

UNIVERSIDAD AUTÓNOMA DE MADRID

ESCUELA POLITÉCNICA SUPERIOR



Grado en Ingeniería Informática

TRABAJO FIN DE GRADO

**HERRAMIENTA DE GESTIÓN DE TRABAJOS DE FIN
DE GRADO:**

**MÓDULO DE GESTIÓN DE COMISIONES DE
EVALUACIÓN**

Jaime Zaballos Díaz-Maroto

Tutor: Iván Cantador Gutiérrez

Enero 2015

Resumen

Con la implantación del Trabajo de Fin de Grado (TFG) en la Escuela Politécnica Superior (EPS) de la Universidad Autónoma de Madrid, en los últimos años el creciente número de estudiantes y TFG, así como el de las correspondientes Comisiones de Evaluación (CCEE), ha originado la necesidad de desarrollar una herramienta software que permita automatizar y facilitar las tareas de gestión involucradas.

Esta herramienta ha de disponer de información del total de docentes que componen cada una de las tres titulaciones de la EPS, indicando las tecnologías específicas de cada uno de ellos, información de los estudiantes que ya han defendido algún TFG o tienen alguno asignado o cancelado, así como información (título, descripción, tutor, ponente...) sobre los TFG que hasta la fecha están ofertados, asignados a estudiantes o ya han sido defendidos. Además, dispone de información sobre las CCEE, con los TFG correspondientes a cada una, sus fechas, lugares y docentes asignados (presidente, vocales y suplentes).

El objetivo principal de la herramienta es poder gestionar toda la información anterior, pudiendo añadir, modificar o eliminar cualquier dato.

Respecto a las CCEE, la herramienta ha de ser capaz de asistir en la generación de las comisiones que tendrán lugar próximamente. Para ello, permitirá seleccionar el grado de la comisión que se quiera generar (GII, DGIM o GITST), el número máximo de TFG que se podrán asignar a cada comisión, el curso y la convocatoria. A partir de aquí, sugerirá una primera aproximación de todas las comisiones, y qué proyectos se han asignado a cada una de ellas. Entonces el usuario (coordinador de TFG) podrá mover los TFG que considere necesarios entre comisiones y continuar el proceso. En concreto, se asignarán los docentes a las comisiones ya creadas cumpliendo una serie de restricciones, por lo que nuevamente se mostrará una primera vista de cómo quedarían las comisiones. Finalmente, tras los cambios oportunos (si es que los hubiera) por parte del usuario, se crearán y almacenarán las comisiones establecidas.

Palabras clave

Herramienta de gestión, Trabajo de Fin de Grado, Comisión de Evaluación, Java, bases de datos.

Summary

Due to the increase in the number of Final Degree Projects (FDP) at Escuela Politécnica Superior (EPS) at Universidad Autónoma de Madrid, in recent years, when it comes to managing the growing number of students, projects, and the corresponding Evaluation Committees (EECC), the need has arisen to develop a software tool capable of automatically performing or supporting the involved management tasks.

This tool avails itself of the total number of professors that make up each one of the three degrees offered by the school, indicating the specific technologies of each one. Also appearing are the students that have assigned or have already defended a FDP, and the FDPs that have been cancelled. In addition, it has the FDP information (title, description, advisor, speaker...) that have been presented up to date. Lastly, it also manages the information associated to the EECC, with their corresponding FDPs, date, location and assigned professors (president, speakers and substitutes).

The main goal of the tool is to automatize or assist the management of each of the aforementioned information, being able to add, modify or eliminate any of the details pertaining to the students, professors, FDPs and EECCs.

With regards to the EECCs, the tool has to be able to generate the upcoming committees that will take place. In order to do this, the users (i.e., the FDP coordinator) chooses the degree of the committees he/she want to generate (GII, DGIM or GITST), the maximum number of FDPs that can be assigned to each committee, the academic year, and the submission deadline and defense. At this point, it has to generate a first approximation of all the committees and the projects assigned to each of them. Then, the user has to be able to move the FDPs they consider necessary and continue with the process. Next, the tool has to assign the professors to each of the existing committees, thus showing once again how the committees would be. Finally, after the user makes the appropriate changes (if any), the tools sets and stores the created committees.

Keywords

Management tool, Final Degree Project, Evaluation Committee, Java, databases.

Índice de contenidos

1. Introducción.....	1
2. Arquitectura del Sistema	4
3. Base de Datos	7
3.1. Diagrama Entidad-Relación	7
3.2. Modelo relacional	7
3.2.1. GRADOS.....	8
3.2.2. TECNOLOGÍAS.....	8
3.2.3. GRADO_TECNOLOGÍAS.....	8
3.2.4. DEPARTAMENTOS	9
3.2.5. DOCENTES	9
3.2.6. DOCENTE_TECNOLOGÍAS.....	10
3.2.7. ESTUDIANTES	10
3.2.8. TFGS.....	11
3.2.9. TFG_TECNOLOGÍAS	12
3.2.10. COMISIONES.....	12
3.2.11. COMISION_TFGS	13
3.3. Recopilación y unificación de datos	13
3.3.1. GII y DGI.....	14
3.3.2. GITST.....	14
4. Herramienta software desarrollada.....	16
4.1. Docentes.....	16
4.2. Estudiantes	18
4.3. Trabajos de Fin de Grado	20
4.4. Estadísticas	21
4.4.1. Tecnología_Tfgs.....	22
4.4.2. Docente_Tfgs.....	22
4.4.3. Departamento_Tfgs	23
4.4.4. Docente_comisión.....	23
4.5. Comisiones de Evaluación	23

5. Módulo de generación de Comisiones de Evaluación.....	26
5.1. Fase de agrupación de Trabajos de Fin de Grado por tecnologías	26
5.1.1. Preparación de datos	26
5.1.2. Algoritmo desarrollado	28
5.2. Fase de asignación de profesores como miembros de comisión.....	32
5.2.1 Preparar datos.....	32
5.2.2. Algoritmo desarrollado	34
6. Conclusiones y trabajo futuro.....	37
Referencias	39

Índice de tablas

Tabla 1: Entidad Grados	8
Tabla 2: Entidad Tecnologías	8
Tabla 3: Entidad Grado_tecnologías	8
Tabla 4: Entidad Departamentos	9
Tabla 5: Entidad Docentes.....	9
Tabla 6_ Entidad Docente_tecnologías	10
Tabla 7: Entidad Estudiantes	10
Tabla 8: Entidad TFGS.....	11
Tabla 9: Entidad TFG_tecnologías.....	12
Tabla 10: Entidad Comisiones.....	12
Tabla 11: Entidad Comisión_tfgs	13

Índice de figuras

Figura 1: Diagrama de Arquitectura.....	4
Figura 2: Diagrama Entidad-Relación base de datos.....	7
Figura 3: Pestaña principal Docentes	16
Figura 4: Botón Guardar pestaña Docentes	17
Figura 5: Botón Añadir pestaña Docentes	17
Figura 6: Filtro búsqueda pestaña Docentes.....	18
Figura 7: Pestaña principal Estudiantes.....	18
Figura 8: Botón Añadir pestaña Estudiantes	19
Figura 9: Filtro búsqueda pestaña Estudiantes	20
Figura 10: Pestaña principal Trabajos de Fin de Grado	20
Figura 11: Botón Información pestaña TFG.....	21
Figura 12: Filtro búsqueda pestaña TFG	21
Figura 13: Estadísticas Tecnología_Tfgs	22
Figura 14: Estadísticas Docente_Tfgs	22
Figura 15: Estadísticas Departamento_Tfgs.....	23
Figura 16: Estadísticas Docente_comisión.....	23
Figura 17: Configuración para generar Comisiones.....	24
Figura 18: Comisiones autogeneradas – Asignar TFG.....	24
Figura 19: Comisiones autogeneradas - Asignar docentes.....	25

Índice de fragmentos de código

Código 1: Método <i>compareTo</i> de la clase <i>TFG_Aux</i>	27
Código 2: <i>Query</i> para obtener los códigos de los TF de un Grado	27
Código 3: <i>Query</i> para obtener los ids de las tecnologías de un TFG	28
Código 4: Primer caso generación comisiones - TFG	28
Código 5: Segundo caso generación comisiones - TFG	29
Código 6: Tercer caso generación comisiones - TFG	30
Código 7: Cuarto caso generación comisiones - TFG	31
Código 8: Método <i>compareTo</i> de la clase <i>Docentes_Aux</i>	32
Código 9: <i>Query</i> para obtener los docentes activos de un departamento	33
Código 10: <i>Query</i> para obtener las participaciones de un docente en comisiones	33
Código 11: <i>Query</i> para obtener las tecnologías de los docentes	33
Código 12: Primer paso generación comisiones - Docentes	34
Código 13: Segundo paso generación comisiones - Docentes	35
Código 14: Tercer paso generación comisiones - Docentes	36

Glosario

TFG: Trabajo de Fin de Grado

SQL: *Structured Query Language*. Lenguaje para hacer consultas en la base de datos

JDBC: *Java DataBase Connectivity*. API que permite la ejecución sobre bases de datos desde el lenguaje de programación Java

GII: Grado en Ingeniería Informática

GITST: Grado en Ingeniería de Tecnologías y Servicios de Telecomunicación

DGIM: Doble Grado en Ingeniería Informática y Matemáticas

II: Departamento de Ingeniería Informática

TEC: Departamento de Tecnología Electrónica y de las Comunicaciones

TTEE: Tecnologías. Cada departamento tiene asociadas unas:

Ingeniería Informática:

- **CO:** Computación
- **IS:** Ingeniería del Software
- **SI:** Sistemas de Información
- **IC:** Ingeniería de Computadores
- **TI:** Tecnologías de la Información

Tecnología Electrónica y de las Comunicaciones:

- **EL:** Sistemas Electrónicos
- **RA:** Radiocomunicaciones
- **SE:** Tratamiento de Señales
- **TE:** Telemática

CAT: Categoría a la que pertenece un docente:

- **CU:** Catedrático de Universidad
- **TU:** Profesor Titular Universidad
- **CDR:** Profesor Contratado Doctor
- **AYD:** Profesor Ayudante Doctor
- **ATP:** Ayudante

CCEE: Comisiones de Evaluación

1. Introducción

Como viene especificado en la guía docente [1]: “El Trabajo de Fin de Grado tiene carácter obligatorio y 12 créditos ECTS. Se realizará al finalizar los estudios de Grado, una vez superadas el resto de asignaturas”. Debe ser presentado y defendido ante un tribunal universitario (en este documento se referirá como Comisiones de Evaluación o CCEE).

La realización del TFG consiste en desarrollar un sistema, una aplicación o servicio informático, o un proyecto de investigación. Además, se puede realizar en un ámbito profesional, con lo que un TFG puede tener un enfoque investigador, profesional, tecnológico o de emprendimiento.

En concreto, en la Escuela Politécnica Superior (EPS) de la Universidad Autónoma de Madrid, actualmente se han de realizar los TFG en las titulaciones de Grado en Ingeniería Informática (GII), Doble Grado en Ingeniería Informática y Matemáticas (DGII) y Grado en Ingeniería de Tecnologías y Servicios de Telecomunicación (GITST). A su vez, dentro de cada grado existen una serie de tecnologías, donde cada proyecto profundizará más o menos en cada una de ellas. En el caso de GII y DGIM, las tecnologías que posee son: Computación, Ingeniería del Software, sistemas de Información, Ingeniería de Computadores y Tecnologías de la Información. Para GITST, se consideran las tecnologías Sistemas Electrónicos, Radiocomunicaciones, Tratamiento de Señales y Telemática.

A la hora de formar las CCEE, se deben agrupar los TFG conforme a sus tecnologías, de la mejor forma posible, para no “mezclar” demasiado las distintas tecnologías. Así pues, una vez formadas las comisiones con los TFG agrupados por tecnologías, se deben asignar los miembros docentes que formarán parte del tribunal. Este estará formado por un presidente (normalmente Catedrático de Universidad o Profesor Titular de Universidad), dos vocales y dos suplentes.

La defensa del TFG por parte del estudiante, consiste en una exposición pública del trabajo, donde los miembros del citado tribunal pueden formular preguntas para juzgar la calidad del proyecto. Tras finalizar la defensa, el tribunal decidirá la calificación a asignar al alumno.

