



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

Facultad de Ingeniería Informática

APLICACIÓN WEB DE SOPORTE AL APRENDIZAJE-SERVICIO

Autores

Daniela-Nicoleta Boldureanu (Grado en Ingeniería del Software)

Victoria Gnatiuk Romaniuk (Grado en Ingeniería Informática)

Jesús Sánchez Granado (Grado en Ingeniería Informática)

Tutores

Simon Pickin

Manuel Montenegro Montes

Índice general

1. Introducción	7
1.1. Antecedentes	7
1.2. Objetivos	7
1.3. Plan de trabajo	8
2. Contexto de la propuesta	11
2.1. Introducción	11
2.2. Elementos que intervienen en un proyecto ApS	11
2.3. Motivación	12
2.4. TFG de partida	13
2.5. Estudio tecnológico	14
3. Modelos de dominio, de datos y de relación	17
3.1. Introducción	17
3.2. Modelo de dominio	18
3.3. Modelo de datos	20
3.4. Modelo relacional	21
3.4.1. Separación de datos	22
3.4.2. Usuarios	22
3.4.3. Anuncios de servicio	22
3.4.4. Colaboración	23
3.4.5. Comunicación	24

Índice de figuras

1.1. PivotalTracker: tareas	9
3.1. Modelo de dominio	19
3.2. Modelo de datos	20
3.3. Diagrama de entidad-relación	21
3.4. Diagrama de entidad-relación - Usuarios	23
3.5. Diagrama de entidad-relación - Anuncios de servicio	24
3.6. Diagrama de entidad-relación - Colaboración	25
3.7. Diagrama de entidad-relación - Comunicación	25

Capítulo 1

Introducción

El objetivo de este *Trabajo de Fin de Grado* (TFG) es la continuación de un proyecto que se empezó a plantear en 2004 y a diseñar, y desarrollar en 2018. Este proyecto se detalla más adelante en el capítulo 2.3. El proyecto 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 moldearles 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. 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 (*Aprendizaje-Servicio*). 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 2 se explica con más detalle que es un proyecto ApS.

1.2. Objetivos

Partiendo del TFG 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 tutores, 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 1.1. 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 5 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 3. Esta fase ha comprendido entre el día 20 de noviembre y el día 26 de febrero.

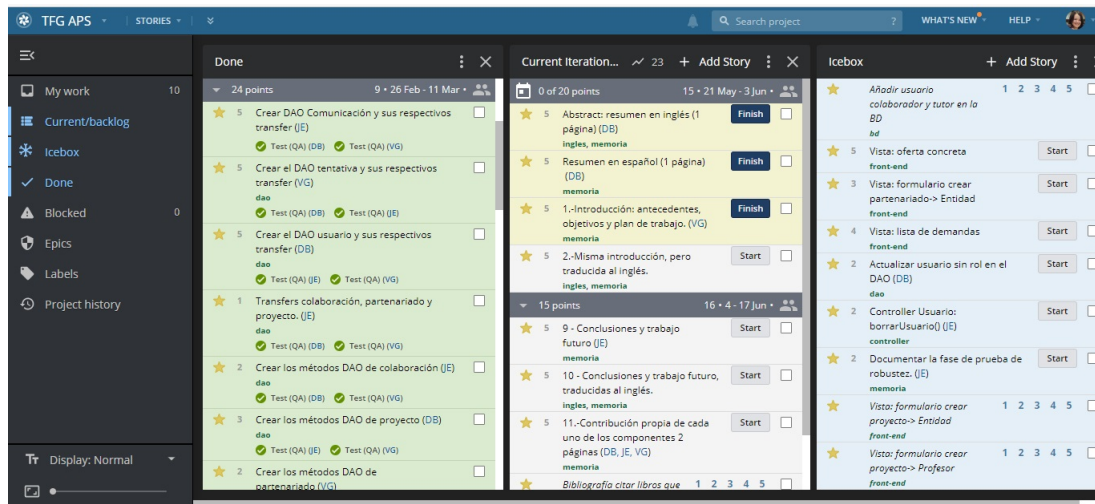


Figura 1.1: PivotalTracker: tareas

- La cuarta fase ha comprendido entre el día 27 de febrero y el día 26 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 cuarta 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 27 de marzo y el día 23 de abril.
- La quinta 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

Contexto de la propuesta

2.1. 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.

2.2. 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 de el 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 realizan reuniones para determinar las características y particularidades del problema. Una vez definidos los términos y condiciones del futuro proyecto, el socio comunitario y el profesor abren el proyecto a los alumnos.
- 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.

