No es casualidad que todas las relaciones que revisamos ya estén en FNBC, ya que con un buen diseño inicial en el que se separan correctamente los conceptos ya se puede conseguir alcanzar este estado de forma natural. Para entenderlo con un ejemplo, para no superar este análisis de normalización deberíamos haber hecho algo

como indicar el nombre del grado universitario o la razón social de la universidad en la relación Student.

- Interview(idInterview, **interview_date**, valid, candidacy, **rrhh_contact**)

En este caso, tenemos que la relación Interview incluye una clave foránea que apunta a Candidacy. Sin embargo, dada la multiplicidad existente entre las candidaturas y las entrevistas tenemos que esta clave foránea también es única y, por lo tanto, una Clave Candidata.

Sin embargo y una vez más, no hemos almacenado en la relación Interview ningún dato relacionado con las candidaturas más allá de su identificador, por lo que volvemos a estar en FNBC. Si hubiéramos indicado el identificador o el nombre del estudiante que asistirá a la entrevista, la situación cambiaría pues ese dato puede obtenerse directamente con el atributo *candidacy* sin necesidad de *idInterview*.

En nuestro modelo, si queremos conocer el nombre del estudiante que participa en la entrevista debemos obtener la candidatura a través de la clave foránea *candidacy* almacenada en Interview y luego obtener los datos del estudiante a través de la clave foránea *student* almacenada en Candidacy.

El resto de las relaciones se encuentran en la misma situación. Almacenan la información mínima y necesaria sobre su relación y luego permiten obtener mucha más información a partir de sus claves foráneas.

En resumen, todas nuestras relaciones se encuentran en la Forma Normal de Boyce-Codd y esto nos garantiza, como ya se indicó al principio de este apartado, que hemos minimizado todo lo posible las redundancias en los datos y los posibles errores de actualización con los que podamos encontrarnos en el futuro.

4. Implementación física

Ahora que ya hemos terminado con el diseño conceptual y el diseño lógico, ya estamos listos para abrir SQLDeveloper y comenzar a construir nueva base de datos. El primer paso será comenzar añadiendo las tablas que representan a las relaciones que hemos definido en la fase de diseño lógico.

En esta fase incluiremos todas las tablas, incluyendo las relacionadas con el manejo de los *logs* y las relacionadas con el repositorio estadístico.

universities			
Esta tabla es donde se almacenarán las universidades dentro de nuestra base de datos. Únicamente está compuesta de tres campos, pues el resto de la información relevante como sus estudiantes, grados ofertados, profesores u ofertas se encuentran en otras tablas en forma de claves foráneas.			
Atributo	Tipo de Datos	Nulo	Descripción
PK cif	VARCHAR(9)	No	El identificador fiscal de una universidad.
name	VARCHAR(40)	No	El nombre de la universidad.
direction	VARCHAR(100)	No	La dirección fiscal de la universidad.

companies			
En esta tabla se almacenarán las empresas que participan en los procesos de prácticas en empresas. Al igual que las universidades, la información que guarda es mínima y la mayor parte se encuentra desperdigada en otras tablas.			
Atributo	Tipo de Datos	Nulo	Descripción
PK cif	VARCHAR(9)	No	El identificador fiscal de una empresa.
name	VARCHAR(40)	No	El nombre de la empresa.
direction	VARCHAR(100)	No	La dirección fiscal de la empresa.
validation_date	DATE	No	La fecha en que la empresa fue validada por el CIC.

degrees			
En esta tabla se almacena la información relativa a los grados universitarios. Se trata de una información importante pues al menos uno de los indicadores estadísticos dependerá de esta tabla.			
Atributo	Tipo de Datos	Nulo	Descripción
PK code	VARCHAR(20)	No	Se trata del identificador único del grado universitario.
name	VARCHAR(100)	No	El nombre del grado universitario.
area	VARCHAR(100)	No	El área de conocimiento al que pertenece el grado.
CHECK (area IN ('Education', 'Political Science', 'Law', 'Science of Security', 'Economy', 'Marketing and Communication', 'Engineering', 'Dessign', 'Medicine', 'Social Sciences', 'Art and Humanity', 'Music', 'Science'))			

degree_offers			
En esta tabla es donde se guarda la relación existente entre las universidades y los grados. O lo que es lo mismo, de aquí podemos conocer que grados son ofertados en cada universidad.			
Atributo	Tipo de Datos	Nulo	Descripción
FK degree	VARCHAR(20)	No	Se trata del identificador único del grado universitario.
FK university	VARCHAR(9)	No	Se trata del identificador único de la universidad.

PRIMARY KEY (degree, university)
FOREIGN KEY (degree) **REFERENCES** degrees(code)
FOREIGN KEY (university) **REFERENCES** universities(cif)

persons		Esta tabla guarda la información de las personas, sin tener en cuenta cual es su rol dentro de la base de datos, por lo que aquí se guarda la información básica de cada persona.		
Atributo	Tipo de Datos	Nulo	Descripción	
PK dni	VARCHAR(9)	No	El identificador único de la persona.	
name	VARCHAR(20)	No	El nombre de la persona.	
surname1	VARCHAR(20)	No	El primer apellido de la persona.	
surname2	VARCHAR(20)	Si	El segundo apellido de la persona.	
birthdate	DATE	No	La fecha de nacimiento de la persona.	
telf.	VARCHAR(20)	No	El teléfono de contacto de la persona.	
direction	VARCHAR(100)	No	La dirección de la persona.	

students		En esta tabla se guarda la información relativa a los estudiantes. Esta tabla referencia también a la tabla personas y es la combinación de ambas la que aporta toda la información sobre un estudiante.		
Atributo	Tipo de Datos	Nulo	Descripción	
PK dni	VARCHAR(9)	No	El identificador único del estudiante.	
code	VARCHAR(20)	No	El código que la universidad asigna al estudiante.	
FK university	VARCHAR(9)	No	El identificador único de la universidad del estudiante.	
FK degree	VARCHAR(20)	No	El identificador único del grado universitario del estudiante.	
FOREIGN KEY (dni) REFERENCES persons(dni),				
FOREIGN KEY (university) REFERENCES universities(cif),				
FOREIGN KEY (degree) REFERENCES degrees(code),				
UNIQUE (code, university)				

professors		En esta tabla se guarda la información relativa a los profesores. Esta tabla referencia también a la tabla personas y es la combinación de ambas la que aporta toda la información sobre un profesor.		
Atributo	Tipo de Datos	Nulo	Descripción	
PK dni	VARCHAR(9)	No	El identificador único del profesor.	
code	VARCHAR(20)	No	El código que la universidad asigna al profesor.	
FK university	VARCHAR(9)	No	El identificador único de la universidad del profesor.	
email	VARCHAR(50)	No	La dirección e-mail de contacto del profesor.	
FOREIGN KEY (dni) REFERENCES persons(dni),				
FOREIGN KEY (university) REFERENCES universities(cif),				
UNIQUE (code, university)				
UNIQUE (email)				

Tal y como está aquí representado, la dirección e-mail es única dentro de la tabla de los profesores. Sin embargo, en realidad la dirección e-mail es única para toda la base de datos.

Sin embargo, desde la sentencia SQL que crea la tabla de profesores no es posible establecer esta restricción de integridad. Más adelante, en los procedimientos almacenados de alta, baja y modificación se implementará esta restricción.

responsibles	En esta tabla se guarda la información relativa a los responsables de las empresas. Esta tabla referencia también a la tabla personas y es la combinación de ambas la que aporta toda la información sobre un responsable.		
Atributo	Tipo de Datos	Nulo	Descripción
PK dni	VARCHAR(9)	No	El identificador único del responsable.
FK company	VARCHAR(9)	No	El identificador único de la empresa del responsable.
FOREIGN KEY (dni) REFERENCES persons(dni), FOREIGN KEY (company) REFERENCES companies(cif)			

RRHH_contacts	En esta tabla se guarda la información relativa a los con contactos de recursos humanos de las empresas. Esta tabla referencia también a la tabla personas y es la combinación de ambas la que aporta toda la información sobre un empleado de recursos humanos.		
Atributo	Tipo de Datos	Nulo	Descripción
PK dni	VARCHAR(9)	No	El identificador único del responsable.
FK company	VARCHAR(9)	No	El identificador único de la empresa del responsable.
email	VARCHAR(50)	No	La dirección e-mail de contacto del trabajador de RRHH.
FOREIGN KEY (dni) REFERENCES persons(dni), FOREIGN KEY (company) REFERENCES companies(cif) UNIQUE (email)			

Al igual que con los profesores, aquí no está totalmente representada la restricción de integridad que se quiere aplicar sobre las direcciones e-mail. Tal y como está ahora, dos direcciones e-mail podrían repetirse entre la tabla de profesores y la de recursos humanos. Esto se solucionará más adelante.

offers	En esta tabla se guarda toda la información relativa a las ofertas que las empresas hacen en cada universidad. La Clave Primaria de esta tabla se generará de forma automática por una secuencia.		
Atributo	Tipo de Datos	Nulo	Descripción
PK idOffer	VARCHAR(10)	No	El identificador único de la oferta.
name	VARCHAR(50)	No	El nombre de la oferta.
description	VARCHAR(500)	No	La descripción de la oferta.
publish_date	DATE	No	La fecha de publicación de la oferta.
FK company	VARCHAR(9)	No	La compañía que publica la oferta.
FK university	VARCHAR(9)	No	La universidad donde se publica la oferta.
FOREIGN KEY (company) REFERENCES companies(cif), FOREIGN KEY (university) REFERENCES universities(cif)			

candidacies				
En esta tabla se guarda la información relativa a uno de los aspectos más importantes y centrales de nuestra base de datos, las candidaturas que los estudiantes realizan para las ofertas.				
Atributo	Tipo de Datos	Nulo	Descripción	
PK idCandidacy	VARCHAR(10)	No	El identificador único de la candidatura.	
valid	NUMBER(1)	Si	Indica si esta candidatura pasó el proceso de validación.	
validation_date	DATE	Si	La fecha en que la candidatura se validó.	
state	VARCHAR(100)	No	El estado actual de la candidatura.	
FK offer	VARCHAR(10)	No	La oferta con la que está relacionada.	
FK student	VARCHAR(9)	No	El estudiante que realiza la candidatura.	
FK responsible	VARCHAR(9)	No	El responsable que revisa la candidatura.	
CHECK (state IN ('Presented', 'Validated', 'Interview', 'Accepted', 'Rejected'))				
FOREIGN KEY (offer) REFERENCES offers(idOffer),				
FOREIGN KEY (student) REFERENCES students(dni),				
FOREIGN KEY (responsible) REFERENCES responsables(dni)				

Hay una restricción de integridad que no se aplica directamente en esta tabla, pero se aplicará más adelante en los procedimientos almacenados de alta, baja y modificación. Se trata del hecho de que el responsable que se aporta aquí debe pertenecer a la empresa que publicó la oferta.

candidacy_requisites				
Cada oferta puede requerir uno o más requisitos que los estudiantes deben aportar. Es en esta tabla donde se deben almacenar.				
Atributo	Tipo de Datos	Nulo	Descripción	
PK idRequisite	VARCHAR(10)	No	El identificador único del requisito.	
requisite	VARCHAR (500)	No	El texto o url que cumplimenta el requisito.	
FK candidacy	VARCHAR(10)	No	La candidatura asociada a este requisito.	
FOREIGN KEY (candidacy) REFERENCES candidacies(idCandidacy)				

interviews				
En esta tabla se guarda la información relativa a las entrevistas realizadas cuando una candidatura ya está llegando a las últimas etapas.				
Atributo	Tipo de Datos	Nulo	Descripción	
PK idInterview	VARCHAR(10)	No	El identificador único de la entrevista.	
interview_date	DATE	No	La fecha de la entrevista.	
valid	NUMBER(1)	Si	Indica si una entrevista ha sido pasada exitosamente.	
FK candidacy	VARCHAR(9)	No	La candidatura asociada a esta entrevista.	
FK rrhh_contact	VARCHAR(9)	No	La persona de recursos humanos que hace la entrevista.	
FOREIGN KEY (candidacy) REFERENCES candidacies(idCandidacy),				
FOREIGN KEY (rrhh_contact) REFERENCES RRHH_contacts(dni)				

De nuevo aquí se querrán aplicar restricciones de integridad que no se pueden apreciar y que se verán en los procedimientos almacenados. Concretamente hay dos, el hecho de que una candidatura solo puede tener una entrevista y el hecho de que la

persona de RRHH debe pertenecer a la compañía que público la oferta de esta candidatura.

interview_coments				
Ya que una entrevista puede tener más de un comentario, es necesario crear una nueva tabla donde se podrán guardar todos ellos.				
Atributo	Tipo de Datos	Nulo	Descripción	
PK	idComment	VARCHAR(10)	No	El identificador único del comentario realizado.
	interview_comment	VARCHAR(500)	No	El comentario en sí mismo.
FK	interview	VARCHAR(9)	No	La entrevista asociada a este comentario.
FOREIGN KEY (interview) REFERENCES interviews(idInterview)				

collaborations				
En esta tabla se almacena la información de la entidad más relevante de la base de datos, las colaboraciones que se realizan entre los alumnos en prácticas y las empresas.				
Atributo	Tipo de Datos	Nulo	Descripción	
PK	idCollaboration	VARCHAR(10)	No	El identificador único de la colaboración.
	init_date	DATE	No	La fecha en que comienza la colaboración.
	duration	NUMBER(10)	No	La duración, en días, de la colaboración.
	remuneration	NUMBER(4)	No	El salario, en euros, de los estudiantes. Cero indica que no es remunerado.
	periodic_report_periodicity	NUMBER(2)	No	La periodicidad, en días, entre los informes periódicos emitidos por el responsable.
FK	student	VARCHAR(9)	No	El estudiante en prácticas.
FK	responsible	VARCHAR(9)	No	El supervisor de la empresa.
FK	Profesor	VARCHAR(9)	No	El profesor que supervisa las prácticas.
FOREIGN KEY (student) REFERENCES students(dni),				
FOREIGN KEY (responsible) REFERENCES responsables(dni),				
FOREIGN KEY (professor) REFERENCES professors(dni)				

