



U N I V E R S I D A D
COMPLUTENSE
M A D R I D

Facultad de Informática

Aplicación web de soporte al
Aprendizaje-Servicio: gestión de ofertas y
demandas de servicio

Web application for supporting Service
Learning: management of service offers
and service requests

Autores

Daniela-Nicoleta Boldureanu (Grado en Ingeniería del Software)
Victoria Gnatiuk Romaniuk (Grado en Ingeniería Informática)
Jesús Sánchez Granado (Grado en Ingeniería Informática)

Directores

Simon Pickin
Manuel Montenegro Montes

Curso 2020/2021

Índice general

1. Introducción	7
1.1. Antecedentes	7
1.2. Objetivos	8
1.3. Plan de trabajo	8
2. Introduction	11
2.1. Background	11
2.2. Objectives	12
2.3. Workplan	12
3. Contexto de la propuesta	15
3.1. Introducción	15
3.2. Elementos que intervienen en un proyecto ApS	15
3.3. Motivación	16
3.4. TFG de partida	17
3.5. Estudio tecnológico	18
4. Modelos de dominio, de datos y de relación	21
4.1. Introducción	21
4.2. Modelo de dominio	22
4.3. Modelo de datos	26
4.4. Modelo relacional	26
4.4.1. Separación de datos	28
4.4.2. Usuarios	28
4.4.3. Anuncios de servicio	30
4.4.4. Colaboración	32
4.4.5. Comunicación	34
5. Contribución	37
5.1. Daniela-Nicoleta Boldureanu	37
5.2. Victoria Gnatiuk Romaniuk	37
5.3. Jesús Sánchez Granado	38

Índice de figuras

2.1. PivotalTracker: tareas	13
4.1. Modelo de dominio	24
4.2. Modelo de datos	25
4.3. Diagrama de entidad-relación	27
4.4. Diagrama de entidad-relación - Usuarios	29
4.5. Diagrama de entidad-relación - Anuncios de servicio	31
4.6. Diagrama de entidad-relación - Colaboración	33
4.7. Diagrama de entidad-relación - Comunicación	35

Capítulo 1

Introducción

El objetivo de este *Trabajo de Fin de Grado* (TFG) es retomar la idea de creación de una comunidad web de proyectos ApS (*Aprendizaje-Servicio*) empezada a nuestro saber, en el año 2004 por Emmanuel Parmentier en su (*Proyecto Fin de Carrera*) PFC. Otros trabajos después siguieron esta trayectoria y el último de ellos fue el TFG de David Jiménez del Rey, este es el proyecto que nosotros continuamos en nuestro TFG. Estos proyectos precedentes se detallan más adelante en el capítulo 3.3. El proyecto objetivo en cuestión consiste en la creación de una plataforma web que permita crear un entorno digital en el que las empresas, principalmente del sector público, y las universidades desarrollen actividades de labor social que ayuden a los alumnos a desarrollar de forma práctica lo aprendido en el aula y a moldearlos como ciudadanos éticos y solidarios.

1.1. Antecedentes

Este trabajo parte del TFG de David Jiménez del Rey, que se desarrolló en la *Universidad Nacional de Educación a Distancia* (UNED) bajo la tutela de Ángeles Manjarrés Riesco y Simon Pickin. Todos los requisitos del TFG nos han sido dados por nuestros directores que actuaban como clientes del proyecto. Esto implica que el funcionamiento de todos los elementos que intervienen en el TFG han sido definidos por nuestros directores junto a colaboradores entendidos en la materia. El proyecto ha sido desarrollado con tecnologías como Node.js, Angular, Express y MySQL.

La plataforma web desarrollada en este proyecto tiene como objetivo ayudar a crear, gestionar y evaluar proyectos ApS. Un proyecto ApS es una práctica académica en la que el alumno aplica las habilidades teóricas aprendidas en las clases en el mundo real, ayudando a su comunidad con todo tipo de tareas. La actividad es planteada y gestionada por uno o varios profesores y un socio comunitario, que es una empresa que está interesada en desarrollar estos proyectos. Al acabar la actividad, el alumno es evaluado por los profesores y es motivado a reflexionar sobre los servicios prestados, con el objetivo de fortalecer la solidaridad y la ética del alumno.

El principal problema de los ApS es acordar los proyectos entre el socio comunitario y los profesores, ya que cada uno tiene una idea muy diferente del proyecto. Aunque el socio comunitario y el profesor quisieran desarrollar el mismo proyecto, lo plantean de formas muy diferentes y es por esto por lo que es difícil emparejarlos. Esta fue la principal motivación de los profesores que empezaron el planteamiento de este proyecto en el 2004. En el capítulo 3 se explica con más detalle que es un proyecto ApS.

1.2. Objetivos

Partiendo del trabajo anterior, nuestros objetivos principales en este TFG fueron continuar el proyecto remodelando la base de datos, rediseñando la aplicación y creando un sistema de emparejamiento de los proyectos planteados por un profesor y los planteados por un socio comunitario. A continuación, se listan los objetivos de este TFG.

- Construir unas bases sólidas del proyecto, creando un modelo de dominio que aclare los conceptos implicados en la aplicación y un modelo de datos que enriquezca el modelo de dominio y plasme cómo la aplicación gestiona la información.
- Crear un modelo relacional que muestre la estructura de la base de datos, facilitando su entendimiento y manejo a los futuros desarrolladores del proyecto.
- Crear una base de datos relacional compleja y rica en detalles.
- Implementar cuatro DAOs (*Objetos de Acceso a Datos*) que realicen la lógica de acceso y gestión de datos, encapsulando el acceso a la base de datos. Crear *transfers* que permiten estructurar y manejar de forma sencilla los datos de la BD.
- Implementar un sistema de *matching* de los proyectos planteados por un profesor y los planteados por un socio comunitario que determina qué porcentaje de encaje tienen.
- Adaptar las páginas de registro y de perfil del usuario al nuevo sistema, e implementar formularios para la creación de ofertas, demandas y partenariados.
- Corregir *bugs* encontrados en el proyecto precedente.

1.3. Plan de trabajo

Establecidos los objetivos anteriores por nuestros directores, lo primero que hicimos fue encontrar una herramienta de gestión de proyectos que nos permitiera organizar el trabajo. Esta herramienta es Pivotal Tracker, ver Figura 2.1. Esta herramienta centrada en la gestión de proyectos de tipo SCRUM nos ha ayudado a crear las tareas, asignarles dificultad, clasificarlas y llevar un control general del trabajo realizado y por realizar. El trabajo realizado en este TFG se puede dividir principalmente en seis grandes fases.

- En la primera fase hemos leído la memoria del TFG de David Jiménez para comprender las raíces del problema a resolver y conocer los detalles del proyecto en el que íbamos a trabajar. En paralelo hemos estado investigando por nuestra cuenta sobre los proyectos ApS y las tecnologías en las que estaba implementado el proyecto, sobre todo Node.js que no conocíamos antes de empezar el TFG. Esta fase ha comprendido desde el día 30 de septiembre hasta el día 6 de noviembre.
- En la segunda fase, hemos explorado el código del anterior TFG para familiarizarnos con él y después hemos realizado pruebas manuales de la solución. Al realizar estas pruebas hemos descubierto algunos *bugs* que posteriormente hemos corregido. Esta fase ha comprendido desde el día 7 de noviembre hasta el día 19 de noviembre.
- La tercera fase ha consistido en el desarrollo de un modelo de dominio y un modelo de datos que ilustran la solución del problema de una forma más detallada y concisa. Por otro lado, hemos estado diseñando la nueva base de datos relacional teniendo en

cuenta la nueva estructura de la aplicación. Esta fase se describe con más detalle en el capítulo 4. Esta fase ha comprendido entre el día 20 de noviembre y el día 26 de febrero.

- La cuarta fase ha comprendido entre el día 27 de febrero y el día 10 de marzo. En esta fase hemos implementado los cuatro DAOs, los **transfers** y hemos adoptado los controladores al nuevo sistema. Esta fase se describe con más detalle en el capítulo ??.
- La quinta fase ha consistido en la creación del sistema de **matching**. Para conocer más detalles sobre esta fase lea el capítulo ??. Esta fase ha comprendido entre el día 11 de marzo y el día 23 de abril.
- La sexta fase ha consistido en el aprendizaje de Angular, tecnología que desconocíamos antes de empezar el TFG y en la creación y adaptación de parte de la interfaz. En concreto la parte de la interfaz afectada se menciona en la sección de objetivos y se explica con más detalle en el capítulo ??. Esta fase ha comprendido entre el día 24 de abril y el día 20 de mayo.

Capítulo 2

Introduction

The objective of this (*Final Degree Project*) FDP is to take up the idea of creating a web community of (*Service-Learning*) SL projects started to our knowledge, in 2004 by Emmanuel Parmentier in his FDP. Other works later followed this path and the last of them was David Jiménez del Rey's FDP, this is the project that we continued in our FDP. These preceding projects are detailed later in the chapter 3.3. The objective of the project consists in the creation of a web platform that allows creating a digital environment in which companies, mainly from the public sector, and universities develop social work activities that help students to develop in a practical way what they have learned in the classroom and to mold them as ethical and caring citizens.