2.3. Motivación

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 *Proyecto Fin de Carrera* 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 [6], 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 [6] 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 [6] 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 [5], 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 [5], 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.

2.4. 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 si, no había razones suficientes para seguir desarrollando la aplicación sobre MongoDB, así que junto con nuestros tutores 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 si. 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.

2.5. 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 3

Modelos de dominio, de datos y de relación

3.1. Introducción

Al empezar con el proyecto, junto con nuestros tutores, 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 no conocen todos los detalles del problema en cuestión que quieren resolver, dentro del marco de los proyectos ApS. Los profesores, por su parte, tienen una idea muy vaga 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 como 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, no solo porque ha cambiado su tipo, que ahora es relacional, sino también porque los conceptos no estaban claros en el anterior TFG, 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.

3.2. 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 3.1.

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. En promotor externo representaríamos 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 o un tutor de la universidad donde se aloja la aplicación. Este profesor es el que más cargos de responsabilidad tiene. Él es el que puede crear las ofertas de servicio y es el responsable de crear los partenariados y los proyectos.
- 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 tramites relacionados con la gestión de los proyectos ApS.

Todo proyecto tiene que definirse en el contexto de un partenariado y este partenariado se origina en la unión de una oferta creada por un profesor, y una demanda creada por un socio comunitario. La creación de la oferta y la demanda se realiza mediante unos formularios en la web.

La aplicación también contempla la creación de 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 “monoparental” del partenariado.

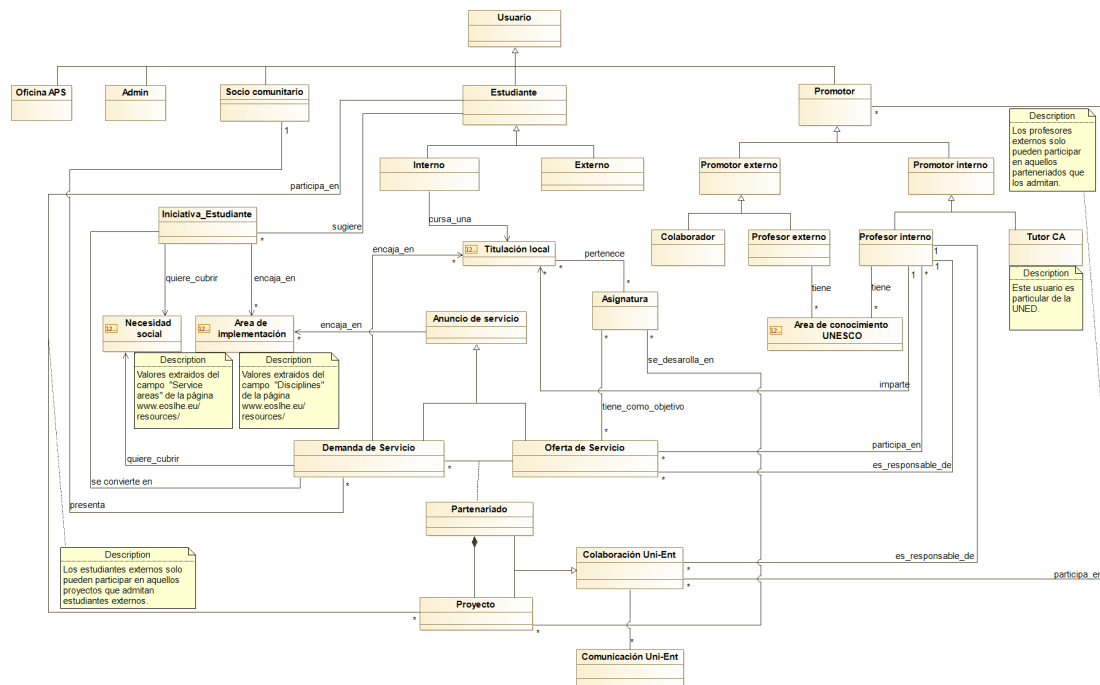


Figura 3.1: Modelo de dominio

En general una demanda de servicio es creada por una 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 posteriormente por un socio comunitario. 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-Ent. Queremos destacar que los atributos de Necesidad social y Área 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.

