



UNIVERSIDAD
COMPLUTENSE
MADRID

Proyecto de Innovación
Convocatoria 2022/2023

Nº de proyecto: 84

Aplicación web progresiva para la realización de
cuestionarios offline: difusión e integración con Moodle

Responsable del proyecto: Manuel Montenegro Montes

Facultad de Informática

Departamento: Sistemas Informáticos y Computación

1. Objetivos propuestos en la presentación del proyecto

Los cuestionarios en línea son una valiosa herramienta en el aprendizaje del alumnado y en la evaluación de las asignaturas. Por un lado, proporcionan un mecanismo eficaz y rápido de autoevaluación, ya que permiten a un estudiante comprobar que ha comprendido los conceptos teóricos mediante la obtención de retroalimentación inmediata. Con respecto a los cuestionarios realizados en papel, los cuestionarios en línea aportan las siguientes ventajas en la enseñanza presencial:

- Aplicación de mecanismos de aleatorización, que dificultan los intentos de copia y fraude.
- Corrección automática o semiautomática (en función del tipo de preguntas planteadas) que facilita la labor de corrección al docente, y además permite obtener a los estudiantes la calificación obtenida de manera instantánea.
- Ahorro de papel, ya que solo es necesario disponer de copias para aquellos estudiantes que no puedan realizar el cuestionario en línea.

Siguiendo en el ámbito de la enseñanza presencial, algunos de los participantes de la presente propuesta de proyecto han aplicado, durante los últimos cursos, metodologías basadas en la modalidad de aula invertida. En esta metodología, los estudiantes se familiarizan con los conceptos teóricos fuera de las horas de clase utilizando material proporcionado por el profesor (en nuestro caso, material audiovisual). Este material se intercala con cuestionarios de autoevaluación que permiten comprobar la correcta comprensión de los contenidos, mientras que las sesiones presenciales se dedican a la realización de actividades en grupo, prácticas de laboratorio y resolución de problemas. En los primeros minutos de algunas sesiones se realiza un breve cuestionario de control, con el fin de asegurar que los estudiantes tienen la preparación mínima necesaria para realizar estas actividades.

La herramienta utilizada para la realización de cuestionarios en línea de manera presencial es, precisamente, la plataforma Moodle utilizada en el Campus Virtual de la UCM. Sin embargo, uno de los problemas que tiene esta herramienta es la necesidad de disponer de una conexión a internet durante todo el proceso de realización del cuestionario. Esta conexión puede verse afectada por varios problemas:

- Sobrecarga en los servidores del Campus Virtual, debido a la gran cantidad de accesos simultáneos. Este riesgo se ha visto mitigado por la existencia de entornos Moodle dedicados exclusivamente a la realización de cuestionarios.
- Problemas de acceso a la red inalámbrica. En las asignaturas con un gran número de estudiantes matriculados, todos los asistentes a la clase se conectan simultáneamente a la red inalámbrica del centro desde el mismo aula, y la conexión con esta red está sujeta a las fluctuaciones habituales en la cobertura.

Estos problemas conllevan varias desventajas:

- Los profesores deben preparar copias del cuestionario en papel para aquellos estudiantes que no puedan acceder al cuestionario en línea, lo que a su vez conlleva vaticinar cuántas copias serán necesarias.

- Se genera una situación de nerviosismo en los estudiantes, ante la amenaza de no poder realizar el cuestionario por motivos ajenos a ellos.
- En el peor de los casos, si estas dificultades de conexión se producen de manera generalizada, podrían llegar a impedir o invalidar el cuestionario.

Para mitigar los problemas relacionados con una repentina desconexión disponemos de las **aplicaciones web progresivas** (en inglés, *Progressive Web Applications* o PWA). Una PWA es una página web que tiene la capacidad de seguir funcionando incluso cuando el dispositivo no está conectado a la red. Además, las PWA funcionan directamente en el navegador web, sin necesidad de instalación en el dispositivo. Entre las PWA más conocidas se encuentran, por ejemplo, Google Drive, Google Docs o Twitter.

