



UNIVERSIDAD
DE MÁLAGA



ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INFORMÁTICA
DANIEL CRUZADO POVEDA

**AUTOMATIZACIÓN DE CORRECCIÓN DE
PROBLEMAS ACADÉMICOS**

**AUTOMATIZATION OF CORRECTION OF
ACADEMIC PROBLEMS**

Realizado por
Daniel Cruzado Poveda

Tutorizado por
Eduardo Guzmán de los Riscos

Departamento
Lenguajes y Ciencias de la Computación

UNIVERSIDAD DE MÁLAGA
MÁLAGA, JUNIO DE 2020

Fecha defensa: 7 de julio de 2020



UNIVERSIDAD
DE MÁLAGA



Resumen

El objetivo de este proyecto es la realización de una aplicación la cual permita, de una forma automatizada, la corrección de sistemas de ecuaciones compatibles determinados (donde se aceptarán sistemas con metodologías de sustitución, igualación y reducción), teniendo en cuenta no exclusivamente la respuesta final que el alumno haya introducido, sino todo el proceso por el cual ha pasado para dar dichas soluciones. Además, estos sistemas son publicados por profesores, los cuales podrán indicar la metodología esperada a seguir, el número de pasos que el alumno llevará y los enunciados y sistemas de ecuaciones que tendrán que resolver. Para poder realizar este proceso, el alumno habrá de introducir el proceso paso a paso que realice para obtener la solución de los sistemas de ecuaciones que esté resolviendo. El sistema procesará estos pasos e irá corrigiendo los resultados. Finalmente el sistema guardará los datos introducidos, y los resultados finales que el alumno haya obtenido en su corrección, pudiendo este revisarlos en la correspondiente página de detalles de la relación resuelta. También se proporcionará al profesor que haya creado la relación de problemas una lista con los datos de los alumnos que hayan resuelto la relación, las notas obtenidas, y la media total de las notas de los alumnos.

Palabras clave: sistema web de evaluación, corrección automática, Flask, Python

Abstract

The objective of this project is the creation of an application that lets, by automatic process, the correction of determined compatible systems (where it will be accepted systems with resolution by substitution, equation and reduction methods), considering not only the final answer given by the student, but the process made to reach that solution. Furthermore, those equation systems are published by teachers, who will specify the resolution method expected to use, the number of steps to be done and the statements and equation systems to be solved. In order to make the whole process, the student will introduce the process made to obtain the solution of the equation system that he corrects, step by step. The system will analyze the steps given and it will correct the results of these. Eventually the system will save the results of the data input and the final results obtained by the student on its correction, being the student able to check them on the details page of the solved problem relationship. Also it will be given to the teacher who created the relationship a list with the students who solved the relation, their marks and the total mean of those marks.

Keywords: web evaluation system, automatic correction, Flask, Python

Índice

Resumen	1
Abstract	1
Índice	1
Introducción	1
1.1 Motivación	1
1.2 Objetivos	1
1.3 Estructura de la memoria	2
1.3.1- Introducción	2
1.3.2- Tecnologías y Herramientas	2
1.3.3- Estudio del caso.....	2
1.3.4- Metodología y planificación	2
1.3.5- Diseño e Implementación	2
1.3.6- Pruebas sobre la aplicación.....	3
1.3.7- Conclusiones y líneas futuras	3
Tecnologías y herramientas	5
2.1 Tecnologías Front-end	6
2.1.1 HTML.....	6
2.1.2 Javascript	6
2.1.3 Bootstrap	6
2.2 Tecnologías Back-end	7
2.2.1 Flask	7
2.2.2 unittest	8
2.3 Sistemas Gestores de Base de Datos	8
2.3.1 MongoDB.....	8
Análisis y Especificación del Problema	9
3.1 Actores principales	10
3.2 Requisitos funcionales	10
3.2.1 Requisitos comunes a ambos actores	10
Tabla 01 - Requisitos funcionales comunes	11
3.2.2 Requisitos asociados al actor <i>alumno</i>	11
Tabla 02 - Requisitos funcionales alumno	12
3.2.3 Requisitos asociados al actor <i>docente</i>	12
Tabla 03 - Requisitos funcionales docente.....	13
3.3 Requisitos no funcionales	13
Tabla 04 - Requisitos no funcionales	14
3.4 Casos de uso	14
Figura 1: CU 01 - Iniciar de sesión	15
Figura 2: CU 02 - Cerrar de sesión	16
Figura 3: CU 03 - Registro en la aplicación	17
Figura 4: CU 04 - Explorar perfil propio	18

Figura 5: CU 05 - Editar perfil.....	19
Figura 6: CU 06 - Explorar relaciones de problemas.....	20
Figura 7: CU 07 - Búsqueda de relaciones mediante el buscador	21
Figura 8: CU 08 - Filtrado de relaciones de problemas.....	22
Figura 09: CU 09 - Seleccionar una relación de problemas para resolver	23
Figura 10: CU 10 - Completar respuestas de una relación de problemas	23
Figura 11: CU 11 - Evaluar resultados de una relación de problemas.....	24
Figura 12: CU 12 - Explorar relaciones de problemas resueltas	25
Figura 13: CU 13 - Seleccionar relación de problemas resuelta	26
Figura 14: CU 14 - Crear una relación de problemas.....	28
Figura 15: CU 15 - Explorar relaciones de problemas publicada por uno mismo	28
Figura 16: CU 16 - Ver detalles de una relación de problemas publicada.....	29
Figura 17: CU 17 - Eliminar una relación de problemas publicada.....	30
Metodología y planificación	31
4.1 Planificación	32
4.2 Metodología de trabajo	32
Figura 18: Primera mitad del diagrama de Gantt de la planificación	33
Figura 19: Segunda mitad del Diagrama de Gantt de la planificación.....	34
Figura 20: Captura de la herramienta Trello	35
Diseño e implementación	37
5.1 Estructura de la aplicación	38
5.1.1 Diseño arquitectónico de la aplicación.....	38
Figura 21: Estructura general del proyecto	38
5.1.2 Diseño conceptual de la aplicación	39
Figura 22: Boceto de ventana inicio de sesión	39
Figura 23: Boceto de ventana de registro	40
Figura 24: Ventana principal del alumno.....	41
Figura 26: Ventana de relaciones de problemas	42
Figura 27: Ventana de relaciones de problemas y sus filtros	43
Figura 28: Ventana de detalles de una relación de problemas	43
Figura 29: Ventana de resolución de una relación de problemas.....	44
Figura 30: Ventana de una relación ya resuelta	44
Figura 31: Ventana de "Mi perfil" con los datos del usuario.....	45
Figura 32: Ventana de edición de perfil del usuario.....	46
Figura 33: Ventana ideada para editar contraseña	46
Figura 34: Ventana de detalles de una relación seleccionada.	47
5.2. Modelo de clases	47
Figura 35: Modelo de la aplicación.....	48
5.3 Implementación de la aplicación	49
Figura 36: Ventana inicial de la aplicación (Inicio de Sesión)	49
Figura 37: Código HTML de la página de Inicio de Sesión.	50
Figura 38: Código Javascript de la página de Inicio de Sesión.....	51
Figura 39: Código Javascript de ejemplo para modificación de elementos HTML.....	51
Figura 40: Ventana de registro de la aplicación	51
Figura 41: Comprobaciones de datos de registro.....	52
Figura 42: Página de inicio mostrada al alumno.....	53
Figura 43: Página de inicio mostrada a un profesor	53
Figura 44: Ejemplo de página con acceso no autorizado por rol.....	54
Figura 45: Página de edición de perfil de usuario	54
Figura 46: Página de relaciones de problemas.....	55
Figura 47: Página de detalles de problema no resuelto	56

Figura 48: Página de detalles de problema resuelto	56
Figura 49: Página de detalles de un profesor	57
Figura 50: Página de relaciones resueltas por un alumno	57
Figura 51: Página de resolución de problemas	58
Figura 52: Página de relaciones de problemas publicadas por profesor	58
Figura 53: Página de publicación de problemas.....	59
Pruebas sobre la aplicación.....	61
6.1 Tipos de prueba	62
6.2 Estructura de pruebas unitarias	62
6.3 Usando unittest	63
6.4 Ejemplo de pruebas unitarias	63
Figura 54: Prueba para un Inicio de sesión correcto	64
Figura 55: Método de inicio de sesión, marcándose la respuesta esperada	64
Figura 56: Resultado correcto de la ejecución de la prueba	64
Conclusiones y líneas futuras	65
7.1 Objetivos y resultados.....	66
7.2 Problemas encontrados.....	66
7.3 Posibles mejoras y futuro del proyecto	67
7.3.1 Posibles mejoras.....	67
7.3.2 Futuro del proyecto.....	68
Referencias	69
Manual de usuario	71
A.1 Introducción:	71
A.2 Organización:	71
Figura 57: Inicio de sesión	72
Figura 58: Registro en la aplicación.....	72
Figura 59: Comprobación de datos en el registro	73
Figura 60: Página de inicio de un alumno.....	74
Figura 61: Página de inicio de un profesor	74
Figura 62: Página de perfil personal	75
Figura 63: Página de edición de perfil	75
Figura 64: Filtrado de relaciones por selector	76
Figura 65: Filtrado de relaciones por buscador	76
Figura 66: Relaciones de problemas.....	76
Figura 67: Filtro y botón eliminar de "Mis relaciones publicadas"	77
Figura 68: Página de "Mis relaciones publicadas"	77
Figura 69: Página de publicación de relaciones.....	78
Figura 70: Datos a rellenar en la relación a publicar	79
Figura 71: Información sobre los pasos.....	79
Figura 72: Creación del sistema de ecuaciones en la relación	80
Figura 73: Página de detalles de relación de un profesor	81
Figura 74: Página de detalles de alumno (relación no resuelta)	82
Figura 75: Página de detalles de alumno (relación resuelta)	83
Figura 76: Página de resolución de relación de problemas.....	84
Figura 77: Pasos posibles para método de sustitución	84
Figura 78: Pasos posibles para método de reducción	84
Figura 79: Pasos posibles para método de igualación.....	84
Figura 80: Página de problemas resueltos por alumno.....	85
Figura 82: Transformar ecuación.....	85

Figura 83: Sustitución incógnita por ecuación.....	86
Figura 84: Fallo de paso de sustitución de incógnita sustituida por ecuación	86
Figura 86: Sustitución de incógnita resuelta en una ecuación	87
Figura 87: Resolución de incógnita.....	87
Figura 88: Operación de dos ecuaciones.....	88
Figura 89. Igualación de dos ecuaciones	88
Manual de Instalación.....	89
Requerimientos:.....	89
Figura 90: Conexión base de datos.....	89
Figura 91: Conectar clúster en la aplicación	90
Figura 92: Conectar a nuestra aplicación	90
Figura 94: Selección de controlador y strin de conexión creado.	91
Figura 95: Ejecución del código con el comando python.	91

1

Introducción

1.1 Motivación

Cada día nos encontramos ante mayores innovaciones a la hora de hacer uso de herramientas educativas. Muchas de ellas ofrecen métodos automatizados para resolver los ejercicios o relaciones de problemas que los docentes proponen al estudiante. No obstante, estas herramientas presentan un problema; no se tiene siempre presente cuál ha sido el proceso que el alumno ha podido llevar a cabo para resolver el problema, dejando la corrección únicamente que la respuesta sea válida o no. Esto provoca una situación injusta a la hora de evaluar al estudiante.

Con esto presente surge la idea de crear una aplicación de corrección de ejercicios (en este caso de sistemas de ecuaciones) la cual evalúe tanto el proceso de resolución que el alumno ha presentado como la respuesta final.

1.2 Objetivos

Este proyecto tiene como objetivo crear una aplicación web que permita de forma automatizada corregir problemas de sistemas de ecuaciones, donde no sólo se tendrá en cuenta la respuesta final del alumno, sino también el proceso que se ha llevado a cabo para la resolución para resolver los mismos. Se pretende con esto una corrección más justa que la que provee la mayoría de correctores automáticos los cuales aceptan exclusivamente una respuesta única que sea válida.

Dentro de la aplicación se propone funcionalidades como que el alumno pueda hacer relaciones de problemas publicadas por los docentes, filtrarlas mediante distintos

criterios y resolverlas. Además, los alumnos contarán con un perfil en el cual podrán ver su información personal y una página para ver qué relaciones de problemas han resuelto ya, junto a su nota evaluada.

Por parte del docente, este podrá publicar relaciones de problemas personalizadas. También contará con una página de perfil con su información personal, una página con las relaciones que ha publicado, donde podrá ver qué alumnos resolvieron la relación, y sus notas respectivas.

1.3 Estructura de la memoria

Esta memoria se encuentra estructurada con los siguientes capítulos:

1.3.1- Introducción

En este capítulo se muestran los aspectos básicos tales como la motivación para llevar a cabo el desarrollo del sistema propuesto, los objetivos que se proponen con la aplicación, y la estructura que compondrá la memoria.