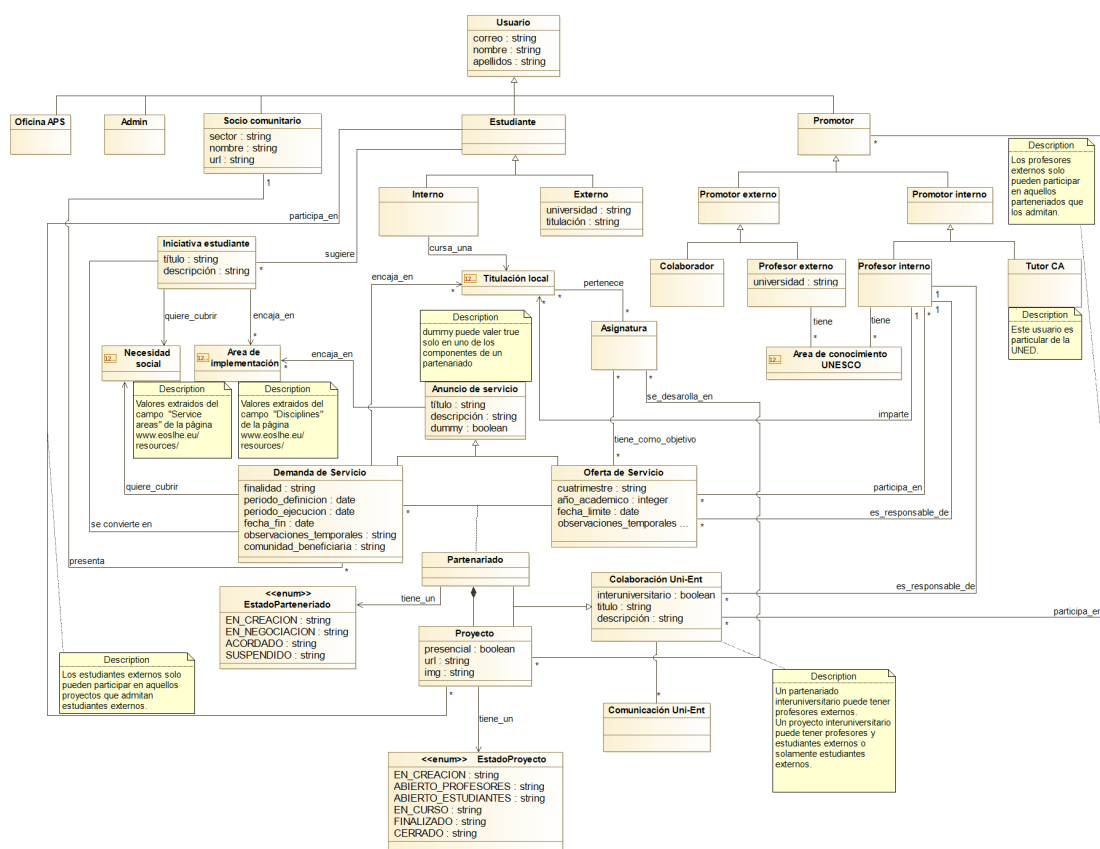


Figura 3.2: Modelo de datos

3.3. 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 3.2.

- El partenariado toma el estado de `EN_CREACIÓN` cuando el profesor se pone en contacto con el socio comunitario o el socio comunitario se pone en contacto con el profesor. Si el receptor de la propuesta contesta y acepta colaborar con el otro, el estado del partenariado pasará a `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

3.4.1. Separación de datos

Una cosa importante a 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 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) 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.

3.4.2. Usuarios

Esta sección es la más compleja debido a la separación que hay que realizar de los usuarios externos e internos. Ver Figura 3.4.

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 tutores 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.

3.4.3. 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 3.5.

- 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.

