

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

Alberto Esparza Pérez
Grado de Ingeniería Informática
Bases de Datos

Jordi Ferrer Duran
Javier Jiménez Pelayo

Febrero 2023



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

FICHA DEL TRABAJO FINAL

Título del trabajo:	<i>Diseño e implementación de la base de datos para una aplicación de control de prácticas en empresas de estudiantes universitarios</i>
Nombre del autor:	<i>Alberto Esparza Pérez</i>
Nombre del consultor/a:	<i>Jordi Ferrer Duran</i>
Nombre del PRA:	<i>Javier Jiménez Pelayo</i>
Fecha de entrega (mm/aaaa):	<i>02/2023</i>
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 en empresa, Gestión del alumnado, Control de universidades.</i>
Resumen del Trabajo (máximo 250 palabras): <i>Con la finalidad, contexto de aplicación, metodología, resultados y conclusiones del trabajo.</i>	
<p><i>Este Trabajo de Fin de Grado pretende dar solución a la necesidad del Consell Interuniversitari de Catalunya (CIC) de poder llevar a cabo un control eficaz del alumnado universitario que realiza prácticas en empresas.</i></p> <p><i>Para ello se diseñará una base de datos a partir de los requisitos planteados por el CIC con los métodos de manipulación de datos necesarios.</i></p> <p><i>Al mismo tiempo se mantendrá un repositorio estadístico que permita conocer en todo momento los indicadores necesarios para ejercer este control.</i></p> <p><i>Por último se deberá permitir una mínima auditoría de las ejecuciones, y resultados de dichas ejecuciones, de los métodos de manipulación de datos.</i></p> <p><i>Se ha utilizado la metodología descrita en la asignatura “Diseño de bases de datos” consistente en 5 fases: Recogida y análisis de requisitos, Diseño conceptual, Diseño lógico, Diseño físico, e Implementación y optimización.</i></p> <p><i>La solución obtenida se ha probado mediante la carga de datos aleatorios y la ejecución de consultas para comprobar la exactitud de los datos del repositorio estadístico.</i></p> <p><i>Se entregan adjuntos los scripts de creación de la estructura de la base de datos así como el de ejecución de la batería de pruebas y los resultados de dichas pruebas.</i></p>	

Abstract (in English, 250 words or less):

This Final Degree Project aims to provide a solution to the need of the Consell Interuniversitari de Catalunya (CIC) to be able to carry out an effective control of university students who are doing an internship in companies.

To this end, a database will be designed based on the requirements exposed by the CIC with the mandatory data manipulation methods.

At the same time, a statistical repository will be created in order to allow at any time to monitor the necessary data indicators.

Finally, a minimum auditing of the executions, and results of said executions, of the data manipulation methods should be allowed.

The methodology used is the same described in the subject "Diseño de bases de datos", consisting of 5 phases: Collection and analysis of requirements, Conceptual design, Logical design, Physical design, and Implementation and optimization.

The solution obtained has been tested by loading random data and running queries to check the accuracy of the data from the statistical repository.

Attached are the scripts needed to create the database structure as well as the execution of the battery of tests and the results of said tests.

Índice

1. Introducción.....	1
1.1. Contexto y justificación del Trabajo.....	1
1.2. Objetivos del Trabajo.....	1
1.3. Enfoque y método seguido.....	2
1.4. Planificación del Trabajo.....	3
1.5. Breve resumen de productos obtenidos.....	8
1.6. Breve descripción de los otros capítulos de la memoria.....	8
2. Análisis de los requisitos.....	10
3. Diseño de la Base de Datos.....	13
3.1. Diseño conceptual.....	13
3.2. Diseño lógico.....	15
3.3. Diseño físico.....	16
4. Implementación.....	20
4.1. Procedimientos ABM.....	20
4.2. Gestión del DataWarehouse.....	34
4.3. Gestión de los logs.....	42
5. Pruebas.....	43
6. Conclusiones.....	51
7. Glosario.....	53
8. Bibliografía.....	55
9. Anexos.....	56

Lista de tablas

Tabla 1: Hitos del proyecto.....	3
Tabla 2: Seguimiento de la planificación.....	5
Tabla 3: Análisis de riesgos.....	7

Lista de figuras

Figura 1: Diagrama de Gantt del TFG completo.....	4
Figura 2: Diagrama de Gantt PEC1 y PEC2.....	5
Figura 3: Diagrama de Gantt PEC3 y Entrega Final.....	5
Figura 4: Modelo conceptual general.....	13
Figura 5: Modelo conceptual repositorio estadístico.....	14
Figura 6: Modelo físico.....	18
Figura 7: Modelo físico del repositorio estadístico.....	19

1. Introducción

1.1. Contexto y justificación del Trabajo

En el ámbito universitario, existe la figura de las prácticas en empresas. En estas prácticas, el alumnado de los últimos cursos de algunos grados realiza prácticas laborales en empresas del ámbito de los estudios realizados.

Mediante estas prácticas se busca que el alumnado universitario tenga un contacto directo y real con el mundo laboral para minimizar la brecha existente entre la formación principalmente teórica de los estudios y el ejercicio profesional.

Estas prácticas nunca deben sustituir una plaza laboral en las empresas, ya que las personas en prácticas se encuentran todavía en periodo formativo y deben aprender y no ocupar una plaza laboral, es por ello que se realizan inspecciones de trabajo periódicas para detectar los posibles abusos que puedan ocurrir.

Para poder realizar un seguimiento del proceso de prácticas las empresas se comprometen a realizar y enviar a la universidad una serie de informes del desarrollo de las prácticas. En el último informe se realizará una evaluación en la que la persona en prácticas recibirá una calificación de aptitud o no aptitud.

Habida cuenta dichas necesidades, el *Consell Interuniversitari de Catalunya (CIC)* ha decidido desarrollar una aplicación que le permita controlar todo el proceso de las prácticas de principio a fin y, de una manera rápida, conocer la situación de todo el alumnado en prácticas.

Se busca controlar todas las etapas del proceso, desde la publicación de las ofertas por parte de las empresas, pasando por el proceso de selección, hasta la valoración final de las prácticas e incluyendo también los informes de cualquier inspección que se haya podido realizar.

En este trabajo se diseñará e implementará una base de datos que dé soporte a la aplicación informática desarrollada por el CIC. Dicha base de datos debe permitir introducir todos los datos necesarios para ejercer el control deseado por parte del CIC así como obtener información a partir de dichos datos que permita conocer en cualquier momento la situación actual.

1.2. Objetivos del Trabajo

El objetivo principal de este trabajo se podría definir como:

- Realizar el diseño, implementación y pruebas de una base de datos que permita gestionar las prácticas en empresas del alumnado universitario de Cataluña. Esta solución debe incluir el almacenamiento de todos los datos necesarios para esta gestión así como la extracción de la

información necesaria para el control del estado actual de dichas prácticas por parte del CIC.

Este objetivo de alto nivel se puede descomponer en una serie de objetivos más detallados:

- Gestión de proyectos. Será necesario definir una planificación del proyecto que permita cumplir con los requisitos en el plazo disponible.
- Extracción de requisitos. Se deben extraer los requisitos del trabajo a realizar a partir del enunciado propuesto.
- Diseño de bases de datos. Se ha de diseñar una base de datos desde cero que cumpla con las necesidades planteadas.
- Explotación de datos. Hay que dotar a la solución obtenida de la capacidad de obtener información útil a partir de los datos introducidos.
- Documentación. Se deberá documentar todo el proceso y generar una memoria del trabajo completo.

Todos estos objetivos buscan integrar los conocimientos adquiridos durante el Grado a través de un proyecto que ofrezca una solución a un problema realista.

1.3. Enfoque y método seguido

A la hora de obtener un producto informático que satisfaga una necesidad por parte de un cliente se pueden optar por diferentes estrategias, cada una de ellas con sus ventajas e inconvenientes.

Adquirir un producto comercial existente

La ventaja más evidente es la inmediatez, si el producto ya existe se puede adquirir y poner en funcionamiento en muy poco tiempo.

Otra ventaja es el precio, los productos existentes en el mercado suelen tener precios inferiores al resto de opciones.

Como desventaja está la capacidad de adaptación de la solución al problema existente. Es difícil que una solución comercial dé respuesta a las necesidades específicas de una organización. Por lo tanto se debe valorar si es viable adaptarse al producto comercial.

Desarrollar un producto nuevo

La principal ventaja es que el producto desarrollado es completamente a medida por lo que probablemente satisfará las necesidades específicas de la organización.

La desventaja es la necesidad de contar con un equipo de desarrollo capacitado para crear el producto.

Otra desventaja es el tiempo, si se debe crear desde cero se debe tener en cuenta que el producto tardará un tiempo en estar disponible.

Contratar el desarrollo de un producto a medida

Esta opción estaría a medio camino entre las dos anteriores.

Como ventaja respecto al desarrollo interno está que no es necesario disponer del equipo de desarrollo ya que se contrata de manera externa.

Como ventaja respecto al producto comercial se puede encontrar que al ser un producto a medida es más probable que se adapte a las necesidades del cliente y satisfaga sus demandas.

Como desventajas, es más caro que un producto comercial y al ser un equipo externo siempre será más difícil transmitir las necesidades y requisitos que con un equipo de desarrollo interno que normalmente conoce la idiosincrasia de la organización.

Para este trabajo se ha optado por desarrollar un producto nuevo desde cero para así controlar todo el ciclo de desarrollo y asegurar que se satisfagan las necesidades específicas del CIC, se parte del supuesto de que el CIC tiene un equipo de desarrollo capacitado para realizar este desarrollo.

1.4. Planificación del Trabajo

Para la planificación del tiempo disponible se deben tener en cuenta los diferentes factores personales de la persona que realiza este TFG. En este caso se debe compaginar con otra asignatura del Grado así como con un trabajo a tiempo completo.

En base a estos factores se realiza una previsión inicial de disponibilidad semanal de 16 horas repartidas en 4 días a la semana, 4 horas cada uno de los días.

Se dividirá el proyecto en 4 hitos adaptados a las PECs en las que se divide la asignatura, lo cual resulta en los siguientes plazos

Tabla 1: Hitos del proyecto

Hito	Inicio	Fin	Horas
Planificación	29/09/22	17/10/22	44
Diseño	18/10/22	21/11/22	80
Implementación	22/11/22	22/12/22	72
Documentación	23/12/22	20/01/23	64
Debate virtual	23/01/23	27/01/23	10

- Planificación del trabajo (44h)
Se debe realizar un análisis previo de las tareas a realizar y generar una planificación de todo el proyecto para posteriormente poder realizar un seguimiento del mismo.
 - Estudio del enunciado (10h)
 - Descomposición en tareas (10h)
 - Planificación del tiempo (10h)
 - Diagrama de Gantt (7h)
 - Análisis de riesgos (7h)

- **Diseño de la base de datos (80h)**
A partir de los requisitos extraídos del enunciado, se debe diseñar una base de datos que satisfaga dichos requisitos.
 - Análisis de requisitos (14h)
 - Creación del diseño conceptual (29h)
 - Creación del diseño lógico (22h)
 - Creación del diseño físico (15h)
- **Implementación de la base de datos (72h)**
Una vez obtenidos los diseños de la etapa anterior, se crearán los scripts que permitan generar los objetos de base de datos, así como insertar los datos de prueba y realizar las pruebas necesarias.
 - Creación de scripts con las tablas (18h)
 - Creación de procedimientos (25h)
 - Creación de scripts con los datos de pruebas (14h)
 - Pruebas de las consultas generadas (15h)
- **Documentación (64h)**
Todo este proceso debe quedar correctamente documentado y se debe generar una memoria exhaustiva para poder defender el trabajo ante el tribunal de evaluación.
 - Completar la memoria (30h)
 - Generar entregables (10h)
 - Preparar presentación (24h)
- **Debate virtual (10h)**
 - Realizar la presentación virtual (5h)
 - Responder preguntas (5h)