1.3.2- Tecnologías y Herramientas

En el capítulo de *Tecnologías y Herramientas* se discuten qué tecnologías y herramientas se van a utilizar en el desarrollo del software, junto a una descripción de las mismas y su uso principal.

1.3.3- Estudio del caso

Este capítulo expondrá los distintos **requisitos funcionales, no funcionales** y **casos de uso** analizados en el proyecto que definirán el cómo se debería comportar la aplicación y qué límites alcanzará.

1.3.4- Metodología y planificación

En este capítulo se expone la metodología usada durante el desarrollo de la aplicación, qué técnicas de la Ingeniería del Software se han aplicado, y la planificación que se ha seguido para la correcta ejecución de las tareas necesarias para llevar el proyecto adelante.

1.3.5- Diseño e Implementación

Este capítulo muestra el diseño que se ha realizado para la aplicación web, su estructura general, la implementación de la misma, junto a las pertinentes

explicaciones sobre cómo funciona y se comporta, y el porqué de las decisiones tomadas a la hora de aplicarlas en el desarrollo.

1.3.6- Pruebas sobre la aplicación

El capítulo donde se tratará el tipo de pruebas que se han realizado para la detección de errores que en un principio podrían pasar desapercibido, y los resultados obtenidos.

1.3.7- Conclusiones y líneas futuras

Este capítulo final discute los aspectos y objetivos que se han podido alcanzar con la aplicación, qué conclusiones se han sacado, qué mejoras se podrían haber hecho, y reflexiones sobre el proyecto y su futuro uso.

2

Tecnologías y herramientas

Este capítulo recoge las tecnologías de las que se ha hecho uso en este proyecto para su desarrollo. Se han clasificado respecto al área de trabajo a la que han pertenecido, que en este caso han sido **Front-end**, **Back-end** y **Sistema Gestor de Base de Datos**. También se hablará de los **frameworks** usados para cada área.

2.1 Tecnologías Front-end

Las tecnologías de Front-end son las destinadas a mostrarse en la capa de presentación. El Front-end compone el punto de con el que la aplicación se informará con el usuario.

Entre estas tecnologías se escogieron:



2.1.1 HTML

HTML es fundamental para el desarrollo de cualquier página web, y en nuestro caso, para el desarrollo de la aplicación web. Será la tecnología base que se usará para mostrar la información que se genere para el usuario.



2.1.2 Javascript

Javascript es un lenguaje de programación interpretado, autodefinido como lenguaje orientado a objetos, que es débilmente tipado y dinámico. Prácticamente todos los navegadores modernos muestran compatibilidad con Javascript. Además de ello, su simplicidad y rápida producción, su enorme cantidad de información y comunidad y su amplia capacidad de posibilidades a la hora de trabajar junto a **HTML** ha hecho que Javascript sea el candidato como lenguaje interpretado para usarse por parte del cliente.



2.1.3 Bootstrap

Bootstrap es un framework de CSS orientado al desarrollo con **HTML**, **Javascript** y **CSS**. Bootstrap ofrece una serie de plantillas que se pueden insertar en nuestro código HTML

por medio de etiquetas específicas. Se ha elegido Bootstrap antes que CSS puro debido a su sencillez a comparación de usar exclusivamente este. Además proporciona resultados mucho más satisfactorios y agradables visualmente que haciendo uso de únicamente HTML en bruto o CSS, sin contar la gran cantidad de información presente del framework, la comunidad que hay detrás, y la gran cantidad de proyectos que hacen uso del mismo.

2.2 Tecnologías Back-end



2.2.1 Flask

Flask es un framework minimalista basado en **Jinja2** que permite la creación de páginas web de forma sencilla. Pensamos que Flask es una buena elección para este proyecto, debido a su simplicidad a la hora de desarrollar (pues su curva de aprendizaje es de lo más sencilla) y su flexibilidad (no se limita a webs básicas, Flask permite mediante ampliaciones crear aplicaciones complejas). Si bien al ser un framework minimalista no presenta un modelo de diseño arquitectónico como el Modelo-Vista-Controlador, que suele ser el más común en ciertos frameworks. Sin embargo, esto no supone un problema, pues cuenta con los conectores necesarios para el acceso a los datos de forma que no se tenga que realizar un modelo como tal en el proyecto. Todo esto le da un valor añadido a la idea de proporcionar simplicidad, y no tener que trabajar con proyectos poco legibles como podría ser con otros frameworks como Django, o en la vertiente de **Java**, proyectos que hagan uso de **JSF** o **JPA**.

Mencionar que la conexión a nuestra base de datos se hará por medio del conector **pymongo**, que no requiere las creaciones de modelos algunos.



2.2.2 unittest

unittest es un framework para la realización de pruebas en Python. Similar a JUnit, permite la automatización de pruebas, agregar pruebas a colecciones, separación de pruebas en el framework, realización de pruebas unitarias, entre otros.

2.3 Sistemas Gestores de Base de Datos



2.3.1 MongoDB

MongoDB es una base de datos no relacional, noSQL, orientada a documentos. La elección de MongoDB se debe al sencillo hecho de la necesidad de un sistema de base de datos el cual proporcione una escalabilidad y rendimiento de alto nivel. Si bien hay sistemas de bases de datos SQL que también proporcionan estas características, su funcionamiento no es el más adecuado para la orientación de este proyecto, que es el de una aplicación web. En proyectos pequeños podría servir, pero en caso de que necesitáramos ampliar los datos a manejar, una base de datos relacional se vería muy afectada ante ello, y no proporcionaría los tiempos de respuesta adecuados que buscamos en la base de datos. Es por ello que necesitamos una base de datos no relacional.

3

Análisis y Especificación del Problema

Este capítulo expone cuáles son las funcionalidades que ofrecerá la aplicación haciendo uso de un análisis respecto a los **requisitos funcionales, no funcionales** y los **casos de uso** que se han observado durante el desarrollo de la misma.

3.1 Actores principales

Esta sección muestra cuáles son los actores principales que interactúan en el sistema. Se puede observar dos actores principales:

Alumno: Podrá resolver distintas relaciones de problemas publicadas, explorar las mismas, ver la información de su perfil personal, y ver qué calificaciones ha obtenido en las relaciones resueltas.

Docentes: Los docentes/profesores podrán publicar relaciones de problemas, explorar las relaciones publicadas, ver su información personal de perfil y ver qué alumnos han resuelto sus relaciones junto a las calificaciones que han obtenido los mismos.

3.2 Requisitos funcionales

Los requisitos funcionales recogerán los aspectos básicos sobre qué acciones podrán realizar los actores del sistema sobre el mismo. Para ello se van a clasificar los requisitos respecto a los propios actores y frente a otros criterios como el que ambos actores puedan realizar las mismas acciones en el sistema.

Los requisitos funcionales se mostrarán en sus respectivas tablas de cada sección.

3.2.1 Requisitos comunes a ambos actores

En estos requisitos se refieren a los actores *alumno* y *docente* como *usuario*.

Requisito	Título	Descripción
RF 01	Iniciar sesión	Un usuario puede iniciar sesión indicando su nombre de usuario y su contraseña.
RF 02	Cerrar sesión	Un usuario puede cerrar sesión pulsando el botón "Cerrar sesión".
RF 03	Registro en la aplicación	Un usuario puede crear una nueva cuenta de usuario indicando en los campos de

		registro su nombre de usuario, nombre real, su contraseña de usuario, un email y un rol de usuario (docente o alumno).
RF 04	Explorar perfil propio	Un usuario puede ver los datos de su perfil pulsando el botón de "Mi perfil" en el menú principal.
RF 05	Editar perfil	Un usuario puede editar los campos de su perfil pulsando el botón "Editar perfil", y confirmar los cambios realizados pulsando el botón "Confirmar cambios".
RF 06	Cambiar la contraseña de usuario	El usuario puede editar la contraseña de su perfil pulsando "Cambiar contraseña"

Tabla 01 - Requisitos funcionales comunes

3.2.2 Requisitos asociados al actor alumno

Requisito	Título	Descripción
RF 07	Explorar relaciones de problemas	Un alumno puede inspeccionar las distintas relaciones de problemas pulsando el botón "Relaciones de problemas" en el menú principal.
RF 08	Búsqueda de relaciones de mediante el buscador	Un alumno puede buscar una relación de problemas indicando el nombre de la misma o el del autor en el buscador.
RF 09	Filtrado de relaciones de problemas	Un alumno puede filtrar las relaciones de problemas mostradas mediante los distintos filtros configurables en el desplegable (nombre de la relación ascendente/descendente, nombre de autor ascendente/descendente y fecha de publicación más antigua/ más reciente).
RF 10	Seleccionar una relación de problemas para resolver	Un alumno puede seleccionar la relación de problemas que quiera resolver y ver sus detalles pulsando sobre su título.
RF 11	Completar respuestas de	Un alumno puede completar una relación de problemas

	una relación de problemas	seleccionada pulsando el botón "Comenzar relación".
RF 12	Evaluar resultados de una relación de problemas	Un alumno puede evaluar sus respuestas a los problemas de la relación pulsando el botón "Evaluar respuestas".
RF 13	Explorar relaciones de problemas resueltas	Un alumno puede ver las relaciones de problemas que ya ha resuelto pulsando el botón "Relaciones de problemas resueltas" en la página de "Mi perfil".
RF 14	Seleccionar relación de problemas resuelta	Un alumno puede seleccionar y ver con más detalle una relación de problemas ya resuelta pulsando el botón "Ver relación de problemas".

Tabla 02 - Requisitos funcionales alumno

3.2.3 Requisitos asociados al actor docente

Requisito	Título	Descripción
RF 15	Crear una nueva relación de problemas	Un docente puede crear una nueva relación de problemas pulsando el botón "Crear nueva relación de problemas" en el Menú principal.
RF 16	Explorar relaciones de problemas publicadas por uno mismo	Un docente puede explorar las relaciones de problemas que haya publicado pulsando el botón "Mis Relaciones de Problemas" en el menú principal.
RF 17	Ver detalles de una relación de problemas publicada	Un docente puede ver los detalles de una relación de problemas que haya publicado pulsando en el título de la misma.
RF 18	Eliminar una relación de problemas	Un docente puede eliminar una relación de problemas que haya publicado pulsando el botón "Eliminar relación".
RF 19	Explorar las relaciones de problemas publicadas	Un docente puede inspeccionar las distintas relaciones de problemas pulsando el botón "Relaciones de problemas" en el menú principal.

RF 20	Búsqueda de relaciones de problemas mediante nombre de relación	Un docente puede buscar una relación de problemas indicando el nombre de la misma en el buscador.
RF 21	Búsqueda de relaciones de problemas mediante nombre de autor	Un docente puede buscar una relación de problemas indicando el nombre del autor en el buscador.
RF 22	Filtrado de relaciones de problemas	Un docente puede filtrar las relaciones de problemas mostradas mediante los distintos filtros mostrados en el desplegable (nombre de la relación ascendente/descendente, nombre de autor ascendente/descendente y fecha de publicación más antigua/ más reciente).

Tabla 03 - Requisitos funcionales docente

3.3 Requisitos no funcionales

Los requisitos no funcionales recogen aspectos que no están relacionados con los actores, si no que son relativos al sistema, tales como rendimiento, tiempos de respuesta, seguridad del sistema, escalabilidad, entre otros.

Requisito	Título	Descripción
RNF 01	Usabilidad del sistema	El usuario debe poder utilizar el sistema de forma intuitiva, y debe poder acceder a las tareas que quiera en no más de 10 clics.
RNF 02	Optimización del tiempo de respuesta	El sistema debe ser funcionalmente óptimo, sin tener tiempos de respuesta demasiado extensos, tomando como máximo 5 segundos en dar un tiempo de respuesta.
RNF 03	Interfaz cómoda e intuitiva	La interfaz de usuario ha de ser agradable para el usuario, sin causar problemas de uso, siendo intuitiva en su disposición de elementos.

RNF 04	Mantenimiento	El sistema debe estar desarrollado de forma que su mantenimiento sea fácil de realizar.
RNF 05	Integración	El sistema podrá integrarse correctamente en el servidor de despliegue en línea que se ofrezca
RNF 06	Escalabilidad	El sistema debe estar desarrollado de forma que se permita realizar ampliaciones en la misma sin ocupar un esfuerzo excesivo.

Tabla 04 - Requisitos no funcionales

3.4 Casos de uso

En este apartado se exponen los casos de uso, que muestran cómo un actor interactúa con el sistema, modelando las entradas y salidas esperadas. Sólo los casos de uso más relevantes o de mayor complejidad vendrán acompañados de un diagrama de flujo.