El objetivo de este proyecto es la **mejora, implantación y difusión de una PWA para la realización de cuestionarios**. Algunos de los participantes de este proyecto trabajaron durante el curso anterior en el desarrollo de dicha aplicación, cuyo funcionamiento se basa en los siguientes principios:

- Los estudiantes pueden descargarse los cuestionarios horas o días antes de su realización. Estos cuestionarios se almacenan en el navegador del dispositivo de manera encriptada, de modo que el estudiante no puede acceder a su contenido hasta obtener la clave que permita descifrarlos.
- En el momento de realizar el cuestionario en clase, el profesor desvela la clave para descifrar los cuestionarios, lo que permite a los estudiantes realizar el cuestionario descargado previamente, incluso si no tienen conexión en ese momento.
- Una vez finalizado el tiempo de realización del cuestionario, el estudiante entrega sus respuestas. Si dispone de conexión en ese momento, las respuestas se almacenan en el servidor y se evalúan automáticamente, proporcionando la calificación al estudiante. Si no dispone de conexión, el estudiante puede obtener una huella digital de las respuestas proporcionadas en forma de código QR. El profesor captura este código para obtener dicha huella. De este modo, cuando el estudiante recupera la conexión puede subir sus respuestas al servidor. La huella digital permite asegurar que el estudiante no ha manipulado sus respuestas tras la realización de la prueba.

De este modo, si un profesor quiere realizar una prueba de evaluación basada en un cuestionario, cualquier estudiante que haya tomado la precaución de descargar el cuestionario antes de la prueba, tendrá la seguridad de que podrá realizarlo independientemente de los problemas de conexión que puedan surgir en el transcurso de la misma.

2. Objetivos alcanzados

El objetivo principal del presente proyecto ha sido la mejora, despliegue y difusión de la aplicación web progresiva de realización de cuestionarios. Este objetivo general se desglosa en cuatro objetivos concretos, descritos en las siguientes secciones.

2.1. Mejoras en la aplicación de realización de cuestionarios

En este proyecto partíamos de un prototipo inicial, con la funcionalidad estrictamente necesaria para determinar la viabilidad de la PWA. Hemos extendido este prototipo con el fin de preparar la PWA para su uso en un entorno real. Las mejoras planteadas con este fin son las siguientes:

- Mejoras relativas a accesibilidad y diseño adaptable.

La PWA está concebida para su uso en el aula a través de una amplia variedad de dispositivos: ordenadores portátiles, móviles, tabletas, etc. cuyas dimensiones físicas pueden ser dispares. Es necesario asegurarse de que los cuestionarios se pueden realizar sin problemas en cualquiera de estos dispositivos, incluso en ausencia de conexión. Durante la ejecución del proyecto hemos ampliado o reemplazado aquellos componentes con problemas de usabilidad en dispositivos pequeños, por otros más adaptables a distintos tipos de dispositivos.

- Inclusión de mecanismos de aleatorización.

Para prevenir situaciones de copia entre los estudiantes que realizan un cuestionario, hemos añadido varias funcionalidades a la PWA que permiten que cada estudiante reciba una variante aleatoria del cuestionario. La aleatorización de los cuestionarios se ha llevado a cabo de varias formas:

- Seleccionando aleatoriamente el orden de las preguntas dentro de un cuestionario.
- Seleccionando aleatoriamente el orden de las posibles respuestas de una pregunta tipo test.
- Introduciendo *preguntas aleatorias*, en las que cada estudiante recibe una pregunta tomada aleatoriamente a partir de un conjunto.

- Uso de formato Markdown, imágenes y fórmulas en el enunciado de las preguntas

Con el fin de que un profesor pueda introducir texto con formato en sus preguntas, la PWA ahora permite introducir los enunciados de las preguntas y las posibles opciones (en una pregunta de tipo test) en formato Markdown. Este formato, además de permitir asignar formato al texto (negrita, itálica, subrayado, etc.) permite incorporar imágenes y fórmulas matemáticas a los enunciados.