Esta forma de evaluar los TFG conlleva realizar la gestión de las CCEE seis veces por curso, ya que existen dos convocatorias ordinarias y una extraordinaria para cada departamento. También el hecho de no tener todos los datos centralizados en un mismo sitio conlleva pérdida de información relativa a algunos TFG defendidos en años anteriores. Además, cada vez son más los proyectos que hay que gestionar,

lo que implica mayor dificultad y mayor empleo de tiempo a la hora de generar las comisiones. Esta herramienta agiliza mucho este aspecto, ya que auto-genera las CCEE. En los dos pasos descritos anteriormente (asociar TFG por tecnologías y asignar docentes) se muestran los primeros resultados, pudiendo el usuario modificar estos aspectos, moviendo los TFG o los docentes de comisión. También mantiene toda la información referente a TFG, comisiones, docentes, estudiantes, por lo que se garantiza que no se perderán datos con el paso del tiempo.

Todo esto queda almacenado en una base de datos relacional, que se ha diseñado y construido en este trabajo. Para introducir los datos correspondientes en la base de datos, en primer lugar hubo que recopilar toda la información necesaria. Así pues, para acceder a las comisiones de los dos primeros años desde la implantación del TFG en la EPS (2012-2013, 2013-2014; en el caso de GII), hubo que recopilar los datos de la página web que entonces se utilizaba para gestionar las CCEE [2], donde también accedían los alumnos para mantenerse informados. Por tanto, los datos correspondientes a las comisiones se extrajeron a la vez que los estudiantes que ya habían defendido sus proyectos, y parte de los datos de los TFG (título, código, tutor). Para los docentes, se accedió a la página web de la EPS [3], donde se pudo completar con nombre completo, departamento, categoría, emails, etc.

Toda esta información recogida se almacenó en la base de datos relacional creada (con el gestor MySQL [4]), a la que se accede mediante JDBC [5] desde el código programado en Java [6] [7].

La aplicación está desarrollada sobre distintos módulos:

- En la capa más baja se encuentra el módulo de acceso a base de datos, en concreto el que se encarga de establecer la conexión (con JDBC), y mantener esta conexión para que el resto de módulos puedan acceder a base de datos.
- Justo por encima de este módulo, se encuentran los *managers*, donde se establecen las consultas SQL necesarias. Hay un manager para cada entidad (TFG, docente, estudiante, CCEE, etc.).
- Las entidades representan un módulo aparte donde se establecen los valores necesarios (atributos Java) para cada clase. Por ejemplo, existen las entidades Grado, Docente o Comisión.
- El siguiente módulo está compuesto por los controladores de las vistas, ya que en la interfaz gráfica se utiliza el Modelo-Vista-Controlador.
- Y por último, en la capa más alta se encuentran las vistas. Aquí es donde se desarrollan gráficamente las distintas pantallas de las que está compuesta la aplicación.

El presente documento está compuesto por seis capítulos. Cada uno de ellos se introduce brevemente a continuación:

1. Capítulo primero: Es la introducción al documento, donde se describe brevemente los distintos apartados por los que está compuesto.
2. Capítulo segundo: Se detalla la arquitectura del sistema, cómo se ha diseñado la aplicación y cada uno de sus módulos.
3. Capítulo tercero: Descripción de la base de datos. Se muestra el diagrama Entidad-Relación sobre el que ha sido construida la base de datos y se explican sus entidades y atributos. Además se detalla la recopilación y unificación de datos.
4. Capítulo cuarto: Se explica la funcionalidad de la herramienta software desarrollada, de Estudiantes, Docentes, Trabajos de Fin de Grado y Comisiones de Evaluación.
5. Capítulo quinto: Se detalla el módulo de creación de Comisiones de Evaluación, los pasos que se siguen a la hora de generar automáticamente las CCEE.
6. Capítulo sexto: Se mencionan las conclusiones extraídas y el posible trabajo futuro.

2. Arquitectura del Sistema

En este apartado se detalla la arquitectura de la herramienta software, de qué partes se compone. En concreto, está formada por módulos, cada uno en una capa diferente. La capa más baja se encarga del acceso a base de datos, mientras que la más alta es la encargada de la interfaz gráfica.

El módulo situado en la capa más baja es el encargado de conectar con la base de datos, la cual utiliza la tecnología MySQL. Este módulo, por tanto, hace uso de JDBC para interactuar con la base de datos. Por encima están los diferentes *managers*, que es donde se encuentran las consultas SQL que necesitan usar los módulos que están por encima. En concreto, los controladores son los que acceden al módulo de los *managers*, para que así el módulo de las vistas sólo se ocupe de la parte gráfica de la herramienta. El módulo *generarComisiones* contiene clases y métodos auxiliares que se utilizan a la hora de generar las CCEE de manera automática. Por último, en el módulo entidades se encuentran las clases que definen cada entidad por separado (Docente, Estudiante, TFG, Comisión,...).

En la siguiente figura se aprecia la distribución de módulos mencionada:

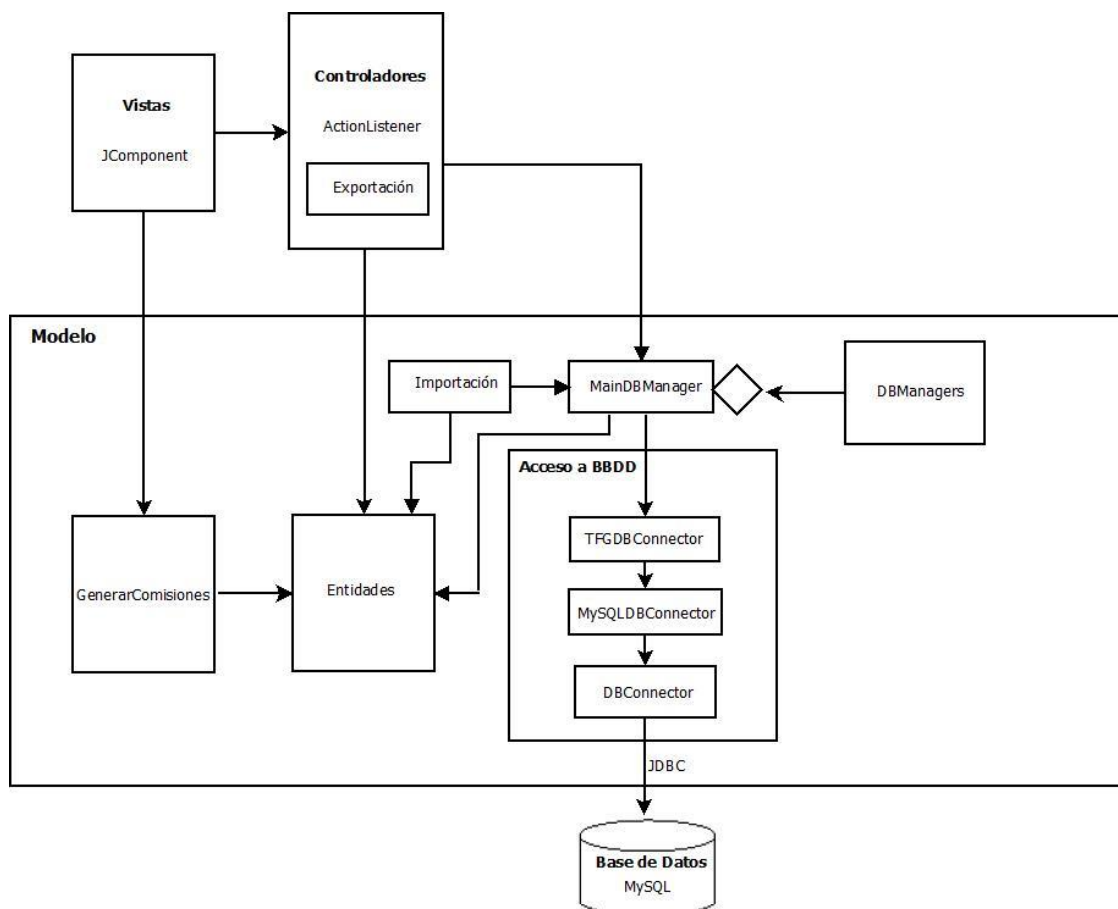


Figura 1: Diagrama de Arquitectura

La parte encargada del acceso a base de datos es el bloque titulado “Acceso a base de datos”. Mediante JDBC, la clase *DBConnector* instaura una conexión genérica con la base de datos. Su instanciación, *MySQLDBConnector*, establece una conexión con el sistema MySQL, que es el que se utiliza en este proyecto. A más alto nivel, encapsulando la clase anterior, *TFGDBConnector* gestiona el acceso a la base de datos de esta herramienta. Por ello, *TFGDBConnector* accederá a *MySQLDBConnector*, y a su vez este accederá a *DBConnector*. Este es el módulo de la capa más baja de la arquitectura del sistema.

En la capa que se encuentra justo por encima están los *DBManagers*. Aquí se define un *DBManager* para cada entidad existente en el sistema. Las entidades (estarían una capa por encima) que se han creado para el sistema son: *Comision*, *Comision_TFG*, *Departamento*, *Docente*, *Docente_Tecnologia*, *Estudiante*, *Grado*, *Grado_Tecnologia*, *TFG*, *TFG_Tecnologia*, *Tecnologia*. Estas entidades representan, codificadas en Java, las entidades que forman la base de datos (explicadas en el siguiente capítulo).

En cada *DBManager* se desarrollan las consultas SQL necesarias para cada entidad (insert, update, select...). Por tanto, para mantener la conexión activa en todo momento de forma unificada, la clase *MainDBManager* contiene todos los *DBManager*, para poder acceder a cada uno de ellos desde aquí.

En el módulo *Importación* también están los *FileReader*. Ocurre lo mismo que con los *DBManagers*, existe un *FileReader* para cada una de las entidades. Estos se encargan de leer datos de un fichero de texto para introducir la información en la base de datos. En el siguiente capítulo se explicará el porqué de estos ficheros.

En la siguiente capa está el módulo de los controladores. Para desarrollar la parte gráfica de la herramienta se ha utilizado el Modelo-Vista-Controlador. En este módulo se encuentran los controladores de cada vista, encargados de la interacción entre unas vistas y otras, haciéndolas visuales o no. Los controladores también hacen uso de la clase *MainDBManager* cuando sea necesario acceder a la base de datos. El módulo *Exportación* se encarga de volcar la información generada (creación o modificación de datos, estadísticas, generación de comisiones...) en un fichero de texto.

Y, en la capa más alta, se hallan las vistas. Estas son la parte visual de la herramienta, donde se generan todas las pantallas. Cada pantalla de la aplicación está codificada en una Vista de este módulo.

Por último, se encuentra el módulo *GenerarComisiones*. Aquí se encuentran las clases Java necesarias para la generación automática de las CCEE. En las clases *GenerarComisiones_TFG* y *GenerarComisiones_Docentes* se encuentran los algoritmos generados para este cometido.

El Modelo del patrón Modelo-Vista-Controlador, por tanto, está formado por los *DBManager/MainDBManager*, *Entidades*, *GenerarComisiones* y la parte correspondiente al acceso a base de datos.

3. Base de Datos

3.1. Diagrama Entidad-Relación

En la figura 2 se muestra el diagrama Entidad-Relación. Este es el diseño que se ha seguido para crear la base de datos de la aplicación. Como ya se ha mencionado, está desarrollada sobre MySQL.