Título	Iniciar sesión
Descripción	El usuario puede iniciar sesión dentro de la aplicación
Pre-condición	El usuario no debe haber iniciado sesión. El usuario ha de tener una cuenta en el sistema.
Post-condición	Garantía de éxito: -El usuario accede a la aplicación con sesión iniciada Garantía de fallo: -El usuario no inicia la sesión
Prioridad	Alta
Escenario principal	
<ol style="list-style-type: none"> 1. El usuario entra en la aplicación 2. El sistema muestra los campos por completar para acceder 3. El usuario introduce su nombre de usuario y su contraseña y pulsa "Iniciar sesión" 4. El sistema valida las credenciales y redirige al usuario al menú principal 	

Escenario alternativo

- 3.b El usuario introduce erróneamente su nombre de usuario o contraseña
 - 3.b.1 El sistema informa que uno de los campos introducidos es erróneo
- 3.c El usuario cancela el inicio de sesión
 - 3.c.1 El usuario sigue en la misma ventana de la aplicación

Diagrama de secuencia

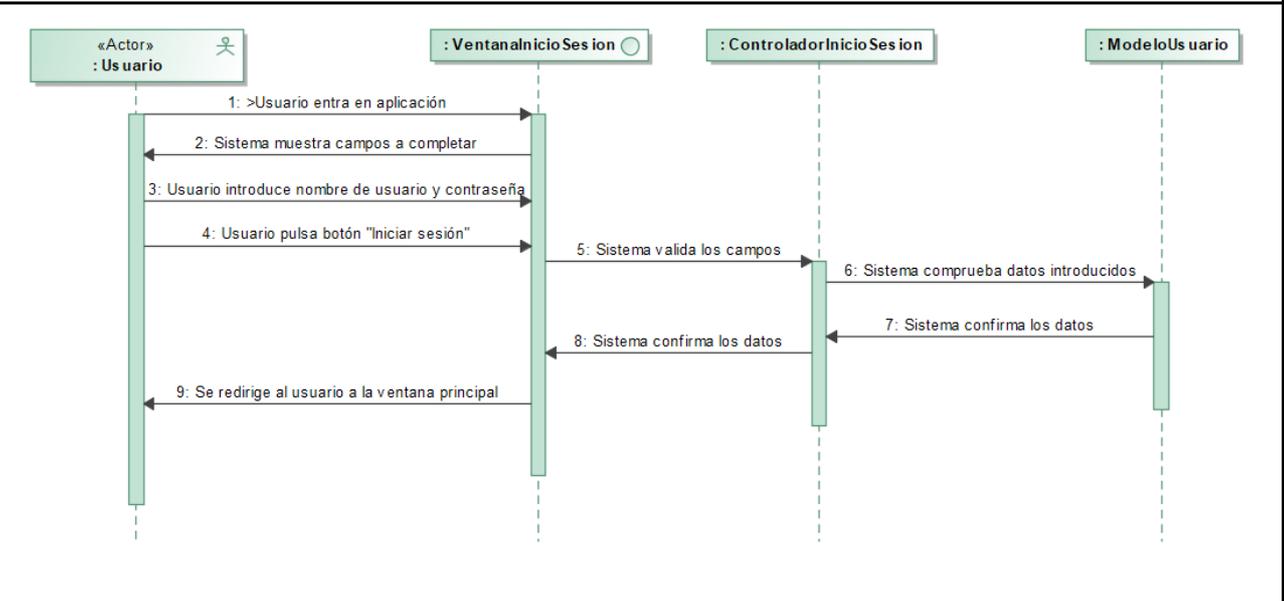


Figura 1: CU 01 - Iniciar de sesión

Título	Cerrar sesión
Descripción	El usuario puede cerrar la sesión dentro de la aplicación
Pre-condición	El usuario debe haber iniciado sesión. El usuario ha de tener una cuenta en el sistema
Post-condición	Garantía de éxito: -El usuario abandona la aplicación cerrando sesión Garantía de fallo: -El usuario mantiene la sesión
Prioridad	Alta
Escenario principal	
<ol style="list-style-type: none"> 1. El usuario pulsa el botón "Cerrar sesión" 2. El sistema cierra la sesión y redirige al usuario a la página de inicio de sesión 	

Escenario alternativo

- 1.b El usuario no pulsa el botón "Cerrar sesión"
- 1.b.1 El usuario se mantiene en la ventana actual
- 3.c El usuario cancela el inicio de sesión
- 3.c.1 El usuario sigue en la misma ventana de la aplicación

Diagrama de secuencia

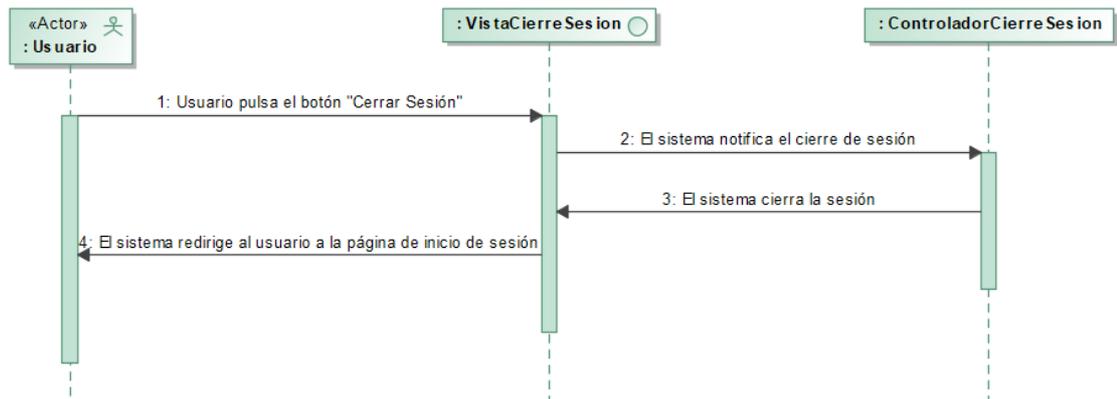


Figura 2: CU 02 - Cerrar de sesión

Título	Registrar usuario
Descripción	El usuario puede crear una cuenta dentro de la aplicación
Pre-condición	El usuario debe haber entrado en la página de registro
Post-condición	Garantía de éxito: -El usuario crea una nueva cuenta Garantía de fallo: -El usuario no crea una nueva cuenta
Prioridad	Alta

Escenario principal

- 1. El usuario rellena los campos "Nombre de usuario", "Nombre completo", "Rol", "Correo electrónico" y "Contraseña"
- 2. El sistema comprueba que los campos obligatorios están rellenos
- 3. El usuario pulsa el botón "Registrarme"
- 4. El sistema crea un nuevo usuario y redirige a la página de inicio de sesión

Escenario alternativo

- 1.b El usuario no rellena todos los campos obligatorios
 - 1.b.1 El sistema indica que faltan campos por completar
- 3.c El usuario cierra la ventana
 - 3.c.1 El sistema no crea el usuario

Diagrama de secuencia

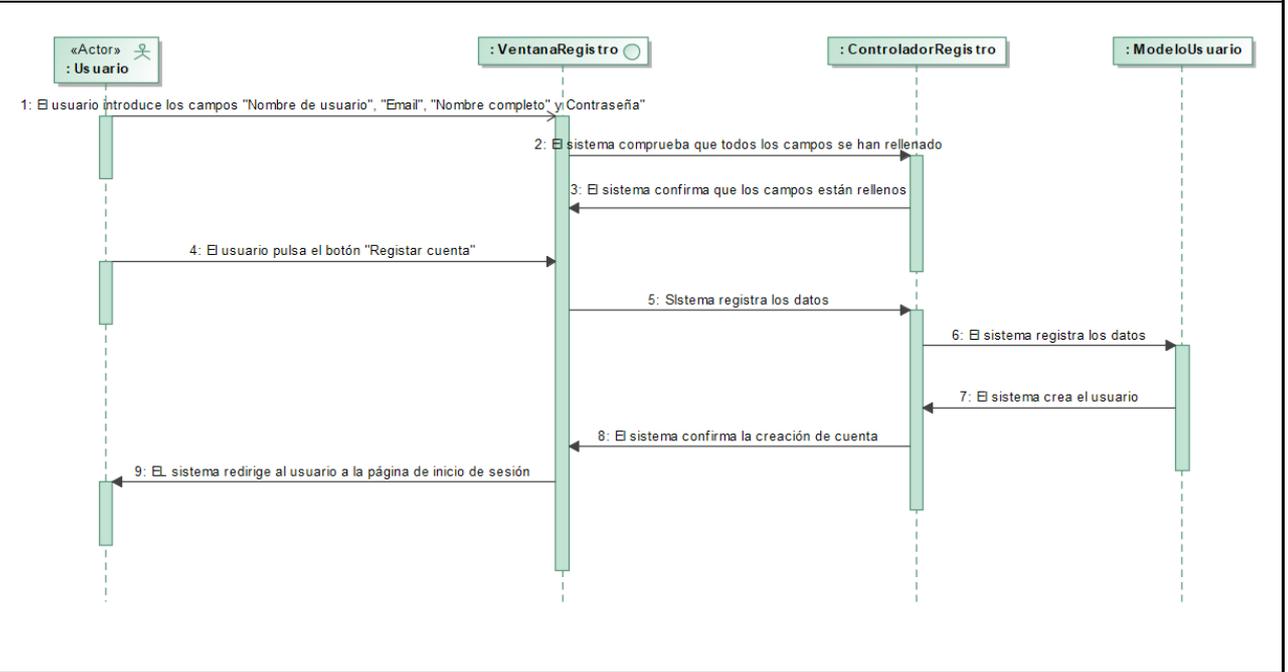


Figura 3: CU 03 - Registro en la aplicación

Título	Explorar perfil propio
Descripción	El usuario puede ver los datos de su propio perfil
Pre-condición	El usuario debe haber iniciado sesión. El usuario ha de tener acceso al botón de "Mi perfil"
Post-condición	Garantía de éxito: -El usuario accede a los datos de perfil con éxito Garantía de fallo: -El usuario se mantiene en la página actual
Prioridad	Alta
Escenario principal	
<ul style="list-style-type: none"> 1. El usuario pulsa el botón "Mi perfil" 2. El sistema redirige al usuario a su perfil de usuario 	

Escenario alternativo
1.b El usuario cierra la ventana 1.b.1 El sistema cierra la aplicación 1.c El usuario cancela el visionado del perfil 1.c.1 El usuario sigue en la misma ventana de la aplicación

Figura 4: CU 04 - Explorar perfil propio

Título	Editar perfil propio
Descripción	El usuario puede editar los datos de su perfil
Pre-condición	El usuario debe haber iniciado sesión. El usuario debe estar en la página de perfil
Post-condición	Garantía de éxito: -El usuario edita los datos de su perfil Garantía de fallo: -El usuario mantiene los datos actuales de su perfil
Prioridad	Media
Escenario principal	
1. El usuario pulsa el botón "Editar mi perfil" 2. El sistema muestra los campos editables del perfil 3. El usuario introduce los campos que quiera editar, que pueden ser "Nombre de usuario", "Contraseña" o "Email" y pulsa "Confirmar cambios". 4. El sistema comprueba los datos y edita el perfil.	
Escenario alternativo	
3.b El usuario deja campos obligatorios en blanco 3.b.1 El sistema indica que hay campos por completar 3.c El usuario cancela la edición de perfil 3.c.1 El usuario sigue en la misma ventana de la aplicación	
Diagrama de secuencia	