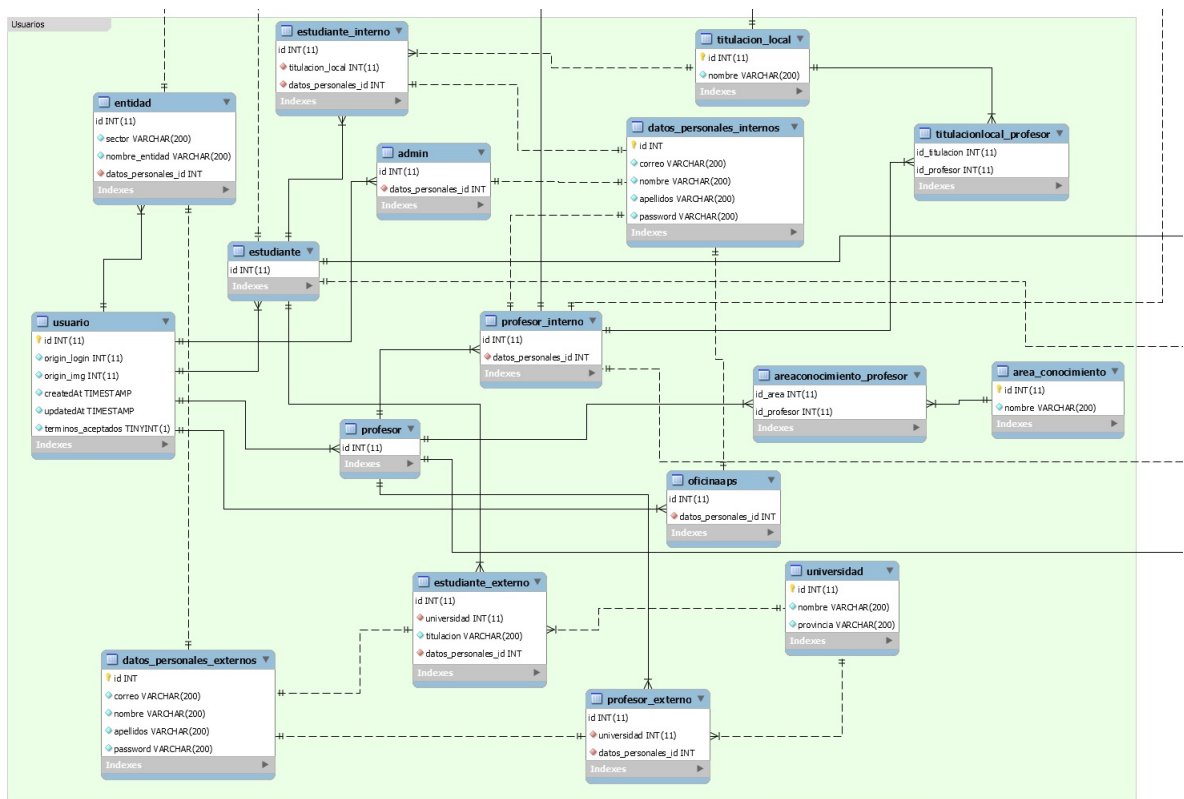


Figura 3.4: Diagrama de entidad-relación - Usuarios

- 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*.
- 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.

3.4.4. 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 3.6.

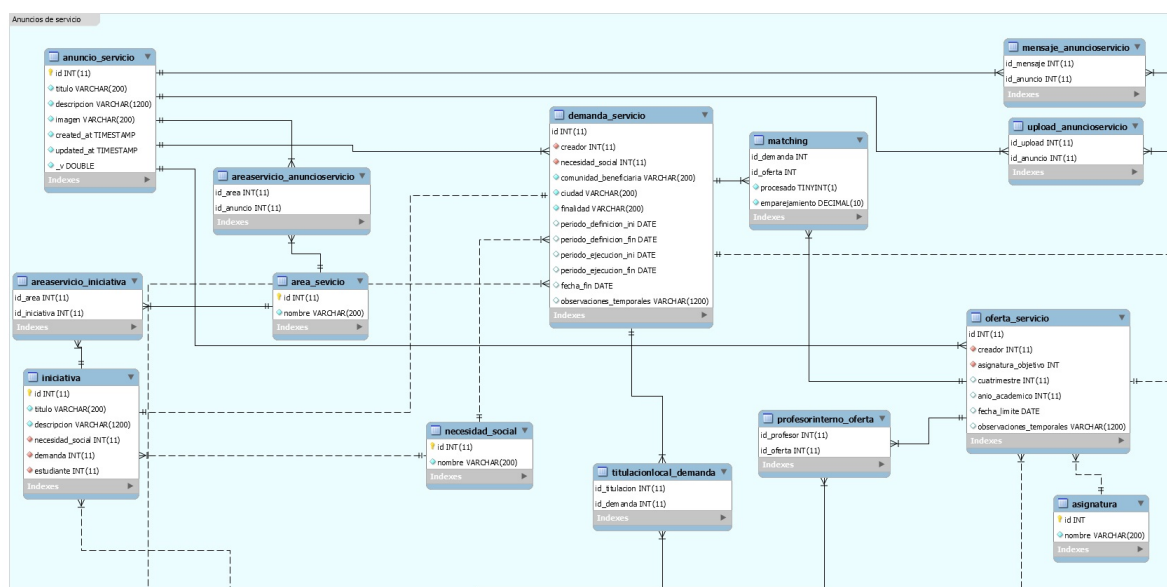


Figura 3.5: Diagrama de entidad-relación - Anuncios de servicio

3.4.5. 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 3.7.