Diagrama de Gantt del TFG completo

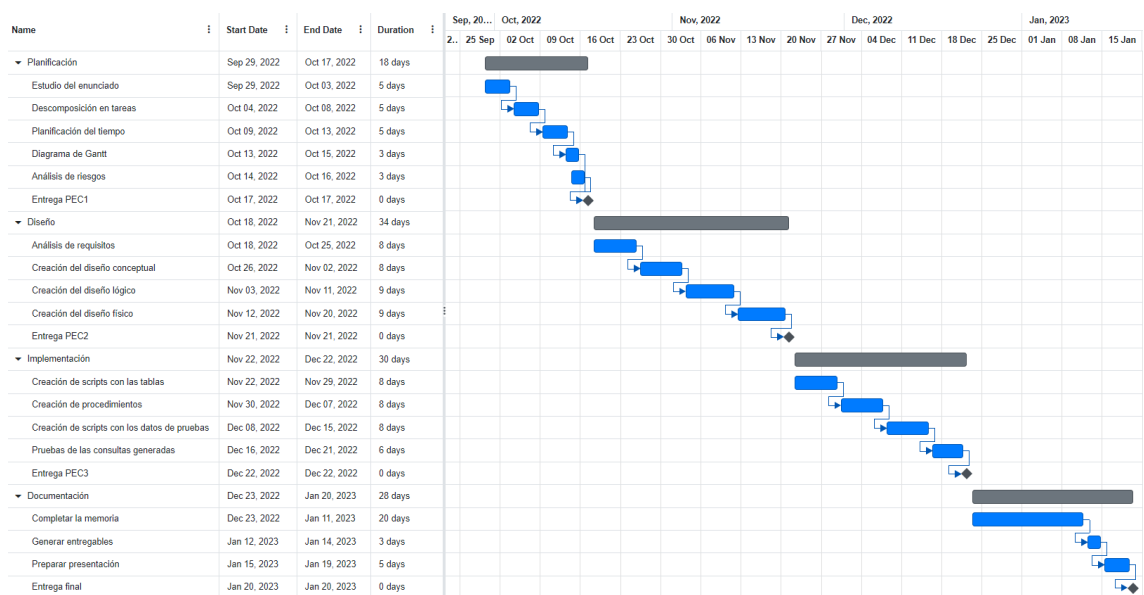


Figura 1: Diagrama de Gantt del TFG completo

Detalle PEC1 y PEC2

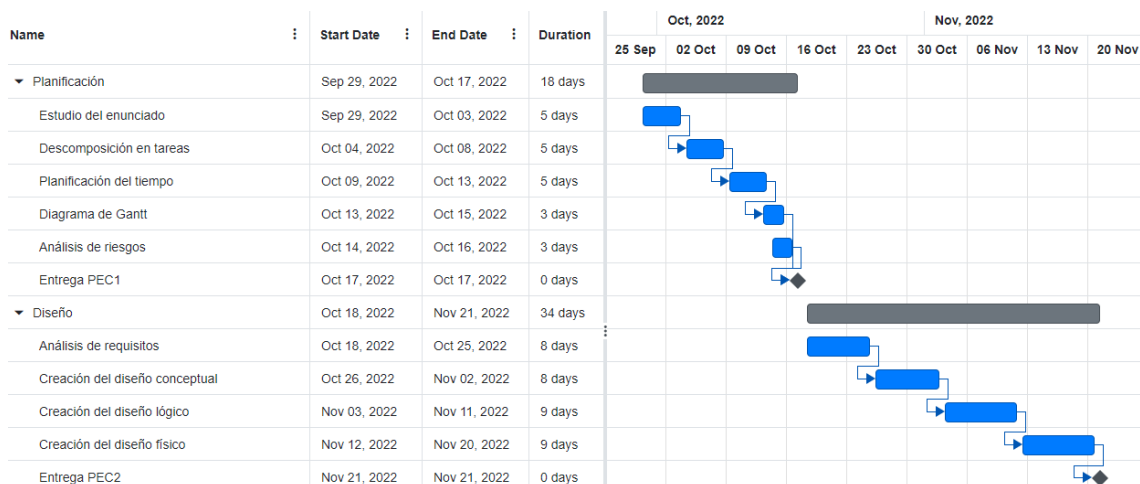


Figura 2: Diagrama de Gantt PEC1 y PEC2

Detalle PEC3 y Entrega Final

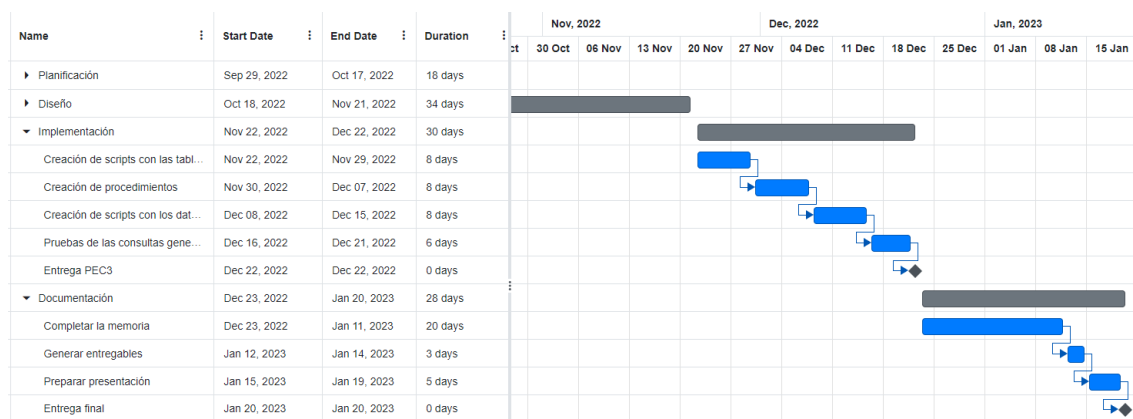


Figura 3: Diagrama de Gantt PEC3 y Entrega Final

1.4.1. Seguimiento de la planificación

Para llevar a cabo un seguimiento de la planificación del proyecto se compararán los objetivos planificados con los objetivos completados en cada uno de los hitos y se planificarán las acciones necesarias para mitigar cualquier desviación resultante en el siguiente hito.

Tabla 2: Seguimiento de la planificación

Objetivo	% Completado	Acciones
Planificación		
Estudio del enunciado	100%	
Descomposición en tareas	100%	
Planificación del tiempo	100%	

Diagrama de Gantt	100%	
Análisis de riesgos	100%	
Diseño		
Análisis de requisitos	100%	
Creación del diseño conceptual	100%	
Creación del diseño lógico	100%	
Creación del diseño físico	100%	
Implementación		
Creación de scripts con las tablas	100%	
Creación de procedimientos	100%	
Creación de scripts con los datos de pruebas	100%	
Pruebas de las consultas generadas	100%	
Documentación		
Completar la memoria	100%	
Generar entregables	100%	
Preparar presentación	100%	

PEC1

Para la primera entrega se completan todas las tareas planificadas dentro del plazo de entrega previsto, por lo que se realiza la entrega completa.

PEC2

Para la segunda entrega también se completan todas las tareas planificadas dentro del plazo y se realiza una entrega completa.

PEC3

En la tercera entrega aparecen varios problemas:

Primero de planificación, se subestima la complejidad de la tarea de creación de procedimientos, de la cual dependen las tareas siguientes.

A esto se une un aumento de la carga de trabajo de la persona que realiza el proyecto que resta parte del tiempo disponible.

Ambos problemas provocan que dicha tarea no se pueda completar lo que, a su vez, causa que las tareas siguientes de carga de datos de prueba y explotación de datos tampoco se puedan completar.

Por ello, llegada la fecha de entrega queda pendiente una parte de los procedimientos de base de datos y gran parte de la creación de datos de prueba y explotación de dichos datos.

Como medidas para mitigar este problema en el próximo (y último) hito se aumentarán las horas de dedicación aprovechando el periodo vacacional y se reducirán las horas de alguna de las tareas del siguiente hito.

Entrega final

En esta última entrega se completan las tareas pendientes del hito anterior a costa de ampliar las horas de dedicación semanales.

Asimismo se logran completar las tareas propias del hito para lograr realizar la entrega de la memoria y todos los productos asociados en el plazo marcado.

1.4.2. Análisis de riesgos

A la hora de prever posibles desviaciones de la planificación y de disponer de un plan de contingencia se han identificado los siguientes riesgos.

Tabla 3: Análisis de riesgos

Riesgo	Probabilidad	Impacto	Contingencia / mitigación
Tecnológicos:			
Fallo en el equipo personal de trabajo.	Media	Bajo	Todo el trabajo se guarda en el Drive de la UOC y sincronizado en dos portátiles que mantienen la copia local actualizada. Bastaría con cambiar de equipo.
Fallo en la conexión a internet.	Baja	Bajo	Se dispone de medios para establecer conexión a internet mediante 3G. En caso de urgencia a la hora de realizar una entrega se puede realizar mediante esta vía alternativa.
Caída de la web de la UOC.	Baja	Crítico	Para evitar un problema con una entrega si la web no está disponible se planificarán todas las entregas con, al menos, un día de antelación.

Laborales:			
Aumento de la carga de trabajo en el ámbito profesional.	Baja	Medio	Para evitar posibles problemas por un aumento en la carga de trabajo (o algún curso obligatorio fuera de horario) que reduzcan la disponibilidad de horas planificadas se podrán aumentar dichas horas reduciendo el tiempo libre disponible (en fines de semana o gastando días libres en el trabajo).
Familiares y de salud:			
Enfermedad incapacitante propia o de algún familiar de primer grado	Baja	Crítico	En caso de hospitalización o problemas graves de salud de algún familiar de primer grado se solicitaría ayuda al consultor o tutor de la asignatura para tratar de minimizar el impacto del tiempo perdido.

1.5. Breve resumen de productos obtenidos

Durante la elaboración de este trabajo se han obtenido los siguientes productos:

- Memoria. Documento en el que se detalla el proceso de elaboración del trabajo.
- Anexos. Documento en que se explica el código de las pruebas realizadas así como los resultados de las consultas.
- Scripts. Conjunto de scripts de base de datos con todo lo necesario para crear la base de datos completa e inicializarla así como los datos de pruebas.
- Autoinforme. Informe de autoevaluación del alumno.
- Vídeo. Presentación en vídeo del trabajo para el tribunal.

1.6. Breve descripción de los otros capítulos de la memoria

Capítulo 2: Análisis de los requisitos

Capítulo en el que se detallan los requisitos obtenidos a partir del enunciado y se exponen las decisiones de diseño tomadas.

Capítulo 3: Diseño de la Base de Datos

Capítulo en el que se detalla el proceso de diseño de la base de datos.

Se divide en tres etapas secuenciales que convierten el conjunto de requisitos obtenido en el capítulo anterior en un modelo válido de base de datos.

Capítulo 4: Implementación

En este capítulo se describen los métodos implementados en los paquetes de base de datos que dotan al sistema de las funcionalidades necesarias para cumplir con los requisitos.

Capítulo 5: Pruebas

Capítulo en el que se describe la metodología seguida para comprobar el correcto funcionamiento del sistema.

2. Análisis de los requisitos

Partiendo del enunciado se obtienen los siguientes requisitos a partir de los que se modelará la solución.

- Se deben poder introducir los datos relevantes de los principales actores: universidades, estudiantes, titulaciones, profesores responsables y empresas que ofrecen posiciones.
- De las empresas se deberá indicar el contacto de RRHH y cada una de las personas que controlarán a los alumnos.
- Las empresas tienen solicitantes internos que serán los responsables de los estudiantes.
- Las universidades reciben las ofertas de las empresas y las publican para sus estudiantes.
- Los estudiantes se apuntan a las ofertas que consideran.
- Las empresas entrevistarán a los candidatos que más les interesen, de dicha entrevista saldrán unos comentarios y una valoración.
- Los estudiantes seleccionados firman una colaboración con la empresa en la que consta la duración, el sueldo y los informes que la empresa deberá presentar.
- Durante la colaboración se realizan los informes pactados. Finalizada esta, se realiza el informe final con el resultado de aptitud.
- Se deben detallar también las inspecciones de trabajo que reciba una empresa.

Aparte del modelo que permita cumplir con todos estos requisitos se deberá crear un repositorio estadístico que permita responder mediante consultas simples a las siguientes consultas:

- 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.
- Top5 de empresas que han ofrecido más plazas para estudiantes en prácticas al CIC.
- Empresa que, entre todos los datos de que disponemos, ha recibido más inspecciones de trabajo (sin considerar el resultado de la inspección).
- Dado un año universitario en concreto, porcentaje de estudiantes, de cualquier universidad, que no superan las prácticas en empresas.
- 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.

Se han tomado las siguientes decisiones de diseño para obtener el modelo conceptual:

- Se define una relación de composición entre Universidad y Estudiante. Se entiende que no pueden existir estudiantes que no pertenezcan a una universidad.
- Se define una relación de composición entre Universidad y ProfesorResponsable. Se entiende que los profesores lo son porque pertenecen a una universidad.
- Se define una relación de composición entre Empresa y Solicitante. Se interpreta que las personas solicitantes que crean las ofertas y supervisan a las personas en prácticas pertenecen a una empresa.
- Se define una relación de herencia entre Informe e InformeFinal. Se decide que los informes finales son una especialización del resto de informes.

Para el diseño del repositorio estadístico se ha optado por modelar 3 entidades que contendrán los resultados actualizados para cada una de las consultas necesarias. Las entidades responden a criterios temporales.

Una entidad ReGeneral con los datos que no tienen que ver con periodos temporales. Una segunda entidad ReAnyActual con los datos del año universitario actual, la cual reseteará los valores calculados cada curso universitario para comenzar de nuevo de 0 mediante un procedimiento programado en la base de datos. Y una tercera entidad ReAnyUniv con los datos agrupados por año universitario que permita consultas en función del año universitario deseado.

Para mantener estas entidades con los valores actualizados serán necesarias otras 4 entidades en las que se mantienen algunos de los datos desagregados de cada una de las fuentes que deben alimentar los resultados agregados. Estas fuentes son: las titulaciones, los profesores, las universidades y las empresas.

Con este sistema para cada operación en el modelo principal de base de datos se actualizarán los datos de cálculo directo en las 3 entidades del repositorio

estadístico, aquellos datos que sean agregados se actualizarán en las 4 entidades intermedias y se agregarán en un paso posterior.