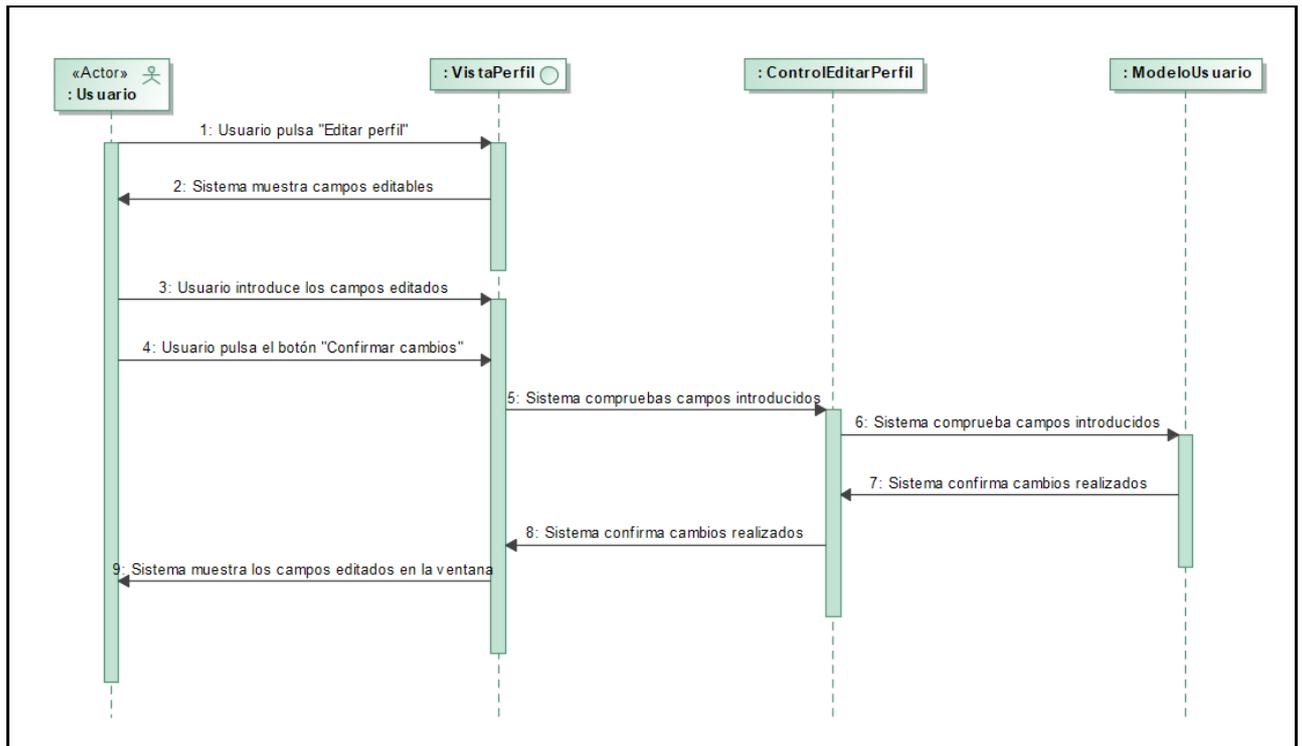


Figura 5: CU 05 - Editar perfil

Título	Explorar relaciones de problemas
Descripción	El alumno puede ver las relaciones de problemas publicadas por los docentes
Pre-condición	El alumno debe haber iniciado sesión. El alumno ha de tener acceso al botón de "Relaciones publicadas"
Post-condición	Garantía de éxito: -El alumno accede a la ventana con las relaciones de problemas Garantía de fallo: -El alumno se mantiene en la página actual
Prioridad	Alta
Escenario principal	
<ol style="list-style-type: none"> 1. El alumno pulsa el botón "Relaciones publicadas" 2. El sistema redirige al usuario a la ventana de relaciones de problemas 	
Escenario alternativo	
<ol style="list-style-type: none"> 1.b El alumno cierra la ventana <ol style="list-style-type: none"> 1.b.1 El sistema cierra la aplicación 1.c El alumno cancela el visionado de las relaciones <ol style="list-style-type: none"> 1.c.1 El alumno sigue en la misma ventana de la aplicación 	

Figura 6: CU 06 - Explorar relaciones de problemas

Título	Búsqueda de relaciones mediante el buscador
Descripción	El alumno puede buscar relaciones de problemas concretas introduciendo los campos en el buscador
Pre-condición	El alumno debe haber iniciado sesión. El alumno debe encontrarse en la página de "Relaciones publicadas"
Post-condición	Garantía de éxito: -El alumno filtra los resultados con los criterios aplicados Garantía de fallo: -Las relaciones de problemas se siguen mostrando
Prioridad	Alta
Escenario principal	
<ol style="list-style-type: none"> 1. El alumno pulsa en el campo de texto del buscador. 2. El sistema espera recibir la entrada. 3. El alumno introduce el nombre de la relación o el nombre del autor de la relación para filtrar y pulsa el botón "Buscar". 4. El sistema filtra las relaciones y las muestra en pantalla. 	
Escenario alternativo	
<ol style="list-style-type: none"> 1.b El alumno cierra la ventana <ol style="list-style-type: none"> 1.b.1 El sistema cierra la aplicación 1.c El alumno cancela la búsqueda <ol style="list-style-type: none"> 1.c.1 El sistema sigue mostrando las mismas relaciones de problemas. 	
Diagrama de secuencia	

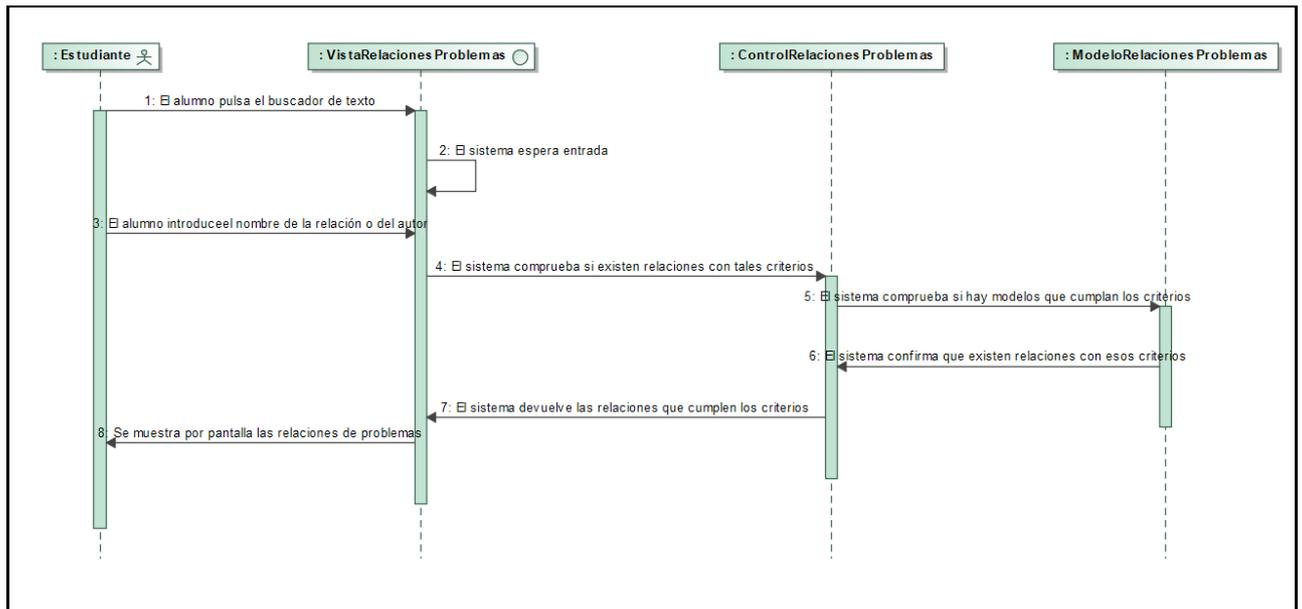


Figura 7: CU 07 - Búsqueda de relaciones mediante el buscador

Título	Filtrado de relaciones de problemas
Descripción	El alumno puede filtrar las relaciones de problemas que se muestren por pantalla seleccionando uno de los filtros de la lista desplegable.
Pre-condición	El alumno debe haber iniciado sesión. El alumno debe encontrarse en la página de "Relaciones publicadas"
Post-condición	Garantía de éxito: -El alumno filtra los resultados con los criterios aplicados Garantía de fallo: -Las relaciones de problemas se mantienen
Prioridad	Baja
Escenario principal	<ol style="list-style-type: none"> 1. El alumno pulsa el botón de "Seleccionar filtro" 2. El sistema despliega la lista de filtros existentes. 3. El alumno selecciona uno de los filtros posibles (nombre de la relación ascendente/descendente, nombre de autor ascendente/descendente y fecha de publicación más antigua/más reciente) y pulsa "Filtrar" 4. El sistema muestra las relaciones de problemas conforme los filtros seleccionados.
Escenario alternativo	

- 1.b El alumno cierra la ventana
 - 1.b.1 El sistema cierra la aplicación
- 1.c El alumno cancela el filtrado
 - 1.c.1 El sistema sigue mostrando las mismas relaciones de problemas.

Diagrama de secuencia

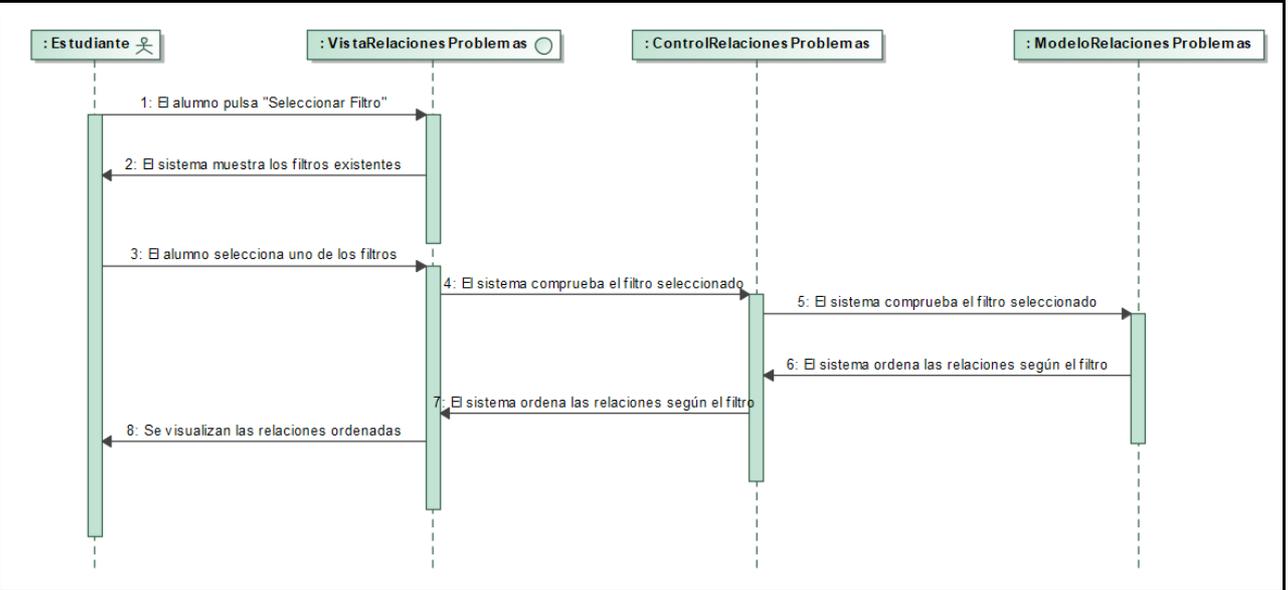


Figura 8: CU 08 - Filtrado de relaciones de problemas

Título	Seleccionar una relación de problemas para resolver
Descripción	El alumno puede seleccionar la relación de problemas que quiera resolver y ver los detalles de la misma
Pre-condición	El alumno debe haber iniciado sesión. El alumno debe encontrarse en la página de "Relaciones publicadas"
Post-condición	Garantía de éxito: -El alumno accede a los detalles de la relación de problemas seleccionada Garantía de fallo: -El alumno se mantiene en la página
Prioridad	Alta
Escenario principal	
<ol style="list-style-type: none"> 1. El alumno pulsa el título de la relación que quiera resolver. 2. El sistema redirige a una ventana con los datos de la relación seleccionada. 	

Escenario alternativo
1.b El alumno cierra la ventana 1.b.1 El sistema cierra la aplicación 1.c El alumno cancela la selección 1.c.1 El alumno se mantiene en la misma ventana

Figura 09: CU 09 - Seleccionar una relación de problemas para resolver

Título	Completar respuestas de una relación de problemas
Descripción	El alumno puede completar una relación de problemas pulsando el botón "Comenzar relación"
Pre-condición	El alumno debe haber iniciado sesión. El alumno debe encontrarse en la página de la relación seleccionada
Post-condición	Garantía de éxito: -El alumno accede a la relación de problemas que quiera resolver Garantía de fallo: -El alumno se mantiene en la página
Prioridad	Alta
Escenario principal	
1. El alumno pulsa el botón "Comenzar a resolver" 2. El sistema redirige al usuario a la ventana con las preguntas de la relación.	
Escenario alternativo	
1.b El alumno cierra la ventana 1.b.1 El sistema cierra la aplicación 1.c El alumno cancela la selección 1.c.1 El alumno se mantiene en la misma ventana	

Figura 10: CU 10 - Completar respuestas de una relación de problemas

Título	Evaluar resultados de una relación de problemas
Descripción	El alumno puede evaluar los datos introducidos en la relación pulsando el botón "Evaluar respuestas"

Pre-condición	El alumno debe haber iniciado sesión. El alumno debe encontrarse en la página de la relación seleccionada
Post-condición	Garantía de éxito: -El alumno obtiene los resultados de la relación de problemas Garantía de fallo: -El alumno se mantiene en la página
Prioridad	Alta

Escenario principal

1. El alumno introduce los campos obligatorios para resolver la relación y pulsa "Evaluar respuestas"
2. El sistema abre una ventana emergente preguntando al usuario si confirma la acción
3. El usuario selecciona "Confirmar"
4. El sistema evalúa los resultados y devuelve la evaluación.