Aquí también se aplicará más adelante una restricción de integridad que ahora mismo no puede verse y es el hecho de que el estudiante y el profesor asignados deben pertenecer a la misma universidad.

collaboration_themes				
Cada colaboración abarcará diferentes temas a tratar y pueden ser más de uno, por lo que se almacenan a parte en esta tabla.				
Atributo	Tipo de Datos	Nulo	Descripción	
PK	idTheme	VARCHAR(12)	No	El identificador único del tema.
	name	VARCHAR(10)	No	Un nombre asignado al tema.
	description	VARCHAR(500)	No	La descripción del tema.
FK	collaboration	VARCHAR(10)	No	La colaboración asociada a este tema.
FOREIGN KEY (collaboration) REFERENCES collaborations(idCollaboration)				

periodic_reports		En esta tabla se almacenará toda la información de los informes periódicos que se emitirán para cada colaboración.		
Atributo	Tipo de Datos	Nulo	Descripción	
PK	idReport	VARCHAR(10)	No	El identificador único del informe.
	report_date	DATE	No	La fecha de emisión.
	report_comment	VARCHAR(500)	No	El texto del informe.
FK	collaboration	VARCHAR(10)	No	La colaboración asociada al informe.
FOREIGN KEY (collaboration) REFERENCES collaborations(idCollaboration)				

inspections		En esta tabla se almacenará toda la información de las inspecciones que se emitirán para cada colaboración.		
Atributo	Tipo de Datos	Nulo	Descripción	
PK	idInspection	VARCHAR(10)	No	El identificador único del informe.
	inspection_date	DATE	No	La fecha de la inspección.
	passed	NUMBER(1)	No	Este valor indica si la inspección está pasada o no.
FK	collaboration	VARCHAR(10)	No	La colaboración asociada al informe.
FOREIGN KEY (collaboration) REFERENCES collaborations(idCollaboration)				

El tipo de datos NUMBER(1) está aquí para representar un valor Booleano en el que 0 es igual a False y 1 es igual a True. En el procedimiento almacenado de inserción de inspecciones se establecerán mecanismos para garantizar que únicamente los valores 0 y 1 se pueden pasar como atributos de *passed*.

final_reports		En esta tabla se almacenará toda la información del informe final de las colaboraciones.		
Atributo	Tipo de Datos	Nulo	Descripción	
PK	idReport	VARCHAR(10)	No	El identificador único del informe final.
	report_date	DATE	No	La fecha de emisión.
	valuation	NUMBER(4, 2)	No	La nota del 1 al 10 de este informe.
	suitable	NUMBER(1)	No	Indica si la empresa considera que la práctica se ha superado exitosamente.
FK	collaboration	VARCHAR(10)	No	La colaboración asociada al informe.
FOREIGN KEY (collaboration) REFERENCES collaborations(idCollaboration)				

Aquí también se aplicarán ciertas restricciones de integridad que todavía no se aprecian. La primera es que una colaboración solo puede tener un informe final y la segunda es que el valor que se puede aportar a *suitable* es 1 o 0.

final_report_comments		En esta tabla se guardarán todos los comentarios que las personas involucradas almacenen sobre los procesos de colaboración.		
Atributo	Tipo de Datos	Nulo	Descripción	
PK	idComment	VARCHAR(10)	No	El identificador único del comentario
	final_report_comment	VARCHAR(500)	No	El comentario en sí mismo.
FK	final_report	VARCHAR(10)	No	El informe final asociado al comentario.

FK	person	VARCHAR(9)	No	La persona que emite el comentario.
FOREIGN KEY (final_report) REFERENCES final_reports(idReport),				
FOREIGN KEY (person) REFERENCES persons(dni)				

Para esta table también se establecerá una restricción de integridad adicional y es que las únicas personas que pueden proporcionar comentarios son el estudiante, profesor y responsable que participaban en la colaboración.

Con esta tabla hemos terminado todas aquellas que están relacionadas con la información básica de la base de datos. Pero esto no significa que hayamos terminado de crear tablas.

logs	En esta tabla se almacenarán los logs que servirán para poder comprobar que procesos están saliendo mal en nuestra base de datos. Esta tabla se irá llenando poco a poco y de forma automáticamente cada vez que se llame a un proceso almacenado.			
Atributo	Tipo de Datos	Nulo	Descripción	
PK idLog	VARCHAR(15)	No	El identificador único del log.	
log_date	TIMESTAMP	No	La fecha y hora en que se registró el log.	
log_process	VARCHAR(100)	No	El nombre del procedimiento que registró el log.	
log_entry	VARCHAR(200)	No	Los valores de entrada que recibió el procedimiento.	
log_return	VARCHAR(200)	No	El resultado final de la ejecución del procedimiento.	

Cuando algo funcione mal, por ejemplo, que se incumpla alguna restricción de integridad a la hora de insertar una nueva fila en una tabla, se podrá revisar en este log y saber con exactitud cuál fue el error.

Esto ayudará a futuros usuarios de la base de datos a saber que está pasando y discernir porque la base de datos está fallando.

general_counters	En esta tabla se almacenarán los contadores de carácter general que se usarán para calcular el repositorio estadístico.			
Atributo	Tipo de Datos	Nulo	Descripción	
PK counter_name	VARCHAR(30)	No	El nombre del contador. Sirve como clave primaria.	
counter_value	NUMBER = 0	Si	El valor del contador, inicializado a 0 en todos los casos.	

Después de insertar esta tabla, el siguiente paso es poblarla e incluir los contadores que vamos a necesitar. Para ello se ejecutan las siguientes líneas:

```
INSERT INTO general_counters (counter_name)
VALUES ('university_counter');
INSERT INTO general_counters (counter_name)
VALUES ('collaborations_counter');
INSERT INTO general_counters (counter_name)
VALUES ('aggregated_duration');
```

```

INSERT INTO general_counters (counter_name)
VALUES ('company_counter');
INSERT INTO general_counters (counter_name)
VALUES ('comp_with_off_last');
INSERT INTO general_counters (counter_name)
VALUES ('comp_with_off_last_not_current');
INSERT INTO general_counters (counter_name)
VALUES ('comp_with_off_current');
INSERT INTO general_counters (counter_name)
VALUES ('counter_uni_collab_this_year');

```

university_year_stats_aux		En esta tabla se almacenarán diferentes contadores y valores relacionados con los años académicos. Estos valores se usarán para calcular varios estadísticos del repositorio.		
Atributo	Tipo de Datos	Nulo	Descripción	
PK academic_year	VARCHAR(10)	No	Se trata del año académico en cuestión. Sirve como clave primaria.	
total_offers	NUMBER = 0	Si	El total de ofertas en un año académico.	
total_candidacies	NUMBER = 0	Si	El total de candidaturas en un año académico.	
total_collaborations	NUMBER = 0	Si	El total de colaboraciones en un año académico.	
aggregated_income	NUMBER = 0	Si	La suma de todos los salarios en un año.	
mean_income	NUMBER = 0	Si	La media de los salarios.	
total_collaborations_ended	NUMBER = 0	Si	Las colaboraciones finalizadas.	
suitable_collaborations	NUMBER = 0	Si	Las colaboraciones finalizadas aprobadas.	
non_suitable_collaborations	NUMBER = 0	Si	Las colaboraciones finalizadas suspendidas.	
non_suitable_percentage	NUMBER = 0	Si	El porcentaje de colaboraciones suspendidas.	
counter_uni_with_collab	NUMBER = 0	Si	Total de universidades con alumnos en prácticas.	

companies_stats_aux		En esta tabla se almacenarán diferentes contadores y valores relacionados con las compañías. Estos valores se usarán para calcular varios estadísticos del repositorio.		
Atributo	Tipo de Datos	Nulo	Descripción	
PK cif	VARCHAR(9)	No	Se trata de la empresa en cuestión.	
total_offers	NUMBER = 0	Si	Total de ofertas emitidas.	
total_collaborations_started	NUMBER = 0	Si	Total de prácticas iniciadas.	
total_collaborations_ended	NUMBER = 0	Si	Total de prácticas finalizadas.	
suitable_collaborations	NUMBER = 0	Si	Total de prácticas aprobadas.	
non_suitable_collaborations	NUMBER = 0	Si	Total de prácticas suspendidas.	
non_suitable_percentage	NUMBER = 0	Si	Porcentaje de prácticas suspendidas.	
total_inspections	NUMBER = 0	Si	Total de inspecciones.	
passed_inspections	NUMBER = 0	Si	Total de inspecciones pasadas.	
failed_inspections	NUMBER = 0	Si	Total de inspecciones falladas.	
failed_inspections_percentage	NUMBER = 0	Si	Porcentaje de inspecciones falladas.	
has_offers_this_year	NUMBER(1) = 0	Si	Si ha emitido ofertas este año.	
had_offers_last_year	NUMBER(1) = 0	Si	Si emitió ofertas el año pasado.	

university_stats_aux		En esta tabla se almacenarán diferentes contadores y valores relacionados con las universidades. Estos valores se usarán para calcular varios estadísticos del repositorio.		
Atributo	Tipo de Datos	Nulo	Descripción	
PK cif	VARCHAR(9)	No	Se trata de la universidad en cuestión.	
has_collaboration_this_year	NUMBER(1) = 0	Si	Si tiene alumnos en prácticas este año.	
total_collaborations_ended	NUMBER = 0	Si	Total de prácticas finalizadas.	
suitable_collaborations	NUMBER = 0	Si	Total de prácticas aprobadas finalizadas.	
non_suitable_collaborations	NUMBER = 0	Si	Total de prácticas suspendidas finalizadas.	
suitable_percentage	NUMBER = 0	Si	Porcentaje de prácticas aprobadas.	

degree_stats_aux		En esta tabla se almacenarán diferentes contadores y valores relacionados con los grados universitarios. Estos valores se usarán para calcular varios estadísticos del repositorio.		
Atributo	Tipo de Datos	Nulo	Descripción	
PK code	VARCHAR(20)	No	El grado en cuestión.	
suitable_collaborations_ended	NUMBER = 0	Si	Total de prácticas aprobadas finalizadas.	

professor_stats_aux		En esta tabla se almacenarán diferentes contadores y valores relacionados con los profesores. Estos valores se usarán para calcular varios estadísticos del repositorio.		
Atributo	Tipo de Datos	Nulo	Descripción	
PK dni	VARCHAR(9)	No	El profesor en cuestión.	
current_year_collabs_supervised	NUMBER = 0	Si	Total de prácticas supervisadas este año.	

indicator_11_aux		En esta tabla se almacenará el total de entrevistas realizadas por cada empresa en cada año. Inicialmente solo tiene la columna cif para representar la empresa. De forma dinámica, se añadirá una nueva columna cada vez que se inserte un nuevo año académico en el sistema.		
Atributo	Tipo de Datos	Nulo	Descripción	
PK cif	VARCHAR(9)	No	La empresa en cuestión.	

unique_indicators		En esta tabla se almacenarán los resultados del repositorio estadístico que son únicos. Se entiende por únicos a que son un único valor como el CIF de una empresa o la media de salarios.		
Atributo	Tipo de Datos	Nulo	Descripción	
PK type_indicator	VARCHAR(15)	No	El identificador único del indicador. Es un identificador numérico y coincide con el orden en que el repositorio es mencionado en el enunciado.	
type_description	TIMESTAMP	No	Una descripción resumida del indicador.	
type_value	VARCHAR(100)	No	El resultado que se ha almacenado en la tabla.	