3. Diseño de la Base de Datos

3.1. Diseño conceptual

Para obtener el modelo conceptual se ha utilizado la metodología centralizada siguiendo una estrategia descendente¹.

A partir de las estructuras principales obtenidas en la fase de análisis de requisitos se ha ido refinando el modelo para obtener el resto de entidades y obtener así el siguiente diseño conceptual para la aplicación.

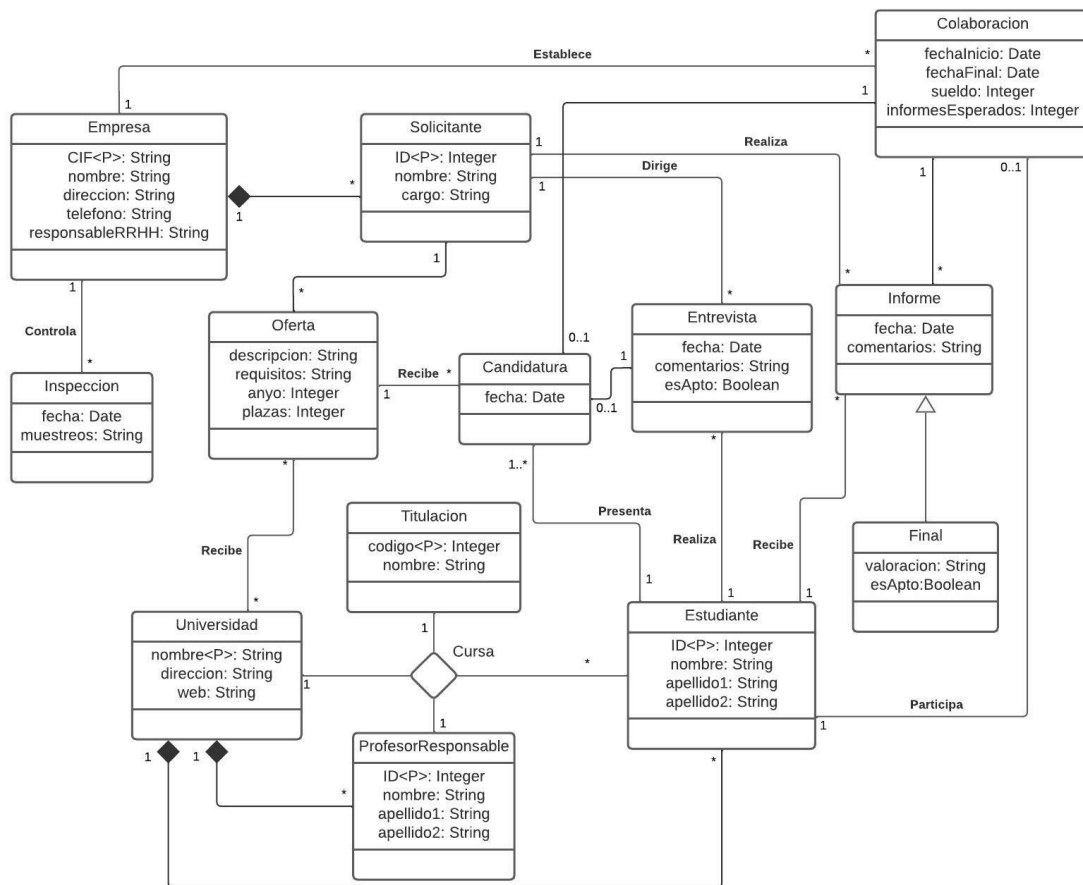


Figura 4: Modelo conceptual general

Partiendo de la entidad que representa a una universidad se establecen relaciones de composición con las entidades que representan a los estudiantes así como a los profesores. Junto con las titulaciones se establece una relación 4-aria para reflejar los estudios que cursa una persona.

Igualmente, las empresas establecen una relación de composición con las personas solicitantes. Éstas son quienes generan las ofertas que se envían a las universidades y a las que los estudiantes presentan candidaturas.

1 Se indican las metodologías y estrategias de diseño en el recurso bibliográfico 1. Diseño conceptual de bases de datos.

Se realizan las entrevistas de algunas de dichas candidaturas dirigidas por las personas solicitantes y en las que participan los estudiantes.

De aquellas candidaturas que hayan tenido una entrevista apta se generan colaboraciones entre empresas y estudiantes. Estas colaboraciones reciben informes de los cuales uno es final.

Por último, las empresas reciben inspecciones.

Para el repositorio estadístico se ha aplicado el mismo criterio pero el modelo obtenido es más simple ya que el objetivo es obtener las consultas con el menor esfuerzo computacional posible.

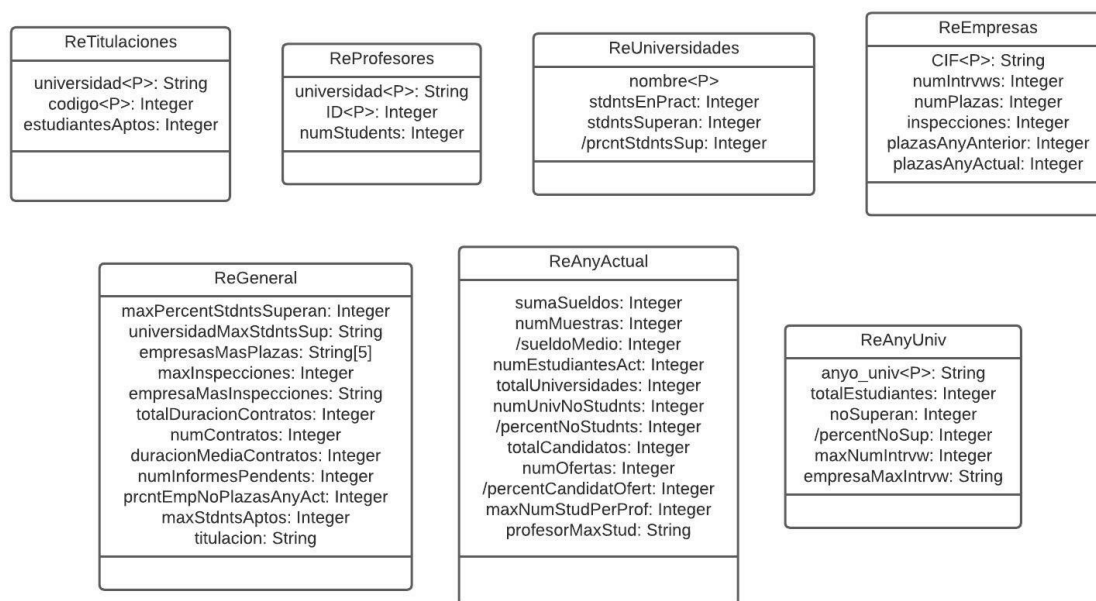


Figura 5: Modelo conceptual repositorio estadístico

Se han creado 3 entidades acotadas por ámbito temporal que contienen en todo momento los valores necesarios para satisfacer las consultas planteadas en el enunciado.

A partir de los procedimientos de modificación de los datos de la aplicación se irán actualizando los valores que pueden calcularse de manera directa mediante operaciones básicas como adición y sustracción.

Para los valores agregados se utilizan entidades con los datos estadísticos de las titulaciones, los profesores, las universidades y las empresas.

En estas entidades se irán actualizando los valores de la misma manera, pero, una vez actualizado el valor derivado del procedimiento de modificación se deberá realizar una operación de agregación para actualizar el valor agregado en la entidad correspondiente.

En el caso de los datos agregados que resultan de comparar diferentes valores no agregados, por ejemplo la empresa con más inspecciones, el criterio de

sustitución del valor agregado es que se supere el valor agregado anterior, en el ejemplo anterior únicamente cambiaría cuando otra empresa tenga más inspecciones que la empresa que tenía el máximo anterior.

3.2. Diseño lógico

A partir del modelo conceptual anterior y, siguiendo unas reglas de transformación², se obtiene el modelo relacional.

Las relaciones binarias 1..* se representarán mediante una clave foránea en el extremo opuesto al que tiene la multiplicidad máxima igual a 1.

Las relaciones binarias *.* se representarán mediante una nueva relación con claves foráneas a las relaciones originales.

Las relaciones binarias 1..1 se representarán mediante una clave foránea en el extremo que tiene la multiplicidad mínima igual a 0.

Las relaciones de composición se representarán como clave foránea en la relación que representa la entidad dependiente. Su clave primaria estará formada por el identificador del tipo de entidad concatenado con la clave foránea.

La generalización se representará con una relación para cada tipo de entidad. Cada relación contendrá las columnas de los atributos correspondientes a su tipo de entidad. En el caso de las relaciones que representan las subclases, además, tendrán el identificador con una restricción de clave foránea que referenciará la relación padre o superclase.

Los tipos de relación de aridad mayor que 2 se representan por relaciones. Lo que es preciso tener en cuenta son las multiplicidades, puesto que si hay alguna conectividad máxima igual a 1, aparecen claves alternativas.

Universidad (nombre, **direccion**, web)

Titulacion (codigo, **nombre**)

ProfesorResponsable (universidad_id, **nombre**, **apellido1**, apellido2)
{universidad} is foreign key to Universidad

Estudiante (universidad_id, **nombre**, **apellido1**, apellido2)
{universidad} is foreign key to Universidad

Cursa (universidad, titulacion, profesor, estudiante)
{universidad} is foreign key to Universidad
{titulacion} is foreign key to Titulacion
{profesor} is foreign key to ProfesorResponsable

² Se puede obtener más información sobre la transformación del modelo conceptual al modelo físico en el recurso bibliográfico 2. Diseño lógico de bases de datos.

{estudiante} is foreign key to Estudiante
(universidad, titulacion, estudiante) es clave alternativa
(universidad, profesor, estudiante) es clave alternativa

Empresa (CIF, **nombre**, **direccion**, **telefono**, **responsableRRHH**)

Inspeccion (ID, **empresa**, **fecha**, **muestreos**)
{empresa} is foreign key to Empresa

Solicitante (empresa, ID, **nombre**, **cargo**)
{empresa} is foreign key to Empresa

Oferta (ID, solicitante, **descripcion**, **requisitos**, **anyo**, **plazas**)
{solicitante} is foreign key to Solicitante

Recibe (universidad, oferta)
{universidad} is foreign key to Universidad
{oferta} is foreign key to Oferta

Candidatura (estudiante, oferta, **fecha**)
{estudiante} is foreign key to Estudiante
{oferta} is foreign key to Oferta

Entrevista (estudiante, candidatura, solicitante, **fecha**, **comentarios**, **esApto**)
{estudiante} is foreign key to Estudiante
{candidatura} is foreign key to Candidatura
{solicitante} is foreign key to Solicitante

Colaboracion (empresa, estudiante, candidatura, **fechaInicio**, **fechaFinal**, **sueldo**, **informesEsperados**)
{empresa} is foreign key to Empresa
{estudiante} is foreign key to Estudiante
{candidatura} is foreign key to Candidatura

Informe (colaboracion, solicitante, estudiante, fecha, **comentarios**)
{colaboracion} is foreign key to Colaboracion
{solicitante} is foreign key to Solicitante
{estudiante} is foreign key to Estudiante

Final (informe, **valoracion**, **esApto**)
{informe} is foreign key to Informe

3.3. Diseño físico

Partiendo del modelo relacional anterior y teniendo en cuenta que se va a implementar esta base de datos en un SGBD Oracle, se transforma dicho modelo en un modelo virtual³ para poderlo implementar.

3 Se puede obtener más información sobre la transformación al modelo virtual en el recurso bibliográfico 3. Diseño físico de bases de datos.

La nomenclatura de las tablas pasa de ser Pascal a ser nombres con todo letras minúsculas y separadas por _.

Los tipos de datos utilizados son:

String se implementa como VARCHAR2 de la longitud máxima estimada de los valores a almacenar.

Integer se implementa como NUMBER con la precisión y escala suficiente estimada.

Date se implementa como DATE.

Boolean se implementa como CHAR con una restricción de que su valor debe ser 0 o 1.

Se crean todas las tablas obtenidas en el punto anterior.

También se deberán generar las claves primarias y los índices para las claves foráneas en cada tabla.

Las tablas universidad, profesor_responsable, estudiante, empresa, inspeccion, solicitante y oferta pasan a tener clave primaria sintética.

Para las tablas con claves sintéticas se generará una secuencia y un trigger de inserción.

Además de las tablas resultantes de la transformación del diseño conceptual se crea una tabla llamada logs para almacenar los logs de las llamadas a los procedimientos ABM de los datos.

Para la implementación física se ha optado por utilizar la plataforma Oracle Cloud mediante una cuenta de Oracle Cloud Free Tier.