Escenario alternativo

- 1.b El alumno cierra la ventana
 - 1.b.1 El sistema cierra la aplicación
- 3.c El alumno cancela la evaluación
 - 3.c.1 El alumno se mantiene en la misma ventana

Diagrama de secuencia

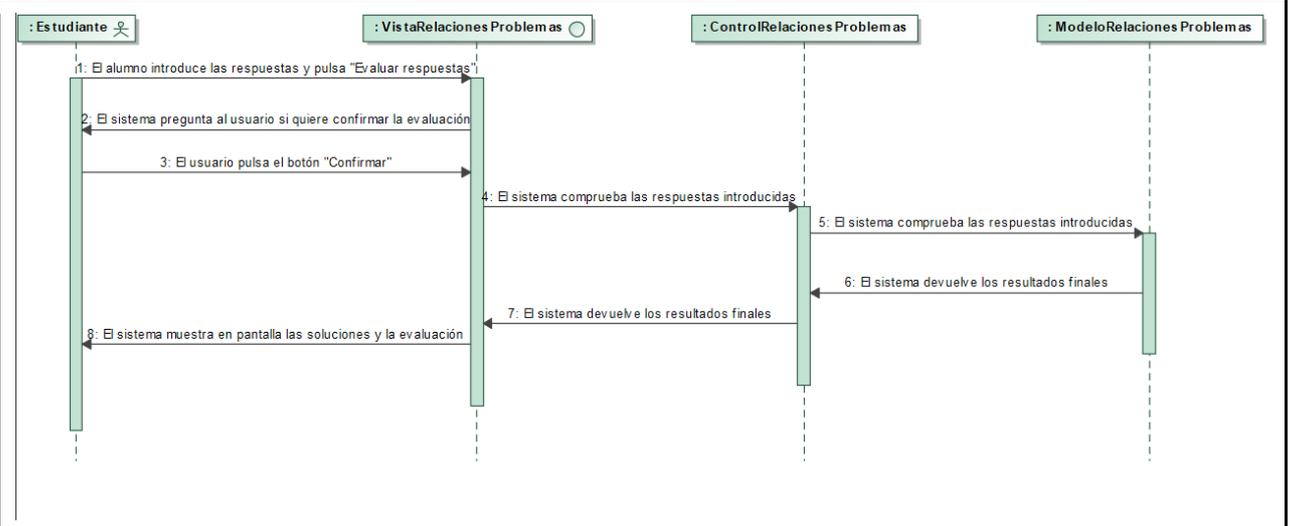


Figura 11: CU 11 - Evaluar resultados de una relación de problemas

Título	Explorar relaciones de problemas resueltas
---------------	--

Descripción	El alumno puede ver las relaciones de problemas que ya haya resuelto
Pre-condición	El alumno debe haber iniciado sesión. El alumno debe poder acceder al botón "Relaciones ya resueltas"
Post-condición	Garantía de éxito: -El alumno accede a las relaciones de problemas que ya haya resuelto Garantía de fallo: -El alumno se mantiene en la página actual
Prioridad	Alta
Escenario principal	
<ol style="list-style-type: none"> 1. El pulsa el botón "Relaciones ya resueltas" del menú principal. 2. El sistema redirige al usuario a la ventana con las relaciones que ya ha resuelto. 	
Escenario alternativo	
<ol style="list-style-type: none"> 1.b El alumno cierra la ventana <ol style="list-style-type: none"> 1.b.1 El sistema cierra la aplicación 1.c El alumno cancela el acceso a las relaciones resueltas <ol style="list-style-type: none"> 1.c.1 El alumno se mantiene en la misma ventana 	

Figura 12: CU 12 - Explorar relaciones de problemas resueltas

Título	Seleccionar relación de problemas resuelta
Descripción	Un alumno puede seleccionar y ver con más detalle una relación de problemas ya resuelta pulsando el botón "Ver relación de problemas".
Pre-condición	El alumno debe haber iniciado sesión. El alumno debe estar en la página "Relaciones ya resueltas"
Post-condición	Garantía de éxito: -El alumno accede a los detalles de la relación resuelta Garantía de fallo: -El alumno se mantiene en la página actual
Prioridad	Alta
Escenario principal	

1. El alumno pulsa el botón "Ver relación de problemas"
2. El sistema muestra en una ventana los distintos detalles de la relación (nombre, autor, fecha de publicación y nota obtenida).

Escenario alternativo

- 1.b El alumno cierra la ventana
 - 1.b.1 El sistema cierra la aplicación
- 1.c El alumno cancela el acceso a los detalles de la relación
 - 1.c.1 El alumno se mantiene en la misma ventana

Diagrama de secuencia

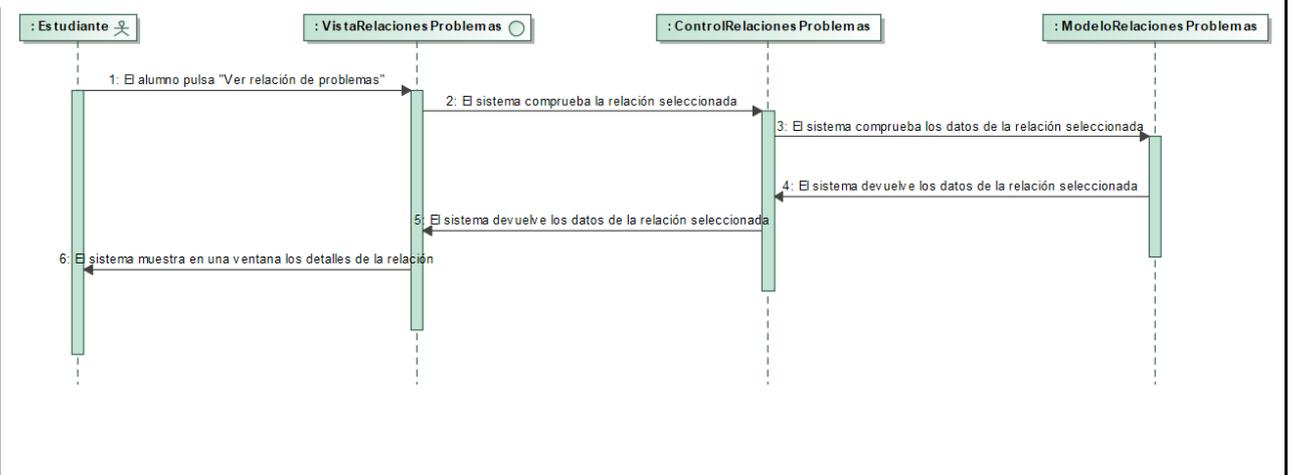


Figura 13: CU 13 - Seleccionar relación de problemas resuelta

Título	Crear una nueva relación de problemas
Descripción	Un docente puede crear una nueva relación de problemas pulsando el botón "Crear nueva relación de problemas" en el Menú principal.
Pre-condición	El docente debe haber iniciado sesión. El docente debe estar en el Menú principal
Post-condición	Garantía de éxito: -El docente crea una nueva relación de problemas. Garantía de fallo: -El docente se mantiene en la misma ventana
Prioridad	Alta
Escenario principal	

1. El docente pulsa el botón "Crear relación de problemas"
2. El sistema muestra un formulario para crear una nueva relación de problemas
3. El docente indica el método de resolución de la relación, los enunciados de los problemas, las matrices de los sistemas de ecuaciones y el resultado final. Después pulsa el botón "Publicar"
4. El sistema pregunta al usuario si confirma la publicación del problema
5. El usuario pulsa el botón "Confirmar"
6. El sistema guarda la relación y la publica

Escenario alternativo

- 1.b El docente cierra la ventana
 - 1.b.1 El sistema cierra la aplicación
- 1.c El docente cancela el acceso a los detalles de la relación
 - 1.c.1 El docente se mantiene en la misma ventana
- 3.b El docente sale de la ventana actual
 - 3.b.1 El sistema comunica al docente si realmente quiere perder los datos completados
 - 3.b.2 El docente confirma la salida de la ventana
 - 3.b.3 El sistema redirige al docente a la anterior ventana
- 5.b El docente pulsa el botón "Cancelar"
 - 5.b.1 El docente se mantiene en la página de formulario
- 5.c El docente no introduce todos los datos necesarios
 - 5.c.1 El sistema informa al docente de los campos faltantes

Diagrama de secuencia

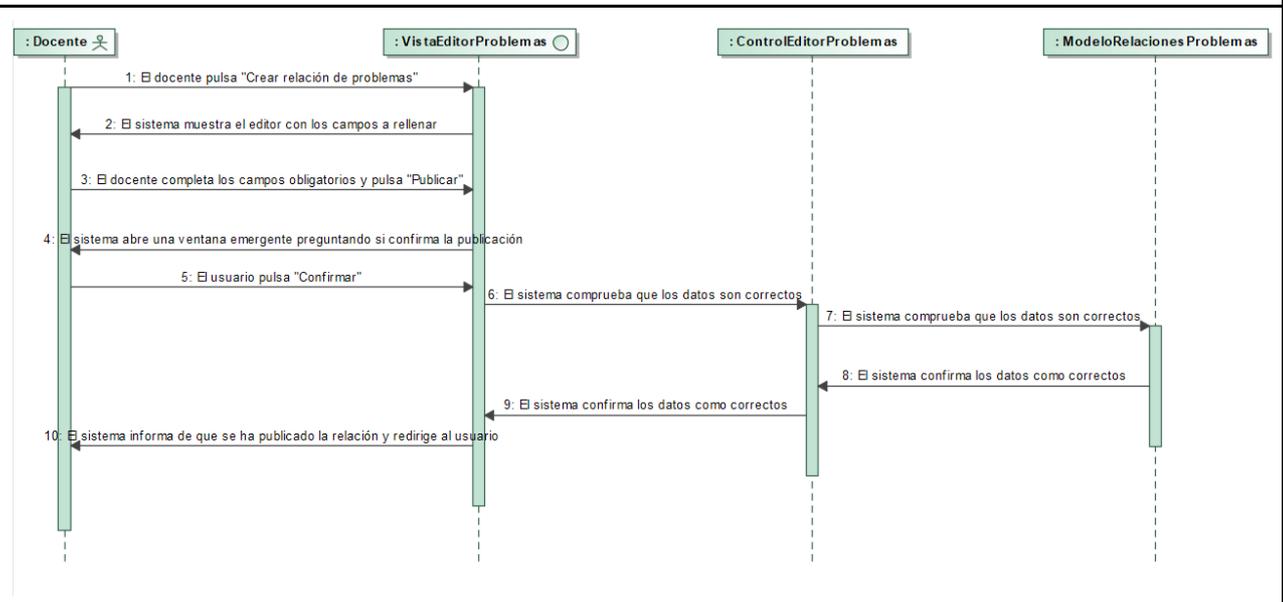


Figura 14: CU 14 - Crear una relación de problemas

Título	Explorar relaciones de problemas publicada por uno mismo
Descripción	Un docente puede explorar las relaciones de problemas que haya publicado él mismo pulsando el botón "Mis Relaciones de Problemas"
Pre-condición	El docente debe haber iniciado sesión. El docente debe tener acceso al botón de "Mis Relaciones de Problemas"
Post-condición	Garantía de éxito: -El docente accede a la ventana de las relaciones que haya publicado Garantía de fallo: -El docente se mantiene en la misma ventana
Prioridad	Alta
Escenario principal	
<ol style="list-style-type: none"> 1. El docente pulsa el botón "Mis Relaciones de Problemas" 2. El sistema redirige al docente a una ventana donde se muestran los problemas que ha publicado. 	
Escenario alternativo	
<ol style="list-style-type: none"> 1.b El docente cierra la ventana <ol style="list-style-type: none"> 1.b.1 El sistema cierra la aplicación 1.c El docente cancela el acceso a las relaciones que haya publicado <ol style="list-style-type: none"> 1.c.1 El docente se mantiene en la misma ventana 	

Figura 15: CU 15 - Explorar relaciones de problemas publicada por uno mismo

Título	Ver detalles de una relación de problemas publicada
Descripción	Un docente puede ver los detalles de una relación de problemas que haya publicado, tales como los alumnos que la resolvieron, sus calificaciones, la media de calificaciones, y los problemas de la misma relación

Pre-condición	El docente debe haber iniciado sesión. El docente debe estar en la ventana de "Mis relaciones"
Post-condición	Garantía de éxito: -El docente accede a los detalles de la relación seleccionada Garantía de fallo: -El docente se mantiene en la misma ventana
Prioridad	Alta
Escenario principal	
<ol style="list-style-type: none"> 1. El docente pulsa el título de la relación de problemas 2. El sistema muestra en una ventana los detalles relativos a la relación 	
Escenario alternativo	
<ol style="list-style-type: none"> 1.b El docente cierra la ventana <ol style="list-style-type: none"> 1.b.1 El sistema cierra la aplicación 1.c El docente cancela el acceso a los detalles de la relación 	