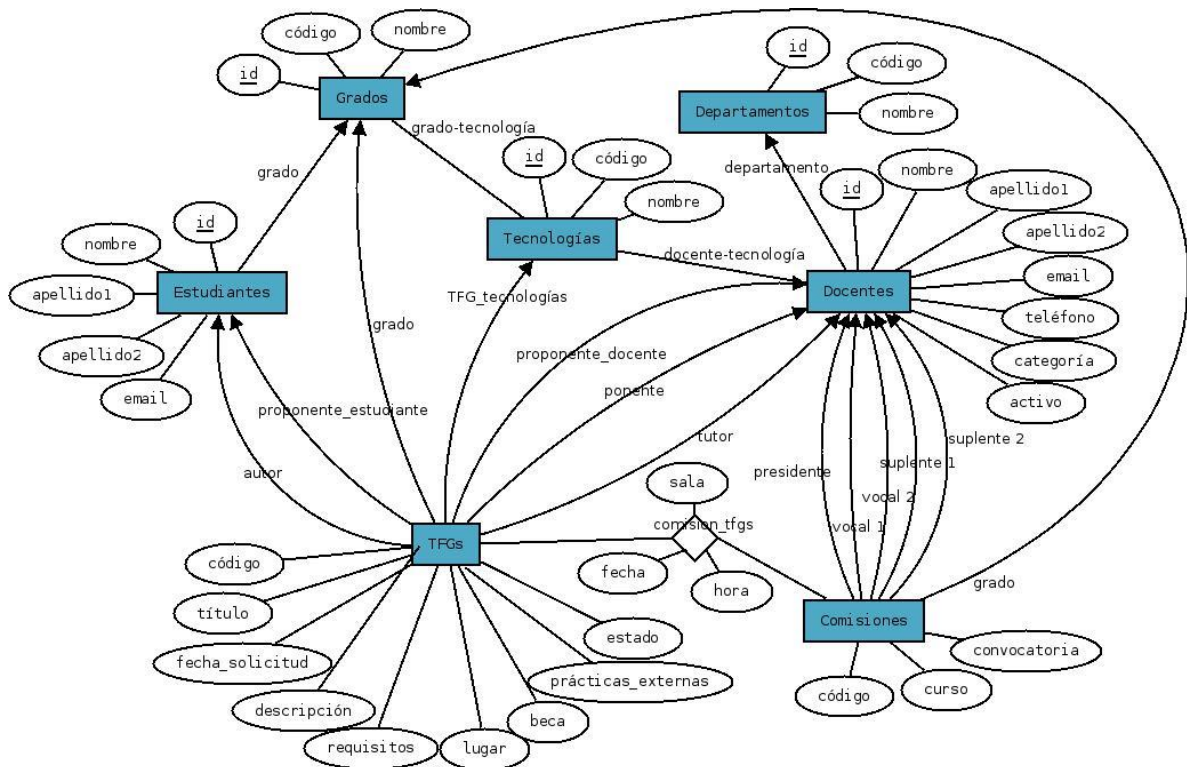


Figura 2: Diagrama Entidad-Relación base de datos

En el siguiente apartado se detalla cada una de las entidades, junto con sus atributos y relaciones.

3.2. Modelo relacional

A continuación se muestran las tablas que componen la base de datos. Junto con cada tabla se describen las entidades y sus atributos correspondientes.

3.2.1. GRADOS

Atributo	Tipo	Descripción
<u>Id</u>	Integer	Clave Primaria
Código	Varchar	Código del Grado
Nombre	Varchar	Nombre completo del Grado

Tabla 1: Entidad Grados

En esta entidad están los 3 grados que actualmente existen en la EPS.

Son Grado en Ingeniería Informática (código GII), Grado en Ingeniería de Tecnologías y Servicios de Telecomunicación (código GITST) y Doble Grado en Ingeniería Informática y Matemáticas (DGIM).

3.2.2. TECNOLOGÍAS

Atributo	Tipo	Descripción
<u>Id</u>	Integer	Clave Primaria
Código	Varchar	Código de la Tecnología
Nombre	Varchar	Nombre completo de la Tecnología

Tabla 2: Entidad Tecnologías

Aquí se encuentran las tecnologías que existen en total en la EPS.

Son Computación (CO), Ingeniería del Software (IS), Sistemas de Información (SI), Ingeniería de Computadores (IC), Tecnologías de la Información (TI), Sistemas Electrónicos (EL), Radiocomunicaciones (RA), Tratamiento de Señales (SE) y Telemática (TE).

3.2.3. GRADO_TECNOLOGÍAS

Atributo	Tipo	Descripción
<u>Id_grado</u>	Grados (Id)	Clave externa que referencia al Id de la tabla Grados
<u>Id_tecnología</u>	Tecnologías (Id)	Clave externa que referencia al Id de la tabla Tecnologías

Tabla 3: Entidad Grado_tecnologías

Esta tabla relaciona los grados con las tecnologías.

Tanto a GII como DGI pertenecen CO, IS, SI, IC y TI.
A GITST pertenecen el resto: EL, RA, SE y TE.

3.2.4. DEPARTAMENTOS

Atributo	Tipo	Descripción
<u>Id</u>	Integer	Clave Primaria
Código	Varchar	Código del Departamento
Nombre	Varchar	Nombre completo del Departamento

Tabla 4: Entidad Departamentos

En esta entidad se encuentran los departamentos de la EPS.

Existen dos: Departamento de Ingeniería Informática (II) y Departamento de Tecnología Electrónica y de las Comunicaciones (TEC).

3.2.5. DOCENTES

Atributo	Tipo	Descripción
<u>Id</u>	Integer	Clave Primaria
Nombre	Varchar	Nombre del Docente
Apellido1	Varchar	Primer apellido
Apellido2	Varchar	Segundo apellido
Email	Varchar	Email del Docente
Teléfono	Integer	Teléfono del Docente
Categoría	Varchar	Categoría del Docente
Departamento	Departamentos(Id)	Clave externa que referencia al Id de la tabla Departamentos
Activo	Boolean	Si actualmente está activo o no

Tabla 5: Entidad Docentes

En esta tabla se almacenan todos los Docentes.

Se guarda su nombre completo, email, teléfono, categoría, departamento (referenciando al Id de la tabla Departamentos) y si está activo. Este último campo es necesario, ya que puede haber algún docente que ofertó algún TFG o aparece como ponente pero que actualmente no está en la EPS.

Las diferentes categorías pueden ser: Catedrático de Universidad (CU), Profesor Titular de Universidad (TU), Profesor Contratado Doctor (CDR), Profesor Ayudante Doctor (AYD), Ayudante (AYU) y Profesor Asociado (ATP).

3.2.6. DOCENTE_TECNOLOGÍAS

Atributo	Tipo	Descripción
Id_docente	Docentes (Id)	Clave externa que referencia al Id de la tabla Docentes
Id_tecnología	Tecnologías (Id)	Clave externa que referencia al Id de la tabla Tecnologías

Tabla 6_ Entidad Docente_tecnologías

Esta tabla relaciona la tabla Docentes con la de Tecnologías.

De esta forma, quedan definidas las tecnologías que tiene cada docente, ya que puede estar asociado a un departamento, pero tener tecnologías de ambos departamentos.

3.2.7. ESTUDIANTES

Atributo	Tipo	Descripción
<u>Id</u>	Integer	Clave Primaria
Nombre	Varchar	Nombre del Estudiante
Apellido1	Varchar	Primer apellido
Apellido2	Varchar	Segundo apellido
Email	Varchar	Email del Estudiante
Grado	Grados (Id)	Clave externa que referencia al Id de la tabla Grados

Tabla 7: Entidad Estudiantes

Entidad donde se guardan los estudiantes.

Se almacena el nombre completo, el email y el grado al que pertenecen. Este último referenciando al Id de la tabla Grados.

3.2.8. TFGS

Atributo	Tipo	Descripción
Codigo	Varchar	Clave Primaria
Grado	Grados(Id)	Clave externa que referencia al Id de la tabla Grados
Título	Varchar	Título del TFG
Fecha_solicitud	Varchar	Fecha en la que se solicitó el TFG
Proponente_estudiante	Estudiantes (Id)	Clave externa que referencia al Id de la tabla Estudiantes
Proponente_docente	Docentes (Id)	Clave externa que referencia al Id de la tabla Docentes
Ponente	Docentes (Id)	Clave externa que referencia al Id de la tabla Docentes
Tutor	Docentes(Id)	Clave externa que referencia al Id de la tabla Docentes
Descripción	Varchar	Descripción del TFG
Requisitos	Varchar	Requisitos del TFG
Lugar	Varchar	Lugar de realización del TFG
Beca	Varchar	Si el TFG dispone de beca
Prácticas_externas	Varchar	Si el TFG se puede realizar a la vez que las prácticas externas
Estado	Varchar	Estado del TFG actual
Autor	Estudiantes (Id)	Clave externa que referencia al Id de la tabla Estudiantes

Tabla 8: Entidad TFGS

En esta tabla se almacenan todos los TFG.

Tiene todos los campos que aparecen en las propuestas de los TFG de la página Moodle [8], es decir (por orden de aparición):

- Proponente: Nombre completo del docente o estudiante que propone el TFG. En este caso es una referencia a la tabla Docentes o Estudiantes. El campo es Proponente_estudiante o Proponente_docente, el que convenga
- Vinculado a prácticas externas: Sí o No
- Título: el título del TFG. Campo título
- Tutor(a): Nombre completo del tutor. Referencia a la tabla Docentes
- Ponente (si procede): Si tiene ponente, aparecerá el nombre completo de este. Es una referencia a la tabla Docentes.
- Descripción y objetivos: campo Descripción
- Requisitos y conocimientos del estudiante: campo Requisitos
- Lugar de realización del proyecto y horario tentativo: campo Lugar
- Beca: campo Beca
- Fecha solicitud: campo Fecha_solicitud

Los campos como Teléfono o Email no aparecen en esta tabla, ya que ya están en la tabla Docentes.

El campo Estado puede obtener los siguientes valores:

- Ofertado: el TFG está ofertado pero no está asignado a ningún estudiante
- Cancelado: el TFG ya no se oferta, y por tanto nunca ha sido defendido
- Asignado: ya se encuentra asignado, por lo que se espera que se defienda en el actual año académico.
- Defendido: el TFG ya ha sido defendido

3.2.9. TFG_TECNOLOGÍAS

Atributo	Tipo	Descripción
Codigo_tfg	TFGS (Codigo)	Clave externa que referencia al Codigo de la tabla TFGS
Id_tecnología	Tecnologías (Id)	Clave externa que referencia al Id de la tabla Tecnologías

Tabla 9: Entidad TFG_tecnologías

Esta tabla relaciona la entidad TFG con la entidad Tecnologías.

Esto es importante cuando se vayan a generar automáticamente las comisiones, para poder agrupar los TFG por tecnologías.

3.2.10. COMISIONES

Atributo	Tipo	Descripción
Codigo	Varchar	Clave Primaria
Curso	Varchar	Curso al que pertenece la Comision
Convocatoria	Varchar	Convocatoria a la que pertenece la Comision
Grado	Grados (Id)	Clave externa que referencia al Id de la tabla Grados
Presidente	Docentes (Id)	Clave externa que referencia al Id de la tabla Docentes
Vocal1	Docentes (Id)	Clave externa que referencia al Id de la tabla Docentes
Vocal2	Docentes (Id)	Clave externa que referencia al Id de la tabla Docentes
Suplente1	Docentes (Id)	Clave externa que referencia al Id de la tabla Docentes
Suplente2	Docentes (Id)	Clave externa que referencia al Id de la tabla Docentes

Tabla 10: Entidad Comisiones

Esta es la entidad donde se almacenan las comisiones, tanto las ya finalizadas como las de la próxima convocatoria (si ya se han autogenerado).

El código de los TFG sigue el patrón AÑO_NÚMERO_TECNOLOGÍAS. Por ejemplo, 1415_007_SI. Se guarda este código, junto con el curso, la convocatoria y el grado al que pertenecen.

Además, tiene cinco referencias a la tabla Docentes para guardar el Presidente de la comisión, así como los dos vocales y los dos suplentes.

3.2.11. COMISION_TFGS

Atributo	Tipo	Descripción
Codigo_comision	Comisiones (Codigo)	Clave externa que referencia al Codigo de la tabla Comisiones
Codigo_tfg	TFGS (codigo)	Clave externa que referencia al Codigo de la tabla TFGS
Sala	Varchar	Sala de la Comision
Fecha	Varchar	Fecha de la Comision
Hora	Time	Hora de la Comision

Tabla 11: Entidad Comisión_tfgs

Esta tabla relaciona las entidades Comisión y TFG.

Por tanto, tiene los códigos (claves primarias) de las dos entidades. También guarda la sala, fecha y hora referentes a dónde y cuándo fue/será la comisión.