El diagrama generado por el DataModeler de Oracle queda:

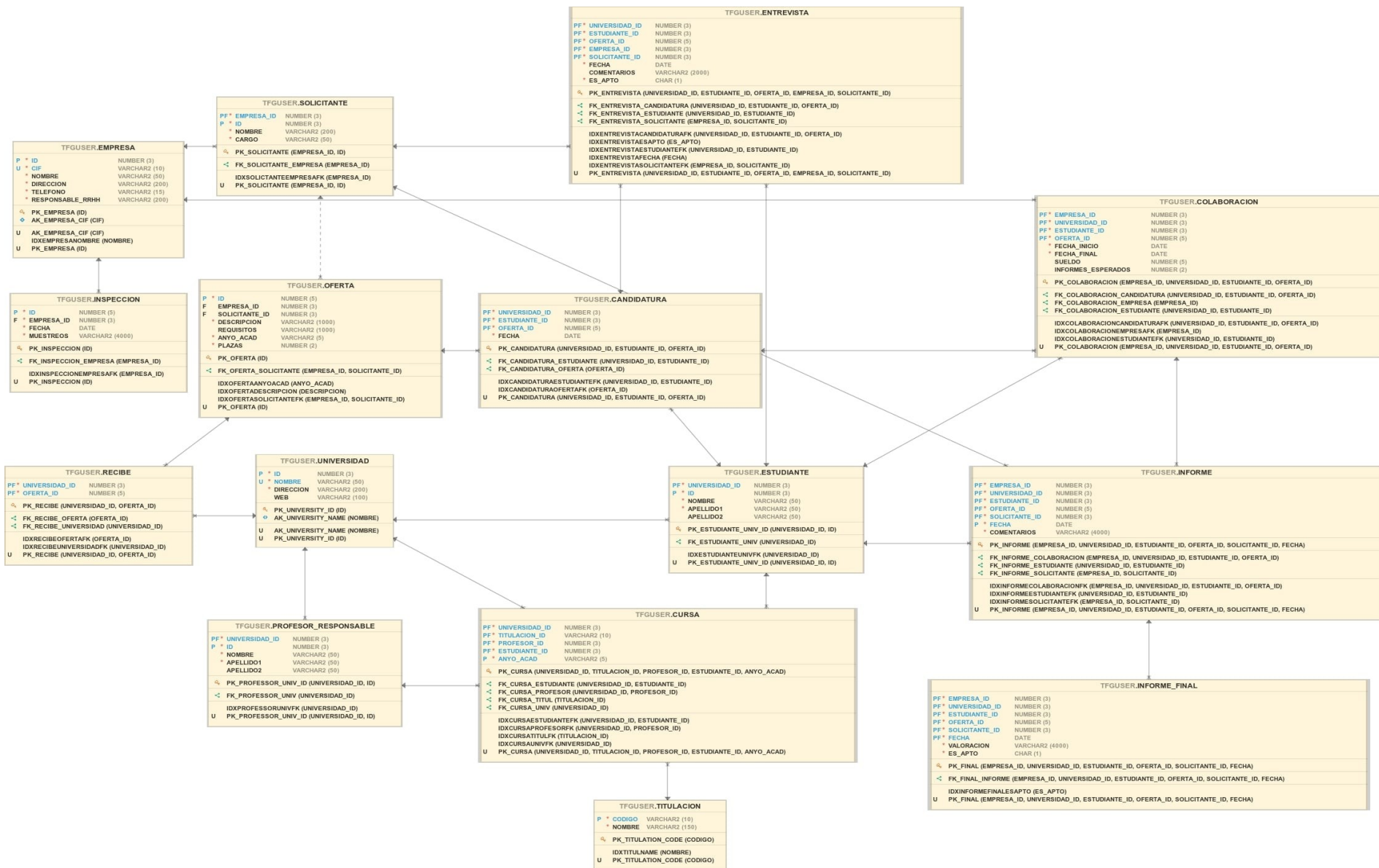


Figura 6: Modelo fisico

Del repositorio estadístico:

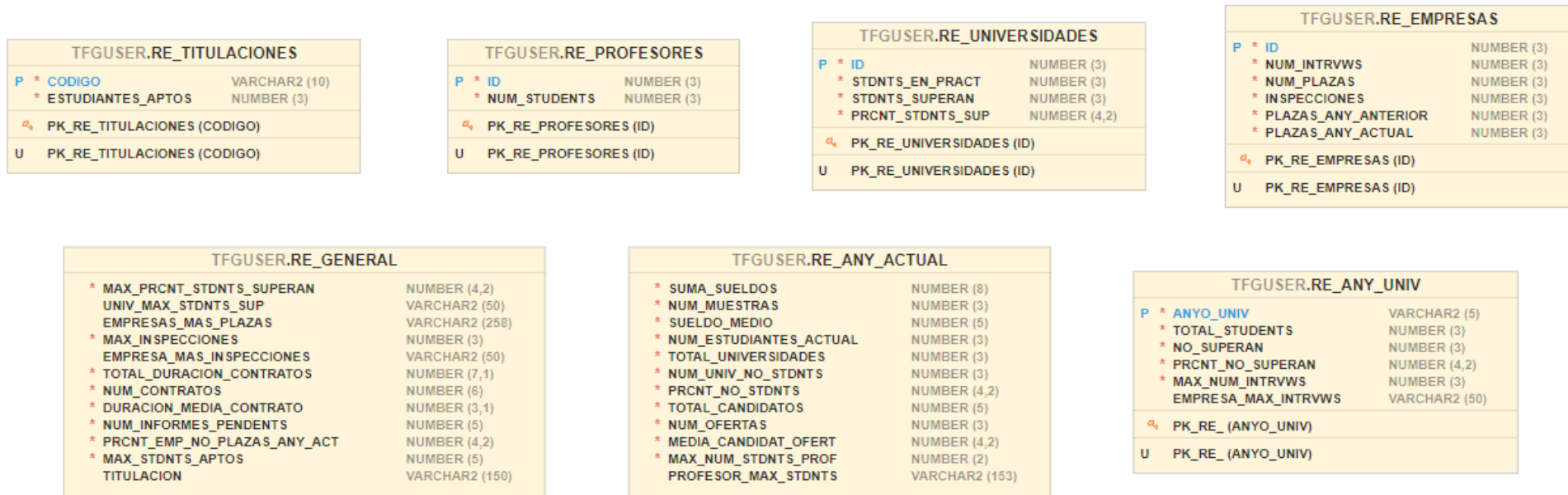


Figura 7: Modelo físico del repositorio estadístico

4. Implementación

La implementación, como ya se ha comentado anteriormente, se realiza sobre una instancia de Oracle Cloud Free Tier creada expresamente para este trabajo.

Se comienza por crear la estructura de tablas mediante el script 01 – Tablas.sql.

Una vez creadas las tablas se crea toda la estructura de índices de éstas mediante el script 02 – Indices.sql.

Para las restricciones de integridad se utilizan dos scripts 03 – PrimaryKeys.sql y 04 – ForeignKeys.sql.

Después se debe ejecutar el script que crea las secuencias de las claves sintéticas 05 – Secuencias.sql y el que crea los triggers que las utilizan en las inserciones 06 – Triggers.sql.

Posteriormente se deberá ejecutar el script 07 – Funciones.sql que crea funciones accesibles desde cualquiera de los paquetes.

Para la creación de los distintos paquetes se deberán ejecutar en orden los scripts 08 – pq_log.sql, 09 – pq_dw.sql y 10 – pq_abm.sql.

Por último se ejecuta el script de inicialización del repositorio estadístico 11 – Inicializacion.sql.

4.1. Procedimientos ABM.

Para la gestión de la operativa ABM sobre los datos se establecen varias restricciones a nivel de las operaciones que se pueden realizar.

Para mantener la integridad del repositorio estadístico no se permite la supresión de los registros creados en las siguiente tablas: universidad, titulacion, profesor_responsable, estudiante, empresa.

A continuación se describen los procedimientos para las operaciones permitidas para cada entidad del modelo conceptual.

4.1.1. Universidad

Se crean dos procedimientos para gestionar las altas y las modificaciones de las universidades. En las modificaciones se permite cambiar el nombre (sin posibilidad de duplicar el de otra universidad existente), la dirección y la página web.

Nombre	altaUniversidad
Descripción	Se crea una universidad nueva
Entradas	nombre, direccion, web
Salidas	Ejecución correcta = 'OK' Fracaso = 'ERROR + TIPO DE ERROR'
Proceso	Se verifican los datos obligatorios Se inserta el registro en la tabla Se llama al procedimiento addUniversidad del DW Se controlan las posibles excepciones Se guarda log de la operación, los parámetros y el resultado

Nombre	modUniversidad
Descripción	Se modifican los datos de una universidad
Entradas	id, nombre, direccion, web
Salidas	Ejecución correcta = 'OK' Fracaso = 'ERROR + TIPO DE ERROR'
Proceso	Se verifica que el id existe Se verifican los datos obligatorios Se modifica el registro en la tabla Si cambia el nombre se llama al procedimiento updUniversidad de actualización del DW Se controlan las posibles excepciones Se guarda log de la operación, los parámetros y el resultado

4.1.2. Titulacion

Se crean dos procedimientos para gestionar las altas y las modificaciones de las titulaciones. En las modificaciones se permite cambiar el nombre de la titulación.

Nombre	altaTitulacion
Descripción	Se crea una titulación nueva
Entradas	codigo, nombre
Salidas	Ejecución correcta = 'OK' Fracaso = 'ERROR + TIPO DE ERROR'
Proceso	Se verifican los datos obligatorios Se inserta el registro en la tabla Se llama al procedimiento addTitulacion del DW Se controlan las posibles excepciones Se guarda log de la operación, los parámetros y el resultado

Nombre	modTitulacion
--------	---------------

Descripción	Se modifican los datos de una titulación
Entradas	codigo, nombre
Salidas	Ejecución correcta = 'OK' Fracaso = 'ERROR + TIPO DE ERROR'
Proceso	Se verifica que la clave primaria existe Se verifican los datos obligatorios Se modifica el registro en la tabla Se llama al procedimiento updTitulacion del DW Se controlan las posibles excepciones Se guarda log de la operación, los parámetros y el resultado

4.1.3. ProfesorResponsable

Se crean dos procedimientos para gestionar las altas y las modificaciones del profesorado. En las modificaciones se permite cambiar el nombre y los apellidos.

Nombre	altaProfesor
Descripción	Se crea un nuevo registro de profesor responsable
Entradas	universidad_id, nombre, apellido1, apellido2
Salidas	Ejecución correcta = 'OK' Fracaso = 'ERROR + TIPO DE ERROR'
Proceso	Se verifica que la universidad exista Se verifican los datos obligatorios Se inserta el registro en la tabla Se llama al procedimiento addProfesor del DW Se controlan las posibles excepciones Se guarda log de la operación, los parámetros y el resultado

Nombre	modProfesor
Descripción	Se modifican los datos de un profesor responsable
Entradas	universidad_id, id, nombre, apellido1, apellido2
Salidas	Ejecución correcta = 'OK' Fracaso = 'ERROR + TIPO DE ERROR'
Proceso	Se verifica que la clave primaria existe Se verifican los datos obligatorios Se modifica el registro en la tabla Se llama al procedimiento updProfesor del DW Se controlan las posibles excepciones Se guarda log de la operación, los parámetros y el resultado

4.1.4. Estudiante

Se crean dos procedimientos para gestionar las altas y las modificaciones de las personas estudiantes. En las modificaciones se permite cambiar el nombre y los apellidos.

Nombre	altaEstudiante
Descripción	Se da de alta a un nuevo estudiante
Entradas	universidad_id, nombre, apellido1, apellido2
Salidas	Ejecución correcta = 'OK' Fracaso = 'ERROR + TIPO DE ERROR'
Proceso	Se verifica que la universidad exista Se verifican los datos obligatorios Se inserta el registro en la tabla Se controlan las posibles excepciones Se guarda log de la operación, los parámetros y el resultado

Nombre	modEstudiante
Descripción	Se modifican los datos de un estudiante
Entradas	universidad_id, id, nombre, apellido1, apellido2
Salidas	Ejecución correcta = 'OK' Fracaso = 'ERROR + TIPO DE ERROR'
Proceso	Se verifica que la clave primaria existe Se verifican los datos obligatorios Se modifica el registro en la tabla Se controlan las posibles excepciones Se guarda log de la operación, los parámetros y el resultado

4.1.5. Cursa

Se crean dos procedimientos para gestionar las altas y las bajas de los estudios que cursa una persona. Dado que todos los campos de la tabla forman parte de la clave, no se permiten modificaciones.

Nombre	altaCursa
Descripción	Se da de alta una persona estudiante en una titulación de una universidad con un profesor responsable en un curso concreto.
Entradas	universidad_id, titulacion_id, profesor_id, estudiante_id, anyo_acad
Salidas	Ejecución correcta = 'OK' Fracaso = 'ERROR + TIPO DE ERROR'
Proceso	Se verifican los datos obligatorios Se inserta el registro en la tabla

	Se controlan las posibles excepciones Se guarda log de la operación, los parámetros y el resultado
--	---

Nombre	bajaCursa
Descripción	Se elimina la relación de cursar estudios
Entradas	universidad_id, titulacion_id, profesor_id, estudiante_id, anyo_acad
Salidas	Ejecución correcta = 'OK' Fracaso = 'ERROR + TIPO DE ERROR'
Proceso	Se comprueba que la clave primaria exista Se comprueba que la persona estudiante no haya presentado una candidatura en este año académico Se elimina el registro de la tabla Se controlan las posibles excepciones Se guarda log de la operación, los parámetros y el resultado

4.1.6. Empresa

Se crean dos procedimientos para gestionar las altas y las modificaciones de las empresas. En las modificaciones se permite cambiar el CIF (sin posibilidad de duplicar el de otra empresa existente), el nombre, la dirección, el teléfono y la persona responsable de RRHH.