Con las mejoras realizadas, podemos afirmar que este objetivo se ha cumplido en su totalidad.

2.2. Despliegue de la aplicación web y pruebas de funcionamiento

Para facilitar el despliegue de la PWA en cualquier servidor, se ha hecho uso de la herramienta *Docker*, que permite crear contenedores virtuales para cada módulo de la aplicación. Se han creado contenedores independientes para el sistema gestor de bases de datos, para el *back-end* de la aplicación, para el *front-end* de la aplicación, y para un

servidor web que hace de puerta de enlace a estos dos últimos contenedores. De este modo, es posible instalar y desplegar la aplicación en cualquier servidor sin necesidad de instalar previamente las dependencias y de configurar el entorno.

Una vez diseñados los contenedores, se ha realizado el despliegue de la PWA en uno de los servidores de la Facultad de Informática. La PWA es accesible a través de la dirección <http://dalila.sip.ucm.es:8000>.

Aunque entre los objetivos planteados inicialmente en el proyecto estaba la realización de pruebas de funcionamiento de la PWA en el aula, las dificultades encontradas al principio del proyecto (ver sección 5) han hecho que estas pruebas hayan tenido que aplazarse a otro curso académico. Durante el presente proyecto sí se han realizado pruebas de funcionamiento a pequeña escala, en el entorno de desarrollo, y se han aplicado técnicas de testing de unidad en el lado del servidor (*back-end*).

2.3. Integración con las herramientas de Moodle

La implementación inicial de la PWA permitía importar preguntas y cuestionarios escritos en un lenguaje de marcado (YAML) distinto de los utilizados por Moodle. Entre los objetivos conseguidos en este proyecto está la posibilidad de importar preguntas a partir de los formatos soportados por Moodle (en particular, Moodle XML). De este modo, los profesores pueden reutilizar sus bancos de preguntas ya existentes sin necesidad de reescribir de nuevo todas las preguntas en el formato de la PWA, facilitando así la adopción por parte de la comunidad de profesores de la universidad.

Por otro lado, muchos profesores utilizan la herramienta *Calificaciones* de Moodle para gestionar y publicar los resultados de las pruebas de evaluación. Sin embargo, la PWA utiliza un sistema propio para almacenar las calificaciones. Por este motivo se ha añadido la funcionalidad de exportar las calificaciones de un cuestionario a un formato CSV, que puede ser importado directamente por Moodle.

2.4. Difusión de la herramienta

Tras la obtención de una primera versión estable, el código fuente de la PWA ha sido publicado en un repositorio Git: <https://github.com/Qwizer-UCM/Qwizer>. Dicho repositorio contiene las instrucciones de instalación y despliegue de la aplicación.

Antes de dar más difusión de la herramienta, por ejemplo a través de un taller o seminario, consideramos necesario realizar previamente una fase de pruebas más exhaustiva, preferiblemente en forma de pruebas piloto en el aula.