Figura 16: CU 16 - Ver detalles de una relación de problemas publicada

Título	Eliminar una relación de problemas
Descripción	Un docente puede eliminar una relación de problemas que haya publicado él mismo
Pre-condición	El docente debe haber iniciado sesión. El docente debe haber entrado en la página de detalles de una de sus relaciones
Post-condición	Garantía de éxito: -El docente elimina la relación de problemas seleccionada Garantía de fallo: -El docente se mantiene en la misma ventana
Prioridad	Alta
Escenario principal	
<ol style="list-style-type: none"> 1. El docente pulsa el botón "Eliminar relación " 2. El sistema abre una ventana emergente preguntando si desea confirmar la eliminación 3. El docente pulsa el botón "Confirmar" 4. El sistema elimina la relación y actualiza la vista de las relaciones publicadas por el docente 	

Escenario alternativo

- 1.b El docente cierra la ventana
 - 1.b.1 El sistema cierra la aplicación
- 3.c El docente cancela la eliminación de la relación de problemas seleccionada
 - 3.c.1 El sistema mantiene la relación de problemas
 - 3.c.2 El usuario se mantiene en la misma ventana

Diagrama de secuencia

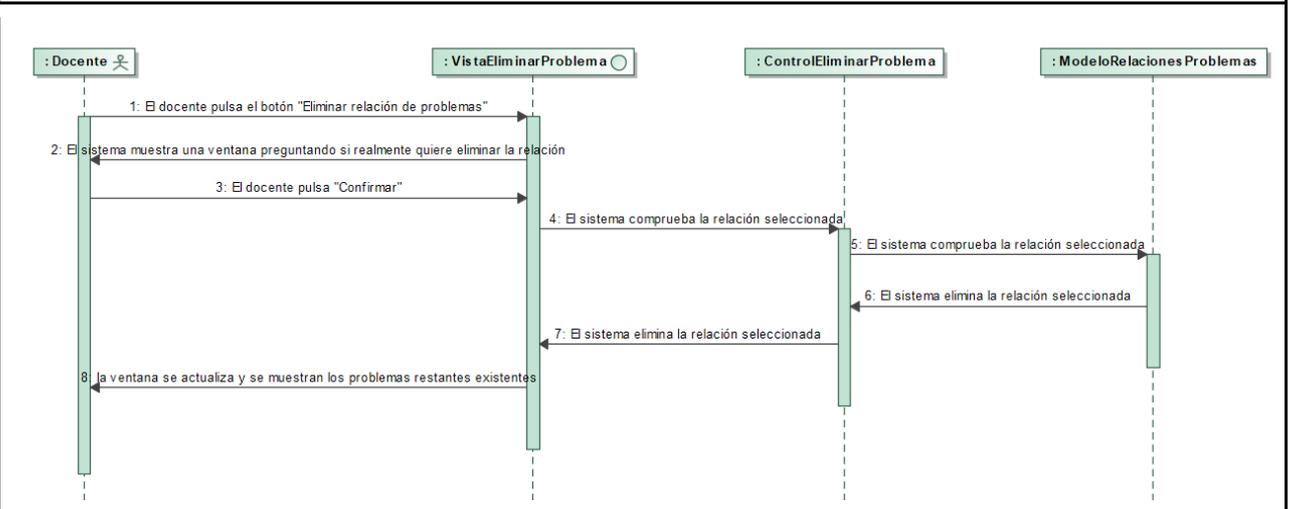


Figura 17: CU 17 - Eliminar una relación de problemas publicadaEl resto de casos de uso relativos al docente tienen el mismo flujo de acciones que en los casos del alumno, por lo que se ha optado a no mostrarlos con el fin de evitar redundancia.

4

Metodología y planificación

En este apartado se verá la planificación que se ha seguido para la ejecución de tareas, junto a la metodología de trabajo que se ha tenido presente.

4.1 Planificación

Para el proceso de planificación de tareas se ha hecho uso de la herramienta **Gantt Project**, programa con el cual se pueden realizar diagramas de Gantt en el que mostrar las tareas existentes, y realizar estimaciones de tiempo sobre la duración de tareas, dependencias entre tareas para la finalización de las mismas, etcétera. En las figuras **18** y **19** se muestran los diagramas de Gantt que se han realizado para este proyecto. Hay que tener en cuenta que estos diagramas son orientativos, y que es una estimación de las tareas, pues algunas han requerido más tiempo del estimado, mientras que otras han requerido de un menor tiempo para su revisión y compleción. Aún así, se ha intentado ajustar en lo posible respecto a la planificación.

4.2 Metodología de trabajo

La metodología de trabajo escogida ha sido una metodología ágil, en concreto, una metodología **incremental iterativa**. En líneas generales, se iban realizando iteraciones de alrededor de 2 semanas, en las cuales se completaban una serie de tareas prioritarias seleccionadas. Las tareas seleccionadas eran clasificadas en tableros conforme el tipo de tarea, las fechas de compleción previstas, y el estado en el que la tarea se encontraba (bien que estuviese propuesto, que estuviese pendiente a realizar, en proceso, lista para revisión y finalmente completa). Finalmente se realizaba un análisis sobre las tareas completadas, las tareas aún pendientes, y las nuevas tareas previstas a realizar, y un análisis del avance del proyecto. Para ellos se ha hecho uso de la herramienta **Trello**, una aplicación web que permite tener un tablero de trabajo donde organizar las tareas. En la **Figura 20** se muestra la interfaz de la aplicación y el uso de la misma.