Nombre	altaEmpresa
Descripción	Se da de alta una empresa en la base de datos
Entradas	cif, nombre, direccion, telefono, responsable_rrhh
Salidas	Ejecución correcta = 'OK' Fracaso = 'ERROR + TIPO DE ERROR'
Proceso	Se verifican los datos obligatorios Se inserta el registro en la tabla Se llama al procedimiento addEmpresa del DW Se controlan las posibles excepciones Se guarda log de la operación, los parámetros y el resultado

Nombre	modEmpresa
Descripción	Se modifican los datos de una empresa existente
Entradas	id, cif, nombre, direccion, telefono, responsable_rrhh
Salidas	Ejecución correcta = 'OK' Fracaso = 'ERROR + TIPO DE ERROR'
Proceso	Se verifica que el id existe Se verifican los datos obligatorios

	Se modifica el registro en la tabla Si cambia el nombre de la empresa se llama al procedimiento updEmpresa de actualización del DW Se controlan las posibles excepciones Se guarda log de la operación, los parámetros y el resultado
--	--

4.1.7. Solicitante

Se crean tres procedimientos para gestionar las altas, modificaciones y bajas de las personas solicitantes. En las modificaciones se permite cambiar el nombre y el cargo. En las bajas se elimina el registro siempre que no existan registros relacionados con el que se quiere eliminar.

Nombre	altaSolicitante
Descripción	Se da de alta una persona solicitante en la base de datos
Entradas	empresa_id, nombre, cargo
Salidas	Ejecución correcta = 'OK' Fracaso = 'ERROR + TIPO DE ERROR'
Proceso	Se verifica que la empresa exista Se verifican los datos obligatorios Se inserta el registro en la tabla Se controlan las posibles excepciones Se guarda log de la operación, los parámetros y el resultado

Nombre	modSolicitante
Descripción	Se modifican los datos de una persona solicitante
Entradas	empresa_id, id, nombre, cargo
Salidas	Ejecución correcta = 'OK' Fracaso = 'ERROR + TIPO DE ERROR'
Proceso	Se verifica que la clave primaria existe Se verifican los datos obligatorios Se modifica el registro en la tabla Se controlan las posibles excepciones Se guarda log de la operación, los parámetros y el resultado

Nombre	bajaSolicitante
Descripción	Se elimina una persona solicitante de la base de datos
Entradas	empresa_id, id
Salidas	Ejecución correcta = 'OK' Fracaso = 'ERROR + TIPO DE ERROR'
Proceso	Se comprueba que la clave primaria exista Se comprueba que la persona solicitante no se encuentre

	relacionada con alguna oferta de prácticas Se elimina el registro de la tabla Se controlan las posibles excepciones Se guarda log de la operación, los parámetros y el resultado
--	---

4.1.8. Oferta

Se crean tres procedimientos para gestionar las altas, modificaciones y bajas de las ofertas. En las modificaciones se permite cambiar la descripción, los requisitos y el número de plazas. En las bajas se elimina el registro siempre que no existan registros relacionados con el que se quiere eliminar.

Nombre	altaOferta
Descripción	Se da de alta una oferta en el sistema
Entradas	empresa_id, solicitante_id, descripcion, requisitos, plazas
Salidas	Ejecución correcta = 'OK' Fracaso = 'ERROR + TIPO DE ERROR'
Proceso	Se verifica que la persona solicitante exista Se verifican los datos obligatorios Se calcula el año académico actual Se inserta el registro en la tabla Se insertan tantos registros como universidades haya en la tabla recibe Se llama al procedimiento addOferta de actualización del DW Se controlan las posibles excepciones Se guarda log de la operación, los parámetros y el resultado

Nombre	modOferta
Descripción	Se modifican los datos de una oferta existente
Entradas	id, descripcion, requisitos, plazas
Salidas	Ejecución correcta = 'OK' Fracaso = 'ERROR + TIPO DE ERROR'
Proceso	Se verifica que la clave primaria existe Se verifican los datos obligatorios Si se reduce el número de plazas se comprueba que no sea inferior al de colaboraciones ya establecidas Se modifica el registro en la tabla Si cambia el número de plazas se llama al procedimiento updOferta de actualización del DW Se controlan las posibles excepciones Se guarda log de la operación, los parámetros y el resultado

Nombre	bajaOferta
Descripción	Se elimina una oferta del sistema

Entradas	id
Salidas	Ejecución correcta = 'OK' Fracaso = 'ERROR + TIPO DE ERROR'
Proceso	Se comprueba que la clave primaria exista Se comprueba que no se hayan presentado candidaturas a dicha oferta Se eliminan todos los registros asociados en la tabla recibe Se elimina el registro de la tabla Se llama al procedimiento delOferta de actualización del DW Se controlan las posibles excepciones Se guarda log de la operación, los parámetros y el resultado

4.1.9. Candidatura

Se crean tres procedimientos para gestionar las altas, modificaciones y bajas de las candidaturas. En las modificaciones se permite cambiar la fecha. En las bajas se elimina el registro siempre que no existan registros relacionados con el que se quiere eliminar.

Nombre	altaCandidatura
Descripción	Se registra una candidatura en el sistema
Entradas	universidad_id, estudiante_id, oferta_id, fecha
Salidas	Ejecución correcta = 'OK' Fracaso = 'ERROR + TIPO DE ERROR'
Proceso	Se verifica que el estudiante exista Se verifica que la oferta exista Se verifican los datos obligatorios Se inserta el registro en la tabla Se llama al procedimiento addCandidatura del DW Se controlan las posibles excepciones Se guarda log de la operación, los parámetros y el resultado

Nombre	modCandidatura
Descripción	Se modifican los datos de una candidatura
Entradas	universidad_id, estudiante_id, oferta_id, fecha
Salidas	Ejecución correcta = 'OK' Fracaso = 'ERROR + TIPO DE ERROR'
Proceso	Se verifica que la clave primaria existe Se verifican los datos obligatorios Se modifica el registro en la tabla Se controlan las posibles excepciones Se guarda log de la operación, los parámetros y el resultado

Nombre	bajaCandidatura
Descripción	Se elimina una candidatura del sistema
Entradas	universidad_id, estudiante_id, oferta_id
Salidas	Ejecución correcta = 'OK' Fracaso = 'ERROR + TIPO DE ERROR'
Proceso	Se comprueba que la clave primaria exista Se comprueba que no se haya realizado una entrevista para dicha candidatura Se elimina el registro de la tabla Se llama al procedimiento delCandidatura del DW Se controlan las posibles excepciones Se guarda log de la operación, los parámetros y el resultado

4.1.10. Entrevista

Se crean tres procedimientos para gestionar las altas, modificaciones y bajas de las entrevistas. En las modificaciones se permite cambiar la fecha, los comentarios o la aptitud. En las bajas se elimina el registro siempre que no existan registros relacionados con el que se quiere eliminar.

Nombre	altaEntrevista
Descripción	Se registra una entrevista realizada en la base de datos
Entradas	universidad_id, estudiante_id, oferta_id, empresa_id, solicitante_id, fecha, comentarios, es_apto
Salidas	Ejecución correcta = 'OK' Fracaso = 'ERROR + TIPO DE ERROR'
Proceso	Se verifica que el estudiante exista Se verifica que la persona solicitante exista Se verifica que la candidatura exista y sea de una oferta realizada por la persona solicitante Se verifican los datos obligatorios Se inserta el registro en la tabla Se llama al procedimiento addEntrevista del DW Se controlan las posibles excepciones Se guarda log de la operación, los parámetros y el resultado

Nombre	modEntrevista
Descripción	Se modifican los datos de una entrevista ya existente
Entradas	universidad_id, estudiante_id, oferta_id, empresa_id, solicitante_id, fecha, comentarios, es_apto
Salidas	Ejecución correcta = 'OK' Fracaso = 'ERROR + TIPO DE ERROR'
Proceso	Se verifica que la clave primaria existe

	<p>Se verifican los datos obligatorios</p> <p>Se comprueba que si pasa de apto a no apto no exista una colaboración ya firmada con la candidatura entrevistada</p> <p>Se modifica el registro en la tabla</p> <p>Se controlan las posibles excepciones</p> <p>Se guarda log de la operación, los parámetros y el resultado</p>
--	--

Nombre	bajaEntrevista
Descripción	Se elimina una entrevista registrada en la base de datos
Entradas	universidad_id, estudiante_id, oferta_id, empresa_id, solicitante_id
Salidas	Ejecución correcta = 'OK' Fracaso = 'ERROR + TIPO DE ERROR'
Proceso	<p>Se comprueba que la clave primaria exista</p> <p>Se comprueba que no exista una colaboración ya firmada en relación a la candidatura entrevistada</p> <p>Se elimina el registro de la tabla</p> <p>Se llama al procedimiento delEntrevista del DW</p> <p>Se controlan las posibles excepciones</p> <p>Se guarda log de la operación, los parámetros y el resultado</p>

4.1.11. Colaboracion

Se crean tres procedimientos para gestionar las altas, modificaciones y bajas de las colaboraciones. En las modificaciones se permite cambiar las fechas de inicio y fin, el sueldo o la cantidad de informes esperados. En las bajas se elimina el registro siempre que no existan registros relacionados con el que se quiere eliminar.

Nombre	altaColaboracion
Descripción	Se registra en la base de datos la firma de un contrato de prácticas entre una empresa y una persona estudiante
Entradas	empresa_id, universidad_id, estudiante_id, oferta_id, fecha_inicio, fecha_final, sueldo, informes_esperados
Salidas	Ejecución correcta = 'OK' Fracaso = 'ERROR + TIPO DE ERROR'
Proceso	<p>Se verifica que la empresa exista</p> <p>Se verifica que el estudiante exista</p> <p>Se verifica que la candidatura exista y haya recibido una entrevista con resultado de aptitud</p> <p>Se verifican los datos obligatorios</p> <p>Se inserta el registro en la tabla</p> <p>Se llama al procedimiento addColaboracion del DW</p> <p>Se controlan las posibles excepciones</p> <p>Se guarda log de la operación, los parámetros y el resultado</p>

Nombre	modColaboracion
Descripción	Se modifican los datos de una colaboración ya registrada en la base de datos
Entradas	empresa_id, universidad_id, estudiante_id, oferta_id, fecha_inicio, fecha_final, sueldo, informes_esperados
Salidas	Ejecución correcta = 'OK' Fracaso = 'ERROR + TIPO DE ERROR'
Proceso	Se verifica que la clave primaria existe Se verifican los datos obligatorios Se comprueba que no se reduzcan la cantidad de informes esperados por debajo de la cantidad de informes ya emitidos Se modifica el registro en la tabla Se llama al procedimiento updColaboracion del DW Se controlan las posibles excepciones Se guarda log de la operación, los parámetros y el resultado

Nombre	bajaColaboracion
Descripción	Se elimina el registro de una colaboración del sistema
Entradas	empresa_id, universidad_id, estudiante_id, oferta_id
Salidas	Ejecución correcta = 'OK' Fracaso = 'ERROR + TIPO DE ERROR'
Proceso	Se comprueba que la clave primaria exista Se comprueba que no existan informes relacionados con la colaboración Se elimina el registro de la tabla Se llama al procedimiento delColaboracion del DW Se controlan las posibles excepciones Se guarda log de la operación, los parámetros y el resultado

4.1.12. Inspeccion

Se crean tres procedimientos para gestionar las altas, modificaciones y bajas de las inspecciones. En las modificaciones se permite cambiar la fecha o los muestreos realizados. En las bajas se elimina el registro.

Nombre	altaInspeccion
Descripción	Se registra una inspección realizada en una empresa
Entradas	empresa_id, fecha, muestreos
Salidas	Ejecución correcta = 'OK' Fracaso = 'ERROR + TIPO DE ERROR'
Proceso	Se verifica que la empresa exista Se verifican los datos obligatorios

	Se inserta el registro en la tabla Se llama al procedimiento addInspeccion del DW Se controlan las posibles excepciones Se guarda log de la operación, los parámetros y el resultado
--	---

Nombre	modInspeccion
Descripción	Se modifican los datos de una inspección ya registrada
Entradas	id, fecha, muestreos
Salidas	Ejecución correcta = 'OK' Fracaso = 'ERROR + TIPO DE ERROR'
Proceso	Se verifica que la clave primaria existe Se verifican los datos obligatorios Se modifica el registro en la tabla Se controlan las posibles excepciones Se guarda log de la operación, los parámetros y el resultado

Nombre	bajaInspeccion
Descripción	Se elimina una inspección registrada en la base de datos
Entradas	id
Salidas	Ejecución correcta = 'OK' Fracaso = 'ERROR + TIPO DE ERROR'
Proceso	Se comprueba que la clave primaria exista Se elimina el registro de la tabla Se llama al procedimiento delInspeccion del DW Se controlan las posibles excepciones Se guarda log de la operación, los parámetros y el resultado

4.1.13. Informe

Se crean tres procedimientos para gestionar las altas, modificaciones y bajas de los informes. En las modificaciones se permite cambiar los comentarios. En las bajas se elimina el registro siempre que no existan registros relacionados con el que se quiere eliminar.