2.1. Background

This work is based on David Jiménez del Rey's FDP, which was developed at the (*National Distance Education University*) NDEU under the tutelage of Ángeles Manjarrés Riesco and Simon Pickin. All the requirements of the FDP have been given to us by our directors who acted as clients of the project. This implies that the functionality of all the elements that intervene in the FDP have been defined by our directors together with collaborators who are knowledgeable in the matter. The project has been developed with technologies such as Node.js, Angular, Express and MySQL.

The web platform developed in this project aims to help create, manage and evaluate SL projects. An SL project is an academic methodology in which the student applies the theoretical skills learned in the classes in the real world, helping their community with all kinds of tasks. The activity is planned and managed by one or more teachers and a community partner, which is a company that is interested in developing these projects. At the end of the activity, the student is evaluated by the teachers and is motivated to reflect on the services provided, with the aim of strengthening the solidarity and ethics of the student.

The main problem of SLs is to agree the projects between the community partner and the teachers since each has a very different idea of the project. Although the community partner and the teacher would like to develop the same project, they pose it in very different ways and that is why it is difficult to pair them. This was the main motivation of the teachers who started planning this project in 2004. Chapter

3 explains in more detail what a SL project is.

2.2. Objectives

Based on the previous work, our main objectives in this FDP were to continue the project by remodeling the database, redesigning the application and creating a matching system of the projects proposed by a teacher and those proposed by a community partner. The objectives of this FDP are listed below.

- Build a solid foundation for the project, creating a domain model that clarifies the concepts involved in the application and a data model that enriches the domain model and reflects how the application manages information.
- Create a relational model that shows the structure of the database, facilitating its understanding and management by future project developers.
- Create a complex and detailed relational database.
- Implement four DAOs (*Data Access Objects*) that perform the logic of access and data management, encapsulating the access to the database. Create transfers that allow the database data to be structured and managed in a simple way.
- Implement a system of matching of the projects proposed by a teacher and those proposed by a community partner that determines what percentage of fit they have.
- Adapt the registration and user profile pages to the new system, and implement forms for the creation of offers, demands and partnerships.
- Correct bugs found in the previous project.

2.3. Workplan

Having established the above objectives by our directors, the first thing we did was find a project management tool that would allow us to organize the work. This tool is Pivotal Tracker, see Figure 2.1. This tool focused on managing SCRUM-type projects has helped us to create tasks, assign them difficulty, classify them and keep a general control of the work done and to be done. The work carried out in this FDP can be divided mainly into six main phases.

- In the first phase we have read David Jiménez's FDP report to understand the roots of the problem to be solved and to know the details of the project we were going to work on. In parallel, we have been investigating on our own about the SL projects and the technologies in which the project was implemented, especially Node.js that we did not know before starting the FDP. This phase has ran from September 30th to November 6th.
- In the second phase, we have explored the code of the previous FDP to become familiar with it and then we have carried out manual tests of the solution. During these tests we have discovered some bugs that we have subsequently corrected. This phase has ran from November 7th to November 19th.

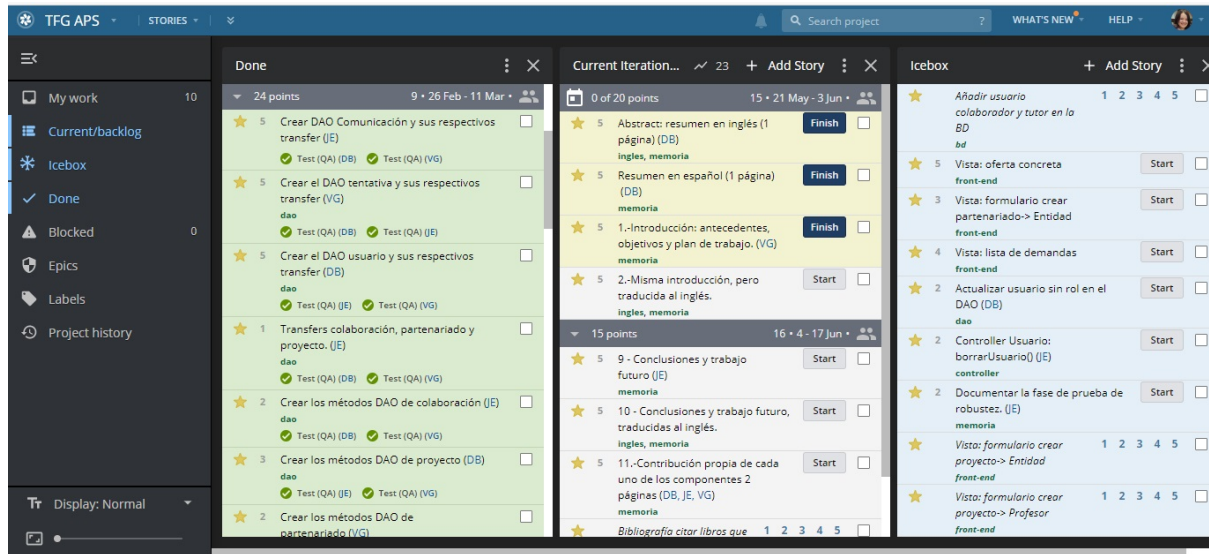


Figura 2.1: PivotalTracker: tareas

- The third phase consisted of developing a domain model and a data model that illustrate the solution of the problem in a more detailed and concise way. On the other hand, we have been designing the new relational database considering the new structure of the application. This phase is described in more detail in chapter 4. This phase has been between November 20th and February 26th.
- The fourth phase ran from February 27th to March 10th. In this phase we have implemented the four DAOs, the transfers and we have adopted the controllers to the new system. This phase is described in more detail in chapter ??.
- The fifth phase consisted of creating the matching system. For more details on this phase, read the chapter ??. This phase has been between March 11th and April 23th.
- The sixth phase consisted of learning Angular, a technology that we were unaware of before starting the FDP, and creating and adapting part of the interface. Specifically, the part of the interface affected is mentioned in the objectives section and explained in more detail in the chapter ??. This phase has been between April 24th and May 20th.

Capítulo 3

Contexto de la propuesta

3.1. Introducción

El ApS es una propuesta educativa que combina aprendizaje y servicios a la comunidad. Los proyectos ApS permiten a los alumnos aprender de una forma más práctica, aplicando sus conocimientos adquiridos en clase mediante la realización de tareas útiles para la comunidad.

Además de dar a los estudiantes la oportunidad de aplicar sus conocimientos en un entorno real, el ApS les impulsa a comprender el funcionamiento de la sociedad y las responsabilidades sociales que estos tienen por formar parte de una sociedad.

Todo proyecto ApS empieza por una iniciativa relacionada con una necesidad social real que implica la ejecución de un servicio para solventarla y tiene como objetivo el aprendizaje y la reflexión del alumno. Para saber más sobre el tema dirijase a la referencia [5].

3.2. Elementos que intervienen en un proyecto ApS

En un proyecto ApS intervienen los siguientes elementos:

- El *alumno* es el individuo que aplica sus conocimientos teóricos en un entorno físico beneficiando a su comunidad. Además de adquirir habilidades prácticas relacionadas con su formación, es importante que se incite al alumno a reflexionar sobre sus actos y el impacto positivo que tienen estos sobre los demás. Esto permite al alumno adquirir compromiso social y desarrollar pensamiento ético, cultivando un ciudadano responsable capaz de mejorar la sociedad de la que forma parte.
- El *socio comunitario* es una empresa pública o de tercer sector que colabora con la institución educativa para resolver un determinado problema social. El socio comunitario suele tener en mente un problema muy concreto, pero no lo suficientemente detallado para la creación de un proyecto educativo. Es por eso que es necesario el partenariado. Es importante que la universidad haga entender al socio comunitario que el ApS no es voluntariado. Por tanto, bajo ningún concepto se puede usar al alumno para la generación de beneficios propios de la empresa o la competencia desleal. El principal objetivo del ApS es formar al alumno introduciéndolo en un entorno real para que este establezca una relación entre lo aprendido en el aula con lo realizado en el proyecto ApS.

- El *profesor* es el individuo que se encarga de guiar al alumno en todo el proceso del proyecto, evaluando sus tareas e incitando al alumno a la reflexión. Además de guiar al alumno en su formación y gestionar el proyecto, ofrece su formación y conocimientos al socio comunitario con la que se colaborará en el proyecto. El profesor se encarga de acordar y organizar los proyectos con el socio comunitario, estableciendo todos los requisitos necesarios para la correcta formación del alumno y el cumplimiento del socio comunitario con los principios del ApS.
- El *partenariado* es una colaboración entre un profesor, o un equipo de profesores, y el socio comunitario. Partiendo de un problema social real y los conocimientos dispuestos por el profesor, el profesor y el socio comunitario determinan las características y particularidades del problema. Una vez definidos los términos y condiciones del futuro proyecto, el profesor abre el proyecto a otros profesores que quieran colaborar en las propuestas y posteriormente el proyecto se abre a los alumnos. Todo partenariado tiene un profesor responsable que pertenece a la universidad en la que se ha desplegado la aplicación.
- El *proyecto* consiste en la ejecución de ciertas tareas realizadas por el alumno que están relacionadas con su formación. Estas tareas permiten al alumno establecer una relación entre lo aprendido en clase y el mundo real. Gracias a estas tareas o servicios, el alumno beneficia a su comunidad, otorgándole una satisfacción personal. El alumno es evaluado de forma continua por el equipo docente. Todo proyecto tiene un profesor responsable que pertenece a la universidad en la que se ha desplegado la aplicación.