3.3. Recopilación y unificación de datos

Todos los datos recopilados han sido extraídos de contenido de acceso público, sin tratar información comprometida, salvaguardando en todo momento la privacidad de docentes y estudiantes.

Esta recopilación se detalla en dos apartados. En primer lugar se explica lo referente a los grados de Ingeniería Informática y Doble Grado en Ingeniería Informática y Matemáticas. Y a continuación se detalla lo referente al Grado en Ingeniería de Tecnologías y Servicios de Telecomunicación.

3.3.1. GII y DGI

La tabla Docentes se extrajo casi en su totalidad de la página web de la EPS [3], obteniendo de aquí el nombre, apellidos, email, teléfono, categoría y el departamento.

Para saber las tecnologías que cada docente tiene asociadas, fue necesario acceder a una hoja Excel que el anterior coordinador de Trabajos de Fin de Grado facilitaba a los docentes donde marcaran las tecnologías pertinentes. Así, los nombres de los primeros docentes añadidos a la Base de Datos se extrajeron de aquí. El resto se han ido añadiendo a medida que aparecen en las propuestas de TFG o en las CCEE posteriores.

Respecto a la tabla Estudiantes, se extrajo a la vez que la de TFG, ya que a la hora de recopilar los datos los estudiantes que se iban a almacenar eran únicamente los que ya habían defendido algún TFG. Por tanto, accediendo a la página web [2] donde se gestionaban y consultaban las CCEE durante los dos primeros años de docencia de TFG (2012-2013, 2013-2014) se extrajeron parte de los datos de la tabla TFG y los datos referentes a estudiantes (nombre, apellidos y grado al que pertenecen). Los emails no se han podido recuperar, por lo que este campo, por ahora, permanece vacío.

Así pues, de TFG se pudo extraer el código, el título y el autor (estudiante). El ponente se pudo recuperar accediendo uno a uno a las memorias presentadas por los estudiantes. Otros datos como la descripción, los requisitos o la fecha de asignación no fue posible recuperarlos, ya que las propuestas de estos TFG no se almacenaron.

Los datos de la tabla Comisiones también fueron extraídos de esta página web: el código de cada comisión, el curso, convocatoria y los docentes que la componen. Para completar la tabla Comision_tfgs, hubo que ir guardando uno a uno los códigos de los TFG, ya que estos no aparecían en orden y había que ir buscando las memorias referentes a cada uno para buscar el ponente del proyecto, en caso de que tuviera.

3.3.2. GITST

En el caso de GITST sólo ha habido un año de presentaciones de TFG, el 2013-2014. Para extraer los datos de estos TFG y CCEE, hubo que acceder a los ficheros PDF proporcionados por el coordinador de este grado. Al igual que ocurre con GII o DGI, había que acceder a las memorias de los TFG correspondiente para extraer datos como el título del trabajo, el tutor y el ponente correspondientes a las tablas Comisiones y TFG.

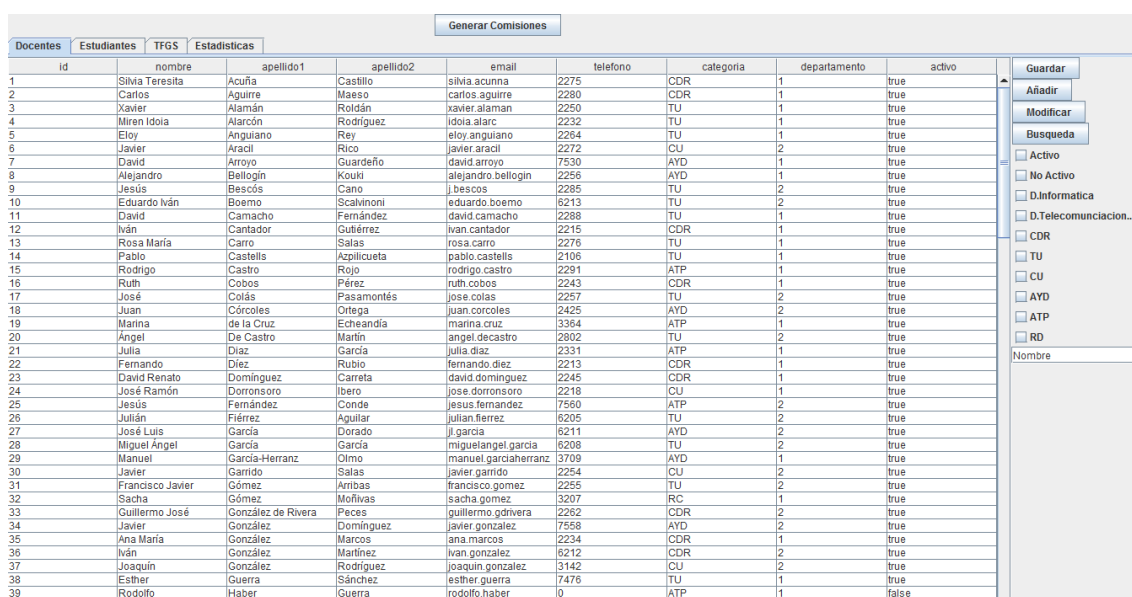
Otros campos como la fecha, hora y lugar donde se celebraron las comisiones, así como el código de la propia comisión, se almacenaban en ficheros de texto extraídos de los emails donde se confirmaban estos datos. Esto era importante para completar la tabla Comision_tfgs.

Los datos pertenecientes a las tablas Docentes y Estudiantes han sido extraídos de igual manera que se ha explicado en el apartado anterior, para el caso de GII y DGI.

4. Herramienta software desarrollada

En este capítulo se describe la funcionalidad de la aplicación. Se subdivide en cinco apartados, explicando en cada uno la funcionalidad de cada una de las cinco partes principales de las que se compone la herramienta software. Para que la explicación sea más clara, se aportan capturas de pantalla.

4.1. Docentes



id	nombre	apellido1	apellido2	email	telefono	categoria	departamento	activo
1	Silvia Teresa	Acuña	Castillo	silvia.acuna	2275	CDR	1	true
2	Carlos	Aguirre	Maeso	carlos.aguirre	2280	CDR	1	true
3	Xavier	Alamán	Roldán	xavier.alaman	2250	TU	1	true
4	Miren Idola	Alarcón	Rodríguez	idola.alarc	2232	TU	1	true
5	Eloy	Anguiano	Rey	eloy.anguiano	2264	TU	1	true
6	Javier	Araçil	Rico	javier.aracil	2272	CU	2	true
7	David	Arroyo	Guardaño	david.arroyo	7530	AYD	1	true
8	Alejandro	Bellogin	Koukl	alejandro.bellogin	2256	AYD	1	true
9	Jesús	Bescós	Cano	j.bescos	2285	TU	2	true
10	Eduardo Iván	Boemo	Scalvinoni	eduardo.boemo	6213	TU	2	true
11	David	Camacho	Fernández	david.camacho	2288	TU	1	true
12	Iván	Cantador	Gutiérrez	ivan.cantador	2215	CDR	1	true
13	Rosa María	Carro	Salas	rosa.carro	2276	TU	1	true
14	Pablo	Castells	Aspigueta	pablo.castells	2106	TU	1	true
15	Rodrigo	Castro	Rojo	rodrigo.castro	2291	ATP	1	true
16	Ruth	Cobos	Pérez	ruth.cobos	2243	CDR	1	true
17	José	Colás	Pasamontés	jose.colas	2257	TU	2	true
18	Juan	Córcoles	Ortega	juan.corcoles	2425	AYD	2	true
19	Marina	de la Cruz	Echeandía	marina.cruz	3364	ATP	1	true
20	Ángel	De Castro	Marín	angel.decastro	2802	TU	2	true
21	Julia	Díaz	García	julia.diaz	2331	ATP	1	true
22	Fernando	Díez	Rubio	fernando.diez	2213	CDR	1	true
23	David Renato	Dominguez	Carreta	david.dominguez	2245	CDR	1	true
24	José Ramón	Dorronsoro	Ibero	jose.dorronsoro	2218	CU	1	true
25	Jesús	Fernández	Conde	jesus.fernandez	7560	ATP	2	true
26	Julían	Fiérez	Aguilar	julian.fierrez	6205	TU	2	true
27	José Luis	García	Dorado	jl.garcia	6211	AYD	2	true
28	Miguel Ángel	García	García	miguelangel.garcia	6208	TU	2	true
29	Manuel	García-Herranz	Olmo	manuel.garciaherranz	3709	AYD	1	true
30	Javier	Garrido	Salas	javier.garrido	2254	CU	2	true
31	Francisco Javier	Gómez	Arribas	francisco.gomez	2255	TU	2	true
32	Sacha	Gómez	Mollivas	sacha.gomez	3207	RC	1	true
33	Guillermo José	González de Rivera	Peces	guillermo.gdrivera	2262	CDR	2	true
34	Javier	González	Domínguez	javier.gonzalez	7558	AYD	2	true
35	Ana María	González	Marcos	ana.marcos	2234	CDR	1	true
36	Iván	González	Martínez	ivan.gonzalez	6212	CDR	2	true
37	Joaquín	González	Rodríguez	joaquin.gonzalez	3142	CU	2	true
38	Esther	Guerra	Sánchez	esther.guerra	7476	TU	1	true
39	Rodolfo	Haber	Guerra	rodolfo.haber	0	ATP	1	false

Figura 3: Pestaña principal Docentes

Esta es la primera pestaña de la herramienta. Como se puede apreciar, se muestran todos los campos de la entidad docente: nombre, apellidos, email, teléfono, categoría, departamento y activo.

En el menú de la derecha aparecen una serie de botones, con los que se puede interactuar con la tabla:

- Guardar: Vuelca los datos en un fichero de texto. Sale una ventana emergente preguntando donde se desea guardar, como se muestra en la siguiente captura de pantalla:



Figura 4: Botón Guardar pestaña Docentes

- Añadir: La función de este botón es agregar un docente a la base de datos. Aparece un formulario, donde se deberán completar los datos solicitados. Se puede salir de esta ventana sin guardar los cambios pulsando en *Cancelar*. Si se presiona *Guardar*, se almacenarán los cambios en la base de datos.

NUEVO DOCENTE	
Activo	D.Informatica
Nombre	Categoria
Apellido1	Apellido2
Email	Telefono
<input type="button" value="Guardar"/> <input type="button" value="Cancelar"/>	

Figura 5: Botón Añadir pestaña Docentes

- **Modificar:** Para modificar cualquier fila, se puede pulsar sobre ella. Se puede ir modificando cualquier campo sin que se almacenen los cambios. Sólo tras pulsar en el botón *Modificar* se guardarán los cambios en la base de datos.
- **Búsqueda:** Los *checkbox* que aparecen debajo de este botón sirven para establecer los filtros de búsqueda deseados. Así pues, se puede elegir entre “activo” y “no activo”, la categoría y el departamento. Además, se puede escribir en el campo de texto para especificar más la búsqueda deseada.