Nombre	altaInforme
Descripción	Se da de alta un informe relativo a una colaboración
Entradas	empresa_id, universidad_id, estudiante_id, oferta_id, solicitante_id, fecha, comentarios
Salidas	Ejecución correcta = 'OK' Fracaso = 'ERROR + TIPO DE ERROR'
Proceso	Se verifica que la colaboración exista Se verifica que la persona solicitante exista

	<p>Se verifica que la persona estudiante exista</p> <p>Se verifican los datos obligatorios</p> <p>Se inserta el registro en la tabla</p> <p>Se llama al procedimiento addInforme de actualización del DW</p> <p>Se controlan las posibles excepciones</p> <p>Se guarda log de la operación, los parámetros y el resultado</p>
--	---

Nombre	modInforme
Descripción	Se modifica un informe ya registrado
Entradas	empresa_id, universidad_id, estudiante_id, oferta_id, solicitante_id, fecha, comentarios
Salidas	Ejecución correcta = 'OK' Fracaso = 'ERROR + TIPO DE ERROR'
Proceso	<p>Se verifica que la clave primaria existe</p> <p>Se verifican los datos obligatorios</p> <p>Se modifica el registro en la tabla</p> <p>Se controlan las posibles excepciones</p> <p>Se guarda log de la operación, los parámetros y el resultado</p>

Nombre	bajaInforme
Descripción	Se elimina un informe de la base de datos
Entradas	empresa_id, universidad_id, estudiante_id, oferta_id, solicitante_id, fecha
Salidas	Ejecución correcta = 'OK' Fracaso = 'ERROR + TIPO DE ERROR'
Proceso	<p>Se comprueba que la clave primaria exista</p> <p>Se comprueba que no exista un informe final asociado</p> <p>Se elimina el registro de la tabla</p> <p>Se llama al procedimiento delInforme de actualización del DW</p> <p>Se controlan las posibles excepciones</p> <p>Se guarda log de la operación, los parámetros y el resultado</p>

4.1.14. InformeFinal

Se crean tres procedimientos para gestionar las altas, modificaciones y bajas de los informes finales. En las modificaciones se permite cambiar los comentarios, la valoración o la aptitud. En las bajas se elimina el registro.

Nombre	altaInformeFinal
Descripción	Se registra un informe final en el sistema
Entradas	empresa_id, universidad_id, estudiante_id, oferta_id, solicitante_id, fecha, comentarios, valoracion, es_apt
Salidas	Ejecución correcta = 'OK'

	Fracaso = 'ERROR + TIPO DE ERROR'
Proceso	<p>Se verifica que la colaboración exista</p> <p>Se verifica que la persona solicitante exista</p> <p>Se verifica que la persona estudiante exista</p> <p>Se verifican los datos obligatorios</p> <p>Se inserta el registro padre en la tabla Informe</p> <p>Se inserta el registro en la tabla</p> <p>Se llama al procedimiento addInformeFinal del DW</p> <p>Se controlan las posibles excepciones</p> <p>Se guarda log de la operación, los parámetros y el resultado</p>

Nombre	modInformeFinal
Descripción	Se modifican los datos de un informe final ya registrado en el sistema
Entradas	empresa_id, universidad_id, estudiante_id, oferta_id, solicitante_id, fecha, comentarios, valoracion, es_apto
Salidas	<p>Ejecución correcta = 'OK'</p> <p>Fracaso = 'ERROR + TIPO DE ERROR'</p>
Proceso	<p>Se verifica que la clave primaria existe</p> <p>Se verifican los datos obligatorios</p> <p>Si cambian los comentarios se modifica el registro padre en la tabla Informe</p> <p>Si cambia la valoración o el resultado de aptitud se modifica el registro en la tabla</p> <p>Se llama al procedimiento updInformeFinal del DW</p> <p>Se controlan las posibles excepciones</p> <p>Se guarda log de la operación, los parámetros y el resultado</p>

Nombre	bajaInformeFinal
Descripción	Se elimina un informe final de la base de datos
Entradas	empresa_id, universidad_id, estudiante_id, oferta_id, solicitante_id, fecha
Salidas	<p>Ejecución correcta = 'OK'</p> <p>Fracaso = 'ERROR + TIPO DE ERROR'</p>
Proceso	<p>Se comprueba que la clave primaria exista</p> <p>Se elimina el registro de la tabla</p> <p>Se elimina el registro padre de la tabla Informe</p> <p>Se llama al procedimiento delInformeFinal del DW</p> <p>Se controlan las posibles excepciones</p> <p>Se guarda log de la operación, los parámetros y el resultado</p>

4.2. Gestión del DataWarehouse.

Para la actualización de los datos del DataWarehouse se crean varios procedimientos mediante los cuales se actualizan las tablas necesarias y se recalculan los valores finales en función de los datos modificados en el origen.

En el caso de los procedimientos de actualización del repositorio estadístico se ha optado por no devolver una salida y no interrumpir la ejecución del procedimiento de ABM inicial. En caso de producirse una excepción durante la ejecución de un método se insertará un registro con la información detallada del error en la tabla *logs_dw* para que pueda ser tratado posteriormente y no interrumpir un procedimiento de modificación de los datos de origen.

Nombre	addUniversidad
Descripción	Se da de alta una universidad en las tablas estadísticas
Entradas	id
Salidas	
Proceso	Se añade una nueva fila en la tabla re_universidades con el id y el resto de valores a 0 Se incrementa en 1 el valor de re_any_actual.total_universidades

Nombre	updUniversidad
Descripción	Se modifica el nombre de la universidad en las tablas estadísticas
Entradas	id, nombre
Salidas	
Proceso	Si el valor de re_general.univ_max_stdnts_sup_id es el del id, se actualiza re_general.univ_max_stdnts_sup con el nombre

Nombre	addTitulacion
Descripción	Se da de alta una titulación en las tablas estadísticas
Entradas	codigo
Salidas	
Proceso	Se añade una nueva fila en la tabla re_titulaciones con el código de la titulación y el resto de valores a 0

Nombre	updTitulacion
Descripción	Se modifica el nombre de la titulación en las tablas estadísticas
Entradas	codigo, nombre
Salidas	

Proceso	Si el valor de re_general.titulacion_id es igual a codigo, se actualiza re_general.titulacion con el nombre
---------	---

Nombre	addProfesor
Descripción	Se da de alta un profesor en las tablas estadísticas
Entradas	id
Salidas	
Proceso	Se añade una nueva fila en la tabla re_profesores con el id y el resto de valores a 0

Nombre	updProfesor
Descripción	Se modifica el nombre de un profesor en las tablas estadísticas
Entradas	id, nombreCompleto
Salidas	
Proceso	Si el valor de re_any_actual.profesor_max_stdnts_id es igual a id, se actualiza re_any_actual.profesor_max_stdnts con nombreCompleto

Nombre	addEmpresa
Descripción	Se da de alta una empresa en el repositorio estadístico
Entradas	id
Salidas	
Proceso	Se añade una nueva fila en la tabla re_empresas con el id y el resto de valores a 0

Nombre	updEmpresa
Descripción	Se modifica el nombre de una empresa en el repositorio estadístico
Entradas	id, nombre
Salidas	
Proceso	Si re_any_univ.empresa_max_intrvws_id es igual a id, se actualiza el valor de re_any_univ.empresa_max_intrvws con nombre Si re_general.empresas_mas_plazas_id contiene id, se actualiza el valor de re_general.empresas_mas_plazas Si re_general.empresa_mas_inspecciones_id es igual a id, se actualiza el valor de re_general.empresa_mas_inspecciones con nombre

Nombre	addOferta
--------	-----------

Descripción	Se modifican los valores del repositorio estadístico en los que intervienen las ofertas
Entradas	idEmpresa, numPlazas
Salidas	
Proceso	Se aumenta en 1 el valor de re_any_actual.num_ofertas Se aumenta en numPlazas el valor de re_empresas.num_plazas correspondiente a idEmpresa Se aumenta en numPlazas el valor de re_empresas.plazas_any_actual correspondiente a idEmpresa Se recalcula re_any_actual.media_candidat_ofert Se recalcula re_general.empresas_mas_plazas Se recalcula re_general.prcnt_emp_no_plazas_any_act

Nombre	updOferta
Descripción	Se modifican los valores del repositorio estadístico para ajustarlos a la nueva cantidad de plazas
Entradas	idEmpresa, varNumPlazas
Salidas	
Proceso	Se aumenta en varNumPlazas el valor de re_empresas.num_plazas correspondiente a idEmpresa Se aumenta en varNumPlazas el valor de re_empresas.plazas_any_actual correspondiente a idEmpresa Se recalcula re_general.empresas_mas_plazas Se recalcula re_general.prcnt_emp_no_plazas_any_act

Nombre	delOferta
Descripción	Se modifican los valores del repositorio estadístico para tener en cuenta que desaparece una oferta
Entradas	idEmpresa, numPlazas
Salidas	
Proceso	Se reduce en 1 el valor de re_any_actual.num_ofertas Se reduce en numPlazas el valor de re_empresas.num_plazas correspondiente a idEmpresa Se reduce en numPlazas el valor de re_empresas.plazas_any_actual correspondiente a idEmpresa Se recalcula re_any_actual.media_candidat_ofert Se recalcula re_general.empresas_mas_plazas Se recalcula re_general.prcnt_emp_no_plazas_any_act

Nombre	addCandidatura
Descripción	Se añade una candidatura al repositorio estadístico
Entradas	

Salidas	
Proceso	Se aumenta en 1 el valor de re_any_actual.total_candidatos Se recalcula re_any_actual.media_candidat_ofert

Nombre	delCandidatura
Descripción	Se elimina la candidatura del repositorio estadístico
Entradas	
Salidas	
Proceso	Se reduce en 1 el valor de re_any_actual.total_candidatos Se recalcula re_any_actual.media_candidat_ofert

Nombre	addEntrevista
Descripción	Se añade una entrevista al repositorio estadístico
Entradas	idEmpresa
Salidas	
Proceso	Se aumenta en 1 el valor de re_empresas.num_intrvws para idEmpresa Se recalculan re_any_univ.max_num_intrvws , re_any_univ.empresa_max_intrvws_id y re_any_univ.empresa_max_intrvws

Nombre	delEntrevista
Descripción	Se elimina la entrevista del repositorio estadístico
Entradas	idEmpresa
Salidas	
Proceso	Se reduce en 1 el valor de re_empresas.num_intrvws para idEmpresa Se recalculan re_any_univ.max_num_intrvws , re_any_univ.empresa_max_intrvws_id y re_any_univ.empresa_max_intrvws

Nombre	addColaboracion
Descripción	Se añaden los datos de una colaboración al repositorio estadístico
Entradas	idUniversidad, idProfesor, sueldo, duracion, numInformes
Salidas	
Proceso	Se aumenta en 1 el valor de re_universidades.stdnts_en_pract para el idUniversidad Se recalcula re_universidades.prcnt_stdnts_sup para el idUniversidad

	<p>Se recalculan re_general.max_prctn_stdnts_superan, re_general.univ_max_stdnts_sup_id y re_general.univ_max_stdnts_sup</p> <p>Se aumenta en 1 el valor de re_profesores.num_students para el idProfesor</p> <p>Se aumenta en 1 el valor de re_any_univ.total_students</p> <p>Se recalcula el valor de re_any_univ.prctn_no_superan</p> <p>Se aumenta en sueldo el valor de re_any_actual.suma_sueldos</p> <p>Se aumenta en 1 el valor de re_any_actual.num_muestras</p> <p>Se recalcula el valor de re_any_actual.sueldo_medio</p> <p>Se aumenta en 1 el valor de re_any_actual.num_estudiantes_actual</p> <p>Se recalcula re_any_actual.num_univ_no_stdnts</p> <p>Se recalcula re_any_actual.prctn_no_stdnts</p> <p>Se recalculan re_any_actual.max_num_stdnts_prof, re_any_actual.profesor_max_stdnts_id y re_any_actual.profesor_max_stdnts</p> <p>Se aumenta en duracion el valor de re_general.total_duracion_contratos</p> <p>Se aumenta en 1 el valor de re_general.num_contratos</p> <p>Se recalcula el valor de re_general.duracion_media_contrato</p> <p>Se aumenta en numInformes el valor de re_general.num_informes_pendents</p>
--	---

Nombre	updColaboracion
Descripción	Se modifican los datos de una colaboración al repositorio estadístico
Entradas	deltaSueldo, deltaDuracion, deltaInformes
Salidas	
Proceso	<p>Se aumenta en deltaSueldo el valor de re_any_actual.suma_sueldos</p> <p>Se recalcula el valor de re_any_actual.sueldo_medio</p> <p>Se aumenta en deltaDuracion el valor de re_general.total_duracion_contratos</p> <p>Se recalcula el valor de re_general.duracion_media_contrato</p> <p>Se aumenta en deltaInformes el valor de re_general.num_informes_pendents</p>