3.3. Motivación

La motivación descrita a continuación ha sido extraída de nuestros directores los cuales conocen el histórico del proyecto:

La experiencia de los profesores que han llevado a cabo, o intentado llevar a cabo, iniciativas de tipo ApS muestra que muchos proyectos potenciales no llegan a realizarse por la dificultad en casar la oferta con la demanda, es decir, cuadrar las necesidades didácticas, organizativas, etc. de la institución educativa que quiere prestar un servicio, con las necesidades y disponibilidad de la organización o comunidad que quiere recibir un servicio. El establecimiento de relaciones de partenariado en el ApS requiere tiempo y energía, siendo el análisis de compatibilidad entre la oferta y la demanda de servicios un aspecto clave. Los profesores con experiencia en iniciativas de tipo ApS, tanto de carácter local como en el contexto de proyectos de cooperación internacional para el desarrollo, se dieron cuenta hace tiempo de que un buen soporte informático podría ser de gran ayuda en la difícil tarea de casar la oferta y la demanda de ApS. Gracias a este soporte se facilitarían la identificación de potenciales partenariados, así como la colaboración entre el prestador y el receptor potenciales del servicio en la tarea de refinar una idea inicial y convertirla en una propuesta de proyecto realista que cumple las necesidades de las dos partes.

Los programas de ApS, ya consolidados en el continente americano, tan solo recientemente empiezan a cobrar fuerza en las instituciones educativas europeas, y en particular en las españolas. Los primeros proyectos de ApS se iniciaron en nuestro país apenas hace 15 años. El escaso valor académico concedido hasta ahora a la práctica del ApS fuera del ámbito de los estudios pedagógicos, ha sido motivo de que los intentos de desarrollar la

aplicación informática descrita hayan surgido siempre en el contexto de PFC/TFG. El desarrollo de una tal aplicación requiere una alta dedicación y no tiene valor de investigación tecnológica, de modo que los profesores técnicamente cualificados no tienen disponibilidad para abordarlo.

El PFC de Ingeniería Industrial de la *Universidad Politécnica de Madrid* (UPM) [4] fue el primer intento, según nuestro conocimiento, de desarrollar una aplicación para el soporte de la definición de PFC de cooperación (y del previo establecimiento de un partenariado Universidad-ONG) y dio como resultado una aplicación basada en el *sistema de gestión de contenidos* (CMS) Plone. Sin embargo, su naturaleza de prototipo, junto con la falta de recursos humanos para su despliegue y administración, hizo que nunca llegara a utilizarse. Ángeles Manjarrés, profesora de la UNED, y Simon Pickin, uno de los directores del presente trabajo, retomaron la idea de [4] en el TFG del grado en Informática de la UNED [7], en el que se desarrolló una aplicación web basada en el popular *framework de gestión de contenidos* (CMF) Drupal. El prototipo desarrollado fue bastante básico e incompleto y tenía restricciones técnicas que limitaban su funcionalidad, pero constituyó un pequeño avance hacia el objetivo. Aunque el propósito del PFC [4] y del TFG [7] fue el soporte de la definición de PFC/TFG de cooperación, este objetivo puede verse como un caso particular del soporte de proyectos ApS y la plataforma informática necesaria es prácticamente idéntica en los dos casos.

Se retomó la iniciativa de [7] en el contexto de una colaboración entre la UCM y el grupo de investigación en innovación docente COETIC (“Grupo de Innovación Docente de la UNED para el Desarrollo de la Competencia Ética y Cívica y las metodologías basadas en la Comunidad en la educación superior”) con el TFG del grado en Educación Social de la UNED [6], centrado en la especificación de una aplicación para el soporte del ApS virtual, y los TFG del grado en Informática de la UNED [3] y [2], centrados en el desarrollo de tal aplicación, los dos últimos codirigidos por Ángeles Manjarrés y Simon Pickin. En [3], después de un intento fallido de continuar con el desarrollo en Drupal empezado en [6], debido a que Drupal resultó ser mucho más cerrado de lo esperado, se desarrolló una nueva aplicación desde cero basada en el stack MyEAN (*MySQL-Express-Angular-Node*). Esta aplicación se describe brevemente en [1]. En [2], se continuó el desarrollo de la aplicación de [3] pero esta vez con el stack MEAN (*MongoDB-Express-Angular-Node*); la substitución de MySQL por MongoDB no se debía a razones técnicas sino a razones de conveniencia y familiaridad del desarrollador. Aunque los títulos de [3] y [2] hacen referencia al ApS “Virtual”, la aplicación desarrollada está pensada para dar servicio a toda la comunidad de ApS.

3.4. TFG de partida

El TFG de David Jiménez presentaba una base a partir de la cual hemos partido para desarrollar nuestra parte de este proyecto. Partiendo del TFG anterior que estaba desarrollado en Angular y Node.js, David Jiménez siguió desarrollando la aplicación hasta conseguir un prototipo de la futura aplicación.

David Jiménez implementó las páginas de registro, *login*, perfil, iniciativa y partenariado. También incluyó diferentes perfiles como el *admin*, el socio comunitario, el profesor y el alumno.

Por razones de comodidad y familiarización con la tecnología, en el TFG anterior se construyó la base de datos en MongoDB. Debido a que los datos de la aplicación son claramente estructurados y muy relacionados entre sí, no había razones suficientes para seguir desarrollando la aplicación sobre MongoDB, así que junto con nuestros directores decidimos cambiar la base de datos a SQL.

Debido a las dificultades provocadas por la pandemia del COVID-19, las bases de la aplicación no quedaron del todo definidas en el TFG anterior y por esta razón hemos tenido que redefinirlas creando un modelo de dominio y de datos que representa todos los elementos del problema y cómo se relacionan entre sí. Diseñamos una base de datos nueva más compleja y rica en detalles que servirá como cimientos para nuestros sucesores, ya que esperamos que llegue el día en que esta plataforma sirva a usuarios reales, los cuales ayudaran a que la educación sea más eficaz y enriquecedora.

3.5. Estudio tecnológico

A continuación, se explicará qué tecnologías tienen el potencial para desarrollar este proyecto y cuáles hemos elegido finalmente. Las razones de las decisiones tomadas sobre el uso de estas tecnologías se detallan en la sección 4.

1. Servidor web:

- ExpressJS es un *framework* basado en NodeJS que permite gestionar el servidor de una forma sencilla. Este *framework* fue utilizado por David Jiménez en el TFG precursor y se ha mantenido.

2. *Backend*:

- NodeJS es entorno basado en JavaScript muy popular. Este entorno fue usado en el anterior TFG y se ha mantenido.

3. *Frontend*

- Angular es un *framework* utilizado en el *frontend* del TFG anterior y se ha decidido mantener.

4. Base de datos:

- MongoDB es un sistema de base de datos no estructurado. Este sistema es el que se estaba usando en el proyecto.
- MySQL: es un sistema de base de datos relacional. Decidimos utilizar este sistema en nuestro TFG.

5. Software de control de versiones:

- Git es el controlador de versiones más conocido y muy eficaz, así que desde el principio supimos que es el software que íbamos a usar.

6. Repositorio:

- GitHub es un repositorio gratuito que permite almacenar todos los archivos relacionados con un proyecto y mantenerlos de forma colaborativa con otros usuarios. Se ha decidido utilizar GitHub por su integración con Git.

- Google Drive: es un contenedor gratuito que permite almacenar cualquier fichero y compartirlo con los demás. En un principio se estudió utilizar para guardar los *Backups* pero se acabó descartando. Al final se ha utilizado para almacenar todo tipo de documentos relacionados con el TFG, excepto el código fuente de la aplicación.

7. Herramientas de organización:

- GitHub Projects es una herramienta que ofrece GitHub que permite crear una organización de proyecto tipo Kanban.
- Trello es una herramienta sencilla estilo Kanban para organizar los proyectos, pero tiene muchas limitaciones en su versión gratuita.
- PivotalTracker es una herramienta de gestión de proyectos basada en Scrum que permite crear *Stories*, asignarles un peso en función de lo compleja que sea la *Story* y ofrece analíticas que permiten analizar el progreso del proyecto. Hemos decidido utilizar esta herramienta para organizarnos porque es una herramienta completa.