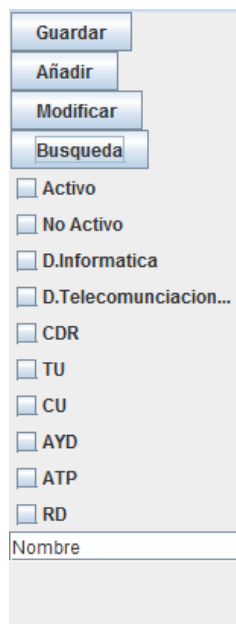


Figura 6: Filtro búsqueda pestaña Docentes

4.2. Estudiantes

Generar Comisiones							
Docentes		Estudiantes	TFGS	Estadísticas			
id	nombre	apellido1	apellido2	email	grado		
1	Cristina	Archilla	Manso		1		
2	Carlos	Delgado	Calle		1		
3	Daniel	Faro	García		1		
4	Cristina	García	Rufes		1		
5	Victor	López	Rivero		1		
6	Pablo	Ortega	Moreno		1		
7	Lidia	París	Cabello		1		
8	Ignacio	del Pozo	Martínez		1		
9	Manuel	Pozo	Muñoz		1		
10	Eloy	de la Rubia	Luenzo		1		
11	Daniel	Santoja	Merino		1		
12	Jorge	Magallón	Villa		1		
13	Daniel	Michaud	Vallinoto		1		
14	Santiago	Pérez	Castro		1		
15	Jorge	Femenía	del Rey		1		
16	Manuel	Konomi	Pilkati		1		
17	Alejandro	Ortiz	Martín		1		
18	Paula	Roquero	Fuentes		1		
19	Miguel	Vázquez	Fernández de Lezeta		1		
20	Rafael	Vindel	Amor		1		
21	Alma	Castillo	Antolín		1		
22	Alfredo	de Cea	Bello		1		
23	Irene	Córdoba	Sánchez		1		
24	Cristina	Martín	Montero		1		
25	Roberto	Rozalén	Calvo		1		
26	Alfonso	Alhambra	Morón		1		
27	Rocio	Castillares	Pérez		1		
28	Chen Da	Liu	Zhang		1		
29	Sofía	Marina	Pepa		1		
30	Cristina	Pruenza	García-Hinojosa		1		
31	Álvaro	García	Delgado		1		
32	Pedro José	García	López		1		
33	Mario	González	Nahón		1		
34	Felipe	Jasmin	Salas		1		
35	Dagmara	Wójcik			1		
36	Adalberto	Plaza	Jurado		1		
37	Alba	Calvo	Ruiz		1		
38	Juan Carlos	Torrado	Vidal		1		
39	Miguel	Moraleda	Ponzola		1		

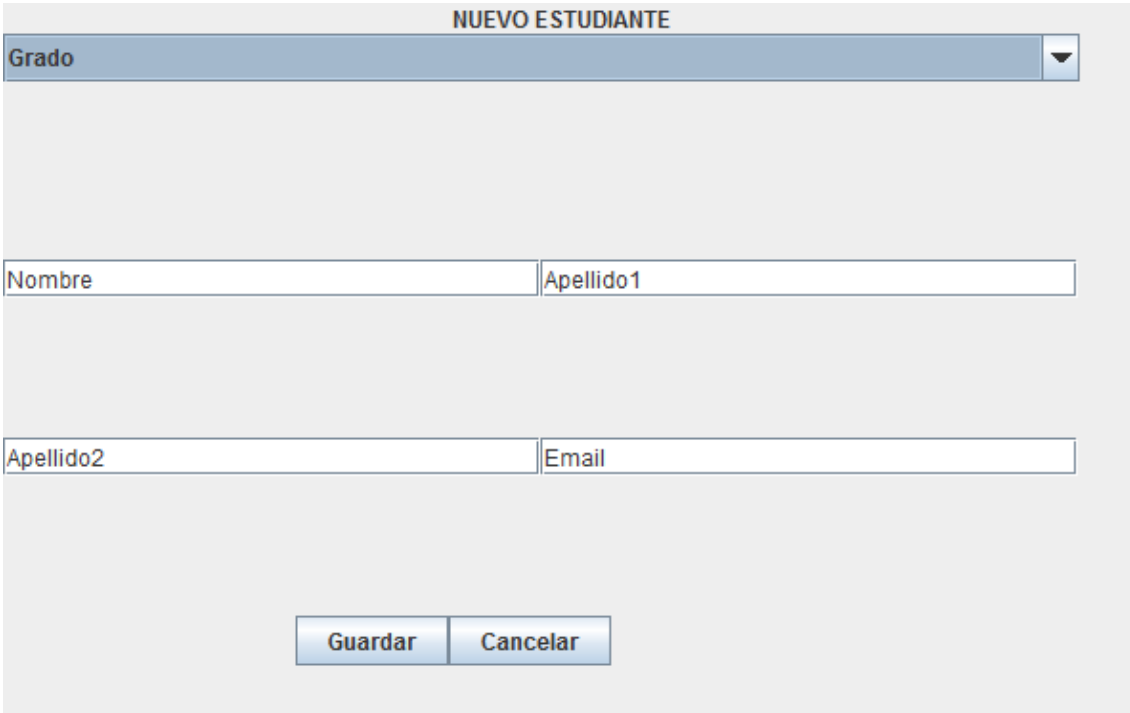
Figura 7: Pestaña principal Estudiantes

La imagen anterior pertenece a la pestaña de Estudiantes. Como se puede apreciar, el diseño es igual que el de Docentes, pero cambiando algunos aspectos.

Los campos que aparecen aquí son el nombre completo del estudiante, su email y el grado al que pertenecen. Como ya se ha explicado en la sección “3.3 Recopilación y unificación de datos”, los email de los estudiantes no se pudieron recuperar, con lo que este campo permanece vacío por ahora.

La funcionalidad de los botones que aparecen a la derecha es la misma que para la pestaña de Docentes. Los cambios están en el botón *Añadir* y en el filtro de búsqueda.

Para el botón *Añadir* aparecen a completar los campos grado, nombre, apellidos y email. Tras pulsar *Guardar* los cambios se almacenarán en la base de datos. Si no se desean guardar los cambios, basta con pulsar el botón *Cancelar*.



El formulario, titulado "NUEVO ESTUDIANTE", contiene un menú desplegable "Grado" en la parte superior. A continuación, hay dos filas de campos de texto: la primera fila tiene "Nombre" y "Apellido1"; la segunda fila tiene "Apellido2" y "Email". En la parte inferior del formulario, hay dos botones: "Guardar" y "Cancelar".

Figura 8: Botón Añadir pestaña Estudiantes

Aquí la búsqueda se puede filtrar por Grado y, al igual que en Docentes, especificar más la búsqueda completando el cuadro de texto.

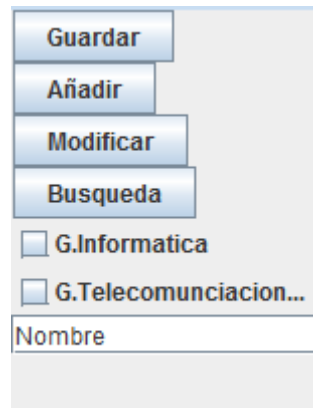


Figura 9: Filtro búsqueda pestaña Estudiantes

4.3. Trabajos de Fin de Grado

Generar Comisiones						
Docentes	Estudiantes	TFGS	Estadísticas			
	codigo	grado	titulo	tutor	estado	
	1415_001_SE	2	Generación de panorámicas de fondo a...	9	ofertado	
	1415_001_TI	1	Creación de una aplicación web para la	7	ofertado	
	1415_002_EL	2	Aplicación Android para autoevaluación d...	10	ofertado	
	1415_002_SITI	1	Desarrollo de una aplicación Web para m...	8	ofertado	
	1415_003_SE	2	Desarrollo de herramienta para la anotac...	125	ofertado	
	1415_003_SITI	1	Desarrollo de un sistema de recogida de...	8	asignado	
	1415_004_SE	2	Desarrollo de interfaces en entornos And...	125	ofertado	
	1415_004_SITI	1	Desarrollo de un sistema de recogida y c...	9	asignado	
	1415_005_ELRA	2	Caracterización de las condiciones exper...	18	ofertado	
	1415_005_TI	1	Generación de panorámicas de fondo a...	9	ofertado	
	1415_006_ELRA	2	Diseño asistido por ordenador de filtros ...	18	ofertado	
	1415_006_SI	1	Herramienta de gestión de Trabajos de F...	12	asignado	
	1415_007_ELRA	2	Diseño asistido por ordenador de transfo...	18	ofertado	
	1415_007_SI	1	Herramienta de gestión de Trabajos de F...	12	asignado	
	1415_008_EL	2	Aplicación Android para la enseñanza de ...	113	ofertado	
	1415_008_TI	1	Desarrollo de herramienta para la anotac...	125	ofertado	
	1415_009_EL	2	Control en FPGA de fuente de alimentaci...	20	ofertado	
	1415_009_TI	1	Desarrollo de interfaces en entornos And...	125	ofertado	
	1415_010_COSI	1	Novedad y diversidad en motores de bus...	14	ofertado	
	1415_010_EL	2	Diseño e implementación de módulos p...	20	asignado	
	1415_011_COSI	1	Recomendación en entornos de red social	14	asignado	
	1415_011_EL	2	Estudio comparativo de plataformas HIL...	20	asignado	
	1415_012_COSITI	1	Explotación de analíticas de aprendizaje ...	16	asignado	
	1415_012_SE	2	Detección de la posición relativa de una ...	111	ofertado	
	1415_013_CO	1	Generación de mallas 3D bidimensionale...	18	ofertado	
	1415_013_SE	2	Identificación automática de materiales u...	111	ofertado	
	1415_014_CO	1	Integración de un eipensolver paraleliza...	18	ofertado	
	1415_014_TE	2	Monitorización del ancho de banda de re...	27	ofertado	
	1415_015_JC	1	Control en FPGA de convertidor conmuta...	20	ofertado	
	1415_015_SE	2	Reconocimiento de imágenes mediante...	28	ofertado	
	1415_016_JC	1	Emulación en hardware del lazo cerrado...	20	ofertado	
	1415_016_SE	2	Seguimiento de personas en sistemas...	28	ofertado	
	1415_017_IS	1	Generación automática de interfaces de...	46	ofertado	
	1415_017_SE	2	Video-seguimiento mediante modelos d...	28	ofertado	
	1415_018_JSSI	1	Visualización escalable de modelos soft...	46	asignado	
	1415_018_SE	2	Detección de personas en grupos...	108	asignado	
	1415_019_COIC	1	Paralelización, optimización y uso de GP...	128	ofertado	
	1415_019_SE	2	Preprocesado para la mejora de eficienci...	108	ofertado	
	1415_020_COTI	1	Data Analytics en energías renovables	24	ofertado	

Figura 10: Pestaña principal Trabajos de Fin de Grado

En la tercera pestaña se encuentran los TFG. Se muestran los campos código, grado, título, tutor y estado.

El botón *Guardar* actúa como en las dos pestañas ya explicadas, exporta a un fichero de texto con la información que se muestra en la pestaña actual.

Lo mismo ocurre con el botón *Modificar*, no cambia su funcionalidad aquí. Es para que los datos que se hayan modificado manualmente se guarden en la base de datos. Es importante resaltar que los datos que se ven en la tabla no estarán guardados en la base de datos hasta que no se pulse este botón.

El botón *Información* amplía la información que se muestra en la pantalla principal. Aparecen los datos que se muestran en la siguiente captura:

Información del TFG	
Código:	1415_007_SI
Grado:	1
Título:	Herramienta de gestión de Trabajos de Fin de Grado: Módulos de gestión de profesorado y comisiones de evaluación
Fecha de solicitud:	30/05/2014
Proponente Estudiante:	—
Proponente Docente:	12
Ponente:	—
Tutor:	12
Descripción:	El objetivo de este trabajo es el diseño e implementación de una herramienta software que automatice varios de los procesos de gestión de los Trabajos de Fin de Grado en la EPS-UAM. Al hacerse en conjunción
Requisitos:	Conocimiento y experiencia en programación Java, diseño y uso de bases de datos relacionales.
Lugar:	Flexible, aunque se valorará el hecho de que el candidato pueda asumir un horario fijo en un laboratorio del edificio B de la escuela.
Beca:	—
Prácticas Externas:	—
Estado:	asignado
Autor:	—

Figura 11: Botón Información pestaña TFG

Son los datos que aparecen en las propuestas de TFG, mostrando adicionalmente campos como Grado o el Estado.

Por último, el botón *Búsqueda* muestra en la tabla principal los datos que se deseen mediante el filtro de búsqueda.

Figura 12: Filtro búsqueda pestaña TFG

Se puede filtrar por Grado o por Estado (asignado, cancelado, ofertado o defendido). Para una búsqueda más específica se puede introducir el código de un TFG en el cuadro de texto.

4.4. Estadísticas

Esta pestaña muestra una serie de estadísticas, recogiendo la información que actualmente se encuentra en la base de datos. Se dan cuatro tablas con estadísticas distintas, cada una de ellas descritas en los siguientes apartados.