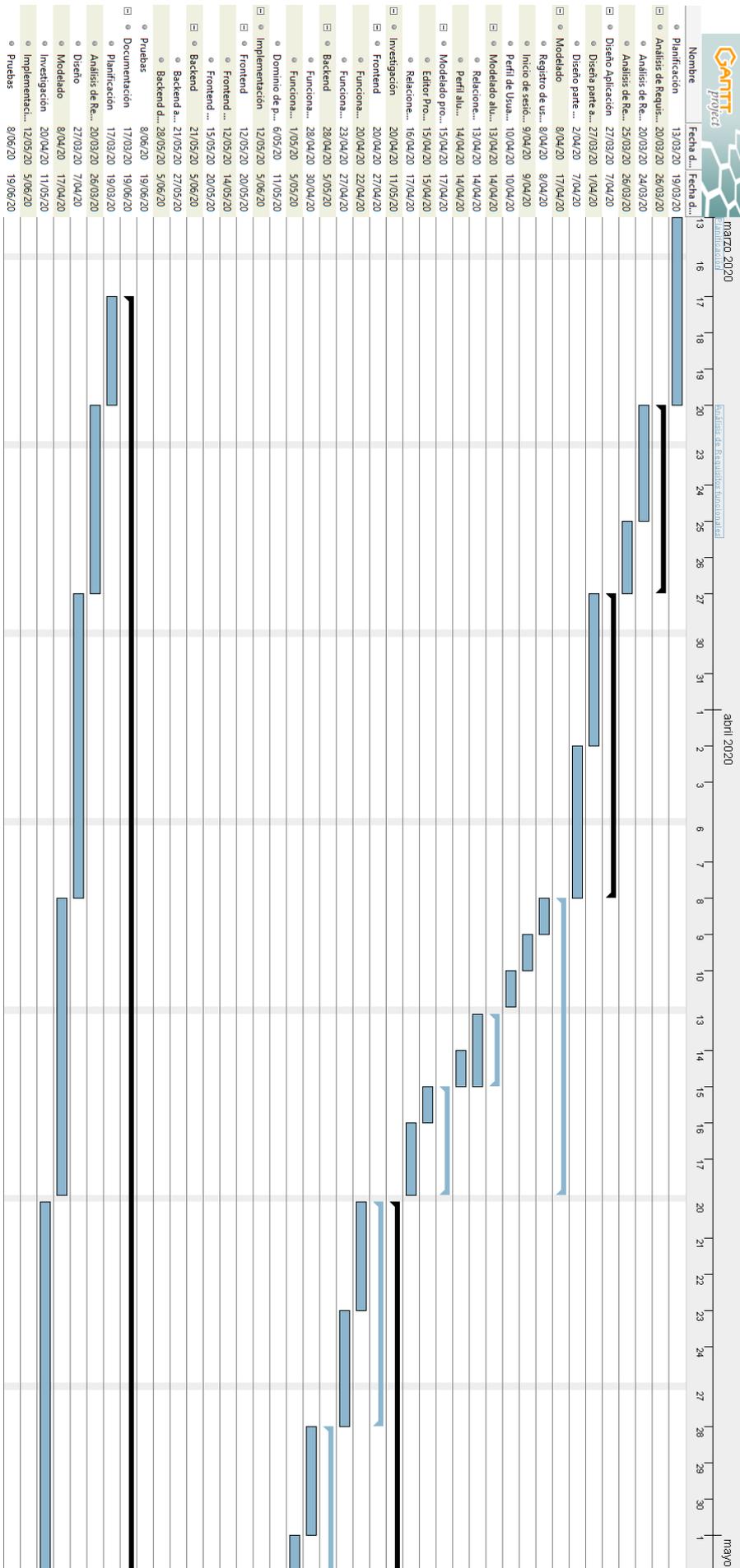


Figura 18: Primera mitad del diagrama de Gantt de la planificación

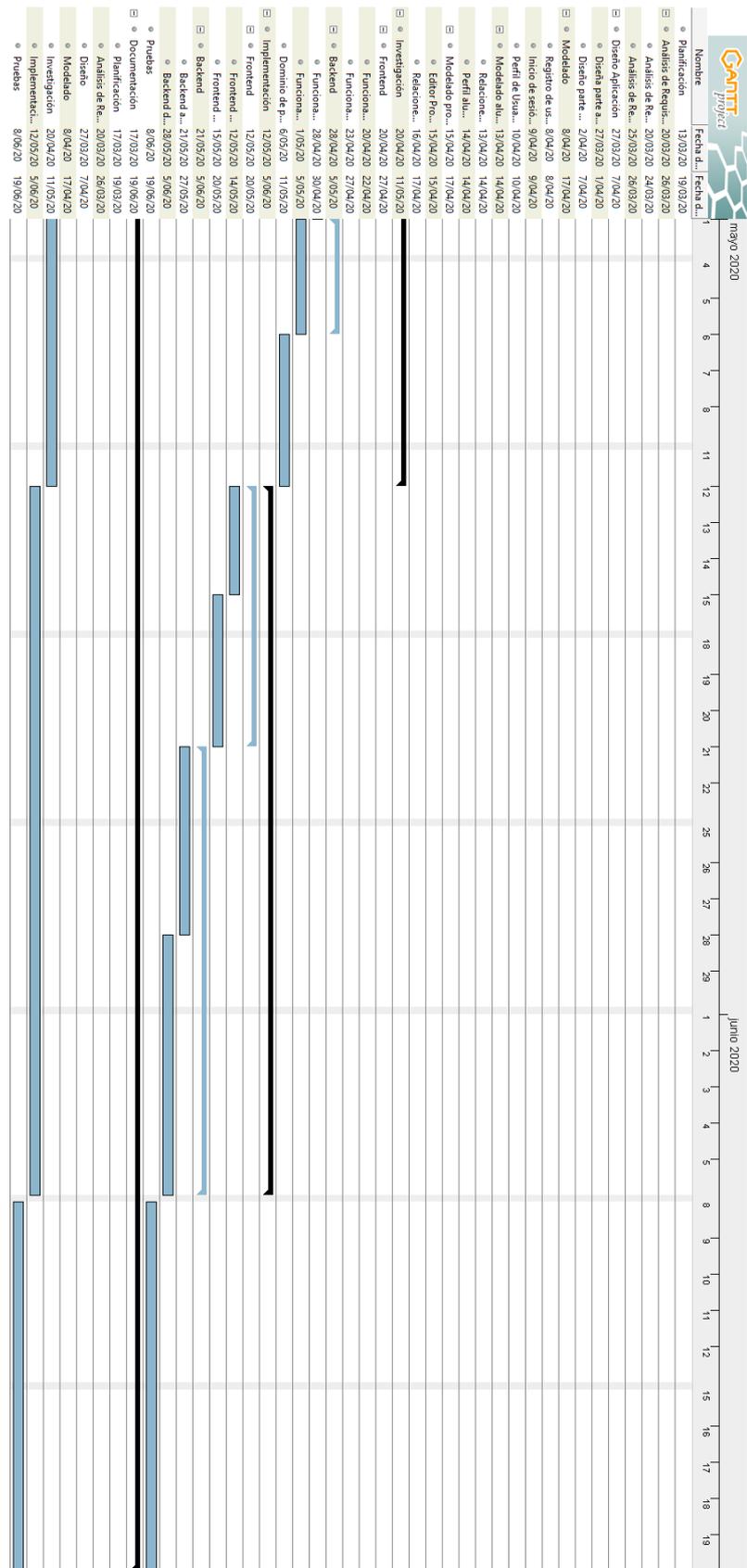


Figura 19: Segunda mitad del Diagrama de Gantt de la planificación

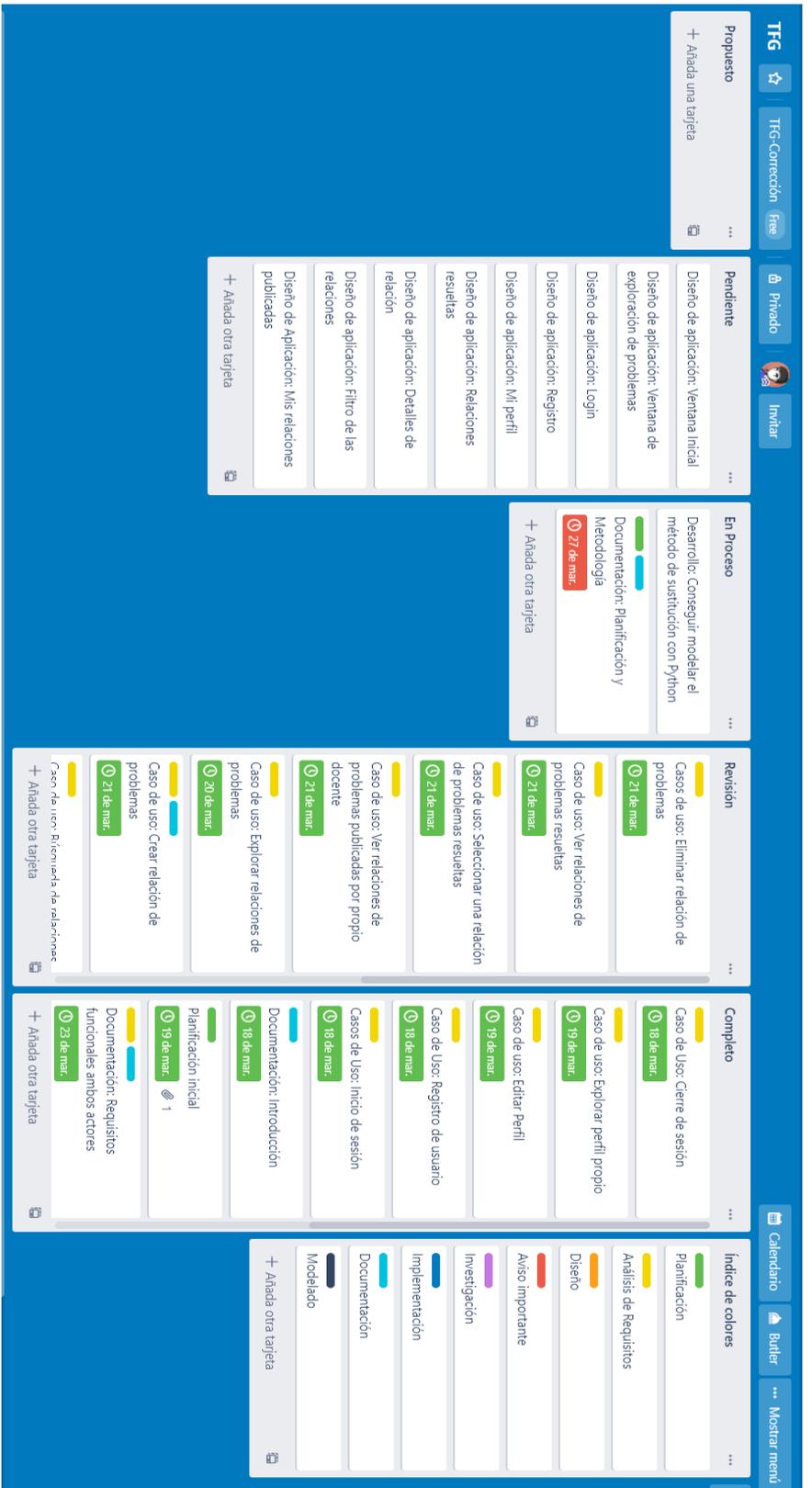


Figura 20: Captura de la herramienta Trello

5

Diseño e implementación

Este apartado trata sobre los aspectos de diseño de la aplicación, su estructura general y modelos de la misma, acompañado de la implementación que se ha realizado, junto a sus explicaciones pertinentes.

5.1 Estructura de la aplicación

5.1.1 Diseño arquitectónico de la aplicación

Para el diseño general de la aplicación, se ha tenido presente la forma en la que trabaja el framework Flask, y sus soluciones. De por sí Flask es minimalista, por lo que elementos como realizar un modelado de objetos orientado a sus esquemas en base de datos no son una necesidad como tal (bien como aparecería en otros frameworks como JSF, JPA, y demás). Aquí se ve una estructura más ajustada. La aplicación se podría dividir en dos componentes principales: frontend (o lado del cliente) y backend unificado (lado del servidor). En la **Figura 21** se ilustra visualmente el concepto.

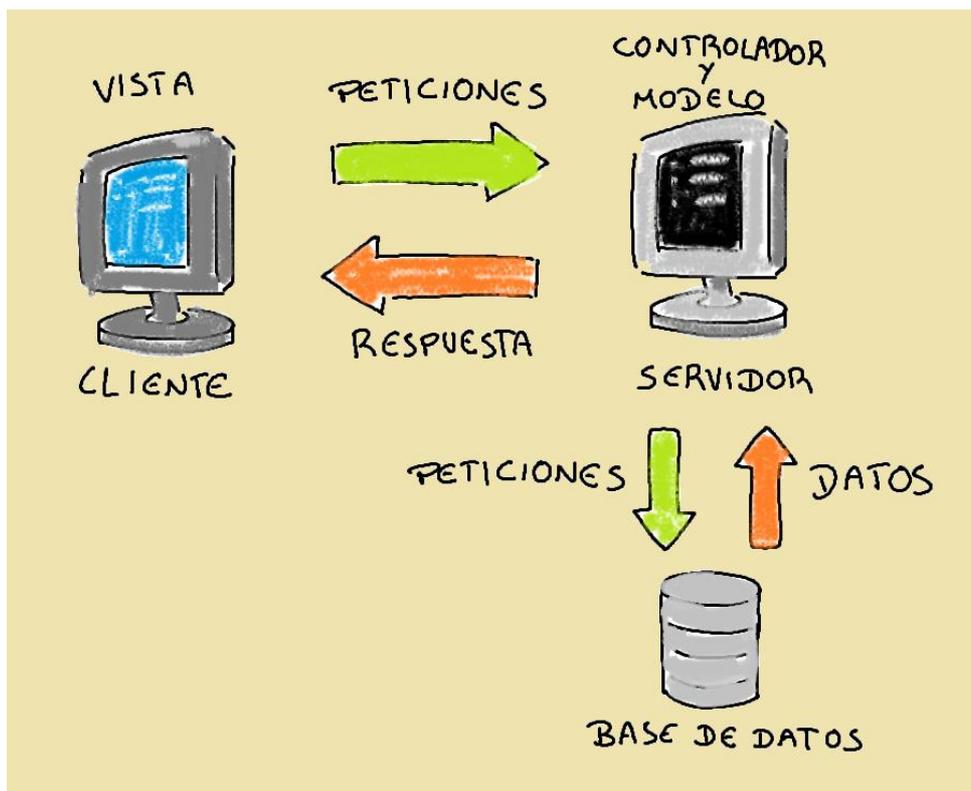


Figura 21: Estructura general del proyecto

Se ha decidido denominar al backend como unificado ya que en él se mezclan ciertos aspectos. Si se comparase con un modelo MVC (Modelo Vista Controlador), se podría decir que nuestro backend representa el Modelo y el Controlador; se tienen las funciones de control en el proyecto, las cuales operan sobre la base de datos mediante unos modelos los

cuales no se especifican, pero se han de conocer para poder trabajar con las estructuras de la base de datos. Más adelante, cuando se llegue al apartado de la implementación, se verán estos aspectos concretos.

5.1.2 Diseño conceptual de la aplicación

La aplicación podría dividirse en cuatro apartados generales si se obvian los roles de los distintos actores del sistema; el **inicio de sesión/registro** en la aplicación, el **menú principal**, las **relaciones de problemas** y sus distintas acciones, el **perfil de usuario** y las **relaciones de problemas ya resueltas/publicadas**.

5.1.2.1 Inicio de sesión y registro de usuario

Esta sección de la aplicación recorre el propio acceso a la misma, con la necesidad de que el usuario tenga un perfil creado para poder acceder a ella. Este es el punto de inicio en la aplicación, el usuario aquí podrá hacer dos cosas, iniciar sesión indicando sus credenciales, o registrarse indicando los datos requeridos. En la **Figura 22** se ilustra cómo sería la página de inicio de sesión, para poder ilustrar mejor la idea, como bien la **Figura 23** ilustra la página de registro. Estas figuras son simples mockups ideados para el diseño de la aplicación, como lo serán el resto de figuras de estos subapartados, y pueden ver variación respecto al resultado de la implementación final.



Figura 22: Boceto de ventana inicio de sesión



Figura 23: Boceto de ventana de registro

5.1.2.2 Menú principal de la aplicación

Una vez el usuario haya iniciado sesión, se le mostrarán los distintos apartados a los que tiene acceso. Dependiendo del tipo de usuario, se mostrarán unos apartados u otros. Para ambos se mostrará la opción de poder acceder al perfil propio de usuario. También ambos tipos de usuarios contarán con un menú superior el cual les permitirá seleccionar las distintas secciones de la página en cualquiera de las ventanas de la aplicación. Dentro de las diferencias entre roles, el usuario alumno podrá ver en la página el botón "Explorar relaciones", como también unos apartados especiales del menú, que en su caso será ver las relaciones de problemas que ya ha resuelto (Relaciones ya resueltas). Por parte del docente, este podrá acceder a un apartado donde podrá publicar relaciones de problemas, y otro donde podrá ver sus relaciones de problemas que él haya publicado. Las figuras **24** y **25** ilustran cómo serían las vistas asociadas al alumno y al profesor, respectivamente.

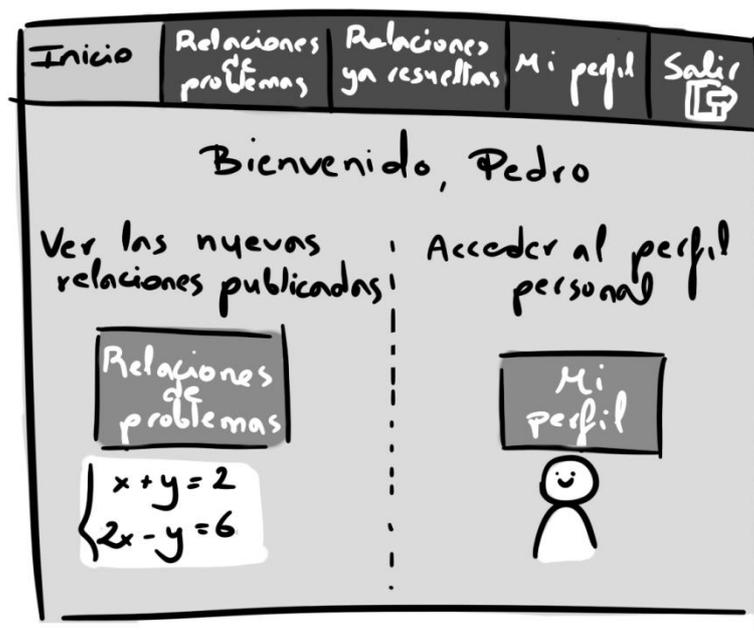


Figura 24: Ventana principal del alumno

5.1.2.3 Relaciones de problemas - Explorar y resolver relaciones

En este apartado se muestra cómo es el diseño de la ventana de "Relaciones de problemas" de la aplicación. Una vez el usuario accede a ella, podrá ver las distintas relaciones de problemas las cuales se encuentran publicadas. Estas se mostrarán ordenadas por fecha de publicación más reciente. También podrá obtener más información de ellas pulsando sobre el título de las mismas, donde se redirigirá a una página con más detalles sobre esta. El usuario podrá filtrar las relaciones que se muestren bien haciendo uso del buscador o bien mediante los filtros los cuales se muestran en el desplegable. En la **Figura 26** se muestra un boceto con el esquema de la página mostrando las relaciones. En la **Figura 27** los distintos filtros de las relaciones, y en la **Figura 28** la página detallada de una relación.