Antes de continuar con más tablas, se realiza la ejecución de las siguientes sentencias SQL para ir llenando la tabla de indicadores únicos:

```

INSERT INTO unique_indicators (type_indicator, type_description)
VALUES (1, 'Universidad con porcentaje más elevado de prácticas superadas.');
```

```

INSERT INTO unique_indicators (type_indicator, type_description)
VALUES (3, 'Empresa que ha recibido más inspecciones.');
```

```

INSERT INTO unique_indicators (type_indicator, type_description)
VALUES (5, 'Sueldo medio que reciben los estudiantes durante este año.');
```

```

INSERT INTO unique_indicators (type_indicator, type_description)
VALUES (6, 'Número de estudiantes que están realizando prácticas.');
```

```

INSERT INTO unique_indicators (type_indicator, type_description)
VALUES (7, 'Porcentaje de universidades sin alumnos en prácticas para este año.');
```

```

INSERT INTO unique_indicators (type_indicator, type_description)
VALUES (8, 'Titulación universitaria con mayor éxito en las prácticas.');
```

```

INSERT INTO unique_indicators (type_indicator, type_description)
VALUES (9, 'Número medio de estudiantes que se presentan a cada oferta.');
```

```

INSERT INTO unique_indicators (type_indicator, type_description)
VALUES (10, 'Duración media de los contratos en prácticas.');
```

```

INSERT INTO unique_indicators (type_indicator, type_description)
VALUES (12, 'Profesor que supervisa más alumnos en el año en curso.');
```

```

INSERT INTO unique_indicators (type_indicator, type_description)
VALUES (13, 'Número total de informes de prácticas pendientes de ser realizados.');
```

```

INSERT INTO unique_indicators (type_indicator, type_description)
VALUES (14, 'Porcentaje de empresas que ofrecieron prácticas el año anterior y no este.');
```

Al finalizar estas sentencias, la tabla ya tiene todos los indicadores con un valor nulo en el campo que marca el valor del indicador. Este campo se irá calculando más adelante una vez se hayan finalizado los procedimientos almacenados y disparadores que se encargarán de ello.

indicator_type_2			
En esta tabla se almacenará el resultado al segundo indicador del repositorio estadístico. Es necesario tenerlo a parte ya que debe guardar una lista con las cinco empresas que más prácticas han ofrecido.			
Atributo	Tipo de Datos	Nulo	Descripción
company	VARCHAR(9)	Si	Se trata del cif de la empresa.

indicator_type_4			
En esta tabla se almacenará el resultado al cuarto indicador del repositorio estadístico. Es necesario tenerlo a parte ya que debe guardar, para cada año académico, el porcentaje de estudiantes que no superan las prácticas.			
Atributo	Tipo de Datos	Nulo	Descripción
year	VARCHAR(4)	Si	El año del que se tiene la información. Es año académico, entendiendo que va de septiembre de un año hasta junio del siguiente. Pero se representa únicamente con el año en que empieza el curso.
value	FLOAT	Si	El valor del porcentaje de estudiantes suspendidos.

indicator_type_11			
En esta tabla se almacenará el resultado al onceavo indicador del repositorio estadístico. Es necesario tenerlo a parte ya que debe guardar, para cada año, las empresas que más entrevistas han hecho.			
Atributo	Tipo de Datos	Nulo	Descripción
year	VARCHAR(4)	Si	El año del que se tiene la información.

company	VARCHAR(9)	Si	El cif de la empresa que ofreció más entrevistas. Aparecerá ordenado en orden descendiente.
---------	------------	----	---

indicator_type_15	En esta tabla se almacenará el resultado al decimoquinto indicador del repositorio estadístico. Es necesario tenerlo a parte ya que debe guardar una lista con las diez empresas en que los estudiantes tienen menso éxito.		
Atributo	Tipo de Datos	Nulo	Descripción
company	VARCHAR(9)	Si	Se trata del cif de la empresa.

indicator_type_16	En esta tabla se almacenará el resultado al decimosexto indicador del repositorio estadístico. Es necesario tenerlo a parte ya que debe guardar una lista con las cinco empresas que más fraudes han cometido en sus contratos.		
Atributo	Tipo de Datos	Nulo	Descripción
company	VARCHAR(9)	Si	Se trata del cif de la empresa.

Ahora ya si que se han finalizado todas las tablas que contendrán la información de nuestra base de datos. Sin embargo, el enunciado especifica que los usuarios finales no deberán hacer operaciones directas sobre estas tablas, por lo que todavía queda mucho trabajo por hacer.

4.1 Procedimientos almacenados ABM

En los requisitos de la base de datos, concretamente en el RNF-03, se especifica que todas las operaciones de ABM (Alta, Baja y Modificación) se harán a través de procedimientos almacenados.

Además, en el RNF-08 se indica que los procedimientos deberán tener un parámetro de salida que será lo que, finalmente, se almacenará en forma de logs para poder mantener un control sobre los procedimientos.

Sin embargo, antes de comenzar con los procedimientos almacenados, crearé una serie de secuencias que servirán para asignar la clave primaria de las tablas: *offers*, *candidacies*, *candidacy_requisites*, *interviews*, *interview_comments*, *collaborations*, *collaboration_themes*, *periodic_reports*, *inspections*, *final_reports*, *final_report_comments* y *log_storeds*.

```
CREATE SEQUENCE seq_id_offers START WITH 1 INCREMENT BY 1;
CREATE SEQUENCE seq_id_candidacies START WITH 1 INCREMENT BY 1;
CREATE SEQUENCE seq_id_candidacy_requisites START WITH 1 INCREMENT BY 1;
CREATE SEQUENCE seq_id_interviews START WITH 1 INCREMENT BY 1;
CREATE SEQUENCE seq_id_interview_comments START WITH 1 INCREMENT BY 1;
CREATE SEQUENCE seq_id_collaborations START WITH 1 INCREMENT BY 1;
CREATE SEQUENCE seq_id_collaboration_themes START WITH 1 INCREMENT BY 1;
CREATE SEQUENCE seq_id_periodic_reports START WITH 1 INCREMENT BY 1;
CREATE SEQUENCE seq_id_inspections START WITH 1 INCREMENT BY 1;
```

```

CREATE SEQUENCE seq_id_final_reports START WITH 1 INCREMENT BY 1;
CREATE SEQUENCE seq_id_final_report_comments START WITH 1 INCREMENT BY 1;
CREATE SEQUENCE seq_id_log_storeds START WITH 1 INCREMENT BY 1;

```

Estas secuencias servirán, durante el uso de los procedimientos almacenados, para asignar la clave primaria de sus respectivas clases mediante el uso de .NEXTVAL, el cual facilita el siguiente valor de la secuencia.

Con todos los preparativos ya realizados, podemos pasar a los procedimientos almacenados para las operaciones de inserción, baja y modificación. No obstante, no se incluye ninguna operación de baja de datos pues se entiende que el CIC desea guardar la información de forma permanente para poder mantener actualizado su repositorio estadístico.

insert_or_update_company	
Descripción	Este procedimiento se utiliza para insertar una nueva empresa o para modificar una que ya existe.
Entrada	<ul style="list-style-type: none"> • p_cif IN VARCHAR2, • p_name IN VARCHAR2, • p_direction IN VARCHAR2, • p_validation_date IN DATE
Funcionamiento	<ol style="list-style-type: none"> 1. Comprueba si el CIF que se ha pasado como valor de entrada existe o no en la base de datos. 2. Comprueba si el CIF que se ha pasado contiene exactamente 9 caracteres. <ol style="list-style-type: none"> 2.1. Si no los tiene, devuelve un error. 3. Si el CIF que se pasó como entrada no existía en la base de datos, crea una nueva compañía. 4. Si el CIF que se pasó como entrada existía en la base de datos, modifica esa empresa. 5. Registra el log con el resultado de la operación.
Log	<ul style="list-style-type: none"> • KO: El CIF debe tener exactamente 9 caracteres. • OK
Salida	<ul style="list-style-type: none"> • RSP OUT VARCHAR2

insert_or_update_university	
Descripción	Este procedimiento se utiliza para insertar una nueva universidad o para modificar una que ya existe.
Entrada	<ul style="list-style-type: none"> • p_cif IN VARCHAR2, • p_name IN VARCHAR2, • p_direction IN VARCHAR2,
Funcionamiento	<ol style="list-style-type: none"> 1. Comprueba si el CIF que se ha pasado como valor de entrada existe o no en la base de datos. 2. Comprueba si el CIF que se ha pasado contiene exactamente 9 caracteres.

	<p>2.1. Si no los tiene, devuelve un error.</p> <p>3. Si el CIF que se pasó como entrada no existía en la base de datos, crea una nueva compañía.</p> <p>4. Si el CIF que se pasó como entrada existía en la base de datos, modifica esa empresa.</p> <p>5. Registra el log con el resultado de la operación.</p>
Log	<ul style="list-style-type: none"> • KO: El CIF debe tener exactamente 9 caracteres. • OK
Salida	<ul style="list-style-type: none"> • RSP OUT VARCHAR2

insert_or_update_degree	
Descripción	Este procedimiento se utiliza para insertar un nuevo grado o para modificar uno que ya existe.
Entrada	<ul style="list-style-type: none"> • p_code IN VARCHAR2, • p_name IN VARCHAR2, • p_area IN VARCHAR2,
Funcionamiento	<ol style="list-style-type: none"> 1. Comprueba si el valor pasado como p_area pertenece a una de las áreas de conocimiento validas. <ol style="list-style-type: none"> 1.1. Devuelve un error si no lo hace. 2. Comprueba si el p_code pasado ya existe en la base de datos. <ol style="list-style-type: none"> 2.1. Crea un nuevo grado si no existe. 2.2. Modifica el grado si ya existe. 3. Genera el log con el resultado.
Log	<ul style="list-style-type: none"> • KO: El Área debe ser valida. • OK
Salida	<ul style="list-style-type: none"> • RSP OUT VARCHAR2

insert_degree_offers	
Descripción	Este procedimiento se utiliza para insertar en la tabla de degree_offers.
Entrada	<ul style="list-style-type: none"> • p_degree IN VARCHAR2, • p_university IN VARCHAR2,
Funcionamiento	<ol style="list-style-type: none"> 1. Comprueba si el grado pasado como parámetro existe. <ol style="list-style-type: none"> 1.1. Devuelve un error si no existe. 2. Comprueba si existe la universidad. <ol style="list-style-type: none"> 2.1. Devuelve un error si no existe. 3. Comprueba si ya hay una combinación de universidad y grado. <ol style="list-style-type: none"> 3.1. Devuelve un error si la combinación existe. 4. Si no se devuelve ningún error, se insertan los datos. 5. Se registra el log con el resultado.
Log	<ul style="list-style-type: none"> • KO: El grado indicado no existe. • KO: La universidad indicada no existe. • KO: Esta combinación ya está incluida. • OK
Salida	<ul style="list-style-type: none"> • RSP OUT VARCHAR2

insert_or_update_student	
Descripción	Este procedimiento se utiliza para insertar o modificar al mismo tiempo los datos que un estudiante tiene en la tabla de estudiantes y de personas.
Entrada	<ul style="list-style-type: none"> • p_dni IN VARCHAR2 • p_name IN VARCHAR2 • p_surname1 IN VARCHAR2 • p_surname2 IN VARCHAR2 • p_birthdate IN DATE • p_telf IN VARCHAR2 • p_direction IN VARCHAR2 • p_code IN VARCHAR2 • p_university IN VARCHAR2 • p_degree IN VARCHAR2
Funcionamiento	<ol style="list-style-type: none"> 1. Comprueba si el grado existe. <ol style="list-style-type: none"> 1.1. Devuelve un error si no lo hace. 2. Comprueba si la universidad existe. <ol style="list-style-type: none"> 2.1. Devuelve un error si no lo hace. 3. Comprueba si la universidad ofrece ese grado. <ol style="list-style-type: none"> 3.1. Devuelve un error si no lo hace. 4. Comprueba si el estudiante existe. <ol style="list-style-type: none"> 4.1. Si no existe, inserta los datos en las tablas personas y estudiantes. 4.2. Si existe, modifica los datos en ambas tablas. 5. Genera un log con el resultado.
Log	<ul style="list-style-type: none"> • KO: El Grado indicado no existe. • KO: La Universidad indicada no existe. • KO: La Universidad no ofrece los estudios indicados. • OK
Salida	<ul style="list-style-type: none"> • RSP OUT VARCHAR2

insert_or_update_professor	
Descripción	Este procedimiento se utiliza para insertar o modificar al mismo tiempo los datos que un profesor tiene en la tabla de profesores y de personas.
Entrada	<ul style="list-style-type: none"> • p_dni IN VARCHAR2 • p_name IN VARCHAR2 • p_surname1 IN VARCHAR2 • p_surname2 IN VARCHAR2 • p_birthdate IN DATE • p_telf IN VARCHAR2 • p_direction IN VARCHAR2 • p_code IN VARCHAR2 • p_university IN VARCHAR2 • p_email IN VARCHAR2

Funcionamiento	<ol style="list-style-type: none"> 1. Comprueba si la universidad existe. <ol style="list-style-type: none"> 1.1.Devuelve un error si no lo hace. 2. Comprueba si el e-mail ya está en la base de datos. <ol style="list-style-type: none"> 2.1.Devuelve un error si lo está. 3. Comprueba si el código ya existe para esa universidad. <ol style="list-style-type: none"> 3.1.Devuelve error si lo hace. 4. Comprueba si el profesor existe. <ol style="list-style-type: none"> 4.1.Si no existe, inserta los datos en las tablas personas y profesores. 4.2.Si existe, modifica los datos en ambas tablas. <ol style="list-style-type: none"> 4.2.1.Si existe, pero intentamos cambiar el código, da un error. 5. Genera un log con el resultado.
Log	<ul style="list-style-type: none"> • KO: La Universidad indicada no existe. • KO: Este e-mail ya está en la base de datos. • KO: Este código ya está registrado en esta universidad. • KO: El código es único y no se puede cambiar. • OK
Salida	<ul style="list-style-type: none"> • RSP OUT VARCHAR2

Es aquí donde se aplica la primera restricción de integridad que vimos que no se podía aplicar en las tablas. Este proceso comprueba en todas las tablas donde puede haber un e-mail y, si lo encuentra, entonces devuelve un error.

insert_or_update_responsible	
Descripción	Este procedimiento se utiliza para insertar o modificar al mismo tiempo los datos que un responsable tiene en la tabla de responsables y de personas.
Entrada	<ul style="list-style-type: none"> • p_dni IN VARCHAR2 • p_name IN VARCHAR2 • p_surname1 IN VARCHAR2 • p_surname2 IN VARCHAR2 • p_birthdate IN DATE • p_telf IN VARCHAR2 • p_direction IN VARCHAR2 • p_company IN VARCHAR2
Funcionamiento	<ol style="list-style-type: none"> 1. Comprueba si la empresa existe. <ol style="list-style-type: none"> 1.1.Devuelve un error si no lo hace. 2. Comprueba si el responsable existe. <ol style="list-style-type: none"> 2.1.Si no existe, inserta los datos en las tablas personas y responsables. 2.2.Si existe, modifica los datos en ambas tablas. 3. Genera un log con el resultado.
Log	<ul style="list-style-type: none"> • KO: La compañía indicada no existe. • OK
Salida	<ul style="list-style-type: none"> • RSP OUT VARCHAR2

insert_or_update_RRHH_contact	
Descripción	Este procedimiento se utiliza para insertar o modificar al mismo tiempo los datos que un trabajador de RRHH tiene en la tabla de rrrh_contacts y de personas.
Entrada	<ul style="list-style-type: none"> • p_dni IN VARCHAR2 • p_name IN VARCHAR2 • p_surname1 IN VARCHAR2 • p_surname2 IN VARCHAR2 • p_birthdate IN DATE • p_telf IN VARCHAR2 • p_direction IN VARCHAR2 • p_email IN VARCHAR2 • p_company IN VARCHAR2
Funcionamiento	<ol style="list-style-type: none"> 1. Comprueba si la empresa existe. <ol style="list-style-type: none"> 1.1.Devuelve un error si no lo hace. 2. Comprueba si el e-mail ya está en la base de datos. <ol style="list-style-type: none"> 2.1.Devuelve un error si lo está. 3. Comprueba si el responsable existe. <ol style="list-style-type: none"> 3.1.Si no existe, inserta los datos en las tablas personas y responsables. 3.2.Si existe, modifica los datos en ambas tablas. 4. Genera un log con el resultado.
Log	<ul style="list-style-type: none"> • KO: La compañía indicada no existe. • KO: Este e-mail ya está en la base de datos. • OK
Salida	<ul style="list-style-type: none"> • RSP OUT VARCHAR2

Aquí aparece de nuevo la restricción de integridad sobre los e-mails. De este modo, no podrá existir un profesor y un trabajado de RRHH con el mismo e-mail.

insert_offers	
Descripción	Este procedimiento inserta las ofertas que hacen las empresas.
Entrada	<ul style="list-style-type: none"> • p_name IN VARCHAR2 • p_description IN VARCHAR2 • p_publish_date IN DATE • p_company IN VARCHAR2 • p_university IN VARCHAR2
Funcionamiento	<ol style="list-style-type: none"> 1. Comprueba si la universidad existe. <ol style="list-style-type: none"> 1.1.Devuelve un error si no lo hace. 2. Comprueba si la empresa existe. <ol style="list-style-type: none"> 2.1.Devuelve un error si no lo hace. 3. Inserta la oferta si no hay errores. 4. Registra el log.
Log	<ul style="list-style-type: none"> • KO: La Universidad indicada no existe.