8. Herramientas UML:

- Diagrams.net es una herramienta de diseño de diagramas *online* la cual permite diseñar varios tipos de diagramas entre los cuales se encuentran diagramas de clases, de flujos, de entidades, etc.
- Modelio es una aplicación de escritorio que permite crear modelos UML complejos, indicando los atributos, los métodos y las relaciones que tienen los elementos entre sí. Debido a que es una herramienta completa y que es una herramienta que ya conocíamos, la hemos elegido para la creación de nuestros modelos.

9. Herramientas para el diseño del modelo relacional de la base de datos:

- phpMyAdmin es una herramienta web que permite gestionar una base de datos SQL. Esta herramienta muestra un modelo relacional de la base de datos muy simple.
- MySQL Workbench es una herramienta de gestión de diseño de base de datos visual que permite crear modelos relacionales complejos. Esta es la aplicación que se ha decidido utilizar.

10. Herramientas para la redacción de la memoria:

- Microsoft Word siendo una herramienta popular y muy conocida para la creación de documentos escritos, fue nuestra primera opción para la redacción de la memoria.
- LaTeX es una herramienta que permite crear documentos profesionales con resultados profesionales. Esta es la herramienta que se ha decidido utilizar.

11. Lenguaje para insertar datos en la BD:

- Python se ha utilizado para insertar valores enumerados en la base de datos.

Capítulo 4

Modelos de dominio, de datos y de relación

4.1. Introducción

Al empezar con el proyecto, junto con nuestros directores, nos hemos dado cuenta de que la aplicación necesitaba ser definida de una forma más detallada. En TFGs anteriores se entendía que los profesores y las entidades definían los proyectos de la misma forma, pero esto no es así. Las entidades tienen un problema en mente que quieren resolver, pero no saben cómo adaptarlo a un proyecto estudiantil. Los profesores, por su parte, tienen una idea abstracta del problema que quieren resolver. Normalmente tienen ciertos conocimientos académicos los cuales quieren aplicar para mejorar el mundo, pero la necesidad social en cuestión no suele estar muy clara.

En el TFG precedente se creó un elemento llamado *Iniciativa* que almacenaba algunas de las características generales que comparten las propuestas del profesor y el socio comunitario. La *Iniciativa* derivaba en un formulario que se les ofrecía a las entidades y a los profesores, pero las entidades y los profesores no plantean los problemas de la misma manera. Las entidades tienen una idea de proyecto clara pero no saben cómo orientarlo al mundo académico y los profesores tienen una idea de proyecto abstracta y lo plantean con el objetivo de encajar en un plan académico, es por esto que no es apropiado utilizar el mismo formulario para ambos.

Para definir con precisión el funcionamiento correcto y completo de la aplicación, se ha creado un modelo de dominio y un modelo de datos. Estos modelos nos permitirán entender acertadamente el funcionamiento de la aplicación, pero lo que es más importante, permitirán transmitir dicho funcionamiento e idea general a otras personas que trabajarán en este proyecto después de nosotros.

Gracias al modelo de dominio podemos entender qué elementos intervienen y cómo interactúan entre sí, además de las restricciones que se presentan en sus interacciones.

Con el modelo de datos podemos conocer información detallada de cada elemento que interviene en la resolución del problema, así como los atributos que posee.

Como se ha rediseñado la base de datos se ha decidido crear un modelo relacional que muestra todas las tablas de la nueva base de datos y sus relaciones. De esta manera,

podemos representar sus especificaciones técnicas para comprender su estructura y funcionamiento.

4.2. Modelo de dominio

El modelo de dominio es un mapa conceptual de la aplicación que permite a cualquier individuo entender su funcionamiento. El modelo de dominio de esta aplicación se puede observar en la Figura 4.1.

Si empezamos mirando nuestro modelo desde arriba podemos observar que hay una clase madre, llamada Usuario y de ella se ramifican otras cinco clases que representan los cinco grupos de usuarios que tiene la aplicación.

- Los estudiantes se dividen en internos y externos. Los internos representan a aquellos estudiantes que pertenecen a la universidad donde está instalada la aplicación y los externos pertenecen a otras universidades.
- El grupo de usuarios promotor se divide en externos e internos por el mismo motivo que los estudiantes.
 - El promotor externo representa al profesor externo, que es aquel profesor que no forma parte de la universidad donde está la aplicación, pero puede participar en un proyecto o partenariado evaluando y guiando a un estudiante externo. El colaborador, por otra parte, es un experto en algún tema en concreto que puede participar en un proyecto o partenariado ofreciendo sus conocimientos o habilidades.
 - El promotor interno puede ser un profesor de la universidad en la que se aloja la aplicación y también puede ser un tutor si pertenece a la UNED, donde existe este rol. El profesor interno puede crear las ofertas de servicio y es el responsable del partenariado y del proyecto.
- El socio comunitario es aquella entidad que hace de socio para la creación del proyecto.
- El admin es el usuario administrador de la aplicación.
- La Oficina APS es un usuario responsable de todos los trámites relacionados con la gestión de los proyectos ApS.

Todo proyecto tiene que definirse en el contexto de un partenariado y este partenariado se puede originar de varias maneras.

Una de ellas es a partir de una oferta de servicio y una demanda de servicio. En este caso el partenariado puede surgir porque una de las dos partes decida aliarse con la otra buscándolo manualmente en la plataforma o porque el sistema de *matching* sugiera la unión de ambas partes. Otra forma es mediante la creación de lo que llamamos partenariados “monoparentales”, es decir, partenariados creados cuando un socio comunitario acepta una oferta creada por un profesor sin haber creado previamente una demanda, o cuando un profesor respalda una demanda creada por un socio comunitario sin haber creado previamente una oferta. En estos casos, la correspondiente oferta (resp. demanda) se creará en el momento de aceptar (resp. respaldar) la oferta (resp. la demanda) y se

utilizará un atributo de la oferta y de la demanda para almacenar la naturaleza “mono-parental” del partenariado.

En general una demanda de servicio es creada por un socio comunitario, pero puede llegar a crearse gracias a un estudiante, el cual previamente ha tenido que crear una iniciativa. Una iniciativa es una propuesta de proyecto de un estudiante, la cual es acogida por un socio comunitario despues de haber sido aprobada por la oficina ApS. Cuando el socio comunitario acoge dicha iniciativa, esta se convierte en una demanda de servicio.

Debido a que los proyectos definidos por los profesores y los definidos por las entidades comparten ciertas características, podemos ver en el diagrama que ambos cuelgan de un elemento padre llamado **Anuncio de servicio** que posee las características comunes de ambos. Lo mismo ocurre con **partenariado** y **proyecto** que cuelgan de **Colaboración Uni-Soc**.

Queremos destacar que los atributos de **Necesidad social** y **Area de implementación**, tan importantes para emparejar ofertas y demandas, han sido obtenidos de la página web www.eoslhe.eu/resources/ Los partenariados y los proyectos presentan ciertas restricciones relacionadas con la participación de alumnos y promotores externos.

En los proyectos solo pueden participar estudiantes externos si el responsable del proyecto acepta estudiantes externos. Los profesores externos solo pueden participar en aquellos partenariados y proyectos que los acepten. Además, en el caso de un proyecto, un participante solo puede tener el perfil de profesor externo si también participan alumnos externos de su universidad, cuya participación tendrá que evaluar; en otro caso, tendrá el perfil de colaborador, independientemente del tipo de puesto que ocupa en su universidad.