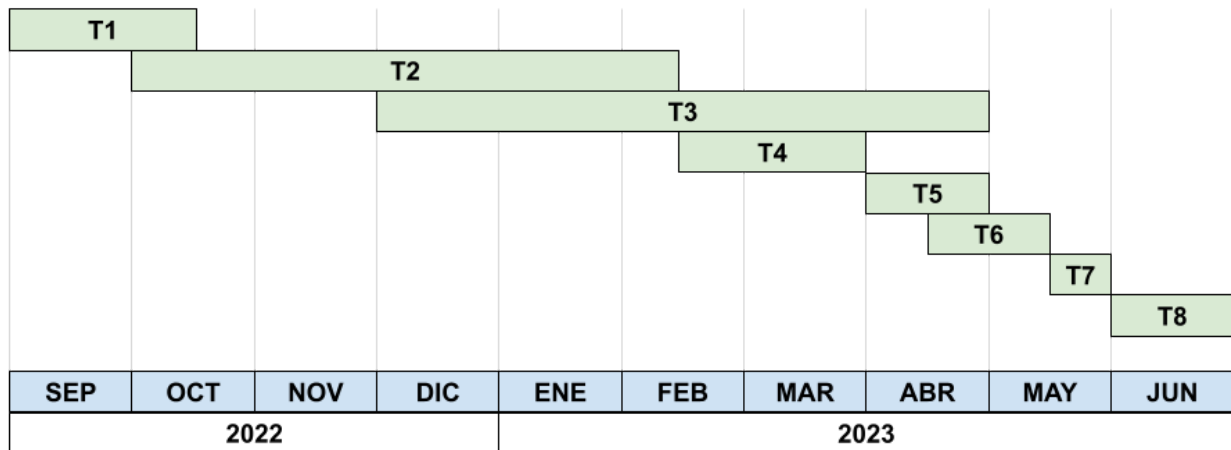
3. Metodología empleada en el proyecto

Las principales tareas del proyecto estaban centradas en el desarrollo de las nuevas funcionalidades de la PWA. Estas tareas se llevaron a cabo en el contexto de un TFG realizado por Vicentiu Tiberius Roman y José Luis Bartha de las Peñas [5]. En este desarrollo, hubo una primera fase de familiarización con el código y una profunda refactorización del *front-end* y del *back-end* de la misma.

A continuación se muestra el listado de tareas realizadas:

- **Tarea 1.** Estudio del código fuente del prototipo inicial y elección de tecnologías.
- **Tarea 2.** Refactorización del prototipo de la PWA y corrección de errores.
- **Tarea 3.** Introducción de mejoras de diseño adaptable.
- **Tarea 4.** Desarrollo de funcionalidades relativas a la aleatorización de cuestionarios.
- **Tarea 5.** Inclusión de formato Markdown, imágenes y fórmulas en los enunciados de las preguntas y opciones.
- **Tarea 6.** Desarrollo de funcionalidades relativas a la integración con Moodle.
- **Tarea 7.** Mantenimiento de contenedores *Docker* para despliegue.
- **Tarea 8.** Despliegue de la aplicación web.

La distribución temporal de las tareas se refleja en el siguiente diagrama:



4. Recursos humanos

El equipo del proyecto está formado por cinco profesores doctores (Enrique Martín, Manuel Montenegro, Adrián Riesco, Rubén Rubio y Alberto Verdejo) y dos estudiantes (Zakaria El Fakhri y Pedro Martínez). Estos dos últimos son los autores del prototipo inicial [1] de la PWA de la que partimos en este proyecto, y han mantenido su colaboración facilitándonos información sobre la arquitectura del sistema que habían implementado.

Además del equipo del proyecto, los estudiantes Vicentiu Tiberius Roman y José Luis Bartha de las Peñas han trabajado en la implementación de las extensiones de este prototipo, como parte de su trabajo de fin de grado [5].

5. Desarrollo de las actividades

5.1. Estudio del prototipo inicial y elección de tecnologías

La elección de tecnologías a utilizar para el desarrollo de la PWA se ha visto influida, en gran medida, por aquellas que ya se utilizaban en el prototipo inicial. En particular, hemos utilizado los siguientes lenguajes y *frameworks*:

- Lenguaje Python, para el desarrollo de la aplicación web en el *back-end*.
- *Framework* Django, para la aplicación web en el lado del servidor. Este *framework* integra un ORM para la gestión de la base de datos y un panel de administración que utilizamos para añadir los usuarios del sistema.
- JavaScript, como lenguaje de desarrollo en el *front-end*.
- React, como biblioteca de manejo de interfaz basado en componentes.
- Service workers, como herramienta fundamental para el desarrollo de aplicaciones web progresivas. Se encargan de realizar las acciones necesarias cuando el navegador no dispone de conexión a internet.