	<ul style="list-style-type: none"> • KO: La Compañía indicada no existe. • OK
Salida	<ul style="list-style-type: none"> • RSP OUT VARCHAR2

insert_new_candicacy	
Descripción	Este procedimiento es el que está a cargo de insertar una candidatura en el sistema, así como de hacer todas las comprobaciones para garantizar su integridad.
Entrada	<ul style="list-style-type: none"> • p_offer IN VARCHAR2 • p_student IN VARCHAR2 • p_responsible IN VARCHAR2
Funcionamiento	<ol style="list-style-type: none"> 1. Comprueba si el estudiante existe. <ol style="list-style-type: none"> 1.1. Devuelve un error si no lo hace. 2. Comprueba si la oferta existe. <ol style="list-style-type: none"> 2.1. Devuelve un error si no lo hace. 3. Comprueba si el responsable existe. <ol style="list-style-type: none"> 3.1. Devuelve un error si no lo hace. 4. Comprueba si el responsable trabaja en la empresa que emitió la oferta. <ol style="list-style-type: none"> 4.1. Devuelve un error si no lo hace. 5. Comprueba si el estudiante estudia en la universidad que recibió la oferta. <ol style="list-style-type: none"> 5.1. Devuelve un error si no lo hace. 6. Si no hay errores, inserta la candidatura. La candidatura se inserta con el estado '<i>presented</i>' y con valores nulos en los campos <i>valid</i> y <i>validation_date</i>. 7. Registra el log.
Log	<ul style="list-style-type: none"> • KO: El estudiante indicado no existe. • KO: La oferta indicada no existe. • KO: El responsable indicado no existe. • KO: El responsable indicado no trabaja en la empresa que emite la oferta. • KO: El estudiante indicado no estudia en la universidad que recibe la oferta. • OK
Salida	<ul style="list-style-type: none"> • RSP OUT VARCHAR2

Aquí podemos ver otra de las restricciones de integridad que se mencionaron anteriormente. Una candidatura no puede ser valida si se le intenta asignar un responsable que no se corresponde a la empresa que emitió la oferta.

insert_candicacy_requisites	
Descripción	Este procedimiento sirve para insertar los requisitos cumplimentados por los estudiantes.
Entrada	<ul style="list-style-type: none"> • p_requisite IN VARCHAR2

	<ul style="list-style-type: none"> • p_candidacy IN VARCHAR2
Funcionamiento	<ol style="list-style-type: none"> 1. Comprueba si la candidatura existe. <ol style="list-style-type: none"> 1.1. Devuelve un error si no existe. 2. Inserta el requisito si no hay errores. 3. Registra el log.
Log	<ul style="list-style-type: none"> • KO: La candidatura indicada no existe. • OK
Salida	<ul style="list-style-type: none"> • RSP OUT VARCHAR2

change_candidacy_state	
Descripción	Este procedimiento sirve para cambiarle el estado a una candidatura. Se utilizará junto a otros procedimientos.
Entrada	<ul style="list-style-type: none"> • p_candidacy IN VARCHAR2 • p_state IN VARCHAR2
Salida	<ul style="list-style-type: none"> • p_return OUT NUMBER
Funcionamiento	<ol style="list-style-type: none"> 1. Comprueba si la candidatura pasada existe. <ol style="list-style-type: none"> 1.1. Devuelve error si no existe. 2. Comprueba que el estado sea uno de los aceptados. <ol style="list-style-type: none"> 2.1. Devuelve error si no lo hace. 3. Comprueba si la candidatura está en estado Aceptada o Rechazada. <ol style="list-style-type: none"> 3.1. Devuelve un error pues esta candidatura ya no puede editarse. 4. Si no hay errores, modifica el estado de la candidatura. 5. Si hace el cambio devuelve el valor 1 y si hay algún error devuelve el valor 0 en p_return. 6. Registra el log
Log	<ul style="list-style-type: none"> • KO: El estado indicado no es correcto. • KO: La candidatura indicada no existe. • KO: La candidatura indicada ya no se puede editar. • OK

Aquí se aplica también una restricción de integridad que no podíamos aplicar en la creación de la tabla. Y es que una candidatura que ya ha sido marcada como Aceptada o como Rechazada ya no puede editarse más, pues ha llegado al final del proceso.

validate_candidacy	
Descripción	Este procedimiento es el que realizan los trabajadores de RRHH encargados de revisar si una candidatura cumple con los requisitos mínimos de la oferta.
Entrada	<ul style="list-style-type: none"> • p_candidacy IN VARCHAR2 • p_valid IN NUMBER • p_date IN DATE
Funcionamiento	<ol style="list-style-type: none"> 1. Se comprueba si la candidatura indicada es correcta o no.

	<ol style="list-style-type: none"> 1.1. Si no existe, devuelve error. 2. Si revisa si el valor p_valid es 0 o 1. <ol style="list-style-type: none"> 2.1. Devuelve error si no lo es. 3. Revisa si la candidatura está rechazada o aceptada. <ol style="list-style-type: none"> 3.1. Si lo estuviera, devuelve un error. 4. Si no hay ningún error. <ol style="list-style-type: none"> 4.1. Se cambia el estado de la candidatura a Validada si p_valid es 1. 4.2. Se cambia el estado de la candidatura a Rechazada si p_valid es 0. 5. Registra el log
Log	<ul style="list-style-type: none"> • KO: La candidatura indicada no existe. • KO: El valor indicado no es válido. • KO: No se puede validar una candidatura que ya está rechazada o aceptada. • OK
Salida	<ul style="list-style-type: none"> • RSP OUT VARCHAR2

insert_new_interview	
Descripción	Este procedimiento es el que se encarga de insertar una nueva entrevista. Además, también cambiará el estado de la candidatura a Entrevista en caso de que todo esté bien.
Entrada	<ul style="list-style-type: none"> • p_date IN DATE • p_candidacy IN VARCHAR2 • p_rrhh_contact IN VARCHAR2
Funcionamiento	<ol style="list-style-type: none"> 1. Comprueba si la candidatura existe. <ol style="list-style-type: none"> 1.1. Devuelve error si no lo hace. 2. Comprueba que el contacto de RRHH exista. <ol style="list-style-type: none"> 2.1. Devuelve error si no lo hace. 3. Comprueba si la candidatura ya tiene entrevista. <ol style="list-style-type: none"> 3.1. Devuelve error si ya tiene entrevista. 4. Comprueba que el contacto de RRHH trabaje en la empresa que publico la oferta de la candidatura. <ol style="list-style-type: none"> 4.1. Devuelve error si no lo hace. 5. Comprueba si la candidatura está aceptada o rechazada. <ol style="list-style-type: none"> 5.1. Devuelve un error si lo estuviera. 6. Si no hay errores, inserta la entrevista y cambia el estado de la candidatura a Entrevista. 7. Registra el log.
Log	<ul style="list-style-type: none"> • KO: La candidatura indicada no existe. • KO: El contacto de RRHH indicado no existe. • KO: Esta candidatura ya tiene una entrevista. • KO: El contacto de recursos humanos indicado no trabaja en la empresa que supervisa esta candidatura. • KO: No se puede concertar una entrevista a una candidatura

	aceptada o rechazada.
	<ul style="list-style-type: none"> • OK
Salida	<ul style="list-style-type: none"> • RSP OUT VARCHAR2

Este procedimiento almacenado realiza cuatro comprobaciones de integridad diferentes. Siendo las más relevantes y que no estaban implementadas también en las tablas que una candidatura solo puede tener una entrevista y que la persona que realiza la entrevista debe trabajar en la empresa que supervisa la candidatura.

insert_interview_comment	
Descripción	Este procedimiento se utiliza para insertar comentarios en una entrevista.
Entrada	<ul style="list-style-type: none"> • p_comment IN VARCHAR2 • p_interview IN VARCHAR2
Funcionamiento	<ol style="list-style-type: none"> 1. Revisa si la entrevista pasada como parámetro existe. <ol style="list-style-type: none"> 1.1. Devuelve error si no existe. 2. Registra el comentario si no hay errores. 3. Registra el log.
Log	<ul style="list-style-type: none"> • KO: La entrevista indicada no existe. • OK
Salida	<ul style="list-style-type: none"> • RSP OUT VARCHAR2

evaluate_interview	
Descripción	Este procedimiento es el que realiza un contacto de RRHH en el momento en que finaliza la entrevista.
Entrada	<ul style="list-style-type: none"> • p_interview IN VARCHAR2, • p_valid IN NUMBER
Funcionamiento	<ol style="list-style-type: none"> 1. Comprueba si la entrevista existe. <ol style="list-style-type: none"> 1.1. Devuelve error si no lo hace. 2. Comprueba que el valor de p_valid sea 1 o 0 <ol style="list-style-type: none"> 2.1. Devuelve error si no lo es. 3. Comprueba si esta entrevista ya ha recibido una evaluación. <ol style="list-style-type: none"> 3.1. Devuelve error si l oha hecho. 4. Modifica la entrevista para indicar el valor de <i>valid</i> y modifica la candidatura. <ol style="list-style-type: none"> 4.1. Si p_valid es 1 la candidatura queda Aceptada. 4.2. Si es 0 queda Rechazada. 5. Registra el log
Log	<ul style="list-style-type: none"> • KO: La entrevista indicada no existe. • KO: El valor indicado no es válido. • KO: No se puede evaluar una entrevista que ya se ha evaluado. • OK
Salida	<ul style="list-style-type: none"> • RSP OUT VARCHAR2

insert_new_collaboration	
Descripción	Este procedimiento inserta una nueva colaboración.
Entrada	<ul style="list-style-type: none"> • p_init_date IN DATE • p_duration IN NUMBER • p_remuneration IN NUMBER • p_periodic_report_periodicity IN NUMBER • p_student IN VARCHAR2 • p_responsible IN VARCHAR2 • p_professor IN VARCHAR2
Funcionamiento	<ol style="list-style-type: none"> 1. Comprueba que el estudiante exista. <ol style="list-style-type: none"> 1.1. Devuelve error si no lo hace. 2. Comprueba que el responsable exista. <ol style="list-style-type: none"> 2.1. Devuelve error si no lo hace. 3. Comprueba que el profesor exista. <ol style="list-style-type: none"> 3.1. Devuelve error si no lo hace. 4. Comprueba que estudiante y profesor sean de la misma universidad. <ol style="list-style-type: none"> 4.1. Devuelve error si no lo hace. 5. Si no hay errores, inserta la colaboración. 6. Inserta el log.
Log	<ul style="list-style-type: none"> • KO: El estudiante indicado no es correcto. • KO: El responsable indicado no es correcto. • KO: El profesor indicado no es correcto. • KO: El profesor no es de la misma universidad que el estudiante. • OK
Salida	<ul style="list-style-type: none"> • RSP OUT VARCHAR2

Entre las comprobaciones que realiza la que más destaca es la que se encarga de comprobar que el estudiante y el profesor sean de la misma universidad, pues no tendría sentido que no lo hiciera.

insert_collaboration_theme	
Descripción	Cada colaboración puede tener más de un tema, por lo que debe guardarse a parte. Este procedimiento es el que los inserta.
Entrada	<ul style="list-style-type: none"> • p_name IN VARCHAR2 • p_description IN VARCHAR2 • p_collaboration IN VARCHAR2
Funcionamiento	<ol style="list-style-type: none"> 1. Comprueba que la colaboración existe. <ol style="list-style-type: none"> 1.1. Devuelve error si no lo hace. 2. Inserta el tema en la colaboración. 3. Inserta el log.
Log	<ul style="list-style-type: none"> • KO: La colaboración indicada no es correcta.