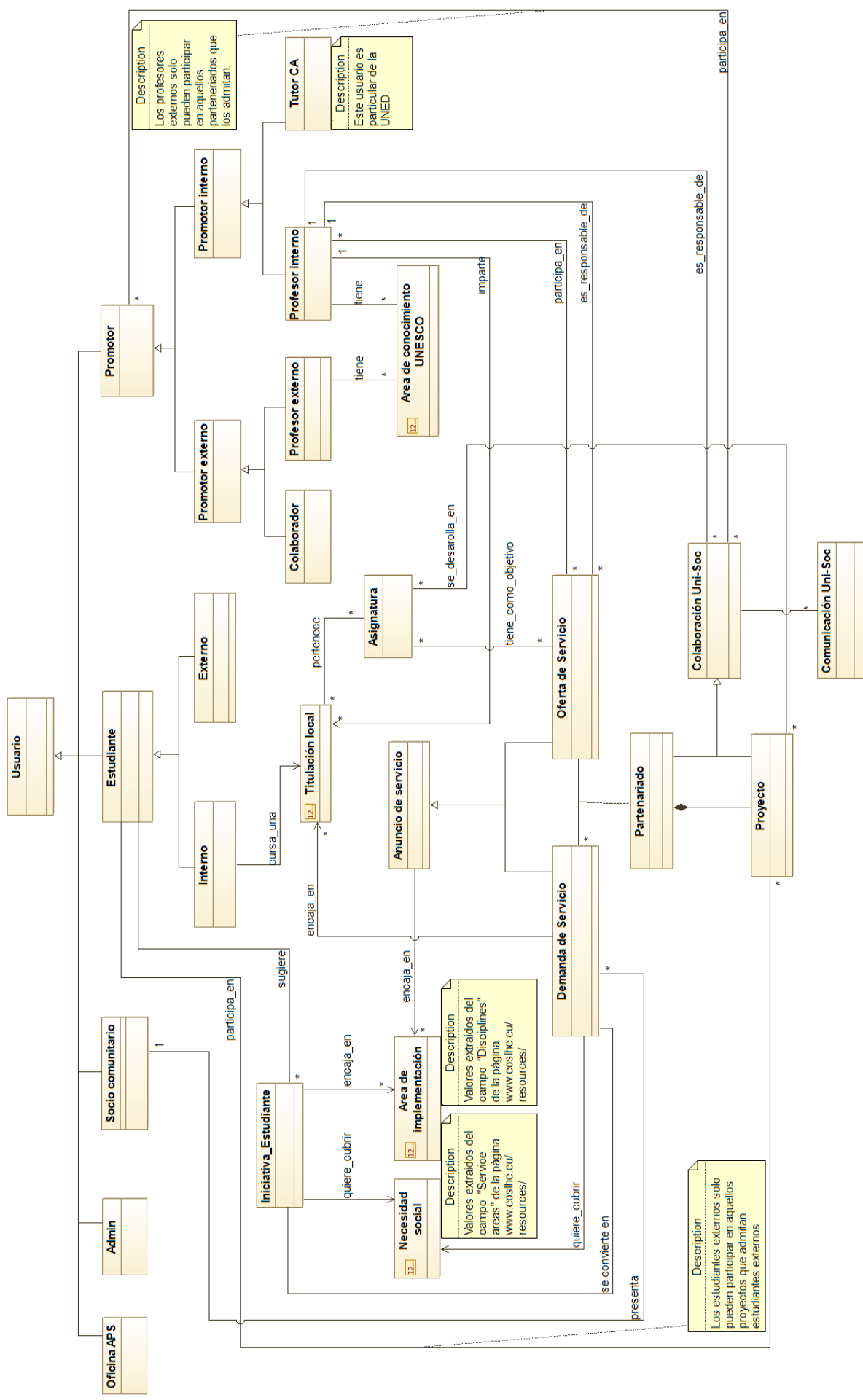


Figura 4.1: Modelo de dominio

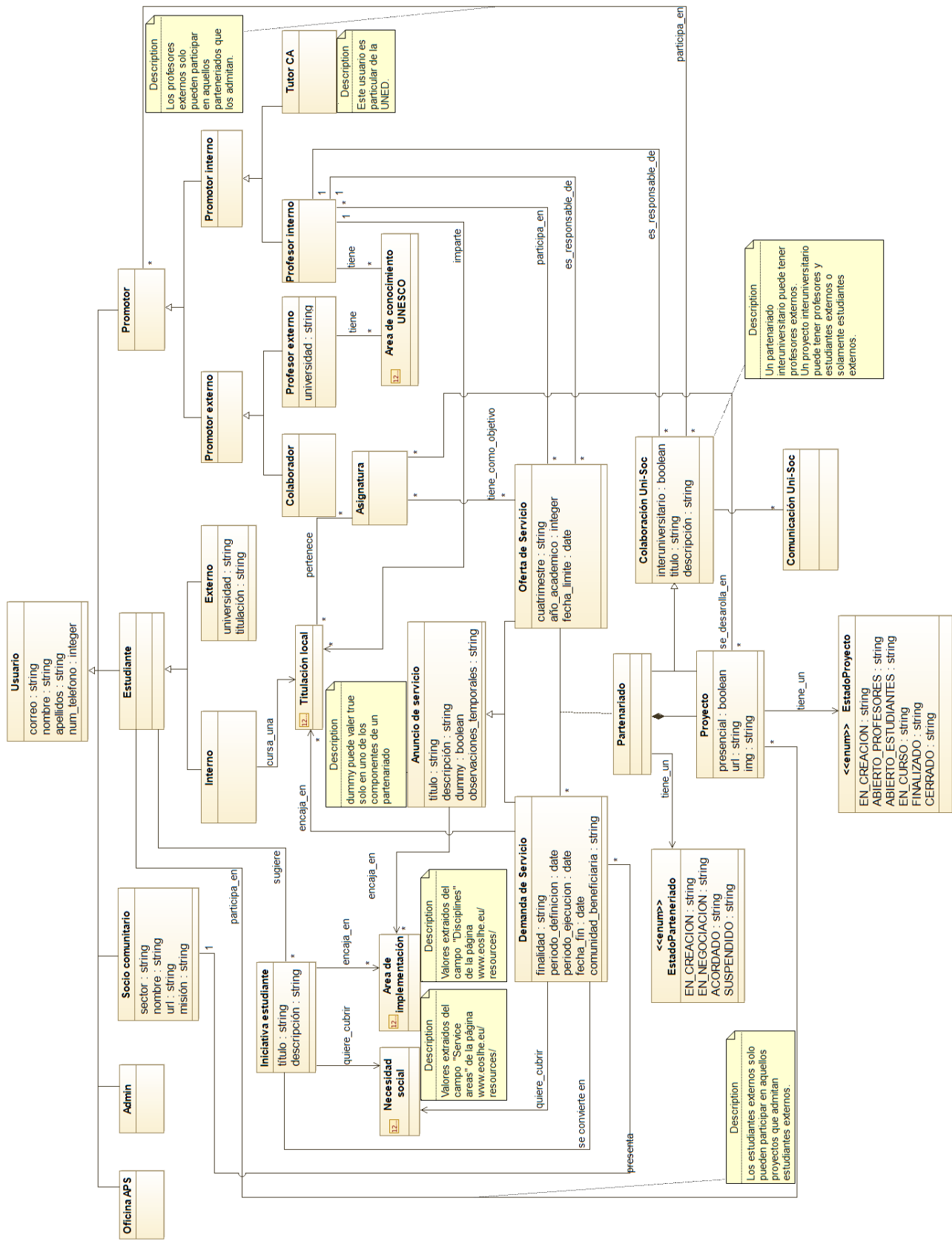


Figura 4.2: Modelo de datos

4.3. Modelo de datos

El modelo de datos describe con más precisión el dominio de la aplicación. Además de algunos atributos importantes como el *interuniversitario*, que indica si el partenariado o el proyecto está abierto a externos, podemos observar los estados que pueden tener el partenariado y el proyecto, representados por dos enumerados. En particular podemos ver que el partenariado puede encontrarse en estados *EN_CREACION*, *EN_NEGOCIACION*, *ACORDADO* y *SUSPENDIDO*. El modelo de datos se puede observar en la Figura 4.2.

- El partenariado se crea en el estado *EN_CREACION* o bien cuando un profesor interno respalda a una demanda presentada previamente por un socio comunitario o acepta un *match* de una oferta y demanda propuesto por la aplicación, o bien cuando un socio comunitario acepta una oferta presentada previamente por un profesor interno. Cuando la otra parte da su asentimiento a la creación del partenariado, su estado pasa a ser *EN_NEGOCIACION*. Cuando el profesor y el socio comunitario terminan de establecer los términos y condiciones del partenariado y ambos están de acuerdo con dichos terminos, el partenariado pasará a estar *ACORDADO*. Si ocurre cualquier discrepancia durante la fase de *EN_CREACION*, *EN_NEGOCIACION* o *ACORDADO* el partenariado puede pasar al estado de *SUSPENDIDO*.
- El proyecto toma los estados *EN_CREACION*, *ABIERTO_PROFESORES*, *ABIERTO_ESTUDIANTES*, *EN_CURSO*, *FINALIZADO* y *CERRADO*. El profesor interno responsable de un partenariado en estado *ACORDADO* puede pedir que se cree un proyecto derivado de este partenariado. El estado inicial de un proyecto es *EN_CREACION*. Cuando el socio comunitario da su asentimiento a la creación del proyecto, su estado pasa a ser *ABIERTO_PROFESORES*. Una vez terminada la definición del proyecto se abre a los alumnos y es allí cuando el proyecto pasa al estado de *ABIERTO_ESTUDIANTES*. Si el proyecto finaliza correctamente, pasará al estado de *FINALIZADO*. Si sucede cualquier imprevisto durante las 4 fases anteriores el proyecto puede pasar al estado *CERRADO*.

4.4. Modelo relacional

Debido a la complejidad de la nueva base de datos compuesta por 46 tablas relacionales, se ha decidido crear un diagrama que sirva como mapa en la gestión de la base de datos. Este modelo relacional se divide en cuatro secciones claramente diferenciadas.

- La sección de los usuarios que contiene tablas relacionadas con información de los usuarios. La sección del **Anuncio de servicio** que contiene todas las tablas que representan información de las ofertas de servicio, las demandas de servicio y las iniciativas.
- La sección de **Colaboración** contiene la información relacionada con los partenariados y los proyectos.
- La sección de Comunicación que contiene las tablas de *mail*, *mensaje*, *upload* y *newsletter*, estas fueron creadas por David Jiménez en los documentos de MongoDB cuya estructura se ha mantenido intacta. En la Figura 4.3 se puede observar el modelo relacional y sus cuatro secciones separadas por colores.