4.4.1. Tecnología_Tfgs

Generar Comisiones						
Docentes	Estudiantes	TFGS	Estadísticas			
Tecnología	Curso 2012/2013	Curso 2013/2014	Curso 2014/2015	TOTAL		
Computación	14	8	47	69	Tecnología_Tfgs	
Ingeniería del Software	1	5	21	27	Docente_Tfgs	
Sistemas de Información	6	4	43	53	Departamento_Tfgs	
Ingeniería de Computadores	1	8	13	22	Docente_comision	
Tecnologías de la Información	7	9	47	63		
Sistemas Electrónicos	0	13	29	42		
Radio comunicaciones	0	8	15	23		
Tratamiento de Señales	0	13	36	49		
Telemática	0	9	6	15		

Figura 13: Estadísticas Tecnología_Tfgs

Se muestran en primer lugar las 9 tecnologías que existen en total en la EPS, en los tres grados. En las siguientes columnas aparecen los tres cursos académicos en que hasta ahora ha habido Trabajos de Fin de Grado.

El objeto de esta pestaña es mostrar el número de TFG que, atendiendo a cada tecnología, aparecen en los distintos cursos. Por ejemplo, en el primer curso, 2012/2013, hubo un total de 14 TFG con la tecnología Computación, mientras que de las tecnologías pertenecientes a GITST no hubo ningún TFG, debido a que aún no existía docencia para la asignatura.

4.4.2. Docente_Tfgs

Generar Comisiones					
Docentes	Estudiantes	TFGS	Estadísticas		
Docente	Curso 2012/2013	Curso 2013/2014	Curso 2014/2015	TOTAL	
Silvia Teresita Acuña Castillo	0	0	0	0	Tecnología_Tfgs
Carlos Aguirre Maeso	0	0	1	1	Docente_Tfgs
Xavier Alamán Roldán	0	0	0	0	Departamento_Tfgs
Miren Idoia Alarcón Rodríguez	0	0	1	1	Docente_comision
Eloy Anguliano Rey	0	0	0	0	
Javier Aracil Rico	0	0	1	1	
David Arroyo Guardado	0	1	4	5	
Alejandro Bellogin Kouki	0	0	3	3	
Jesús Bescós Cano	0	0	2	2	
Eduardo Iván Boemo Scalvinoni	0	0	1	1	
David Camacho Fernández	0	0	0	0	
Iván Cañalder Gutiérrez	0	0	4	4	
Rosa María Carro Salas	0	0	0	0	
Pablo Castells Azpilicueta	1	0	2	3	
Rodrigo Castro Rojo	0	0	0	0	
Ruth Cobos Pérez	0	0	1	1	
José Colás Pasamontés	0	0	0	0	
Juan Córdiles Oitega	0	2	3	5	
Marina de la Cruz Echeandia	0	0	0	0	
Ángel De Castro Martín	0	3	2	5	
Julia Diaz Gardá	0	0	0	0	
Fernando Díez Rubio	0	0	2	2	
David Renato Domínguez Carreta	0	0	0	0	
José Ramón Drottonero Ibero	0	0	1	1	
Jesús Fernández Conde	0	0	0	0	
Julian Fierrez Aguilár	0	0	0	0	
José Luis García Dorado	0	1	1	2	

Figura 14: Estadísticas Docente_Tfgs

En esta tabla se muestran las estadísticas referentes a Docentes y TFG. En cada columna aparece el número de TFG que cada docente tiene para cada curso.

4.4.3. Departamento_Tfgs

Generar Comisiones				
Docentes	Estudiantes	TFGS	Estadísticas	
Departamento	Curso 2012/2013	Curso 2013/2014	Curso 2014/2015	TOTAL
Departamento de Ingeniería Informática	16	26	94	136
Departamento de Tecnología Electrónica y de las Comunicaciones	0	31	76	107

Figura 15: Estadísticas Departamento_Tfgs

Aquí se muestran las estadísticas que relacionan los Departamentos con los TFG. Aparecen los dos departamentos de la EPS, Departamento de Ingeniería Informática y Departamento de Tecnología Electrónica y de las Comunicaciones. En cada columna se muestra un año académico diferente, mientras que en la última se muestra el total, es decir, la suma de todos los años.

4.4.4. Docente_comisión

Generar Comisiones				
Docentes	Estudiantes	TFGS	Estadísticas	
Docente	Curso 2012/2013	Curso 2013/2014	Curso 2014/2015	TOTAL
Silvia Teresita Azuña Castillo	0	0	0	0
Carlos Aguirre Maeso	0	1	0	1
Xavier Alamán Roldán	0	1	0	1
Miren Idoia Alarcón Rodríguez	0	1	0	1
Eloy Anguiano Rey	0	1	0	1
Javier Aracil Rico	0	0	0	0
David Arroyo Guardado	0	2	0	2
Alejandro Ballogin Kouki	0	1	0	1
Jesús Bescós Cano	0	1	0	1
Eduardo Iván Boemo Scalvinoni	0	0	0	0
David Camacho Fernández	0	1	0	1
Iván Cantador Guillén	0	2	0	2
Rosa María Carro Salas	0	1	0	1
Pablo Castells Azpilicueta	0	1	0	1
Rodrigo Castro Rojo	0	0	0	0
Ruth Cobos Pérez	0	0	0	0
José Colás Pasamontés	0	0	0	0
Juan Córcoles Ortega	0	2	0	2
Marina de la Cruz Echeandía	0	0	0	0
Ángel De Castro Martín	0	1	0	1
Julia Diaz Garcia	0	0	0	0
Fernando Díez Rubio	0	3	0	3
David Renato Domínguez Carreta	0	1	0	1
José Ramón Dorronsoro Ibero	0	1	0	1
Jesús Fernández Conde	0	0	0	0
Julían Fíérez Aguilár	0	0	0	0
José Luis García Dorado	0	0	0	0

Figura 16: Estadísticas Docente_comisión

La última pestaña de estadísticas muestra la relación entre los docentes y las comisiones. Es decir, se muestra el número de veces que un docente apareció en una comisión, y en la última columna, el total de veces que ha asistido a una CCEE.

4.5. Comisiones de Evaluación

La última parte que queda por explicar es donde se generan las Comisiones de Evaluación.

Tras pulsar el botón de *Generar Comisiones* aparece la pantalla para configurar los diferentes aspectos de estas:

The image shows a configuration form with the following elements:

- Grado:** A dropdown menu with 'GII' selected.
- Número máximo de TFGs por comisión:** A numeric input field with '8' and increment/decrement buttons.
- Curso:** A dropdown menu with '2014-2015' selected.
- Convocatoria:** A dropdown menu with 'Febrero' selected.

Figura 17: Configuración para generar Comisiones

Se puede elegir entre los grados GII y GITST.

El número máximo de TFG por comisión indica cuántos TFG se pueden dar como máximo en una comisión.

Por último se escoge el curso y la convocatoria.

Tras pulsar en *Siguiente*, aparece la siguiente pantalla:

The interface is divided into several sections:

- TFGs sin asignar:** A list on the left containing '1415_051_COIC' and '1415_004_SITI'. Below it is a dropdown menu showing 'TFG1415-II-01' and an 'Asignar' button.
- Comisiones:** A grid of nine commission boxes, each with a 'Quitar' button below it.
 - TFG1415-II-01:** 1415_137_CO, 1415_128_CO, 1415_114_CO, 1415_113_CO, 1415_059_CO, 1415_036_CO, 1415_035_CO, 1415_033_CO
 - TFG1415-II-02:** 1415_144_JS, 1415_135_JS, 1415_119_JS, 1415_118_JS, 1415_090_JS, 1415_088_JS, 1415_136_JSSI, 1415_127_JSSI
 - TFG1415-II-03:** 1415_007_SI, 1415_006_SI, 1415_133_SI, 1415_130_SI, 1415_129_SI, 1415_140_COSI, 1415_138_COSI, 1415_099_COSI
 - TFG1415-II-04:** 1415_110_IC, 1415_092_IC, 1415_091_IC, 1415_048_IC, 1415_141_ICTI, 1415_065_ICTI
 - TFG1415-II-05:** 1415_055_TI, 1415_054_TI, 1415_045_TI, 1415_143_TI, 1415_142_TI, 1415_134_TI, 1415_109_TI, 1415_056_COTI
 - TFG1415-II-06:** 1415_098_COSI, 1415_097_COSI, 1415_096_COSI, 1415_038_COSI, 1415_011_COSI, 1415_069_COC, 1415_057_COC
 - TFG1415-II-07:** 1415_037_COTI, 1415_120_SITI, 1415_105_SITI, 1415_101_SITI, 1415_032_SITI, 1415_031_SITI, 1415_030_SITI
 - TFG1415-II-08:** 1415_062_JSSI, 1415_061_JSSI, 1415_042_JSSI, 1415_018_JSSI, 1415_139_JSTI, 1415_126_JSTI, 1415_087_JSTI, 1415_066_JSTI
 - TFG1415-II-09:** 1415_003_SITI, 1415_093_COISSI, 1415_060_COISSI, 1415_012_COISITI, 1415_070_COISITI, 1415_117_COISITI, 1415_115_COISSICTI
- Buttons:** 'Siguiente' and 'Cancelar' buttons at the bottom.

Figura 18: Comisiones autogenerated – Asignar TFG

En la parte izquierda de la captura aparecen los TFG que de momento no se han asignado a ninguna comisión. Para asignarlos, se elige el nombre de la comisión deseada en el desplegable y se pulsa el botón *Asignar*. Si se intenta asignar un TFG a alguna comisión que ya está llena, aparecerá un mensaje indicando esta situación: “La comisión seleccionada está llena”.

Para eliminar un TFG de alguna comisión, se puede seleccionar el TFG deseado y pulsar el botón *Quitar*, tras lo cual este aparecerá en la parte izquierda, en la lista de “TFG sin asignar”.

Los nombres de las comisiones se generan de forma automática, de acuerdo con el patrón TFGCurso-Grado-XX. Curso y grado se escogieron en la pantalla de configuración, y XX pertenece al número de comisión, empezando en 01. Por ejemplo, la primera comisión generada para el curso 2014-2015 del Grado en Ingeniería Informática sería TFG1415-II-01.

Cuando todos los TFG estén asignados en alguna comisión, se habilitará el botón *Siguiente*. Tras pulsar este botón aparece la última pantalla:

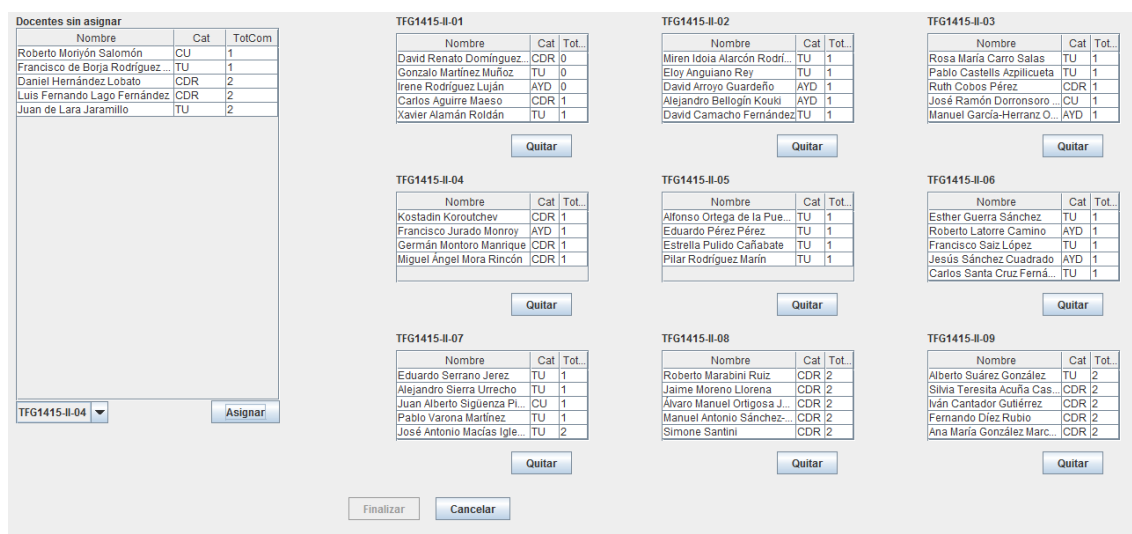


Figura 19: Comisiones autogeneradas - Asignar docentes

La mecánica de esta pantalla es similar a la de asignar TFG. En este caso, una vez asignados los TFG a las comisiones, hay que asignar los docentes.