	<ul style="list-style-type: none"> • OK
Salida	<ul style="list-style-type: none"> • RSP OUT VARCHAR2

insert_periodic_report	
Descripción	Este es el procedimiento que inserta los informes periódicos.
Entrada	<ul style="list-style-type: none"> • p_report_date IN DATE • p_report_comment IN VARCHAR2 • p_collaboration IN VARCHAR2
Funcionamiento	<ol style="list-style-type: none"> 1. Comprueba que la colaboración existe. <ol style="list-style-type: none"> 1.1.Devuelve error si no lo hace. 2. Inserta el informe periódico en la colaboración. 3. Inserta el log.
Log	<ul style="list-style-type: none"> • KO: La colaboración indicada no es correcta. • OK
Salida	<ul style="list-style-type: none"> • RSP OUT VARCHAR2

insert_new_inspection	
Descripción	Este es el procedimiento que inserta las inspecciones.
Entrada	<ul style="list-style-type: none"> • p_inspection_date IN DATE • p_passed IN NUMBER • p_collaboration IN VARCHAR2
Funcionamiento	<ol style="list-style-type: none"> 1. Comprueba si la colaboración existe. <ol style="list-style-type: none"> 1.1.Devuelve error si no lo hace. 2. Comprueba que el valor pasado en p_passed sea 0 o 1. <ol style="list-style-type: none"> 2.1.Devuelve error si no lo es. 3. Si no hay errores, inserta la inspección. 4. Genera el log con el resultado.
Log	<ul style="list-style-type: none"> • KO: La colaboración indicada no es correcta. • KO: El valor que se ha pasado no es correcto. • OK
Salida	<ul style="list-style-type: none"> • RSP OUT VARCHAR2

insert_final_report	
Descripción	Este es el procedimiento que inserta el informe final.
Entrada	<ul style="list-style-type: none"> • p_report_date IN DATE • p_valoration IN NUMBER • p_suitable IN NUMBER • p_collaboration IN VARCHAR2
Funcionamiento	<ol style="list-style-type: none"> 1. Comprueba si la colaboración indica existe. <ol style="list-style-type: none"> 1.1.Devuelve error si no existe. 2. Comprueba si la colaboración ya dispone de informe final. <ol style="list-style-type: none"> 2.1.Devuelve error si tiene informe final. 3. Comprueba que el valor p_suitable sea 1 o 0 <ol style="list-style-type: none"> 3.1.Devuelve error si es otro valor.

	4. Si no hay errores, inserta un informe final suspendido o aprobado.
Log	<ul style="list-style-type: none"> • KO: La colaboración indicada no es correcta. • KO: La colaboración indicada ya tiene un informe final. • KO: El valor pasado no es correcto. • OK
Salida	<ul style="list-style-type: none"> • RSP OUT VARCHAR2

insert_new_final_report_comment	
Descripción	Este es el procedimiento que inserta comentarios en el informe final.
Entrada	<ul style="list-style-type: none"> • p_final_report_comment IN VARCHAR2 • p_final_report IN VARCHAR2 • p_person IN VARCHAR2
Funcionamiento	<ol style="list-style-type: none"> 1. Comprueba si el informe final existe. <ol style="list-style-type: none"> 1.1. Devuelve error si no lo hace. 2. Comprueba que la persona existe. <ol style="list-style-type: none"> 2.1. Devuelve error si no lo hace. 3. Comprueba que la persona pertenezca al proceso. <ol style="list-style-type: none"> 3.1. Devuelve error si no lo hace. 4. Si no hay errores, inserta el comentario en el informe final.
Log	<ul style="list-style-type: none"> • KO: El informe final indicado no existe. • KO: La persona indicada no existe. • KO: La persona indicada no pertenece a este proceso. • OK
Salida	<ul style="list-style-type: none"> • RSP OUT VARCHAR2

Y con esto queda terminado el diseño de los procedimientos almacenados de alta, baja y modificación. Como se puede ver, hemos aprovechado que teníamos que generar estos procesos para cumplir con todas las restricciones de integridad que era necesario aplicar a nuestra base de datos.

4.2 El Repositorio Estadístico

Para actualizar las respuestas que el repositorio estadístico debe dar también se usarán procedimientos almacenados y, mayormente, disparadores. Sin embargo, el RNF-05 nos dice que el repositorio estadístico debe devolver sus respuestas en un tiempo constante 1.

Para lograr este tiempo, la idea general es que toda la información de actualice de forma individual cada vez que se produzca un cambio en la base de datos. Un cambio como, por ejemplo, insertar una nueva universidad.

Cada vez que se produzca un movimiento en la BD que sea susceptible de cambiar el repositorio, un disparador recogerá dicho movimiento y actualizará si es necesario el estadístico afectado.

Por ejemplo, el primer estadístico nos dice que queremos saber la Universidad que, con todos los datos disponibles, tiene el porcentaje más alto de estudiantes que superan las prácticas en empresas. Por lo tanto, el disparador debería ejecutarse cada vez que se inserta un informe final, pues es ahí donde se determina si un estudiante es apto o no.

Así pues, cuando se haga la llamada al repositorio estadístico, este simplemente deberá devolvernos el valor o valores que ya tiene guardado, obteniendo así el tiempo constante 1 que se nos pedía en RNF-05.

Antes de comenzar a calcular el repositorio, primero hay que crear dos procedimientos almacenados que servirán para conocer en qué año universitario nos encontramos. Después de todo, no es tan fácil como coger el año en el que nos encontramos, pues un año académico comienza después de verano y termina antes del verano siguiente.

check_academic_year	
Descripción	Este procedimiento obtiene el año académico a partir de una fecha.
Entrada	<ul style="list-style-type: none"> • p_date IN DATE
Salida	<ul style="list-style-type: none"> • p_academic_year OUT VARCHAR2
Funcionamiento	Revisa el mes de la fecha insertada como parámetro. Si el mes es superior o igual a julio, entonces el año académico es el que comienza el año de la fecha y termina el siguiente año. Por el contrario, si el mes es junio o anterior, el año académico será el que comenzó el año anterior y termina este año.

insert_new_university_year	
Descripción	Este procedimiento obtiene el año académico a partir de una fecha.
Entrada	<ul style="list-style-type: none"> • p_date IN DATE
Funcionamiento	Obtiene el año académico utilizando el procedimiento <i>check_academic_year</i> . Una vez obtenido, inserta una nueva fila con el valor del año académico en las tablas <i>indicator_type_4</i> , <i>indicator_type_11</i> y <i>university_year_stats_aux</i> . También inserta una nueva columna con el año académico en la tabla <i>indicator_11_aux</i> .

A continuación, se describe los disparadores que mantendrán actualizado en todo momento el repositorio estadístico.

add_company_trigger	
Tabla	companies
Disparador	AFTER INSERT / FOR EACH ROW
Acción	Inserta el CIF de la compañía en la tabla <i>companies_stats_aux</i> y en la tabla <i>indicator_11_aux</i> . También incrementa en uno el contador <i>company_counter</i> de la tabla <i>general_counters</i> .

add_university_trigger	
Tabla	universities
Disparador	AFTER INSERT / FOR EACH ROW
Acción	Inserta el CIF de la universidad en la tabla <i>university_stats_aux</i> e incrementa el contador <i>university_counter</i> de la tabla <i>general_counters</i> .

add_degree_trigger	
Tabla	degrees
Disparador	AFTER INSERT / FOR EACH ROW
Acción	Inserta el código del grado universitario en la tabla <i>degree_stats_aux</i> .

add_professor_trigger	
Tabla	professors
Disparador	AFTER INSERT / FOR EACH ROW
Acción	Inserta el DNI del profesor en la tabla <i>profesor_stats_aux</i> .

stat1_trigger	
Tabla	final_reports
Disparador	AFTER INSERT / FOR EACH ROW
Acción	<p>En la tabla <i>university_stats_aux</i> busca el CIF de la universidad a la que pertenece el profesor que supervisa la colaboración de prácticas que recibe el informe final. Luego incrementa el contador <i>total_collaborations_ended</i> en uno. En función de si el informe final es favorable o no, incrementa también <i>suitable_collaborations</i> o <i>non_suitable_collaborations</i>. En ambos casos actualiza también <i>suitable_percentage</i> calculándolo con los nuevos valores.</p> <p>Por último, comprueba si ha habido un cambio en la universidad con más prácticas aprobadas y, si lo ha habido, actualiza la tabla <i>unique_indicators</i> para guardar el CIF de la nueva universidad con un porcentaje más elevado de prácticas aprobadas.</p>

stat2_trigger	
Tabla	offers
Disparador	AFTER INSERT / FOR EACH ROW
Acción	En la tabla <i>companies_stats_aux</i> busca el CIF de la empresa que ha publicado la oferta e incrementa el contador <i>total_offers</i> en uno. Luego limpia la tabla <i>indicator_type_2</i> para, posteriormente, insertar en ella la

nueva lista del TOP 5 de empresas con más ofertas publicadas en el CIC.

stat_3_and_16_trigger

Tabla	inspections
Disparador	AFTER INSERT / FOR EACH ROW
Acción	<p>Primero busca el CIF de la compañía que ha recibido la inspección. Luego incrementa en uno <i>total_inspections</i> en la tabla <i>companies_stats_aux</i> para la empresa que ha recibido la inspección.</p> <p>Dependiendo de si la inspección esta pasada o fallada, actualiza también <i>passed_inspections</i> o <i>failed_inspections</i>. Con los nuevos datos actualiza el valor de <i>failed_inspections_percentage</i>.</p> <p>Finalmente, actualiza el indicador único número 3 insertando la empresa con más inspecciones y actualiza la lista del indicador 16 que contiene la lista de empresas que ha fallado en más del 10% de las inspecciones.</p>

stat4_trigger

Tabla	final_reports
Disparador	AFTER INSERT / FOR EACH ROW
Acción	<p>Primero obtiene el año académico en que se ha emitido el informe final. Luego actualiza los contadores en la tabla <i>university_year_stats_aux</i>. En todos los casos actualiza <i>total_collaborations_ended</i> y <i>non_suitable_percentage</i>. En función de si la valoración final es positiva o no, también actualiza <i>non_suitable_collaborations</i> y <i>suitable_collaborations</i>.</p> <p>Como <i>non_suitable_collaborations</i> ha cambiado, va a la tabla <i>indicator_type_4</i> que contiene el porcentaje de suspensos de cada año universitario y actualiza la información del año académico en cuestión.</p>

stat5_trigger

Tabla	collaborations
Disparador	AFTER INSERT / FOR EACH ROW
Acción	<p>Lo primero que hace es obtener el año académico de la colaboración y también el año académico en curso. Luego en la tabla <i>academic_year_stats_aux</i> actualiza el contador <i>total_collaborations</i>, <i>aggregated_income</i> y <i>mean_income</i>.</p> <p>Para acabar, si el año académico de la colaboración es el año académico en curso, entonces debe actualizar el quinto indicador y añadir la nueva media de la remuneración que acaba de calcular.</p>

stat6_trigger1

Tabla	collaborations
Disparador	AFTER INSERT / FOR EACH ROW

Acción	Este disparador no usa tablas auxiliares. Lo que hace es tomar el valor que ya hay en el indicador 6 en <i>unique_indicators</i> . Si este valor es nulo por ser la primera inserción, lo cambia por un 1. Si el valor no es nulo, entonces lo incrementa en uno. Subiendo en uno así el número de estudiantes en prácticas.
---------------	--

stat6_trigger2	
Tabla	final_reports
Disparador	AFTER INSERT / FOR EACH ROW
Acción	Como en el caso anterior, lo que hace es tomar el valor del indicador 6 y decrementarlo en 1. Pues la emisión de un informe final implica que un estudiante de prácticas ha finalizado dichas prácticas.

stat7_trigger	
Tabla	collaborations
Disparador	AFTER INSERT / FOR EACH ROW
Acción	<p>Lo primero que hace el disparador es obtener el CIF de la universidad para la que trabaja el profesor que supervisa la colaboración. Luego también mira si esa universidad tiene o no al menos una colaboración este año.</p> <p>En caso de que no tuviera ninguna colaboración, cambia el contador para indicar que ahora si tiene una colaboración. A parte de esto, también incrementa en uno el contador de la tabla <i>university_year_stats_aux</i> con nombre <i>counter_uni_with_collab</i> que mide cuantas universidades tienen estudiantes en prácticas para ese año.</p> <p>Por último, si el año de la colaboración es también el año actual en curso, utiliza estos nuevos datos para calcular el porcentaje de universidades que no tienen alumnos en prácticas. Para calcularlo lo que hace es dividir las universidades con alumnos en prácticas entre el total de universidades en el sistema. Luego le resta este valor a 1 y obtiene así el porcentaje de universidades sin alumnos en prácticas, manteniendo así actualizado el indicador 7.</p>

stat8_trigger	
Tabla	final_reports
Disparador	AFTER INSERT / FOR EACH ROW
Acción	<p>El primer paso es obtener el código del grado universitario que está estudiando el alumno que ha obtenido el informe final en sus prácticas.</p> <p>Una vez obtenido, en la tabla <i>degree_stats_aux</i> se incrementa el contador <i>suitable_collaborations_ended</i> en una unidad. El siguiente paso es comprobar si el grado universitario que acabamos de incrementar es el mismo que ya está en el indicador estadístico 8 ya que, de serlo, no es necesario comprobar si ha habido una actualización.</p>

En el caso de que no lo fuera, entonces el disparador revisa si el contador que acabamos de incrementar ahora supera el contador del grado universitario que tenía el primer lugar en el indicador 8. Si lo tuviera, entonces lo actualiza.