Figura 4.3: Diagrama de entidad-relación

4.4.1. Separación de datos

Una cosa importante para destacar es que, en un entorno real, la aplicación cogería parte de los datos de los usuarios de la base de datos de la universidad en la que se despliegue la aplicación. Es por esto por lo que los datos de los usuarios se han separado en dos grupos; internos y externos. Los usuarios internos son aquellos que pertenecen a la universidad en la que se ha desplegado la aplicación y por ello utilizan SSO (Single Sign-On) de la universidad para acceder a la aplicación. Esta separación de datos se ha hecho con el propósito de facilitar la transición entre la base de datos del prototipo y la base de datos de la aplicación real.

En concreto, los datos que son afectados por esta separación son el correo electrónico, el nombre, los apellidos y la contraseña que deberían alojarse en la base de datos de la universidad.

4.4.2. Usuarios

Esta sección es la más compleja debido a la separación que hay que realizar de los usuarios externos e internos. Ver Figura 4.4.

Un usuario interno es aquel profesor, tutor, estudiante, administrador o representante de la oficina ApS que forma parte de la universidad en la que se despliega la plataforma y por tanto tiene sus datos personales dentro de ella. Debido a que, en un despliegue real tendrían parte de los datos que necesitamos para la aplicación en el sistema interno de la universidad, hay que tratarlos de manera diferente a los usuarios externos, que son aquellos que no pertenecen a la universidad. Los colaboradores y directores que aquí mencionamos se pueden ver representados en el modelo de datos y de dominio, pero no se observan aquí porque no nos dio tiempo a integrarlos en la base de datos.

Debido a esta separación, todos los usuarios comparten una tabla común, llamada usuario, que contiene datos exclusivos de la cuenta de la plataforma. Después tenemos tablas que contienen datos particulares de cada tipo de usuario. Estas son las tablas del *socio comunitario*, estudiante interno, estudiante externo, *admin*, profesor interno, profesor externo y oficina ApS.

Cada uno de estos usuarios poseen una tabla que almacena sus datos personales, haciendo diferenciación entre internos y externos. La tabla de datos personales internos es una tabla creada para la simulación de la aplicación. Una vez la aplicación sea desplegada en un entorno real, esta tabla será eliminada y los datos personales se obtendrán haciendo consultas a la base de datos de la universidad.

Por otra parte, podemos observar tablas secundarias que representan características de los usuarios como por ejemplo, la universidad del profesor y estudiante interno, las áreas de conocimiento UNESCO de los profesores y las titulaciones que imparten los profesores o cursan los alumnos.

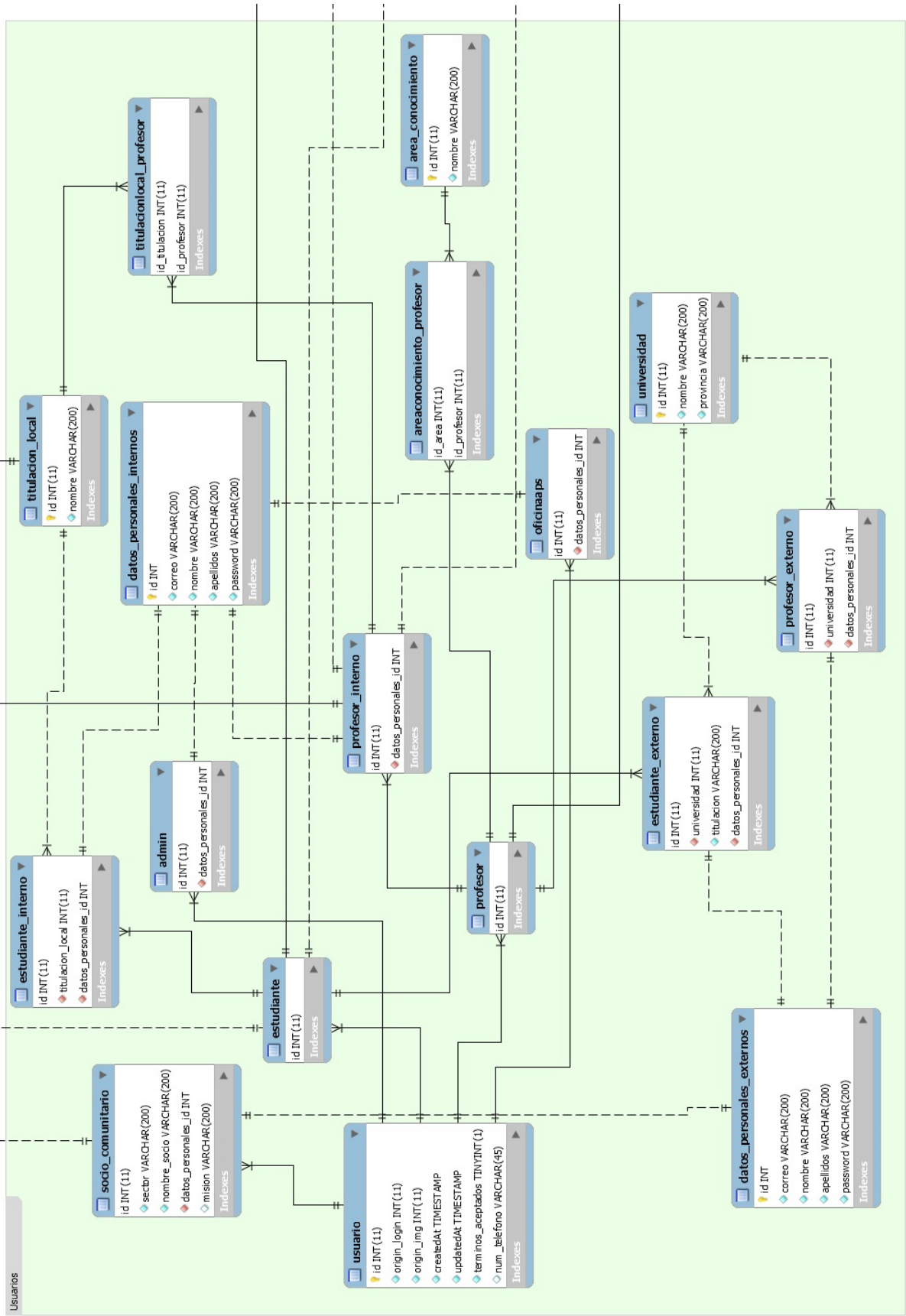


Figura 4.4: Diagrama de entidad-relación - Usuarios

4.4.3. Anuncios de servicio

En este conjunto de tablas podemos encontrar las pertenecientes a la demanda de servicio, la oferta de servicio y la iniciativa. Ver Figura 4.5.

- La iniciativa es una propuesta de proyecto realizada por un estudiante. Esta propuesta debe ser validada por la oficina ApS, que estudiará si la propuesta es viable para ser adoptada por un socio comunitario. Una vez validada la iniciativa, puede ser adoptada por un socio comunitario que desee realizar el proyecto.
- La demanda de servicio es creada por un socio comunitario y define una necesidad específica que quiere cubrir. Esta necesidad social es representada por los elementos de un enumerado alojado en la tabla de necesidad_social. Esta demanda
- La oferta de servicio es creada por un profesor interno y suele tener menos detalles que la demanda porque suele ser una propuesta más genérica.
Cuando una oferta y una demanda son procesadas por el sistema de *matching* se crea una entrada en la tabla *matching* almacenando los *ids* de ambos elementos y el porcentaje de emparejamiento que tienen.
Tanto demanda de servicio como oferta de servicio están conectadas a mensajes y *uploads* porque estos permiten la comunicación con las personas interesadas en las propuestas.