Nombre	delColaboracion
Descripción	Se eliminan los datos de una colaboración al repositorio estadístico
Entradas	idUniversidad, idProfesor, sueldo, duracion, numInformes
Salidas	
Proceso	Se reduce en 1 el valor de re_universidades.stdnts_en_pract

	<p>para el idUniversidad Se recalcula re_universidades.prcnt_stdnts_sup para el idUniversidad Se recalculan re_general.max_prcnt_stdnts_superan, re_general.univ_max_stdnts_sup_id y re_general.univ_max_stdnts_sup Se reduce en 1 el valor de re_profesores.num_students para el idProfesor Se reduce en 1 el valor de re_any_univ.total_students Se recalcula el valor de re_any_univ.prcnt_no_superan Se reduce en sueldo el valor de re_any_actual.suma_sueldos Se reduce en 1 el valor de re_any_actual.num_muestras Se recalcula el valor de re_any_actual.sueldo_medio Se reduce en 1 el valor de re_any_actual.num_estudiantes_actual Se recalcula re_any_actual.num_univ_no_stdnts Se recalcula re_any_actual.prcnt_no_stdnts Se recalculan re_any_actual.max_num_stdnts_prof, re_any_actual.profesor_max_stdnts_id y re_any_actual.profesor_max_stdnts Se reduce en duracion el valor de re_general.total_duracion_contratos Se reduce en 1 el valor de re_general.num_contratos Se recalcula el valor de re_general.duracion_media_contrato Se reduce en numInformes el valor de re_general.num_informes_pendents</p>
--	--

Nombre	addInspeccion
Descripción	Se añade una inspección al repositorio estadístico
Entradas	idEmpresa
Salidas	
Proceso	<p>Se aumenta en 1 el valor de re_empresas.inspecciones para el idEmpresa Se recalculan re_general.max_inspecciones, re_general.empresa_mas_inspecciones_id, re_general.empresa_mas_inspecciones</p>

Nombre	dellInspeccion
Descripción	Se elimina la inspección al repositorio estadístico
Entradas	idEmpresa
Salidas	
Proceso	<p>Se reduce en 1 el valor de re_empresas.inspecciones para el idEmpresa Se recalculan re_general.max_inspecciones, re_general.empresa_mas_inspecciones_id,</p>

	re_general.empresa_mas_inspecciones
--	-------------------------------------

Nombre	addInforme
Descripción	Se añade el informe al repositorio estadístico
Entradas	
Salidas	Ejecución correcta = 'OK' Fracaso = 'ERROR + TIPO DE ERROR'
Proceso	Se reduce en 1 el valor de re_general.num_informes_pendents

Nombre	dellInforme
Descripción	Se elimina el informe al repositorio estadístico
Entradas	
Salidas	Ejecución correcta = 'OK' Fracaso = 'ERROR + TIPO DE ERROR'
Proceso	Se aumenta en 1 el valor de re_general.num_informes_pendents

Nombre	addInformeFinal
Descripción	Se añaden los valores del informe final al repositorio estadístico
Entradas	idUniversidad, idTitulacion, esApto
Salidas	
Proceso	<p>Se reduce en 1 el valor de re_any_actual.num_estudiantes_actual</p> <p>Si es no apto:</p> <ul style="list-style-type: none"> • Se aumenta en 1 el valor de re_any_univ.no_superan • Se recalcula el valor de re_any_univ.prcnt_no_superan <p>Si es apto:</p> <ul style="list-style-type: none"> • Se aumenta en 1 el valor de re_universidades.stdnts_superan para el idUniversidad • Se recalcula re_universidades.prcnt_stdnts_sup para el idUniversidad • Se recalculan re_general.max_prcnt_stdnts_superan, re_general.univ_max_stdnts_sup_id y re_general.univ_max_stdnts_sup • Se aumenta en 1 el valor de re_titulaciones.estudiantes_aptos para el idTitulacion • Se recalculan re_general.max_stdnts_aptos, re_general.titulacion_id y re_general.titulacion

Nombre	updInformeFinal
Descripción	Se modifica el valor de aptitud de un informe final

Entradas	idUniversidad, idTitulacion, esApto
Salidas	
Proceso	Se llama a delInformeFinal pasando el parámetro !esApto Se llama a addInformeFinal con el parámetro esApto

Nombre	delInformeFinal
Descripción	Se eliminan los valores de un informe final en el repositorio estadístico
Entradas	idUniversidad, idTitulacion, esApto
Salidas	Ejecución correcta = 'OK' Fracaso = 'ERROR + TIPO DE ERROR'
Proceso	Se aumenta en 1 el valor de re_any_actual.num_estudiantes_actual Si es no apto: <ul style="list-style-type: none"> • Se reduce en 1 el valor de re_any_univ.no_superan • Se recalcula el valor de re_any_univ.prcnt_no_superan Si es apto: <ul style="list-style-type: none"> • Se reduce en 1 el valor de re_universidades.stdnts_superan para el idUniversidad • Se recalcula re_universidades.prcnt_stdnts_sup para el idUniversidad • Se recalculan re_general.max_prcnt_stdnts_superan, re_general.univ_max_stdnts_sup_id y re_general.univ_max_stdnts_sup • Se aumenta en 1 el valor de re_titulaciones.estudiantes_aptos para el idTitulacion • Se recalculan re_general.max_stdnts_aptos, re_general.titulacion_id y re_general.titulacion

Nombre	resetCurso
Descripción	Procedimiento al que llama el job anual de base de datos que resetea los valores para cada curso académico
Entradas	anyo_univ
Salidas	
Proceso	Se crea una nueva fila en la tabla re_any_univ con la clave anyo_univ y el resto de valores a 0 Se inicializa la tabla re_any_actual Se actualiza el valor del campo prcnt_emp_no_plazas_any_act de la tabla re_general con el valor 100 Para cada fila de la tabla re_empresas se actualiza el valor del campo plazas_any_anterior con el del campo plazas_any_actual y se ponen a 0 los valores de num_intrwvs y plazas_any_actual Para cada fila de la tabla re_profesores se actualiza el valor del

	campo num_students a 0
--	------------------------

4.3. Gestión de los logs.

Para facilitar y unificar la gestión de los logs se crea un paquete con métodos para guardar entradas en las tablas de logs con los datos del método ejecutado.

Nombre	guardaLog
Descripción	Procedimiento para guardar una entrada en la tabla de logs
Entradas	operacion, parametros, resultado
Salidas	
Proceso	Se guarda una entrada en la tabla Logs con los valores pasados como parámetros junto con los datos del usuario que ejecuta la acción y la fecha y hora de ejecución.

Nombre	guardaLogDw
Descripción	Procedimiento para guardar una entrada en la tabla de logs del repositorio estadístico
Entradas	operacion, parametros, resultado
Salidas	
Proceso	Se guarda una entrada en la tabla logs_dw con los valores pasados como parámetros junto con los datos del usuario que ejecuta la acción y la fecha y hora de ejecución.

5. Pruebas

Para comprobar el funcionamiento de los métodos implementados se han realizado pruebas ejecutando cada método y comprobando que la acción se refleja en las tablas de la base de datos.

A la hora de comprobar el correcto funcionamiento del repositorio estadístico se ha creado un script mediante el que se simula la ejecución normal durante 3 cursos insertando alumnos y prácticas de manera aleatoria.

Para cada una de las consultas planteadas en el enunciado se crea la consulta directa sobre el repositorio estadístico y otra que calcula los mismos valores sobre las tablas de datos. En cualquier momento se pueden ejecutar ambas consultas y los resultados deben ser iguales.

A continuación se exponen las consultas SQL tanto sobre el repositorio estadístico (en tiempo constante 1) como la análoga sobre las tablas de datos así como los resultados de dichas consultas tras una de las ejecuciones del script de pruebas.

- 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.

```
SELECT univ_max_stdnts_sup FROM re_general;
```

	UNIV_MAX_STDNTS_SUP
1	Universidad Pompeu Fabra (UPF)

```
SELECT u.nombre
FROM
      (SELECT      c.universidad_id,      COUNT(1)      total,
SUM(TO_NUMBER(NVL(i.es_apto,      '0')))      aptos,
NVL(SUM(TO_NUMBER(NVL(i.es_apto,      '0'))) / NULLIF(COUNT(1), 0), 0)
prcnt
FROM colaboracion c
LEFT JOIN informe_final i ON (
      c.empresa_id = i.empresa_id
      AND c.universidad_id = i.universidad_id
      AND c.estudiante_id = i.estudiante_id
      AND c.oferta_id = i.oferta_id
)
GROUP BY c.universidad_id) prcnt
JOIN universidad u ON prcnt.universidad_id = u.id
ORDER BY prcnt.prcnt DESC, u.id
FETCH FIRST 1 ROWS ONLY;
```

	NOMBRE
1	Universidad Pompeu Fabra (UPF)

- Top5 de empresas que han ofrecido más plazas para estudiantes en prácticas al CIC.

```
SELECT empresas_mas_plazas FROM re_general;
```

	EMPRESAS_MAS_PLAZAS
1	SEAT, Lidl, Grifols, Vueling Airlines, Caprabo

```
SELECT LISTAGG(nombre, ', ')
FROM
  (SELECT o.empresa_id, e.nombre, SUM(o.plazas) plzs
   FROM oferta o
   JOIN empresa e ON o.empresa_id = e.id
   GROUP BY o.empresa_id, e.nombre
   ORDER BY SUM(o.plazas) DESC, o.empresa_id
   FETCH FIRST 5 ROWS ONLY);
```

	LISTAGG(NOMBRE,',')
1	SEAT, Caprabo, Vueling Airlines, Grifols, Lidl

- Empresa que, entre todos los datos de que disponemos, ha recibido más inspecciones de trabajo (sin considerar el resultado de la inspección).

```
SELECT empresa_mas_inspecciones FROM re_general;
```

	EMPRESA_MAS_INSPECCIONES
1	SEAT

```
SELECT e.nombre
FROM inspeccion i
JOIN empresa e ON i.empresa_id = e.id
GROUP BY i.empresa_id, e.nombre
ORDER BY COUNT(1) DESC, i.empresa_id
FETCH FIRST 1 ROWS ONLY;
```

	NOMBRE
1	SEAT

- Dado un año universitario en concreto, porcentaje de estudiantes, de cualquier universidad, que no superan las prácticas en empresas. *Se muestran los resultados para los 3 años de los que hay datos.

```
SELECT prcnt_no_superan FROM re_any_univ WHERE anyo_univ = '<AÑO-UNIVERSITARIO>';
```

	ANYO_UNIV	PRCNT_NO_SUPERAN
1	21-22	53,85
2	22-23	13,16
3	20-21	62,5

```
SELECT NVL(SUM(DECODE(i.es_apto, '0', 1, 0)) / NULLIF(COUNT(1), 0) * 100, 0)
FROM colaboracion c
JOIN oferta o ON c.oferta_id = o.id
LEFT JOIN informe_final i ON (
  c.empresa_id = i.empresa_id
  AND c.universidad_id = i.universidad_id
  AND c.estudiante_id = i.estudiante_id
  AND c.oferta_id = i.oferta_id
)
WHERE o.anyo_acad = '<AÑO-UNIVERSITARIO>';
```

	'20-21'	NVL(SUM(D	'21-22'	NVL(SUM(DEC	'22-23'	NVL(SUM(DECC
1	20-21	62,5	21-22	53,846	22-23	13,158

- En el año universitario en curso, sueldo medio que reciben los estudiantes durante sus prácticas.

```
SELECT sueldo_medio FROM re_any_actual;
```

	SUELDO_MEDIO
1	692

```
SELECT NVL(FLOOR(AVG(c.sueldo)), 0)
FROM colaboracion c
JOIN oferta o ON c.oferta_id = o.id
WHERE o.anyo_acad = anyoActual();
```

	NVL(FLOOR(AVG(C.SUELDO)),0)
1	692

- 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.

```
SELECT num_estudiantes_actual FROM re_any_actual;
```

	NUM_ESTUDIANTES_ACTUAL
1	28

```
SELECT COUNT(1)
FROM colaboracion c
JOIN oferta o ON c.oferta_id = o.id
WHERE o.anyo_acad = anyoActual()
AND NOT EXISTS (SELECT 1
FROM informe_final i
WHERE c.empresa_id = i.empresa_id
AND c.universidad_id = i.universidad_id
AND c.estudiante_id = i.estudiante_id
AND c.oferta_id = i.oferta_id);
```

	COUNT(1)
1	28

- Durante el último año universitario, porcentaje de universidades controladas por el CIC que no tienen estudiantes en prácticas.

```
SELECT prcnt_no_stdnts FROM re_any_actual;
```