stat9_trigger1	
Tabla	offers
Disparador	AFTER INSERT / FOR EACH ROW
Acción	<p>El primer paso es obtener el año académico de la oferta, pues para el estadístico solo nos importa el año en curso. En cualquier caso y sea el año que sea, incrementaremos el contador <i>total_offers</i> de la tabla <i>university_year_stats_aux</i> en una unidad.</p> <p>Si el año universitario de la oferta es igual al año universitario en curso, se recalcula el porcentaje de alumnos que se presentan a cada oferta dividiendo el número total de candidaturas por el número total de ofertas.</p> <p>No hay riesgo de obtener error por dividir por cero ya que el total de ofertas es el divisor y no puede darse el caso de que se activen los disparadores sin que haya al menos ya una oferta en el sistema, pues para insertar una candidatura primero debe haber una oferta.</p> <p>El nuevo valor calculado se almacena en la tabla <i>unique_indicators</i>.</p>
stat9_trigger2	
Tabla	candidacies
Disparador	AFTER INSERT / FOR EACH ROW
Acción	El funcionamiento de este disparador es exactamente el mismo que el de <i>stat9_trigger1</i> , la única diferencia es que aquí lo hace con el dividendo de la división, las candidaturas.

stat10_trigger	
Tabla	collaborations
Disparador	AFTER INSERT / FOR EACH ROW
Acción	<p>Al comenzar, este trigger actualiza dos contadores generales de la tabla <i>general_counters</i>. Concretamente, actualiza <i>collaborations_counter</i> y <i>aggregated_duration</i>. El primero lo incrementa en uno y al segundo le suma la duración en minutos de la nueva colaboración.</p> <p>El siguiente paso es hacer una división entre <i>aggregated_duration</i> y <i>collaborations_counter</i>. El resultado de esta división es el indicador estadístico 10 que se almacenará en <i>unique_indicators</i>.</p>

stat11_trigger	
Tabla	interviews

Disparador	AFTER INSERT / FOR EACH ROW
Acción	<p>El primer paso es tomar el CIF de la empresa que realiza la entrevista a partir del responsable de recursos humanos que la realizará. También obtiene el año académico actual para saber que valor tendrá que actualizar.</p> <p>Una vez el disparador ya sabe que empresa y que año académico debe actualizar, va a la tabla <i>stat11_trigger</i> e incrementa para esa empresa y para ese año académico el contador de entrevistas realizadas.</p> <p>Luego, para ese año en concreto obtiene cual es la empresa que ha realiza más entrevistas y lo actualiza en la tabla <i>indicator_type_11</i>.</p>

stat12_trigger	
Tabla	collaborations
Disparador	AFTER INSERT / FOR EACH ROW
Acción	<p>El indicador 12 solo tiene en cuenta los datos del año en curso. Por lo tanto, lo primero que hará será obtener el año académico actual y el año académico de la colaboración.</p> <p>Solo si ambos coinciden, incrementará en <i>profesor_stats_aux</i> el contador <i>current_year_collabs_supervised</i> en uno. Después de incrementarlo, también actualizará en <i>unique_indicators</i> el indicador estadístico 12 con el profesor que más prácticas supervisa.</p>

stat13_trigger1	
Tabla	collaborations
Disparador	AFTER INSERT / FOR EACH ROW
Acción	<p>Asumiendo que el número de informes finales pendientes de emitir por parte de los responsables es igual al número de colaboraciones restándole el número total de informes finales, entonces el funcionamiento de este disparador es exactamente el mismo que el de <i>stat6_trigger1</i>. Pues para el sexto estadístico se consideró que el numero total de estudiantes actualmente en prácticas era igual al numero de colaboraciones menos el número de informes finales.</p>

stat13_trigger2	
Tabla	collaborations
Disparador	AFTER INSERT / FOR EACH ROW
Acción	<p>Su funcionamiento es exactamente el mismo que el de <i>stat6_trigger2</i>. La única diferencia es que el estadístico sobre el que se realiza la actualización es el 3.</p>

stat14_trigger	
Tabla	offers
Disparador	AFTER INSERT / FOR EACH ROW

Acción	<p>El primer paso es obtener el año académico de la oferta, el año académico actual y el año académico anterior al actual.</p> <p>Si el año académico de la oferta es igual al año académico anterior al actual, entonces comprueba si la compañía tuvo o no ofertas el año pasado. Si no tuvo ofertas, entonces actualiza <i>had_offers_last_year</i> de <i>companies_stats_aux</i> para indicar que esta empresa si tuvo ofertas el año pasado, ya que acaba de insertar una.</p> <p>Del mismo modo, también incrementa en uno el contador <i>comp_with_off_last</i> de <i>general_counters</i>. Este contador indica cuantas empresas tuvieron ofertas el año pasado. Luego comprueba si la empresa en cuestión tiene ofertas en el año en curso. Si no tiene, incrementa en uno el contador <i>comp_with_off_last_not_current</i> que indica el número de empresas que tuvieron ofertas el año pasado y no tienen este año.</p> <p>En resumen, si la empresa no tenía ofertas el año pasado, se indica ahora que si tiene. Si además no tiene ofertas este año, se incrementa el numero de empresas con ofertas el año pasado y no el actual.</p> <p>En el caso de que la oferta sea de este año, entonces el valor que se incrementa es <i>has_offers_last_year</i>. Además, si la empresa tuviera ofertas el año anterior, entonces lo que se hace es decrementar en 1 el contador <i>comp_with_off_last_not_current</i>. Pues con este cambio habría una empresa menos que tuviera ofertas el año pasado y no tuviera ofertas en el actual.</p> <p>Por último y cuando ya se han actualizado los contadores, se procede a dividir el valor de <i>comp_with_off_last_not_current</i> entre el total de empresas que está guardado en el contador <i>company_counter</i> de la tabla <i>general_counters</i>. Obteniendo así el porcentaje de empresas que el año pasado publicaron ofertas y no lo han hecho este año.</p>
---------------	--

stat15_trigger	
Tabla	final_reports
Disparador	AFTER INSERT / FOR EACH ROW
Acción	<p>El primer paso, obtener el CIF de la empresa a partir del responsable que supervisa la colaboración que ha recibido el informe final.</p> <p>Si el informe es positivo se incrementa <i>suitable_collaborations</i> y si se ha suspendido se incrementa <i>non_suitable_collaborations</i>. En ambos casos, se incrementa también <i>total_collaborations_ended</i>. Con estos datos, se incrementa también <i>non_suitable_percentage</i>. Estos cuatro contadores son de <i>companies_stats_aux</i>.</p> <p>Por último, se actualiza la lista del TOP 10 de empresas con una mayor</p>

ratio de prácticas suspendidas. El valor ya está calculado previamente en *non_suitable_percentage*. Para ello, solo hay que utilizar un SELECT que ordene la tabla por ese contador y luego limitar los resultados a los diez primeros.

La implementación física ya casi está terminada. Sin embargo, todavía queda una cosa más por hacer. Hay que pensar en que ocurre cuando termina un año universitario y comienza el siguiente.

Muchos de los estadísticos del repositorio hay que reiniciarlos. También hay que reiniciar e incluso mover algunos de los contadores de las tablas auxiliares. Por no hablar de que hay que insertar nuevas filas de años académicos en las tablas *university_year_stats_aux*, *indicator_type_4* y *indicator_type_11*. Además, en la tabla *indicator_11_aux* hay que insertar una columna con el nuevo año.

Para ello, se genera un procedimiento almacenado.

new_academic_year_starting	
Descripción	Este procedimiento está a cargo de gestionar todo lo que ocurre cuando se inicia un nuevo año académico.
Funcionamiento	<p>Obtiene el valor guardado en <i>comp_with_off_current</i> que indica las empresas con ofertas este año. El valor contenido se lo asigna a <i>comp_with_off_last</i> y <i>comp_with_off_last_not_current</i>. Esto es así ya que el número de empresas con ofertas este año ahora pasa a ser el número de empresas con ofertas el año pasado, pues estamos cambiando de año.</p> <p>Luego reinicia a 0 el valor de <i>compt_with_off_current</i>. Al estar recién cambiado el año, todavía no hay ninguna oferta.</p> <p>En la tabla <i>companies_stats_aux</i> asigna el valor de <i>has_offers_this_year</i> a <i>had_offers_last_year</i> y luego asigna el valor 0 a <i>has_offers_this_year</i>.</p> <p>En la tabla <i>university_stats_aux</i> pone a 0 el valor de <i>has_collaboration_this_year</i>. Hace lo mismo con la tabla <i>professor_stats_aux</i> y el contador <i>current_year_collabs_supervised</i>.</p> <p>Por último, hace un llamado a <i>insert_new_university_year</i> pasándole como parámetro la fecha SYSDATE. Si recordamos, este procedimiento almacenado insertaba las filas y la columna con el nuevo año académico en las tablas en las que debía indicarse el cambio de año.</p>

Esta función debe llamarse una vez al año, justo cuando ha terminado un año académico. Se podría hacer manualmente, pero yo he optado por implementar el siguiente código:

```
BEGIN
  DBMS_SCHEDULER.CREATE_JOB (
    job_name => 'new_academic_year_starting_job',
    job_type => 'PLSQL_BLOCK',
    job_action => 'BEGIN new_academic_year_starting; END;',
    start_date => '01-07-2023 00:00:00',
    repeat_interval => 'FREQ=YEARLY;BYMONTH=7;BYMONTHDAY=1',
    enabled => TRUE
  );
END;
/
```

De este modo, se ha definido un trabajo que se ejecutará todos los años el día 1 de julio. Este trabajo simplemente llamará al procedimiento que se encargará de gestionar todo lo relativo al cambio de año.

Para el primer año de funcionamiento o si se quieren registrar datos históricos, será necesario insertar los años académicos manualmente con el procedimiento que se definió para tal efecto. Sin embargo, para los años siguientes simplemente habrá que dejar que llegue el día indicado.

5. Fase de pruebas

La implementación física de la base de datos ya ha finalizado de forma exitosa. Sin embargo, antes de entregarla al CIC es necesario comprobar que funciona tal y como queremos que lo haga. Para ello se realizará una serie de pruebas a lo largo de este capítulo.

Cuando el capítulo haya finalizado, podremos confirmar que la BD funciona correctamente y que está lista para su entrega.

Tal y como se menciona en RNF-03, la BD debe funcionar mediante procedimientos almacenados. Además, tal y como ya se ha mencionado en otros apartados, la actualización del repositorio estadístico se realiza mediante disparadores.

Por lo tanto, el juego de pruebas se centrará en observar que los procedimientos almacenados que se han diseñado permiten la correcta gestión de la base de datos, que hacen un tratamiento correcto de las excepciones provocadas por los incumplimientos de las restricciones de integridad y que los disparadores activados por estos procedimientos actualizan correctamente el repositorio estadístico.

Procedimiento	Descripción	Resultado
insert_new_university_year	<p>Este procedimiento recibe una fecha, comprueba cual es el año académico de esa fecha y luego la inserta como una fila en las tablas <i>university_year_stats_aux</i>, <i>indicator_type_4</i> e <i>indicator_type_11</i>. Además, la inserta como una columna en la tabla <i>indicator_11_aux</i>.</p> <p>Para la prueba se insertarán las fechas 30/09/2022, 30/09/2021 y 30/09/2020. De este modo, tendremos el año académico en curso, el anterior y el anterior a ese.</p> <p>Las comprobaciones que se realizan son observar que se hayan insertado correctamente esos años en las tablas indicadas.</p>	Satisfactorio
insert_or_update_company	<p>Insertamos 10 empresas diferentes en el sistema. Las comprobaciones que hacemos son las siguientes:</p> <ul style="list-style-type: none"> • Comprobar la tabla <i>Companies</i> • Comprobar la tabla <i>log_storeds</i> para revisar que haya devuelto OK las 10 veces. • Comprobar que el contador <i>company_counter</i> de la tabla <i>general_counters</i> es 10. • Comprobar que se han insertado los 10 CIFs a la tabla <i>companies_stats_aux</i>. • Comprobar que se han insertado los 10 CIFs a la tabla <i>indicator_11_aux</i>. 	Satisfactorio

insert_or_update_company	<p>Este método también sirve para modificar empresas. Por lo que modificamos una de ellas para cambiarle la dirección. Hacemos las siguientes comprobaciones:</p> <ul style="list-style-type: none"> • La dirección se actualiza correctamente. • El log ha devuelto OK. • El contador <i>company_counter</i> sigue siendo 10. • No se ha insertado ninguna fila en las tablas correspondientes. 	Satisfactorio
insert_or_update_company	<p>Intentamos insertar una empresa con un CIF que no tiene exactamente 9 caracteres. Comprobamos lo siguiente:</p> <ul style="list-style-type: none"> • No se inserta la empresa. • El log devuelve KO con el motivo correcto. • El contador <i>company_counter</i> sigue siendo 10. • No se ha insertado ninguna fila en las tablas correspondientes. 	Satisfactorio
insert_or_update_university	<p>Insertamos cinco nuevas universidades en el sistema. Las comprobaciones que hacemos son las siguientes:</p> <ul style="list-style-type: none"> • Las universidades están en <i>universities</i>. • Los cinco logs han devuelto OK. • Los CIF de las universidades están insertados en <i>university_stats_aux</i>. • El contador <i>university_counter</i> de la tabla <i>general_counters</i> es 5. 	Satisfactorio
insert_or_update_university	<p>Modificamos una empresa para cambiarle la dirección.</p> <ul style="list-style-type: none"> • Comprobamos que la dirección es correcta. • El log es OK. • El contador <i>university_counter</i> sigue siendo 5. • No ha habido cambios en <i>university_stats_aux</i>. 	Satisfactorio
insert_or_update_university	<p>Insertamos una universidad con un CIF que no tiene exactamente 9 caracteres.</p> <ul style="list-style-type: none"> • No se inserta la empresa. • El log devuelve KO con el motivo correcto. • El contador <i>university_counter</i> sigue siendo 5. • No ha habido cambios en <i>university_stats_aux</i>. 	Satisfactorio
insert_or_update_degree	<p>Insertamos ocho grados universitarios nuevos.</p> <ul style="list-style-type: none"> • Se insertan correctamente en la tabla <i>degrees</i>. • Los ocho logs son OK. • Se insertan sus códigos en la tabla <i>degree_stats_aux</i>. 	Satisfactorio
insert_or_update_degree	<p>Nos damos cuenta de que hemos escrito un error en el nombre de uno de ellos y lo corregimos.</p> <ul style="list-style-type: none"> • El nombre ahora es correcto. • El log es OK. • No hay cambios en <i>degree_stats_aux</i>. 	Satisfactorio
insert_or_update_degree	<p>Insertamos un nuevo grado cuya área de estudios no forma parte de una de las áreas que no forman parte de la lista de áreas validas.</p> <ul style="list-style-type: none"> • El grado no se inserta. • El log es KO con el motivo correcto. • No hay cambios en <i>degree_stats_aux</i>. 	Satisfactorio