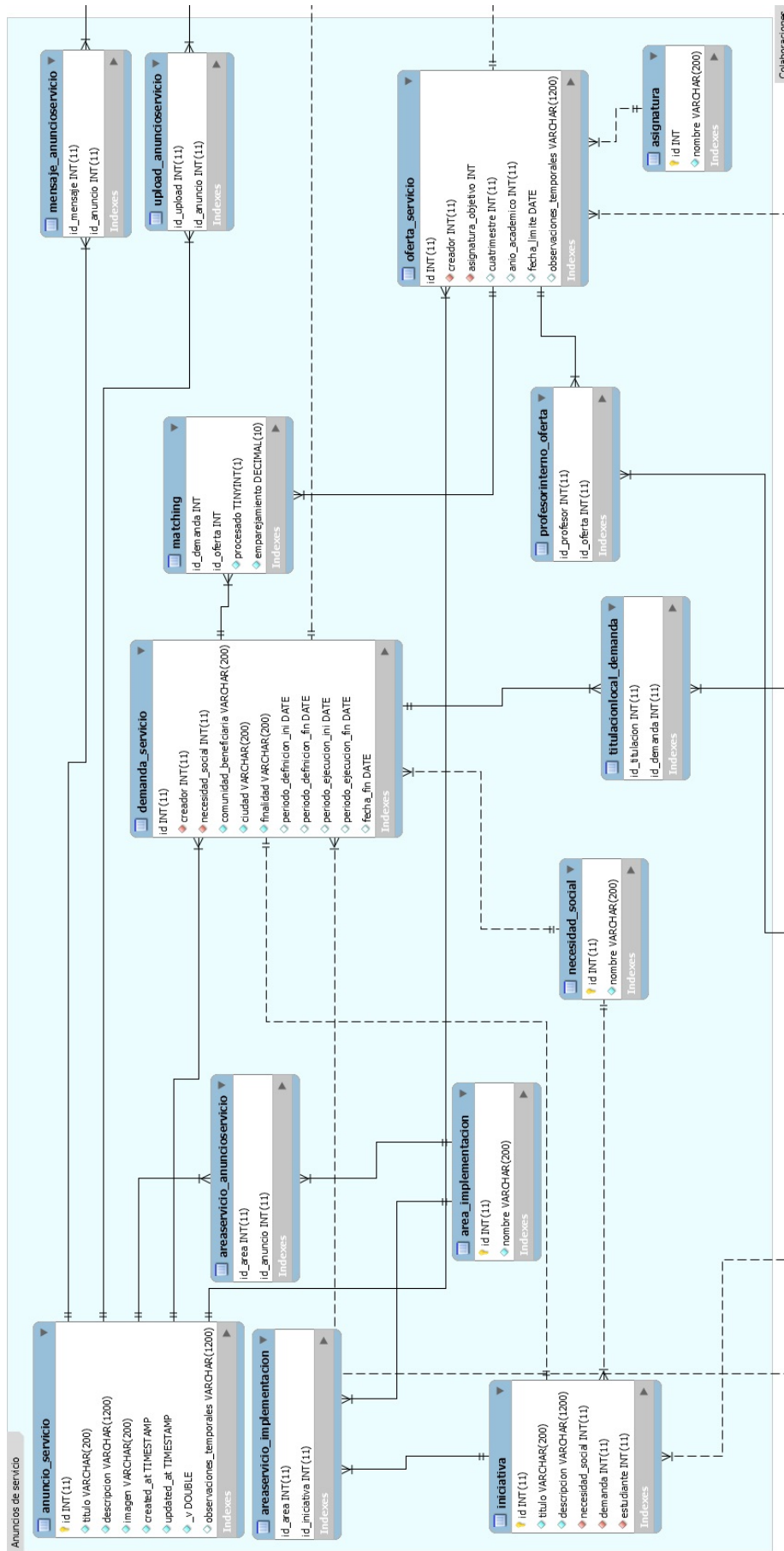


Figura 4.5: Diagrama de entidad-relación - Anuncios de servicio

4.4.4. Colaboración

El partenariado es el segundo paso en la creación de un proyecto. Esta tabla contiene los *ids* de la demanda y la oferta que la componen.

Un proyecto ApS no puede existir sin un partenariado previo y es por eso por lo que tiene un identificador del partenariado a partir del cual se creó. El proyecto posee estudiantes y por ello tiene una conexión con los mismos.

Tanto proyecto como partenariado necesitan un sistema de comunicación y es por eso por lo que tienen tablas intermedias que los conectan a mensaje y *uploads*. Ver Figura 4.6.

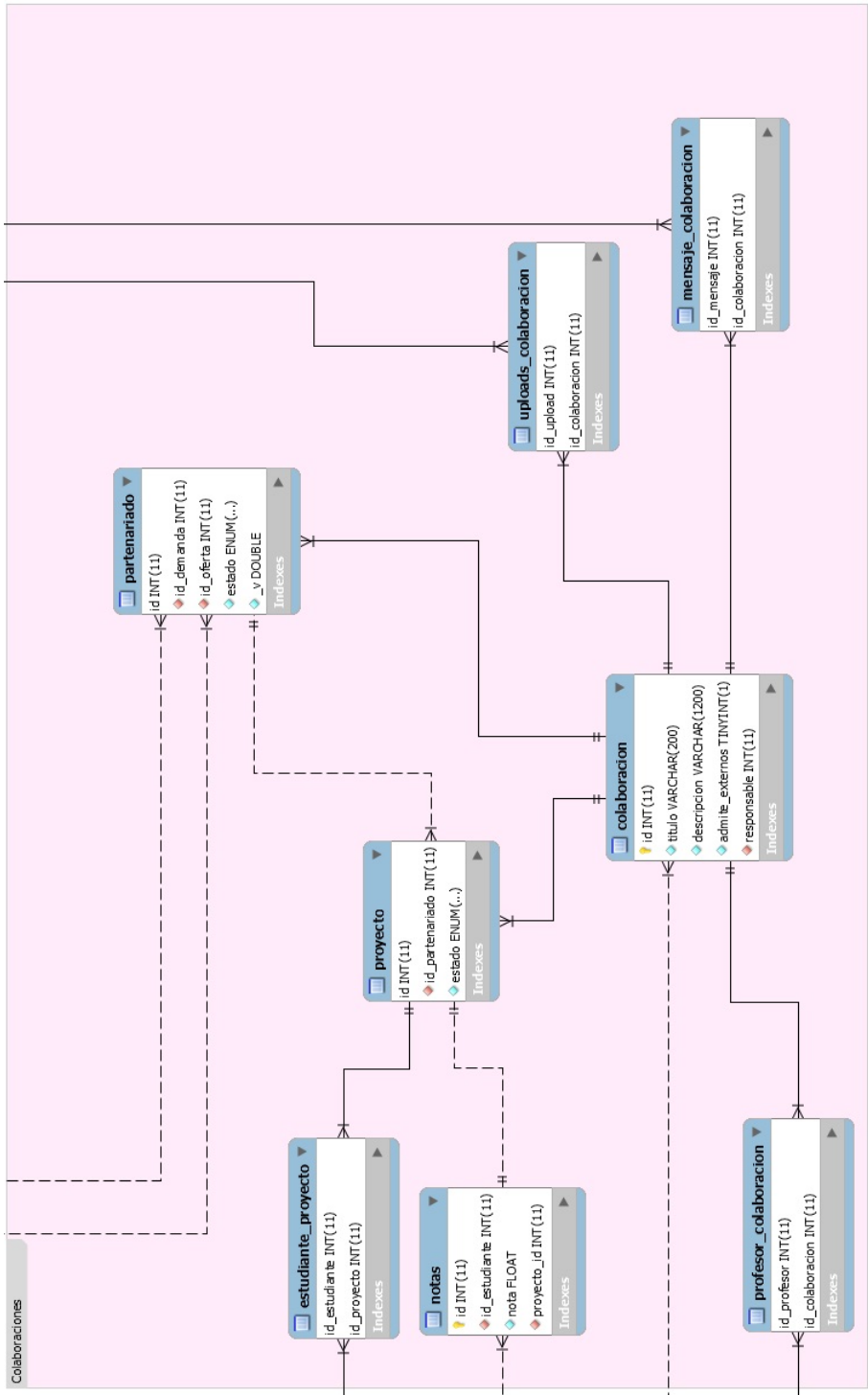


Figura 4.6: Diagrama de entidad-relación - Colaboración

4.4.5. Comunicación

Las funcionalidades de estas tablas no han sido del todo definidas, ya que en un principio se pensó que podrían comunicar al equipo docente con los socios comunitarios en los partenariados y los proyectos, pero también se podrían usar para la comunicación con los creadores de las ofertas y las demandas. Debido a que el funcionamiento no ha sido estudiado aun y que no hemos trabajado con estas tablas en nuestro TFG, se ha mantenido la misma estructura que había definido David Jiménez en sus colecciones de MongoDB adaptándola al modelo relacional. Ver Figura 4.7.

- La tabla de mensajes conecta con las ofertas de servicio, las demandas de servicio, los partenariados y los proyectos porque todos estos necesitan de los mensajes para poder comunicarse.
- La tabla *upload* almacena la información de los ficheros e imágenes subidos tanto en ofertas de servicio, como demandas de servicio, como partenariados y proyectos.
- La tabla *mail* y *newsletter* no han sido conectadas con ninguna otra tabla porque no se han tenido en cuenta para el desarrollo de este TFG, pero representan los correos electrónicos internos de la aplicación y las noticias periódicas enviadas a los usuarios.