5.2. Refactorización del código inicial y corrección de errores

Inicialmente se planteó un despliegue y puesta en marcha del prototipo en fase de pruebas antes de desarrollar nuevas funcionalidades. Sin embargo, el prototipo no era lo suficientemente estable como para poder realizar pruebas en el aula. Por este motivo, se hizo necesario repasar exhaustivamente el código fuente y refactorizar algunos componentes de la PWA. Se necesitaron varios meses para llevar a cabo esta tarea. Algunas de los cambios realizados fueron:

- Uso de componentes en React basados en funciones, al contrario que el diseño basado en clases del prototipo, considerado obsoleto en la versión actual de React.
- Actualización de bibliotecas (por ejemplo, Django) a sus últimas versiones.
- Reescritura del enrutador (Router) en el lado del cliente.
- Reescritura de los manejadores de rutas en Django.
- Uso de transacciones en la base de datos, para mantener su integridad en caso de errores.

5.3. Introducción de mejoras de diseño adaptable

El prototipo inicial utilizaba la biblioteca Bootstrap [3] para la interfaz gráfica en el lado del cliente. Esta biblioteca ya permite diseñar una interfaz adaptable a varios dispositivos, por lo que hemos mantenido esta decisión de diseño durante el desarrollo del presente proyecto. No obstante, había ciertos componentes en el prototipo que no se visualizaban correctamente en dispositivos de dimensiones reducidas (como, por ejemplo, teléfonos móviles). Estos componentes involucraban, sobre todo, la realización de cuestionarios y la paginación de las preguntas contenidas en los mismos. Por este motivo, hemos sustituido estos componentes por versiones adaptables, solventando estos problemas de usabilidad.

5.4. Desarrollo de funcionalidades relativas a la creación de cuestionarios

Esta tarea comprendía uno de los objetivos más complejos del proyecto en cuanto a arquitectura. En términos generales, se trataba de que, cuando un profesor elabora un cuestionario, pueda determinar si cada estudiante recibe una variante distinta del cuestionario para dificultar situaciones de copia. Estas variantes pueden generarse en varias dimensiones:

- **Aleatorización de las posibles respuestas en preguntas de opción múltiple.** El profesor puede especificar, para una o varias preguntas de opción múltiple (esto es, aquellas en las que el estudiante debe elegir una opción correcta entre varias propuestas) que se reordenen aleatoriamente las distintas opciones en el momento en el que el estudiante se descargue el cuestionario, de modo que cada estudiante las reciba en un orden distinto. Además, es posible disponer de opciones con posición fija. Esto es útil cuando la pregunta tiene una posible respuesta de tipo “*Ninguna de las anteriores*”, las cuales han de mostrarse siempre en último lugar.
- **Aleatorización de las preguntas de un cuestionario.** En este caso, no se trata de reordenar aleatoriamente las posibles respuestas en una pregunta de opción múltiple, sino se trata de reordenar las distintas preguntas (independientemente de su tipo) dentro de un cuestionario. De este modo, cada estudiante recibe una permutación distinta cuando se descarga el cuestionario.
- **Preguntas aleatorias.** En este caso, estamos hablando de un tipo específico de pregunta dentro de un cuestionario. Al crear una pregunta aleatoria, el profesor selecciona, dentro de un banco de preguntas, qué preguntas pueden aparecer en ese lugar. Cuando el estudiante se descarga esa pregunta aleatoria dentro de un cuestionario, el sistema selecciona aleatoriamente una de las preguntas asociadas.

La principal complejidad técnica que conlleva esta funcionalidad radica en el diseño de la base de datos. Antes de incorporar mecanismos de aleatorización, cada cuestionario se almacenaba en la base de datos con sus correspondientes preguntas. Sin embargo, dado que ahora cada estudiante recibe una variante distinta del cuestionario, es necesario almacenar todas las variantes en la base de datos, para que luego cada estudiante pueda revisar sus respuestas en el mismo orden en el que le aparecieron cuando realizó el cuestionario. Por este motivo, se introdujo en el diseño relacional el concepto de *instancia* de cuestionario, entendiendo como tal una posible variante del mismo. En el anexo 6.1 puede consultarse el modelo entidad-relación resultante.