insert_degree_offers	<p>Insertamos multiples ofertas a cada universidad. La idea es que cada universidad ofrezca una combinación de grados diferentes. En total se hacen 22 inserciones.</p> <ul style="list-style-type: none"> • Comprobamos que haya 22 inserciones en la tabla <i>degree_offers</i>. • Comprobamos que los 22 logs son OK. 	Satisfactorio
insert_degree_offers	<p>Insertamos una oferta para una universidad que no existe.</p> <ul style="list-style-type: none"> • No se realiza inserción en <i>degree_offers</i>. • El log muestra KO con el motivo correcto. 	Satisfactorio
insert_degree_offers	<p>Insertamos una oferta para un grado que no existe.</p> <ul style="list-style-type: none"> • No se realiza inserción en <i>degree_offers</i>. • El log muestra KO con el motivo correcto. 	Satisfactorio
insert_degree_offers	<p>Insertamos una oferta que ya existe en el sistema.</p> <ul style="list-style-type: none"> • No se realiza inserción en <i>degree_offers</i>. • El log muestra KO con el motivo correcto. 	Satisfactorio
insert_or_update_responsible	<p>Insertamos dos responsables para cada compañía. Un total de 20 responsables.</p> <ul style="list-style-type: none"> • Se realizan 20 inserciones en la tabla <i>responsibles</i> y en la tabla <i>persons</i>. • Los 20 logs son OK. 	Satisfactorio
insert_or_update_responsible	<p>Modificamos uno de los responsables que ya existen y le cambiamos el nombre.</p> <ul style="list-style-type: none"> • Se modifica el nombre correctamente. • El log es OK. 	Satisfactorio
insert_or_update_responsible	<p>Se intenta insertar un responsable a una empresa que no existe.</p> <ul style="list-style-type: none"> • No se realiza ninguna inserción. • El log es KO con el motivo correcto. 	Satisfactorio
insert_or_update_RRHH_contact	<p>Se inserta un contacto de RRHH para cada empresa.</p> <ul style="list-style-type: none"> • Se comprueba que hay 10 inserciones en <i>Persons</i> y en <i>RRHH_contacts</i>. • Los 10 logs son OK. 	Satisfactorio
insert_or_update_RRHH_contact	<p>Se modifica el nombre de un contacto RRHH.</p> <ul style="list-style-type: none"> • El nombre se ha actualizado. • El log es OK. 	Satisfactorio
insert_or_update_RRHH_contact	<p>Se intenta insertar un contacto de RRHH para una empresa que no existe.</p> <ul style="list-style-type: none"> • No se realiza ninguna inserción. • El log es KO con el motivo correcto. 	Satisfactorio
insert_or_update_RRHH_contact	<p>Se intenta insertar un contacto de RRHH con un e-mail ya registrado en la base de datos.</p> <ul style="list-style-type: none"> • No se realiza ninguna inserción. • El log es KO con el motivo correcto. 	Satisfactorio
insert_or_update_professor	<p>Insertamos dos profesores para cada universidad. Diez en total.</p> <ul style="list-style-type: none"> • Comprobamos que hay 10 inserciones en <i>professors</i> y en <i>persons</i>. • Los 10 logs son correctos. • Los DNI se han insertado en la tabla <i>professor_stats_aux</i>. 	Satisfactorio

insert_or_update_professor	<p>Modificamos el nombre de uno de los profesores.</p> <ul style="list-style-type: none"> • La modificación se realiza. • El log es OK. 	Satisfactorio
insert_or_update_professor	<p>Insertamos un profesor a una universidad inexistente.</p> <ul style="list-style-type: none"> • No se realiza la inserción. • El log es KO con el motivo correcto. 	Satisfactorio
insert_or_update_professor	<p>Insertamos un profesor con un email que ya existe en el sistema.</p> <ul style="list-style-type: none"> • No se realiza la inserción. • El log es KO con el motivo correcto. 	Satisfactorio
insert_or_update_professor	<p>Insertamos un profesor con un código repetido.</p> <ul style="list-style-type: none"> • No se realiza la inserción. • El log es KO con el motivo correcto. 	Satisfactorio
insert_or_update_professor	<p>Intentamos modificar el código de un profesor.</p> <ul style="list-style-type: none"> • No se realiza la inserción. • El log es KO con el motivo correcto. 	Satisfactorio
insert_or_update_student	<p>Insertamos cinco alumnos para cada universidad, 25 en total.</p> <ul style="list-style-type: none"> • Se realizan 25 inserciones en las tablas <i>students</i> y <i>persons</i>. • Los 25 logs son OK. 	Satisfactorio
insert_or_update_student	<p>Modificamos el nombre de un estudiante.</p> <ul style="list-style-type: none"> • El nombre se modifica. • El log es OK. 	Satisfactorio
insert_or_update_student	<p>Insertamos un estudiante en una universidad que no existe.</p> <ul style="list-style-type: none"> • No se realiza la inserción. • El log es KO con el motivo correcto. 	Satisfactorio
insert_or_update_student	<p>Insertamos un estudiante con un grado universitario que no existe.</p> <ul style="list-style-type: none"> • No se realiza la inserción. • El log es KO con el motivo correcto. 	Satisfactorio
insert_or_update_student	<p>Insertamos un estudiante con un grado que no se imparte en su universidad.</p> <ul style="list-style-type: none"> • No se realiza la inserción. • El log es KO con el motivo correcto. 	Satisfactorio
insert_or_update_student	<p>Intentamos modificar el código de un estudiante.</p> <ul style="list-style-type: none"> • No se realiza la actualización. • El log es KO con el motivo correcto. 	Satisfactorio
insert_offers	<p>Insertamos múltiples ofertas para diferentes empresas, universidades y años académicos. 25 ofertas en total.</p> <ul style="list-style-type: none"> • Se insertan las 25 ofertas en <i>offers</i>. • Los 25 logs son OK. 	Satisfactorio
insert_offers	<p>Insertamos una oferta para una universidad que no existe.</p> <ul style="list-style-type: none"> • No se realiza la inserción. • El log es KO con el mensaje correcto. 	Satisfactorio
insert_offers	<p>Insertamos una oferta para una empresa que no existe y comprobamos el error.</p> <ul style="list-style-type: none"> • No se realiza la inserción. • El log es KO con el mensaje correcto. 	Satisfactorio

insert_new_candidacy	<p>Insertamos múltiples candidaturas a las diferentes ofertas. 47 candidaturas en total.</p> <ul style="list-style-type: none"> • Se realizan las 47 inserciones. • Los 47 logs son OK. 	Satisfactorio
insert_new_candidacy	<p>Intentamos inserta una candidatura con un estudiante que no existe.</p> <ul style="list-style-type: none"> • No se realiza la inserción. • El log es KO con el mensaje correcto. 	Satisfactorio
insert_new_candidacy	<p>Intentamos insertar una candidatura a una oferta que no existe.</p> <ul style="list-style-type: none"> • No se realiza la inserción. • El log es KO con el mensaje correcto. 	Satisfactorio
insert_new_candidacy	<p>Intentamos insertar una candidatura con un responsable que no existe.</p> <ul style="list-style-type: none"> • No se realiza la inserción. • El log es KO con el mensaje correcto. 	Satisfactorio
insert_new_candidacy	<p>Intentamos insertar una candidatura con un responsable que trabaja en otra empresa.</p> <ul style="list-style-type: none"> • No se realiza la inserción. • El log es KO con el mensaje correcto. 	Satisfactorio
insert_new_candidacy	<p>Intentamos insertar una candidatura con un estudiante que estudia en otra universidad.</p> <ul style="list-style-type: none"> • No se realiza la inserción. • El log es KO con el mensaje correcto. 	Satisfactorio
insert_candidacy_requisites	<p>Insertamos un requisito a cada candidatura. 47 en total.</p> <ul style="list-style-type: none"> • Se realizan las 47 inserciones. • Los 47 logs son OK. 	Satisfactorio
insert_candidacy_requisites	<p>Insertamos un requisito a una candidatura que no existe.</p> <ul style="list-style-type: none"> • No se realiza la inserción. • El log es KO con el mensaje correcto. 	Satisfactorio
validate_candidacy	<p>Emulando el trabajo que haría el contacto de RRHH, revisamos si las candidaturas son correctas y las validamos o rechazamos en consecuencia.</p> <ul style="list-style-type: none"> • Se realizan las 47 modificaciones. • Aparecen 94 logs OK, pues este procedimiento usa el procedimiento <i>change_candidacy_state</i> que tiene su propio log. 	Satisfactorio
validate_candidacy	<p>Intentamos validar una candidatura que no existe.</p> <ul style="list-style-type: none"> • No se realiza ninguna modificación. • El log es KO con el mensaje correcto. 	Satisfactorio
validate_candidacy	<p>Intentamos validar una candidatura con un valor que no es ni 0 ni 1.</p> <ul style="list-style-type: none"> • No se realiza ninguna modificación. • El log es KO con el mensaje correcto. 	Satisfactorio
validate_candidacy	<p>Intentamos validar una candidatura que ya ha sido validada.</p> <ul style="list-style-type: none"> • No se realiza modificación alguna. • Tenemos dos logs KO con el mensaje correcto. Uno para el método <i>validate_candidacy</i> y otro para el método <i>change_candidacy_state</i>. 	Satisfactorio

insert_new_interview	<p>Siguiendo con el proceso, emulando a los responsables solicitamos entrevistas para algunas de las candidaturas. 20 en total.</p> <ul style="list-style-type: none"> • Comprobamos que se insertan 20 entrevistas. • Comprobamos que el estado de las 20 candidaturas cambia a 'Entrevista'. • Comprobamos que los 40 logs son OK. Se incluyen los logs de <i>change_candidacy_state</i>. 	Satisfactorio
insert_new_interview	<p>Se intenta concertar una entrevista para una candidatura que no existe.</p> <ul style="list-style-type: none"> • No se realiza ninguna inserción en <i>interviews</i>. • No realiza ninguna actualización en <i>candidacies</i>. • El log es KO con el mensaje correcto. 	Satisfactorio
insert_new_interview	<p>Se intenta concertar una entrevista con un contacto de RRHH que no existe.</p> <ul style="list-style-type: none"> • No se realiza ninguna inserción en <i>interviews</i>. • No realiza ninguna actualización en <i>candidacies</i>. • El log es KO con el mensaje correcto. 	Satisfactorio
insert_new_interview	<p>Se intenta concertar una entrevista para una candidatura que ya tiene una entrevista concertada.</p> <ul style="list-style-type: none"> • No se realiza ninguna inserción en <i>interviews</i>. • No realiza ninguna actualización en <i>candidacies</i>. • El log es KO con el mensaje correcto. 	Satisfactorio
insert_new_interview	<p>Se intenta concertar una entrevista con un contacto de RRHH que trabaja en otra empresa diferente.</p> <ul style="list-style-type: none"> • No se realiza ninguna inserción en <i>interviews</i>. • No realiza ninguna actualización en <i>candidacies</i>. • El log es KO con el mensaje correcto. 	Satisfactorio
insert_new_interview	<p>Intentamos concertar una entrevista a una candidatura que está rechazada.</p> <ul style="list-style-type: none"> • No se realiza inserción ni modificación alguna. • Tenemos dos logs KO con los mensajes correctos. Uno para <i>insert_new_interview</i> y otro para <i>change_candidacy_state</i>. 	Satisfactorio
insert_interview_comment	<p>Insertamos un comentario de entrevista en la tabla <i>interview_comments</i> para cada una de las entrevistas.</p> <ul style="list-style-type: none"> • Se realizan 20 inserciones en la tabla. • Los 20 logs son OK. 	Satisfactorio
insert_interview_comment	<p>Intentamos insertar un comentario para una entrevista que no existe.</p> <ul style="list-style-type: none"> • No se realiza ninguna inserción. • El log es KO con el mensaje correcto. 	Satisfactorio
evaluate_interview	<p>Emulando el trabajo de los contactos de RRHH, decidimos que entrevistas las consideramos validas y cuales no, teniendo en cuenta que valida implica que la candidatura queda aceptada y no valida implica que queda rechazada.</p> <p>Aceptamos 15 candidaturas y rechazamos 5.</p> <ul style="list-style-type: none"> • Se realizan 20 modificaciones en la tabla 	Satisfactorio