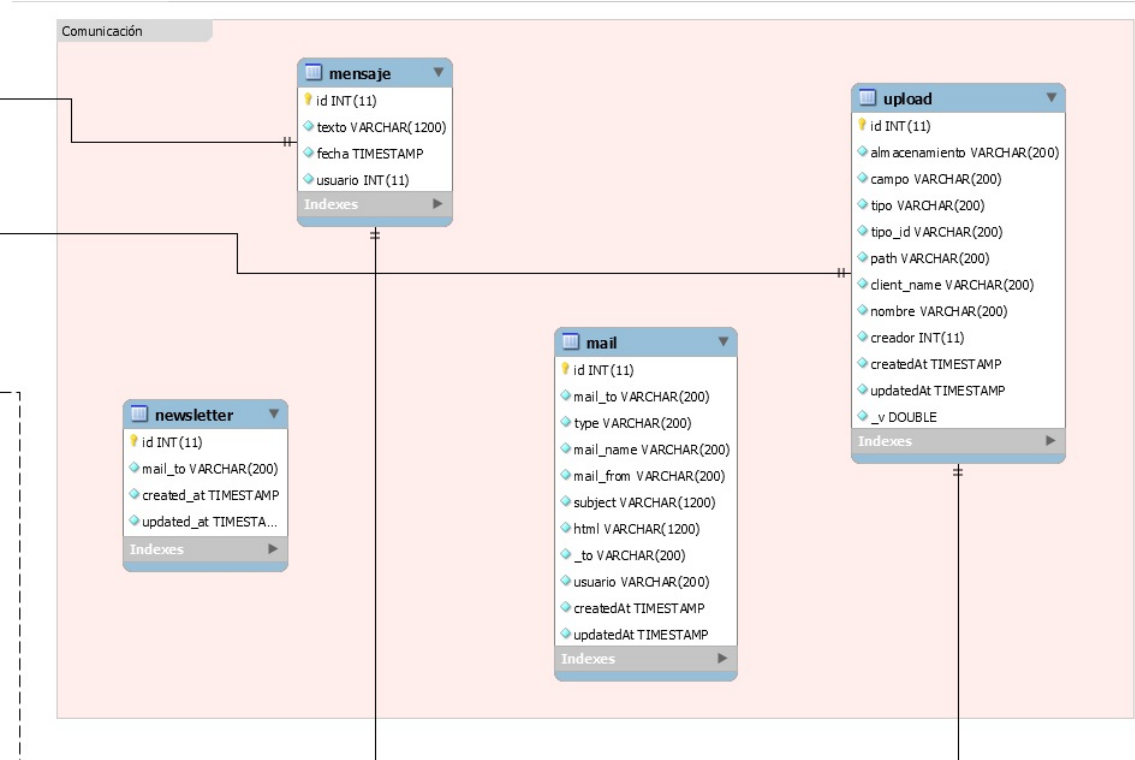


Figura 4.7: Diagrama de entidad-relación - Comunicación

Capítulo 5

Contribución

A continuación, se detallan las contribuciones de cada uno de los componentes del grupo ordenados por orden alfabético.

5.1. Daniela-Nicoleta Boldureanu

5.2. Victoria Gnatiuk Romaniuk

La primera fase era una fase de investigación así que en esta fase todos hemos investigado y no hay ninguna tarea individual que destacar.

En la segunda fase del proyecto Victoria se ha encargado de arreglar algunos *bugs* del trabajo anterior. Los *bugs* eran fáciles de arreglar, uno de ellos era que en el formulario de registro en los campos correo, nombre y apellidos se pedía al usuario que introdujera lo mismo, que era el nombre. Otra cosa que no era un *bug* si no más bien una mejora, era que al editar el perfil de usuario se pedía volver a aceptar las condiciones de uso, cosa que veíamos innecesaria.

En la tercera fase del TFG Victoria se ha encargado de continuar el modelo de dominio y el modelo de datos usando Modelio. En paralelo ha creado las tablas **profesor_externo**, **iniciativa**, **anuncio_servicio**, **demanda_servicio**, **oferta_servicio**, **partenariado**, **proyecto**, **colaboracion**, **newsletter** y **entidad_demanda** en la base de datos. También ha creado las tablas que alojan los valores de los enumerados como la tabla **area_implementación**, **necesidad_social**, **area_conocimiento**, **universidad**, **titulacion_local** y las tablas intermedias que permiten conectar las tablas de los enumerados con las tablas que las referencian. Con crear las tablas nos referimos a crear la estructura, definiendo los nombres de los campos y los tipos de datos, y creando restricciones entre las tablas.

Una vez todas las tablas fueron creadas usando MySQL Workbench 8.0 CE, Victoria creó el modelo relacional a partir del fichero SQL. Después hubo que distribuir todas las tablas en los cuatro grupos explicados en el capítulo 4.3 para que todas las tablas fueran debidamente visibles.

En la fase cuatro Victoria se ha encargado de desarrollar el DAO llamado *Tentativa* que implementa el acceso a la base de datos del grupo de tablas llamado *Anuncios de servicio* que se puede ver en la Figura 4.5. La creación de este DAO ha supuesto la necesidad de la creación de tres *transfers* que alojan los datos del elemento Anuncio de servicio, Demanda

de servicio y Oferta de servicio. Los daos han sido implementados usando Knex.js, una librería que facilita las consultas a la base de datos. En el DAO *Tentativa* se crearon las funciones CRUD, 3 de cada tipo concretamente. Dos de ellas eran las de la oferta y la demanda y la tercera, la cual siempre se llama desde las otras dos, es la del anuncio de servicio. También se creó el método para obtener todas las iniciativas, todas las demandas y todas las ofertas. Además de estos métodos, se han creado otros necesarios para la creación de los formularios de crear oferta y demanda.

Después de finalizar la implementación del DAO *Tentativa* Victoria desarrolló en los métodos CRUD del elemento *Partenariado* perteneciente al DAO *Colaboración*. Este DAO implementa las tablas pertenecientes al grupo *Colaboración* que se puede ver en esta Figura 4.6. Después Victoria implementó en el DAO *Usuarios* dos métodos para obtener cualquier tipo de usuario con el id y con el correo del usuario. Estos métodos eran necesarios para la adaptación de la página de registro y *login* al nuevo sistema. Junto con Jesús, Victoria adaptó el *login* al nuevo sistema.

En la fase cinco Victoria se ha encargado de crear el método que determina la similitud que tiene los campos **area_implementación**, perteneciente a la *Oferta de servicio* y **area_conocimiento**, perteneciente a la *Demanda de servicio*. Para esto se ha tenido que crear un Excel donde los valores de los enumerados de **area_implementación** se han colocado en la primera fila y los valores de los enumerados de **area_conocimiento** se han colocado en la primera columna. Posteriormente se han colocado X para marcar la similitud entre los valores, esta similitud ha sido determinada por Victoria. Después de crear el Excel se ha creado un *script* con Python que leía este Excel e insertaba los emparejamientos en una tabla de la base de datos, de la cual se leerá posteriormente para determinar si estos dos enumerados pueden ser emparejados. A continuación, se creó el método que compraba estos enumerados. Se hizo este mismo proceso para determinar el emparejamiento entre **area_implementación**, perteneciente a la *Oferta de servicio* y **titulacion_local**, perteneciente a la *Demanda de servicio*.

Además de estos métodos para determinar el *matching*, Victoria implementó otros dos. Uno de ellos determinaba si las dos **areas_implementación**, pertenecientes a Oferta de servicio y Demanda de servicio coinciden. El segundo método determinaba si coincidían las titulaciones de la *Demanda de servicio* y las titulaciones que imparte el profesor que ha creado la *Oferta de servicio*.

En la fase cinco Victoria se ha encargado de implementar el formulario para crear la oferta de servicio y también ha desarrollado la parte *back-end* del formulario para crear el partenariado por parte del profesor y por parte del socio comunitario. Conjuntamente con todo esto Victoria ha ido actualizando los modelos de dominio y de datos a medida que iban sufriendo cambios a causa de los cambios sugeridos por los directores del TFG. También ha mantenido actualizado el diagrama relacional a medida que la base de datos iba sufriendo cambios.

5.3. Jesús Sánchez Granado

Bibliografía

- [1] Natalia Rodriguez-Fernández Angeles Manjarrés Riesco, Simon James Pickin y Héctor Alonso Meana. Virtu@l-aps: Soporte tecnológico para el aprendizaje-servicio virtual. *Revista Iberoamericana de Educación a Distancia*, Vol. 23, Núm. 1, 2020.
- [2] David Jiménez del Rey. Desarrollo de una comunidad web para el soporte virtual del aprendizaje-servicio 3. *Directores: Ángeles Manjarrés Riesco UNED, Simon Pickin UCM. Titulación: Grado en Ingeniería Informática de la UNED. Octubre 2020.*
- [3] Héctor Alonso Meana. Aplicación de soporte a una comunidad educativa interesada en el aps virtual. *Directores: Ángeles Manjarrés Riesco UNED, Simon Pickin UCM. Titulación: Grado en Ingeniería Informática de la UNED. Junio 2018.*
- [4] Emmanuel Parmentier. Diseño e implementación de un servicio de conexión de la oferta y la demanda de proyectos de fin de carrera de cooperación. *Director: Carlos Mataix UPM Titulación: Ingeniería Industrial de la UPM. 2004.*
- [5] M.Á.S. Rego, A.S. Losada, and M.L. Moledo. *Aprendizaje-servicio y misión cívica de la universidad: Una propuesta de desarrollo.* Ediciones Octaedro, 2015.
- [6] Natalia Rodriguez. Virtu@l-aps. *Director: Juan García Gutierrez, UNED. Titulación: Grado en Educación Social. 2017.*
- [7] Sol Lozano Serrano. Desarrollo de una comunidad virtual de profesores en el Ámbito de la educación para el desarrollo humano y sostenible. *Directores: Ángeles Manjarrés Riesco UNED, Simon Pickin UCM. Titulación: Ingeniería Informática de la UNED. Sept. 2015.*