Para cada docente se muestra su nombre y apellidos, su categoría y el número de veces que ha participado en una comisión. De esta forma, se escogerán primero los que tengan un número más bajo.

Cada comisión debe estar compuesta por cinco docentes: presidente, dos vocales y dos suplentes. En las tablas que se muestran en la pantalla aparecen en orden.

Tras quitar y asignar los docentes que se desee, una vez que todas las comisiones estén compuestas por los cinco docentes, el botón *Finalizar* aparecerá habilitado. Tras pulsar este botón, se hace el guardado en base de datos. En concreto, se actualizan las tablas Comisiones y Comisión_TFG.

5. Módulo de generación de Comisiones de Evaluación

En este capítulo se detalla a más bajo nivel la manera en que se generan automáticamente las Comisiones de Evaluación.

Este proceso se desarrolla en dos fases. La primera consiste en agrupar los TFG conforme a las tecnologías que estos llevan asociadas. Una vez todos los TFG pertenezcan a una comisión y, por tanto, formadas todas las comisiones, empieza la segunda fase. Esta asigna los cinco docentes que conforman un tribunal a cada comisión, atendiendo a las tecnologías de que estos disponen para que coincida con la agrupación por tecnologías resuelta en la fase anterior.

5.1. Fase de agrupación de Trabajos de Fin de Grado por tecnologías

Esta fase se desarrolla en la clase llamada *GenerarComisiones_TFG*. Esta clase necesita los parámetros establecidos en la primera pantalla de configuración de Comisiones, como se vio en la ilustración 17. Por tanto, se le pasan estos parámetros al constructor: grado, número máximo de TFG por comisión, curso, convocatoria y el *MainDBManager*, para realizar las consultas que correspondan.

5.1.1. Preparar datos

Como complemento a la clase *GenerarComisiones_TFG*, fue necesario crear una clase auxiliar nombrada *TFG_Aux*. Esto es debido a que la entidad TFG no dispone de los objetos necesarios para agrupar los TFG por tecnologías.

Así pues, esta clase está formada por el código del TFG, un *array* de las tecnologías pertenecientes al TFG y la comisión a la que este TFG pertenecerá. Además, esta clase implementa la interfaz *Comparable*, sobrescribiendo su método *compareTo*.

El objetivo es almacenar en un mismo *array* todos los TFG que compondrán la comisión, para poder comparar todos los ids de las tecnologías de cada TFG y ordenarlos. El criterio de ordenación es: Se ordenan de menor a mayor y, además, el que menos tecnologías tenga irá primero.

Por ejemplo, siguiente grupo de ids de tecnologías de varios TFG

{2} {1,3} {1} {2,4,5} {2,3} {5} {1,2} {2,3,4}

quedaría ordenado así:

{1} {2} {5} {1,2} {1,3} {2,3} {2,3,4} {2,4,5}

Para ordenar los Ids de las tecnologías de esta forma, se ha implementado el siguiente método *compareTo*:

```
/**
 * Compara dos objetos TFG_Aux
 *
 * @param other el objeto TFG_Aux a comparar
 */
@Override
public int compareTo(TFG_Aux other) {
    int long1 = this.getId_tecnologias().size();
    int long2 = other.getId_tecnologias().size();

    if (long1 < long2) {
        return -1;
    } else if (long1 > long2) {
        return 1;
    }

    for (int i = 0; i < long1; i++) {
        int elem1 = this.getId_tecnologias().get(i);
        int elem2 = other.getId_tecnologias().get(i);

        if (elem1 != elem2) {
            return elem1 < elem2 ? -1 : 1;
        }
    }

    return -1;
}
```

Código 1: Método *compareTo* de la clase *TFG_Aux*

En primer lugar, se ordena por número de elementos. Es decir, el que tenga un menor número de tecnologías irá posicionado antes en el *array*.

En caso de que el número de tecnologías sea el mismo, entonces se comparan uno a uno los elementos de cada *array* de tecnologías, quedando primero el que tenga los números más bajos.

Para formar el *array* compuesto de *TFG_Aux*, primero hay que acceder a la base de datos y extraer la información necesaria. Por tanto, haciendo uso de la clase *MainDBManager*, se accede a *TFGDBManager* para acceder a la tabla *TFGS* y así obtener los códigos de los TFG que se defenderán en la próxima comisión. La *query* queda de la siguiente forma:

```
SELECT codigo FROM tfg.tfgs WHERE estado='Asignado' AND grado IN
(SELECT id FROM tfg.grados WHERE codigo ='GII');
```

Código 2: *Query* para obtener los códigos de los TF de un Grado

En este caso, obtendría todos los códigos de los TFG del Grado en Ingeniería Informática que tienen estado *Asignado*. El grado “GII” o “GITST” se pasa como parámetro al método que contiene esta *query*. Este resultado queda almacenado en un *array*.

A continuación se recorre este *array* de códigos para extraer las tecnologías de esos TFG mediante el acceso a la tabla *TFG_Tecnologías*. La *query* es la siguiente:

```
SELECT id_tecnologia FROM tfg.tfg_tecnologias WHERE codigo_tfg = "1415_007_SI";
```

Código 3: *Query* para obtener los ids de las tecnologías de un TFG

El *codigo_tfg* se pasa como parámetro al método que contiene la consulta. El resultado se guarda en un *array* de *integer*.

Por tanto, ya se tienen los datos necesarios para formar el *array* de *TFG_Aux*, pasándole al constructor el código extraído anteriormente y el *array* de ids de tecnologías extraído ahora.

A continuación, se ordena este *array* haciendo uso del método *compareTo* desarrollado anteriormente, haciendo la llamada: *Collections.sort(array_TFG_Aux)*.

5.1.2. Algoritmo desarrollado

Tras tener todos los datos preparados, se comienzan a generar las Comisiones de Evaluación. Como el *array* ya está ordenado, el primer TFG de la primera comisión va a ser el primer elemento del *array*. Entonces, el primer caso es el siguiente:

```
for (i=0 ; i<array_TFG_Aux.size(); i++){
    if (array_comisiones.isEmpty()){ //primer caso
        array_TFG_Aux.get(i).setNumComision(0);
        contCom[0]++;
        array_comisiones.add(array_TFG_Aux.get(i));
        comisionCreada = true;
    }else{
```

Código 4: Primer caso generación comisiones - TFG

Como se explicó anteriormente, la clase *TFG_Aux* tiene un atributo para guardar el número de comisión que se le asigna. Por tanto, para este primer caso se le asocia la comisión número 0. Este número es sólo a nivel interno, debido a que las CCEE deben empezar en 01.

La variable *contCom* es un *array* de enteros que sirve para ir guardando el número de TFG que hay en cada comisión, quedando en la posición X el contador de la comisión número X.

Array_comisiones es de tipo *TFG_AUX*, y aquí se irán guardando los TFG que ya se les ha asignado comisión.

La variable *comisionCreada* es un *flag* que cambia su valor a *true* cuando un TFG se asigna a una comisión, para así poder salir del bucle actual sin sobrepasar el límite máximo de número de TFG por comisión.

El siguiente caso a comprobar es cuando el siguiente TFG se quiere asignar a una comisión que ya existe (con esto se refiere a que se va a insertar en una comisión donde coinciden el tipo de tecnologías) y que a su vez no está llena. Esto se implementa así:

```
}else{
    for (int tec : array_TFG_Aux.get(i).getId_tecnologias()){
        for (j=0; j<array_comisiones.size(); j++){
            comisionCreada = false;
            int numC = array_comisiones.get(j).getNumComision();
            for (int tecCom : array_comisiones.get(j).getId_tecnologias()){
                for (k=0; k<contCom.length ; k++){
                    if (tec == tecCom && !comisionCreada){
                        //si ya existe la comision y no esta llena
                        if (k==numC && contCom[numC]<numMaxTFGs){
                            array_TFG_Aux.get(i).setNumComision(numC);
                            contCom[k]++;
                            array_comisiones.add(array_TFG_Aux.get(i));
                            comisionCreada = true;
                            break;
                        }
                    }
                }
            }
            if (comisionCreada)
                break;
        }
        if (comisionCreada)
            break;
    }
    if (comisionCreada)
        break;
}
```

Código 5: Segundo caso generación comisiones - TFG

En rasgos generales, lo que hace aquí el algoritmo es comparar el siguiente TFG de la lista (tecnología a tecnología) con las comisiones ya creadas, para encontrar similitudes en sus respectivas tecnologías.

Lo más importante a resaltar de este fragmento de código son las partes enmarcadas en azul y verde. Primero extraemos en la variable *numC* el número de comisión del TFG con el que se está comparando actualmente.

En el recuadro azul se comprueba que haya coincidencia con alguna de las tecnologías y que el TFG no esté asignado todavía a ninguna comisión (esto podría haber ocurrido en el primer paso). El recuadro verde comprueba que la comisión ya existe y que no está llena. Esto significa que no necesita crear una nueva comisión, por lo que asigna el TFG actual a esta comisión y sale del primer bucle con la ayuda de *comisionCreada*.

El siguiente paso del algoritmo ocurre cuando no se ha cumplido el paso anterior. Es decir, si el TFG no ha encontrado una comisión con similitudes en sus tecnologías o, que si la ha encontrado pero estaba llena. En esta parte se comprueba si estaba llena y, en caso afirmativo, crear una nueva:

```

if (!comisionCreada){ //si no existia o estaba llena
    for (int tec : array_TFG_Aux.get(i).getId_tecnologias()){
        for (j=0; j<array_comisiones.size(); j++){
            comisionCreada = false;
            int numC = array_comisiones.get(j).getNumComision();
            for (int tecCom : array_comisiones.get(j).getId_tecnologias()){
                for (k=0; k<contCom.length ; k++){
                    if (tec == tecCom){
                        //Si estaba llena...
                        if (contCom[numC]==numMaxTFGs){
                            //... intenta crear una
                            if (contCom[k]==0){
                                array_TFG_Aux.get(i).setNumComision(k);
                                contCom[k]++;
                                array_comisiones.add(array_TFG_Aux.get(i));
                                comisionCreada = true;
                                break;
                            }
                        }
                    }
                }
            }
        }
        if (comisionCreada)
            break;
    }
    if (comisionCreada)
        break;
}
if (comisionCreada)
    break;
}
}

```

Código 6: Tercer caso generación comisiones - TFG

Sigue dentro del primer *else*.

Lo resaltado en el recuadro azul es la parte que comprueba si la comisión a la que se quiere asignar el TFG actual estaba llena. En caso afirmativo, entra en el fragmento marcado en verde. Aquí busca la posibilidad de crear alguna comisión sin superar el número máximo de comisiones que se pueden crear. Si puede la crea, y si no avanza al último paso del algoritmo.

Por tanto, el cuarto y último caso es cuando el TFG actual no ha encontrado ninguna similitud de tecnologías con ninguna comisión y, por tanto, intenta crear una nueva (aún dentro del primer *else*):

```
if (!comisionCreada){//Si no existia
    for (j=0; j<array_comisiones.size(); j++){
        comisionCreada = false;
        for (k=0; k<contCom.length ; k++){
            //Si no existia intenta crear una
            if (contCom[k]==0){
                array_TFG_Aux.get(i).setNumComision(k);
                contCom[k]++;
                array_comisiones.add(array_TFG_Aux.get(i));
                comisionCreada = true;
                break;
            }
        }
        if (comisionCreada)
            break;
    }
}
```

Código 7: Cuarto caso generación comisiones - TFG

En este paso el recuadro verde hace lo mismo que en el anterior, pero aquí no ha habido una previa comprobación de similitud de tecnologías ya que, llegado a este paso del algoritmo, implica que no ha habido ninguna y se debe crear una comisión nueva obligatoriamente.

En caso de que un TFG no se asigne a una comisión, este se quedará en el *array* original, *array_TFG_Aux*, mientras que los demás quedarán en el *array* denominado *array_comisiones*. Por tanto, todos los que se hayan asignado a una comisión se borran del primero. De esta forma, los del primer *array* aparecerán en la parte izquierda de la pantalla mostrada en la figura 18, donde se deberán asignar manualmente a una comisión.