5.5. Inclusión de formato Markdown, imágenes y fórmulas

Con el fin de que un profesor pueda introducir texto con formato en los enunciados de las preguntas de sus cuestionarios, se ha introducido la posibilidad de expresar los enunciados en formato Markdown [2]. Además, este formato permite la introducción de imágenes y fórmulas matemáticas expresadas en LaTeX. Para poder visualizar correctamente estas tareas se utilizaron las siguientes bibliotecas compatibles con React:

- *react-markdown*: para la visualización de textos escritos en Markdown.
- *remark-math* y *rehype-katex*: para la visualización de fórmulas matemáticas.

Se puso especial cuidado en que las bibliotecas de renderizado de fórmulas fueran autocontenidas, y no hicieran uso de servicios externos alojados en la nube. Esto es importante en una PWA, donde un potencial corte de conexión imposibilita el uso de estos servicios.

Una de las consecuencias de incorporar imágenes a los cuestionarios es el tener que cambiar el mecanismo de almacenamiento local (esto es, proporcionado por el navegador web) utilizado para almacenar las preguntas de un cuestionario cuando un estudiante se las descarga. El almacenamiento estándar utilizado a tal efecto (*localStorage*) tiene la suficiente capacidad como para almacenar el texto de las preguntas y sus opciones, pero no para almacenar imágenes de gran tamaño. Por este motivo, fue necesario utilizar *IndexedDB* como mecanismo alternativo de almacenamiento local.

5.6. Desarrollo de funcionalidades relativas a la integración con Moodle

Se ha implementado un mecanismo de importación de preguntas en formato Moodle XML. De este modo, es posible utilizar en la PWA preguntas provenientes de un banco de preguntas ya existente en Moodle. De momento se admiten solamente los tipos de preguntas soportados en la PWA, que son las preguntas de opción múltiple, y las preguntas de respuesta corta.

Por otro lado, dado que Moodle permite exportar los listados de calificaciones en formato CSV, la PWA permite leer esos listados y modificarlos inyectando las notas de un determinado cuestionario, para así volver a ser importados desde Moodle.