	<p><i>candidacies.</i></p> <ul style="list-style-type: none"> • Quedan 15 candidaturas aceptadas y 5 rechazadas. • Tenemos 40 logs OK. Incluyendo los del método <i>change_candidacy_state</i>. 	
evaluate_interview	<p>Intentamos evaluar una entrevista que no existe.</p> <ul style="list-style-type: none"> • No se realiza ninguna modificación. • El log es KO con el mensaje correcto. 	Satisfactorio
evaluate_interview	<p>Intentamos evaluar una entrevista con un valor que no es ni 0 ni 1.</p> <ul style="list-style-type: none"> • No se realiza ninguna modificación. • El log es KO con el mensaje correcto. 	Satisfactorio
evaluate_interview	<p>Se intenta evaluar una entrevista que ya ha sido evaluada.</p> <ul style="list-style-type: none"> • No se realiza modificación alguna. • Hay dos mensajes KO en el log. Siendo uno para <i>evaluate_interview</i> y el otro mensajes para <i>change_candidacy_state</i>. 	Satisfactorio
insert_new_collaboration	<p>Vamos a insertar una colaboración para cada una de las candidaturas aceptadas. Un total de 15 colaboraciones.</p> <ul style="list-style-type: none"> • Se realizan 15 inserciones. • Tenemos 15 logs OK. • En la tabla <i>general_counters</i> el contador <i>collaboration_counter</i> es 15. 	Satisfactorio
insert_new_collaboration	<p>Insertamos una colaboración para un estudiante que no existe.</p> <ul style="list-style-type: none"> • No se realiza la inserción. • El log es KO con el mensaje correcto. 	Satisfactorio
insert_new_collaboration	<p>Insertamos una colaboración con un responsable que no existe.</p> <ul style="list-style-type: none"> • No se realiza la inserción. • El log es KO con el mensaje correcto. 	Satisfactorio
insert_new_collaboration	<p>Insertamos una colaboración con un profesor que no existe.</p> <ul style="list-style-type: none"> • No se realiza la inserción. • El log es KO con el mensaje correcto. 	Satisfactorio
insert_new_collaboration	<p>Insertamos una colaboración con un estudiante y un profesor de diferentes universidades.</p> <ul style="list-style-type: none"> • No se realiza la inserción. • El log es KO con el mensaje correcto. 	Satisfactorio
insert_collaboration_theme	<p>Insertamos un tema de colaboración para cada colaboración que hemos creado. 15 en total.</p> <ul style="list-style-type: none"> • Se realizan 15 inserciones. • Disponemos de 15 logs OK. 	Satisfactorio
insert_collaboration_theme	<p>Insertamos un tema en una colaboración que no existe.</p> <ul style="list-style-type: none"> • No se realiza la inserción. • El log es KO con el mensaje adecuado. 	Satisfactorio
insert_periodic_report	<p>Insertamos tres informes periodicos en las colaboraciones de los años anteriores. Pues en nuestra prueba las colaboraciones del año 2022-2023 acaban de empezar.</p>	Satisfactorio

	<ul style="list-style-type: none"> • Se realizan 27 inserciones. • Tenemos 27 logs OK. 	
insert_periodic_report	<p>Insertamos un informe periódico en una colaboración que no existe.</p> <ul style="list-style-type: none"> • No se realiza inserción alguna. • Tenemos un log KO con el mensaje adecuado. 	Satisfactorio
insert_new_inspection	<p>Como antes, insertamos tres inspecciones a cada colaboración de años anteriores. Las de este año aun no han tenido tiempo.</p> <ul style="list-style-type: none"> • Se realizan 27 inserciones. • Tenemos 27 logs OK. 	Satisfactorio
insert_new_inspection	<p>Intentamos insertar una inspección a una colaboración que no existe.</p> <ul style="list-style-type: none"> • No se realiza la inserción. • El log es KO con el mensaje correcto. 	Satisfactorio
insert_new_inspection	<p>Intentamos insertar una inspección con un valor que no es 0 o 1.</p> <ul style="list-style-type: none"> • No se realiza la inserción. • El log es KO con el mensaje correcto. 	Satisfactorio
insert_final_report	<p>Insertamos un informe final a cada una de las colaboraciones de años anteriores.</p> <ul style="list-style-type: none"> • Se realizan 9 inserciones. • Tenemos 9 logs OK. 	Satisfactorio
insert_final_report	<p>Insertamos un informe final en una colaboración que no existe.</p> <ul style="list-style-type: none"> • No se realiza la inserción. • El log es KO con el mensaje correcto. 	Satisfactorio
insert_final_report	<p>Insertamos un informe final en una colaboración que ya tiene informe final.</p> <ul style="list-style-type: none"> • No se realiza la inserción. • El log es KO con el mensaje correcto. 	Satisfactorio
insert_final_report	<p>Insertamos un informe final con una valoración que no es 1 o 0.</p> <ul style="list-style-type: none"> • No se realiza la inserción. • El log es KO con el mensaje correcto. 	Satisfactorio
insert_new_final_report_comment	<p>Insertamos un comentario a cada uno de los informes finales. Intercalamos entre estudiante, profesor y responsable para probarlos todos. 9 en total.</p> <ul style="list-style-type: none"> • Se realizan 9 inserciones. • Tenemos 9 logs OK. 	Satisfactorio
insert_new_final_report_comment	<p>Insertamos un comentario en un informe que no existe.</p> <ul style="list-style-type: none"> • No se realiza la inserción. • Tenemos un log KO con el mensaje adecuado. 	Satisfactorio
insert_new_final_report_comment	<p>Insertamos un comentario de una persona que no existe.</p> <ul style="list-style-type: none"> • No se realiza la inserción. • Tenemos un log KO con el mensaje adecuado. 	Satisfactorio
insert_new_final_report_comment	<p>Insertamos un comentario de una persona que no ha participado en la colaboración.</p> <ul style="list-style-type: none"> • No se realiza la inserción. • Tenemos un log KO con el mensaje adecuado. 	Satisfactorio

Con estas pruebas queda claro que los procedimientos almacenados que hemos diseñado funcionan como se esperaban. Sin embargo, todavía queda una cosa más por poner a prueba.

Concretamente, hablamos del repositorio estadístico. Tal y como está diseñada la base de datos, el hecho de haber hecho esta prueba ya ha estado activando los disparadores y configurando el resultado final que el repositorio debe mostrar.

Para saber si los disparadores que mantienen actualizado el repositorio estadístico están funcionando como se espera es necesario comprobar los valores que, tras ejecutar las pruebas, han quedado almacenados en las correspondientes tablas.

Para ello, se ha diseñado una serie de consultas que utilizan funciones de agregación y agrupación que se pueden ejecutar y darán el mismo resultado que el que hay almacenado en el repositorio estadístico. Estas consultas no se apoyan en ningún contador, todos los valores los obtienen con funciones de agregación.

Si se ejecutan cada una de estas consultas, puede comprobarse que el resultado que arrojan es exactamente el mismo que ha quedado guardado en las tablas que almacenan los resultados del repositorio estadístico.

Por lo tanto, también podemos confirmar que, a la finalización del juego de pruebas, se puede concluir que el diseño del repositorio estadístico es correcto.

6. Conclusiones

Pueden sacarse tres conclusiones principales a la finalización de este proyecto. La primera está relacionada con la consecución de los objetivos planteados por el CIC y las otras dos con la planificación del propio proyecto.

En primer lugar y quizás lo más importante, a la finalización del proyecto se han conseguido cumplir todos los objetivos propuestos durante la fase de diseño que, a su vez, cumplen con todos los requisitos planteados por el CIC.

También se ha conseguido alcanzar el objetivo adicional planteado en el punto 2.3 de ampliar el alcance del repositorio estadístico con el objetivo de servir como punto de inicio a un futuro servicio de valoración interna de las empresas participantes en el sistema de prácticas universitarias.

En segundo lugar, se puede concluir que la decisión de abordar el proyecto con una metodología en cascada fue un acierto, pues las decisiones de diseño más relevantes se tomaron durante las fases de planificación, lo que alivió la carga de trabajo en las fases de implementación de la base de datos que consistieron únicamente en una traducción literal del modelo relacional y lógico a un modelo físico.

Como tercera y última conclusión, es muy importante mencionar que haber asumido en un inicio que apenas habría contratiempos y retrasos fue un error. Finalmente, el proyecto estuvo plagado de factores externos que retrasaron el cumplimiento de la planificación del trabajo.

Junto a estos factores externos también hay que incluir un error detectado en la implementación física del repositorio estadístico que, por la naturaleza en cascada del proyecto, se extendió a todas las fases y provocó un retraso importante hasta lograr corregirlo.

En cualquier caso, esto demuestra que el planteamiento original de reservar tiempo para posibles contratiempos fue un acierto, pues a excepción de un pequeño retraso en la tercera entrega parcial del proyecto se consiguió llegar a todas las fechas límites que se habían definido.

Ya para acabar y relacionado con las posibles líneas de trabajo futuro, creo que sería muy interesante seguir ampliando las funcionalidades que permiten la valoración interna de las empresas. Por ejemplo, incluyendo procedimientos almacenados que permitan impedir a ciertas empresas la publicación de nuevas ofertas.

7. Glosario

- **Atributo:** Propiedad que tienen todas las entidades de un mismo tipo de entidad.
- **BBDD:** Acrónimo de Bases de datos.
- **BD:** Acrónimo de Base de datos.
- **CIC:** Acrónimo de Consell Interuniversitari de Catalunya
- **Clave Foránea:** Atributo o conjunto de atributos de una tabla que referencia a otra tabla con el objetivo de relacionarlas. Este atributo o conjunto de atributos es igual a la Clave Primaria de la segunda tabla.
- **Clave Primaria:** Atributo o conjunto de atributos único que sirve para identificar de forma unívoca a una entidad.
- **Diseño conceptual:** Fase que tiene como objetivo crear un esquema conceptual de alto nivel, independiente de la tecnología que se usará y que representa los requisitos, especificaciones y restricciones del proyecto.
- **Diseño físico:** Fase en que se adapta el esquema lógico al SGBD en el que se construirá la BD.
- **Diseño lógico:** Fase en que se transforma el modelo conceptual en un esquema lógico dependiente de la tecnología que se usará para la implementación final.
- **Metodología en cascada:** Proceso de gestión de proyectos consistente en desglosar todas las etapas del proyecto en una forma descendente cuya principal característica es que una etapa debe finalizarse completamente antes de pasar a la siguiente.
- **Relación:** Asociación entre diferentes entidades.
- **SGBD:** Acrónimo de Sistema de Gestión de Bases de Datos
- **SQL:** Acrónimo de Lenguaje de Consulta Estructurada
- **Subclase:** Entidad específica que hereda los atributos y relaciones de una superclase general, lo que permite especializarla.
- **Superclase:** Entidad general que agrupa atributos y relaciones que son comunes para diferentes entidades específicas.
- **UML:** Acrónimo de Lenguaje Unificado de Modelado

8. Bibliografía

1. Documento BOE-A-2014-8138 [En línea] [Consulta: 12 de octubre de 2022]. Disponible en: <https://www.boe.es/buscar/doc.php?id=BOE-A-2014-8138>
2. Documento BOE-A-1981-16532 [En línea] [Consulta: 12 de octubre de 2022]. Disponible en: <https://www.boe.es/buscar/doc.php?id=BOE-A-1981-16532>
3. BURGUES ILLA, Xavier y CUARTERO OLIVERA, Josep. *Diseño lógico de bases de datos* [En línea]. Barcelona: UOC, 2020 [Consulta: 13 de octubre de 2022]. Disponible en: https://discovery.biblioteca.uoc.edu/permalink/34CSUC_UOC/1asfcbc/alma991000621189706712
4. CASAS ROMA, Jordi. *Introducción al diseño de bases de datos* [En línea]. Barcelona: UOC, 2019 [Consulta: 13 de octubre de 2022]. Disponible en: https://discovery.biblioteca.uoc.edu/permalink/34CSUC_UOC/1asfcbc/alma991000621159706712
5. PRADEL MIQUEL, Jordi y RAYA MARTOS, José. *Ingeniería de requisitos* [En línea]. Barcelona: UOC, 2016 [Consulta: 02 de noviembre de 2022]. Disponible en: http://materials.cv.uoc.edu.eu1.proxy.openathens.net/daisy/Materials/PID_00230177/pdf/PID_00230179.pdf
6. CABRÉ i SEGARRA, Blai; CASAS ROMA, Jordi; COSTAL COSTA, Dolors; JUANOLA JUANOLA, Pere; RIUS GAVIDIA, Àngels y SEGRET i SALA, Ramón. *Disseny físic de bases de dades* [En línea]. Barcelona: UOC, 2012 [Consulta: 05 de diciembre de 2022]. Disponible en: https://discovery.biblioteca.uoc.edu/permalink/34CSUC_UOC/1asfcbc/alma991000796147306712
7. MARTÍN ESCOFET, Carme. *El lenguaje SQL I* [En línea]. Barcelona: UOC, 2019 [Consulta: 05 de diciembre de 2022]. Disponible en: https://discovery.biblioteca.uoc.edu/permalink/34CSUC_UOC/1asfcbc/alma991000597089706712
8. CASANY GUERRERO, M. Josep; RODRÍGUEZ GONZÁLEZ, M. Elena y URPÍ TUBELLA, Toni. *El lenguaje SQL II* [En línea]. Barcelona: UOC, 2019 [Consulta: 05 de diciembre de 2022]. Disponible en: https://discovery.biblioteca.uoc.edu/permalink/34CSUC_UOC/1asfcbc/alma991000621259706712
9. *2 Functions to Get the Year from a Date in Oracle* [En línea] [Consulta: 17 de diciembre de 2022]. Disponible en: <https://database.guide/2-functions-to-get-the-year-from-a-date-in-oracle/>
10. *Oracle CONCAT* [En línea] [Consulta: 19 de diciembre de 2022]. Disponible en: <https://www.oracletutorial.com/oracle-string-functions/oracle-concat/>
11. *How to return a value from stored procedure.* [En línea] [Consulta: 9 de enero de 2023]. Disponible en: <https://community.oracle.com/tech/developers/discussion/1032039/how-to-return-a-value-from-stored-procedure>
12. *How to add columns dynamically to an existing table.* [En línea] [Consulta: 10 de enero de 2023]. Disponible en: <https://community.oracle.com/tech/developers/discussion/3599156/how-to-add-columns-dynamically-to-an-existing-table>
13. *Scheduler jobs in Oracle.* [En línea] [Consulta: 13 de enero de 2023]. Disponible en: <https://dbaclass.com/article/scheduler-jobs-oracle/>