5.2. Fase de asignación de profesores como miembros de comisión

Tras finalizar la fase anterior, y una vez que todos los TFG estén asignados a una comisión, ya sea manual o automáticamente, comienza la siguiente fase. Esta fase, desarrollada en la clase *GenerarComisiones_Docentes*, trata de asignar docentes a las comisiones ya existentes, prestando atención a la similitud de tecnologías de los profesores con los TFG de las comisiones.

5.2.1 Preparación de datos

Para el desarrollo de esta fase, se necesita el uso de dos nuevas clases auxiliares, llamadas *Docentes_Aux* y *Comisiones_Aux*. Esto es debido a que, como en el caso de la primera fase, las entidades *Docente* y *Comisión* no disponen de los datos suficientes para asignar los docentes a las comisiones ya creadas.

La primera, *Docentes_Aux* tiene los atributos *id*, *nombre*, *apellido1*, *apellido2*, *categoría* y un *array* de las tecnologías asociadas al docente. También tiene el número de comisiones totales a las que el docente ya ha asistido. Esta clase implementa la interfaz *Comparable*, quedando el método *compareTo* de esta forma:

```
/**
 * Compara dos objetos Docentes_Aux
 *
 * @param other el objeto Docentes_Aux a comparar
 */
@Override
public int compareTo(Docentes_Aux other) {
    int numCom1 = this.numComisionesTotales;
    int numCom2 = other.getNumComisionesTotales();

    if (numCom1 < numCom2){
        return -1;
    } else if (numCom1 >= numCom2){
        return 1;
    }

    return -1;
}
```

Código 8: Método *compareTo* de la clase *Docentes_Aux*

simplemente compara dos objetos *Docentes_Aux* atendiendo al número de veces que el docente participó en alguna comisión, dejando en menor posición el número menor.

La clase auxiliar *comisiones_Aux* tiene como atributos el nombre de la comisión, un *array* de los docentes que forman el tribunal y un *array* con los TFG que forman la comisión. De este modo, en esta clase quedarán guardadas las comisiones completas.

En primer lugar, hay que acceder a la base de datos para extraer los atributos pertenecientes a la clase *Docentes_Aux*. En la clase *DBManager* se accede a la tabla *Docentes*, cuya consulta SQL es la siguiente:

```
SELECT id, nombre, apellido1, apellido2, categoria FROM tfg.docentes
WHERE (categoria='CU' OR categoria='TU' OR categoria='CDR' OR categoria='AYD')
AND (departamento=1 AND activo=true);
```

Código 9: Query para obtener los docentes activos de un departamento

Las comisiones tienen que estar formadas por los docentes con las categorías Catedrático de Universidad, Profesor Titular de Universidad, Profesor Contratado Doctor y Profesor Ayudante Doctor. Por tanto, la *query* busca los docentes con estas categorías. Además, hace la búsqueda por departamento (pasado como parámetro al método que contiene la *query*) y sólo extrae los que estén activos.

Para poder completar los atributos de la clase *Docentes_Aux*, aún nos falta el número de veces que ha participado el docente en alguna comisión y las tecnologías de que dispone. Para extraer el número de participaciones en comisiones, se accede a la tabla *Comisiones*:

```
SELECT COUNT(*) AS totalNumCom FROM tfg.comisiones
WHERE presidente=1 OR vocal1=1 OR vocal2=1;
```

Código 10: Query para obtener las participaciones de un docente en comisiones

No se tendrán en cuenta los que hayan aparecido como suplentes. El id del docente, en este caso un 1, se pasa como parámetro al método que contiene esta sentencia SQL.

Y ahora, para tener todos los atributos de la clase *Docentes_Aux*, tan sólo falta asociarle las tecnologías a cada docente, accediendo a la tabla *Docente_tecnologías*:

```
SELECT id_tecnologia FROM tfg.docente_tecnologias WHERE id_docente=1;
```

Código 11: Query para obtener las tecnologías de los docentes

donde el *id_docente* se pasa como parámetro al método que lanza la consulta SQL.

Una vez que se tienen todos los atributos de la clase *Docentes_Aux*, se ejecuta la llamada *Collections.sort(array_docentes_Aux)* para ordenar los docentes por número de apariciones en las comisiones de evaluación.

A continuación se introduce el *array* de los *TFG_AUX* en el *array* creado de *Comisiones_Aux*, llamado *array_comisionesCompletas*. De esta forma, ya tenemos en cada comisión creada en la fase anterior los TFG que se le asignaron.

5.2.2. Algoritmo desarrollado

Ahora sólo falta asignar los docentes a las comisiones previamente creadas. Como en el apartado anterior, se va a describir el algoritmo por pasos.

```
//Se introducen en el array en orden:
// Presidente, Vocal1, Vocal2, Suplente1, Suplente2
ArrayList<Docentes_Aux> docentesConComision;
ArrayList<Integer> tecnologiasComision;

for (Comisiones_Aux ca : array_comisionesCompletas) {
    tecnologiasComision = new ArrayList();
    docentesConComision = new ArrayList(5);
    //Extraer tecnologias de la comision
    for (int i=0; i<ca.getTfgs().size(); i++){
        for (int tec : ca.getTfgs().get(i).getId_tecnologias()) {
            if (!tecnologiasComision.contains(tec)) {
                tecnologiasComision.add(tec);
            }
        }
        Collections.sort(tecnologiasComision);
    }
}
```

Código 12: Primer paso generación comisiones - Docentes

Este primer paso consiste en formar un *array* que contenga el total de tecnologías que tiene una comisión. Así que coge los TFG uno a uno para ir metiendo las tecnologías, sin repetirse, en *tecnologiasComision*. Se ordena para, más adelante en el algoritmo, hacer más efectiva la comparación de tecnologías con los docentes. Aquí también se crea el *array docentesConComision*, donde se irán guardando los docentes que se asignen a cada comisión.

El siguiente paso, que sigue aún dentro del bucle *for* principal, es éste:

```
//Buscar al presidente, que debe ser CU o TU a ser posible
int i;
boolean docenteAsignado = false;

for (i=0 ; i<array_docentes_Aux.size(); i++){
    Docentes Aux d = array_docentes_Aux.get(i);
    if (d.getCategoria().equals("CU") || d.getCategoria().equals("TU")){
        for (int tec : tecnologiasComision){
            for (int tecD : d.getTecnologias()){
                if (tec == tecD){
                    //El presidente se coloca en la primera posicion del array
                    docentesConComision.add(d);
                    docenteAsignado = true;
                    break;
                }
            }
            if (docenteAsignado)
                break;
        }
    }
    if (docenteAsignado)
        break;
}
if (docenteAsignado)
    //Los array_docentes_Aux que ya esten en una comision se eliminan del array
    array_docentes_Aux.remove(i);
```

Código 13: Segundo paso generación comisiones - Docentes

El primer docente que se asigna a la comisión es el presidente, ya que este debe tener la categoría de Catedrático de Universidad o Profesor Titular Universidad. Esto se comprueba en el recuadro azul. En la parte remarcada en verde, es donde se comprueba la similitud de tecnologías entre la comisión (*tec*) y los docentes (*tecD*). Una vez encontrado el docente con las características requeridas, este se añade al *array docentesConComision*. Al final, si se ha encontrado un docente que asignar, se elimina de *array_docentes_aux* para que no se vuelva a insertar el mismo docente.

El siguiente, y último, paso del algoritmo es similar al anterior (aún sigue dentro del bucle *for* principal):

```

//Buscar a los vocales y suplentes
for (int numD=1; numD<5; numD++){
    docenteAsignado = false;
    for (i=0 ; i<array_docentes_Aux.size(); i++){
        Docentes_Aux d = array_docentes_Aux.get(i);
        for (int tec : tecnologiasComision){
            for (int tecD : d.getTecnologias()){
                if (tec == tecD){
                    //Se colocan en orden: presidente, vocales y suplentes
                    docentesConComision.add(numD,d);
                    docenteAsignado = true;
                    break;
                }
            }
            if (docenteAsignado)
                break;
        }
        if (docenteAsignado)
            break;
    }
    if (docenteAsignado)
        //Los docentes_Aux que ya esten en una comision se eliminan del array
        array_docentes_Aux.remove(i);
}

ca.setDocentes(docentesConComision);
}

```

Código 14: Tercer paso generación comisiones - Docentes

Esta es la parte que asigna el resto de docentes a la comisión: los vocales y los suplentes. En el recuadro azul se aprecia cómo *numD* es la variable que lleva la cuenta de los cuatro docentes restantes. De esta forma, una vez que se hayan encontrado similitudes en las tecnologías, se introduce el docente en el *array docentesConComision*. Para después saber qué puesto corresponde a cada docente, se introducen en orden, con lo que primero estará el presidente, después los dos vocales y por último los dos docentes. Esto se puede apreciar en el recuadro marcado en verde.

Por último, en el recuadro amarillo se asignan los cinco docentes a la comisión actual, ya que se está recorriendo el *array_comisionesCompletas*. Una vez hecho esto, vuelve al principio del bucle (la llave que aparece dentro de este recuadro es la que cierra el bucle *for* principal), crea de nuevo los *arrays docentesConComision* y *tecnologiasComision*, y vuelve a empezar desde el principio, hasta que se hayan recorrido todas las comisiones.

De esta forma, las Comisiones de Evaluación quedan completas: formada por unos TFG con unas tecnologías similares, y asignados unos docentes acorde también a esas tecnologías.

6. Conclusiones y trabajo futuro

En el trabajo realizado se ha conseguido alcanzar el objetivo que se perseguía: desarrollar una herramienta software capaz de automáticamente resolver o facilitar la resolución manual de tareas que a día de hoy hay en la EPS a la hora de gestionar todo lo relacionado con los TFG, incluyendo la automatización del proceso de generación de las comisiones de evaluación.

La herramienta desarrollada va a ser utilizada de inmediato por los coordinadores de los TFG de los dos departamentos de la EPS, el de Ingeniería Informática y el de Tecnología Electrónica y de las Comunicaciones.

Las pruebas realizadas han sido de validación, a medida que se iban implementando cada uno de los módulos. El proceso de evaluación ha sido manual, comprobando que la herramienta mostraba los resultados esperados y corrigiendo los errores que pudiera haber, componente a componente, para finalmente obtener resultados de manera global, con la aplicación ya completa.

Queda como posible trabajo futuro la captura de datos por parte de la herramienta, ya que ahora se hace de forma manual o a través de formularios web. Sería de gran utilidad integrar esto en la herramienta para así completar aún más su funcionalidad.

Por último, se podría hacer una aplicación web con el fin de que todos los docentes pudieran tener acceso a ella para que, directamente sobre la herramienta, introdujeran las propuestas de los TFG y así evitar parte de la inserción de datos en la herramienta por parte del coordinador. Además, se podría implementar también una aplicación móvil, pudiendo acceder así desde casi cualquier terminal a la herramienta desarrollada.

Referencias

1. Guía docente de la asignatura TFG de la EPS:
<http://www.uam.es/ss/Satellite/EscuelaPolitecnica/es/estudios/grado-3/grado-en-ingenieria-informatica-3/guiadocente/detalle/curso-4-2014-2015.htm>
2. Comisiones de Evaluación del Grado en Ingeniería Informática:
<http://ir.ii.uam.es/~fdiez/TFG/tribunales/CCEE.html>
3. Directorio de la Escuela Politécnica Superior, Búsqueda general:
<http://www.uam.es/ss/Satellite/EscuelaPolitecnica/es/la-escuela-2/Page/sinContenido/directorio.htm>
4. Página web de MySQL: <http://www.mysql.com>
5. The Java Database Connectivity:
<http://www.oracle.com/technetwork/java/javase/jdbc/index.html>
6. Página web oficial de Java: <https://www.java.com/es/>
7. Oracle – Java: <http://www.oracle.com/technetwork/java/index.html>
8. Página Moodle de la UAM referente a los Trabajos de Fin de Grado:
<https://moodle.uam.es/course/view.php?id=9104>