5.7. Mantenimiento de contenedores Docker y despliegue en un servidor

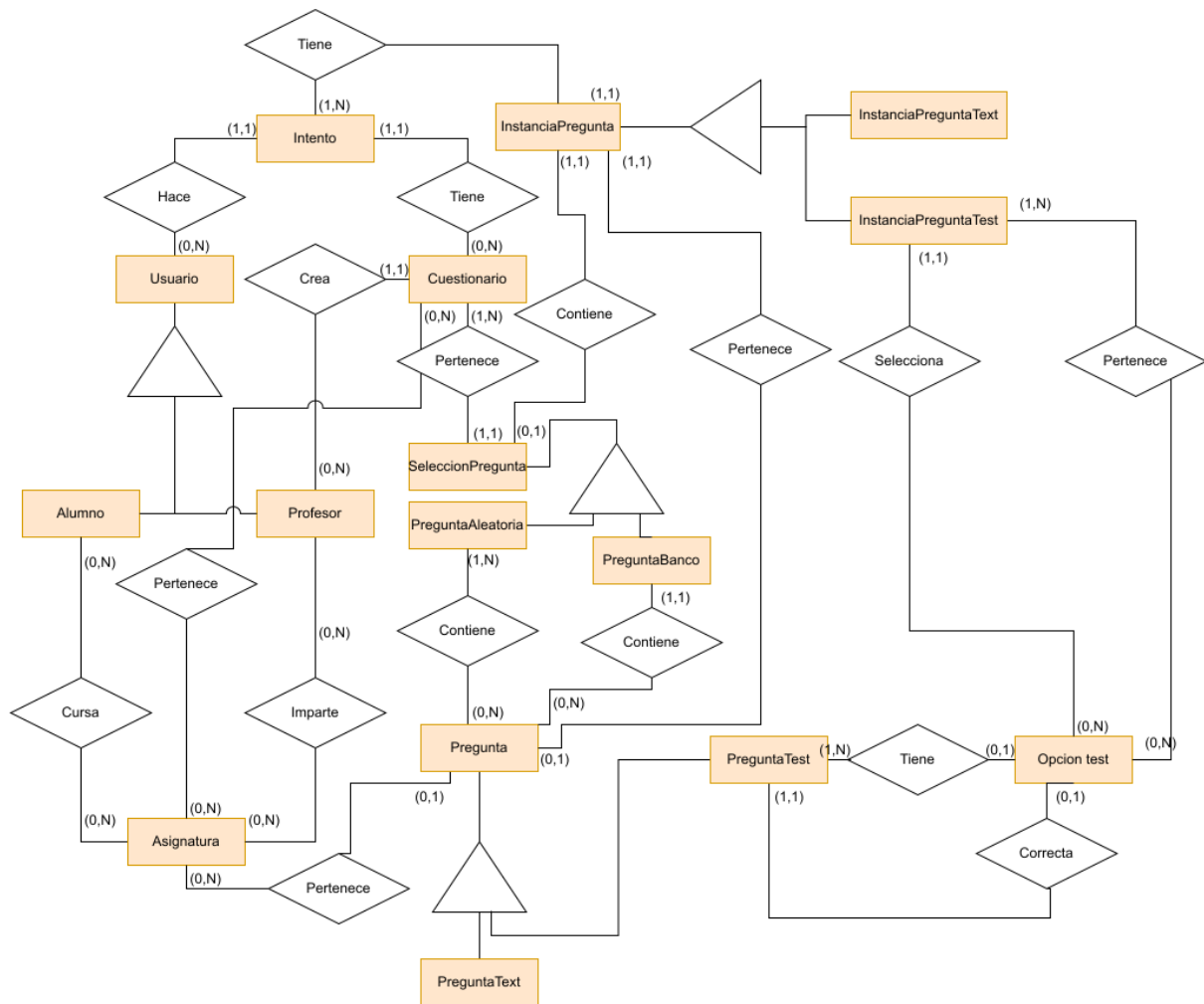
Para facilitar la instalación de la PWA en un servidor, se hizo uso de la herramienta *Docker compose* [4]. Esta herramienta permite definir máquinas virtuales autocontenidas. Estas máquinas virtuales, además, están interconectadas. El listado de máquinas se muestra a continuación:

- `db`: Contiene una instalación del sistema gestor de bases de datos PostgreSQL.
- `pgadmin`: Herramienta de administración de bases de datos PostgreSQL.
- `back`: Código de la aplicación, ejecutándose en el lado del servidor (*back-end*).
- `front`: Código de la aplicación destinado a ejecutarse en el lado del cliente (*front-end*).
- `back-dev` y `front-dev`: Versiones de desarrollo de las dos máquinas anteriores.
- `nginx`: Servidor web que enruta las peticiones HTTP(S) al *front-end* y al *back-end*, según corresponda.

6. Anexo

6.1. Modelo entidad relación

La siguiente figura ha sido tomada del Trabajo de Fin de Grado de Vicentiu Tiberius Roman y José Luis Bartha [5]



6.2. Bibliografía

- [1] Zakaria El Fakhri Ouajih, Pedro Martínez Gamero. *Aplicación web progresiva para la realización de cuestionarios*. Trabajo de Fin de Grado. Facultad de Informática. Universidad Complutense de Madrid (2022).
- [2] John Grüber. *Markdown: syntax*. <https://daringfireball.net/projects/markdown/syntax>
- [3] Jörg Krause. *Introducing Bootstrap 4: Create powerful applications using Bootstrap 4.5*. ISBN 978-1-4842-6202-3. Apress (2020)
- [4] Jeff Nickoloff, Stephen Kuenzli. *Docker in Action, Second Edition*. ISBN 978-1-6172-9476-1. Manning Publications (2019)
- [5] Vicentiu Tiberius Roman, José Luis Bartha de las Peñas. *Integración de Qwizer con Moodle*. Trabajo de Fin de Grado. Facultad de Informática. Universidad Complutense de Madrid (2023).