- 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.

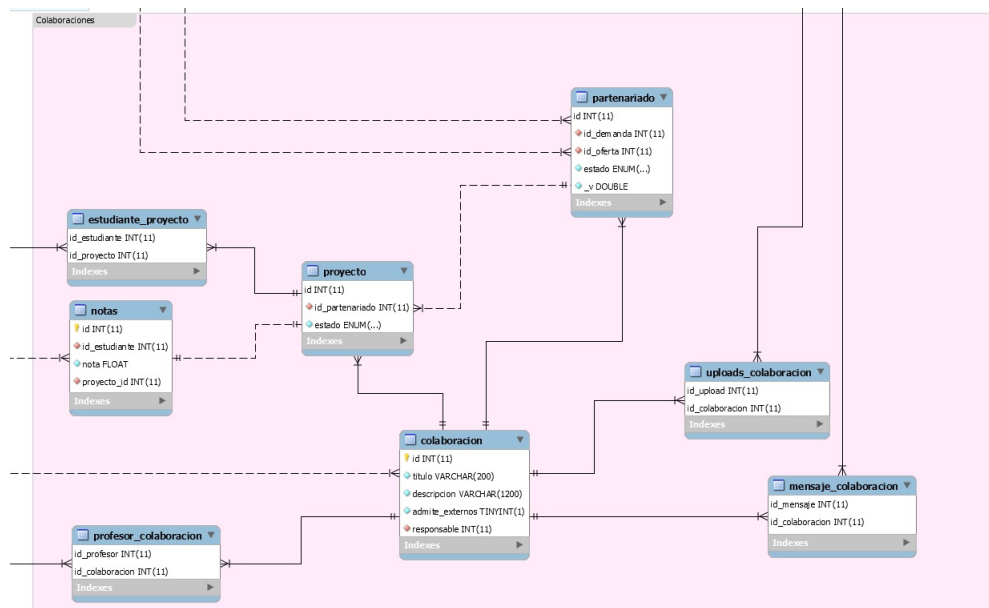


Figura 3.6: Diagrama de entidad-relación - Colaboración

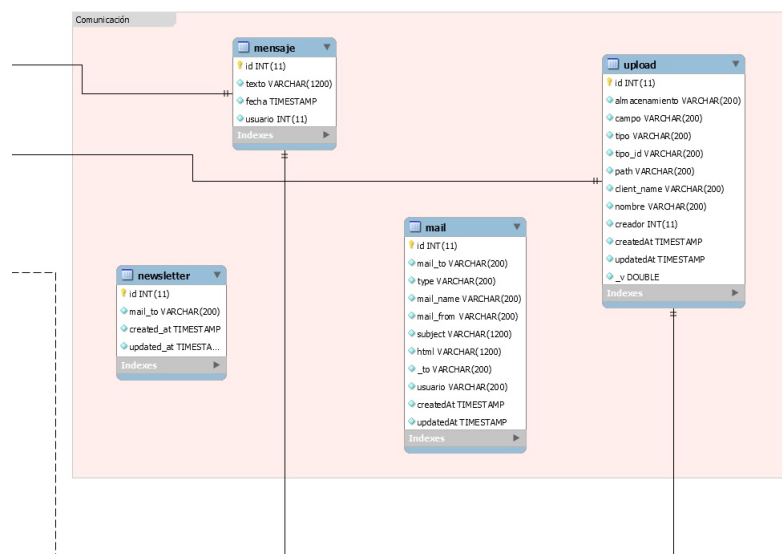


Figura 3.7: Diagrama de entidad-relación - Comunicación

Bibliografía

- [1] Virtu@l-aps: Soporte tecnológico para el aprendizaje-servicio virtual. *Revista Iberoamericana de Educación a Distancia*, Vol. 23, Núm. 1, 2020.
- [2] D. J. del Rey. Desarrollo de una comunidad web para el soporte virtual del aprendizaje-servicio iii.
- [3] H. A. Meana. Aplicación de soporte a una comunidad educativa interesada en el aps virtual.
- [4] E. 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] N. Rodriguez. Virtu@l-aps.
- [6] S. L. Serrano. Desarrollo de una comunidad virtual de profesores en el Ámbito de la educación para el desarrollo humano y sostenible.