	PRCNT_NO_STDNTS
1	25

```
SELECT NVL((COUNT(1) - COUNT(universidad_id)) / NULLIF(COUNT(1), 0) *
100, 0)
FROM universidad u
LEFT JOIN (
SELECT DISTINCT c.universidad_id
FROM colaboracion c
JOIN oferta o ON c.oferta_id = o.id
LEFT JOIN universidad u ON c.universidad_id = u.id
WHERE o.anyo_acad = anyoActual()
) prt ON u.id = prt.universidad_id;
```

	NVL((COUNT(1)-COUNT(UNIVERSIDAD_ID))/NULLIF(COUNT(1),0)*100,0)
1	25

- 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.

```
SELECT titulacion FROM re_general;
```

	TITULACION
1	Ingeniería Náutica y Transporte Marítimo

```
SELECT t.nombre
FROM informe_final i
JOIN oferta o ON i.oferta_id = o.id
JOIN cursa c ON (
    i.universidad_id = c.universidad_id
    AND i.estudiante_id = c.estudiante_id
    AND o.anyo_acad = c.anyo_acad
)
JOIN titulacion t ON c.titulacion_id = t.codigo
WHERE i.es_apto = '1'
GROUP BY t.codigo, t.nombre
ORDER BY COUNT(1) DESC, t.codigo
FETCH FIRST 1 ROWS ONLY;
```

	NOMBRE
1	Ingeniería Náutica y Transporte Marítimo

- Número medio de estudiantes que se presentan a cada oferta recibida por las diferentes universidades durante el año universitario en curso.

```
SELECT media_candidat_ofert FROM re_any_actual;
```

	MEDIA_CANDIDAT_OFERT
1	2,31

```
SELECT NVL(COUNT(c.oferta_id) / NULLIF(COUNT(DISTINCT o.id), 0), 0), 0)
FROM oferta o
LEFT JOIN candidatura c ON o.id = c.oferta_id
WHERE o.anyo_acad = anyoActual();
```

	NVL(COUNT(C.OFERTA_ID)/NULLIF(COUNT(DISTINCTO.ID),0),0)
1	2,313

- Teniendo en cuenta todos los datos de que se dispone, duración media de los contratos de estudiantes con las empresas.

```
SELECT duracion_media_contrato FROM re_general;
```

	DURACION_MEDIA_CONTRATO
1	9,1

```
SELECT NVL(AVG(MONTHS_BETWEEN(fecha_final, fecha_inicio)), 0)
FROM colaboracion;
```

	NVL(AVG(MONTHS_BETWEEN(FECHA_FINAL,FECHA_INICIO)),0)
1	9,059

- Empresa que, en un año en concreto, ha realizado el mayor número de entrevistas. *Se muestran los resultados para los 3 años de los que hay datos.

```
SELECT empresa_max_intrvws FROM re_any_univ WHERE anyo_univ = '<AÑO-UNIVERSITARIO>';
```

	ANYO_UNIV	EMPRESA_MAX_INTRVWS
1	21-22	SEAT
2	22-23	Caprabo
3	20-21	Caprabo

```
SELECT m.nombre
FROM entrevista e
JOIN oferta o ON e.oferta_id = o.id
JOIN empresa m ON e.empresa_id = m.id
WHERE o.anyo_acad = '<AÑO-UNIVERSITARIO>'
GROUP BY m.id, m.nombre
ORDER BY COUNT(1) DESC, m.id
FETCH FIRST 1 ROWS ONLY;
```

	'20-21'	NOMBRE	'21-22'	NOMBRE	'22-23'	NOMBRE
1	20-21	Caprabo	21-22	SEAT	22-23	Caprabo

- En el año universitario en curso, profesor que lleva un mayor número de estudiantes que están realizando prácticas en empresas.

```
SELECT profesor_max_stdnts FROM re_any_actual;
```

	PROFESOR_MAX_STDNTS
1	Plinio Osuna Enríquez

```
SELECT TRIM(p.nombre || ' ' || p.apellido1 || ' ' || p.apellido2)
FROM colaboracion c
JOIN oferta o ON c.oferta_id = o.id
JOIN cursa cr ON (
  c.universidad_id = cr.universidad_id
  AND c.estudiante_id = cr.estudiante_id
  AND o.anyo_acad = cr.anyo_acad
)
JOIN profesor_responsable p ON cr.profesor_id = p.id
WHERE o.anyo_acad = anyoActual()
GROUP BY p.id, p.nombre, p.apellido1, p.apellido2
ORDER BY COUNT(1) DESC, p.id
FETCH FIRST 1 ROWS ONLY;
```

	TRIM(P.NOMBRE ' ' P.APELLIDO1 ' ' P.APELLIDO2)
1	Plinio Osuna Enríquez

- En el momento de ejecutar la consulta, número total de informes de prácticas pendientes de ser realizados por las personas responsables de las empresas.

```
SELECT num_informes_pendents FROM re_general;
```

	NUM_INFORMES_PENDENTS
1	118

```
SELECT SUM(c.informes_esperados - NVL(i.num, 0))
FROM colaboracion c
LEFT JOIN (
  SELECT empresa_id, universidad_id, estudiante_id, oferta_id,
COUNT(1) num
  FROM informe
  GROUP BY empresa_id, universidad_id, estudiante_id, oferta_id
) i ON (
  c.empresa_id = i.empresa_id
  AND c.universidad_id = i.universidad_id
  AND c.estudiante_id = i.estudiante_id
  AND c.oferta_id = i.oferta_id
);
```

	SUM(C.INFORMES_ESPERADOS-NVL(I.NUM,0))
1	118

- 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.

```
SELECT prcnt_emp_no_plazas_any_act FROM re_general;
```

	PRCNT_EMP_NO_PLAZAS_ANY_ACT
1	0

```
SELECT NVL(SUM(hits) / NULLIF(COUNT(1), 0) * 100, 0)
FROM (
      SELECT e.id, CASE WHEN MAX(Decode(o.anyo_acad,
TO_CHAR(TO_NUMBER(SUBSTR(anyoActual(), 1, 2)) - 1) || '-' ||
SUBSTR(anyoActual(), 1, 2), 1, 0)) - MAX(Decode(o.anyo_acad,
anyoActual(), 1, 0)) > 0 THEN 1 ELSE 0 END hits
      FROM empresa e
      LEFT JOIN oferta o ON e.id = o.empresa_id
      GROUP BY e.id
);
```

	NVL(SUM(HITS)/NULLIF(COUNT(1),0)*100,0)
1	0

En el Anexo 1 se detalla el proceso mediante el que se obtienen estos datos de pruebas.

6. Conclusiones

Un trabajo de estas características tiene múltiples facetas. Por un lado está la parte meramente técnica, la cual conlleva una determinada dificultad y un volumen de trabajo. Por otra parte está la planificación y gestión del tiempo disponible (todo un semestre para realizar el trabajo). Y, por último, está la redacción de la memoria y las presentaciones necesarias.

En cuanto a la complejidad técnica, los conocimientos obtenidos en el grado así como la experiencia previa han hecho que no supusiera un gran problema, pero el volumen de trabajo sí que ha sido un problema a la hora de completar el trabajo a tiempo.

La parte de la planificación ha supuesto un problema grave. Respecto a la planificación inicial se han producido importantes retrasos que han obligado a aumentar el tiempo de dedicación en las últimas etapas del trabajo. También la gestión del tiempo es algo que se debe controlar muy bien mediante un seguimiento más exhaustivo.

En el caso de este trabajo, se infravaloró la dedicación necesaria para la implementación de los métodos de gestión de los datos así como para la redacción de la memoria. Pese a las recomendaciones del tutor el seguimiento no fue todo lo efectivo que debería haber sido por lo que la desviación se detectó tarde.

La falta de práctica en la redacción del tipo de documentación requerida en el trabajo también ha afectado negativamente a la gestión del tiempo disponible.

Todo esto ha supuesto un gran aprendizaje, para poder llevar a cabo un proyecto de estas características se debe mejorar mucho en cuanto a la planificación y, además, se debe aplicar un seguimiento efectivo para detectar los desvíos lo antes posible y minimizar el impacto.

Los objetivos iniciales se han cubierto, si bien es cierto que con más esfuerzo del inicialmente previsto. Lo que no se ha podido lograr es superar dichos objetivos e implementar funcionalidades más allá de las mínimas exigidas en el enunciado.

La planificación planteada durante la primera entrega del trabajo se pudo seguir durante la segunda entrega pero para la tercera no fue suficiente y en la entrega final fue necesario ampliar la dedicación para poder completar el trabajo en el plazo marcado.

La metodología de trabajo sí que se ha mantenido, pero el esfuerzo necesario para completar cada uno de los hitos ha sido mucho más alto del inicialmente previsto.

Como líneas de trabajo futuro quedan varias.

- La más evidente es la necesidad de desarrollar una aplicación con interfaz gráfica que permita la interacción con la base de datos de una

manera práctica. Utilizar directamente los procedimientos de base de datos desde consola no es práctico. Este punto quedaba fuera del alcance de este proyecto ya en el propio enunciado, pero es una necesidad obvia.

- Otro punto que ha quedado fuera de este trabajo es la gestión de los permisos en la base de datos. Se podría crear un rol que únicamente disponga de permisos de ejecución de los distintos procedimientos implementados y que no pueda realizar acciones de manipulación de datos directamente en las tablas para evitar discrepancias entre los datos del modelo y los del repositorio estadístico. Todos los usuarios necesarios para las personas que accedan a la aplicación se podrían crear con este rol y así garantizar la integridad de la base de datos.
- También se deberá crear un trabajo de base de datos que, una vez al año, ejecute el método `pq_dw.resetCurso` con el nuevo curso académico para inicializar las tablas del repositorio estadístico con los valores para el nuevo curso.
- Otra línea de trabajo interesante es crear un trabajo periódico de base de datos que revise los logs de los métodos de actualización del repositorio estadístico para que, en el caso de producirse errores, se lleven a cabo tareas automatizadas de mitigación de dichos errores. Ya sea recalculando parte de los datos estadísticos o enviando una alerta a un operador humano para su revisión.

7. Glosario

atributo Propiedad que interesa representar de un tipo de entidad.

base de datos Programa capaz de almacenar gran cantidad de datos, relacionados y estructurados, que pueden ser consultados rápidamente de acuerdo con las características selectivas que se deseen.

clase Nombre que reciben los tipos de entidades en el modelo UML.

clave foránea Campo o combinación de campos en una tabla que se refieren a una clave primaria de otra tabla.

clave primaria Campo o combinación de campos que identifica de forma única a cada fila de una tabla.

conectividad Expresión del tipo de correspondencia entre las entidades de una relación.

data warehouse (Almacén de datos o repositorio de datos) Colección de datos orientada a un determinado ámbito, que ayuda a la toma de decisiones en la entidad en la que se utiliza.

diagrama de Gantt Herramienta gráfica para exponer el tiempo de dedicación previsto para diferentes tareas a lo largo de un periodo total determinado.

diseño conceptual Etapa del diseño de una base de datos que obtiene una estructura de la información de la futura base de datos independiente de la tecnología que se quiere emplear.

diseño lógico Etapa del diseño de una base de datos que parte del resultado del diseño conceptual y lo transforma de manera que se adapte al modelo de sistema gestor de bases de datos con el cual se desea implementar la base de datos.

entidad Objeto del mundo real que podemos distinguir del resto de objetos y del cual nos interesan algunas propiedades.

generalización/especialización Construcción que permite reflejar que existe un tipo de entidad general, llamada superclase, que se puede especializar en diferentes tipos de entidades más específicas, llamadas subclases.

grado de una relación Número de entidades que asocia la relación.

log Grabación secuencial de todos los acontecimientos que afectan a un proceso particular.

script Fragmento de código con instrucciones ordenadas y estructuras de control básicas.

secuencia objeto de base de datos que permite la generación automática de valores

SGBD Sigla de sistema de gestión de bases de datos. Tipo de software específico dedicado a servir de interfaz entre la base de datos, el usuario y las aplicaciones que lo utilizan.

8. Bibliografía

1. Casas Roma, J. y Cuartero Olivera, J. (2020). *Diseño conceptual de bases de datos* (5ª ed.). Fundació Universitat Oberta de Catalunya.
2. Burgués Illa, X. y Cuartero Olivera, J. (2020). *Diseño lógico de bases de datos* (5ª ed.). Fundació Universitat Oberta de Catalunya.
3. Cabré i Segarra, B., Casas Roma, J., Costal Costa, D., Juanola Juanola, P., Plana Vallvé, I., Rius Gavidia, À. y Segret i Sala, R. (2020). *Diseño físico de bases de datos* (7ª ed.). Fundació Universitat Oberta de Catalunya.
4. Oracle (2022) *Oracle Help Center*. Disponible en: <https://docs.oracle.com> (consultado: 15 de diciembre de 2022).
5. Pradel Miquel, J. y Raya Martos, J. (2020). *Análisis UML* (4ª ed.). Fundació Universitat Oberta de Catalunya.
6. Marín Amatller, A. (2020). *Exposición de contenidos en vídeo* (2ª ed.). Fundació Universitat Oberta de Catalunya.
7. Beneito Montagut, R. *Presentación de documentos y elaboración de presentaciones*. Fundació Universitat Oberta de Catalunya.

9. Anexos

Anexo 1: Pruebas del repositorio estadístico.