Para resolver una relación, hay que encontrarse en la página de la información detallada de la misma (tómese de referencia la **Figura 28**). Una vez el alumno quiera comenzar a resolver la relación, tendrá que pulsar el botón "Comenzar a resolver relación". Esto hará que se redirija a una página con las preguntas y los formularios para responder a estas, haciendo uso de una serie de formularios y pasos a seguir e indicar (**Figura 29**). Una vez rellene los datos, podrá evaluar las respuestas.

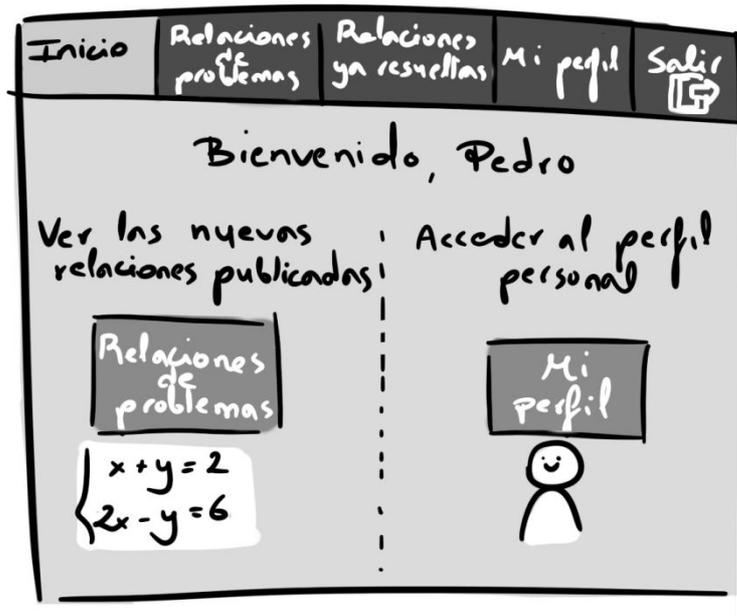


Figura 25: Ventana principal del docente

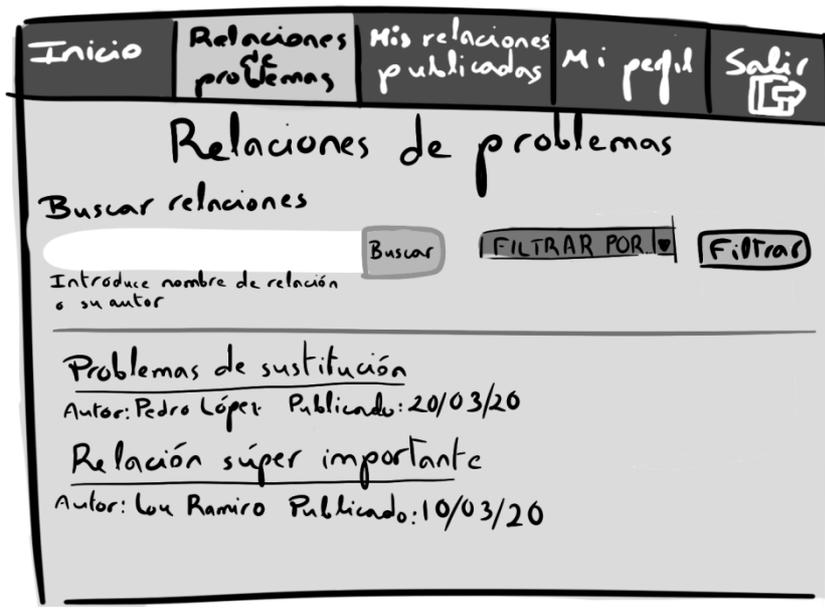


Figura 26: Ventana de relaciones de problemas



Figura 27: Ventana de relaciones de problemas y sus filtros

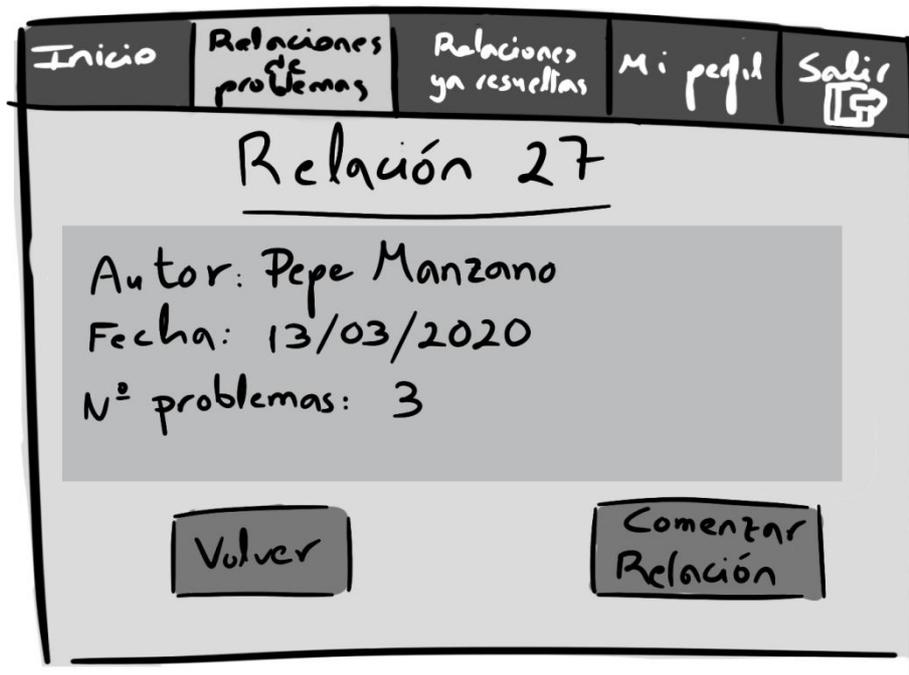


Figura 28: Ventana de detalles de una relación de problemas

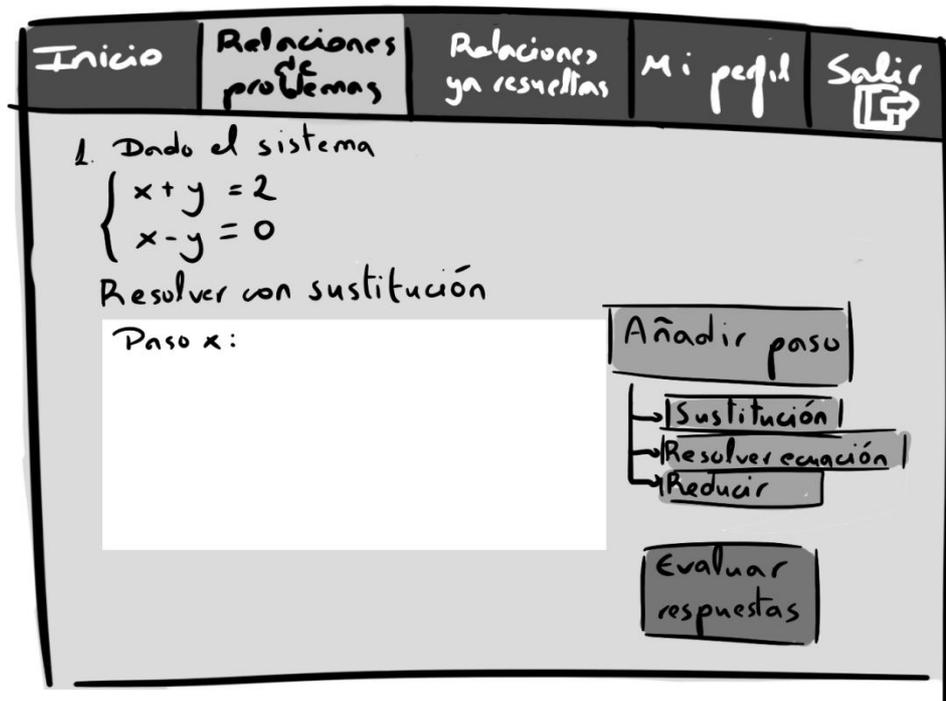


Figura 29: Ventana de resolución de una relación de problemas

5.1.2.4 Exploración de relaciones de problemas ya resueltas

En este apartado de la aplicación el alumno podrá ver las relaciones que ya ha resuelto, y las respuestas que dio en estas junto a los pasos de los ejercicios realizados. La exploración de la página es prácticamente igual que en la parte de "Relaciones de problemas" de la aplicación, donde el único cambio visible observable es a la hora de ver los detalles de la relación (Figura 30), bien como se ha indicado antes.

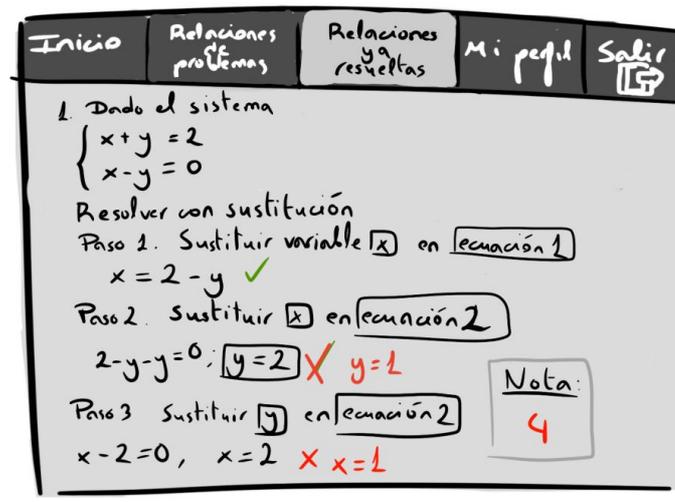


Figura 30: Ventana de una relación ya resuelta

5.1.2.5: Perfil de usuario y edición de datos del perfil

Un usuario (ya sea alumno o docente) podrá ver los datos de su perfil accediendo a la pestaña de "Mi perfil" en el menú principal. Aquí podrá ver sus datos personales (entre ellos su nombre de usuario, su nombre completo, su email y su rol de usuario) (**Figura 31**). Además podrá editar estos datos pulsando el botón "Editar perfil", que redirigirá al usuario a una página donde podrá editar su nombre de usuario, su email y su contraseña, y que podrá confirmar los cambios que quiera pulsando finalmente "Confirmar cambios" (figuras 32 y 33). Si se han guardados los cambios, el usuario será notificado con una alerta de la página.

Nota: Los diseños de esta página han variado respecto a la implementación final, unificando la edición de perfil y la de contraseña como bien se ha indicado antes.



Figura 31: Ventana de "Mi perfil" con los datos del usuario



Figura 32: Ventana de edición de perfil del usuario

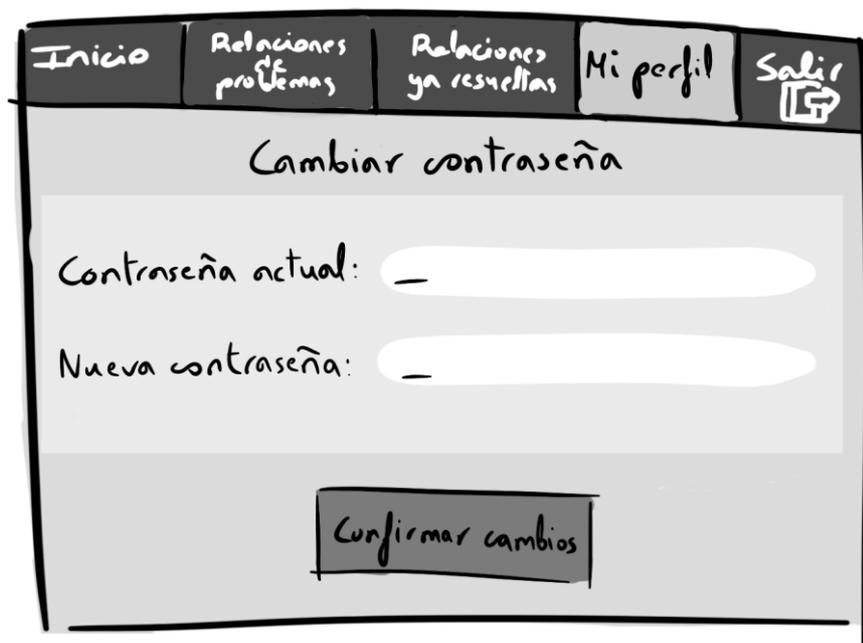


Figura 33: Ventana ideada para editar contraseña

5.1.2.6. Mis relaciones problemas publicadas

Esta ventana es accesible exclusivamente por parte de los docentes. La página mostrará las relaciones que el usuario haya publicado, de una forma similar a como se muestran en la ventana de "Relaciones de problemas", pudiendo también aplicar los filtros de búsqueda por título página (a excepción del de buscar por nombre de autor, el cual no tiene sentido alguno). Lo que cambia respecto a las otras

ventanas es la ventana de detalles cuando se selecciona una relación. En ella se mostrará el nombre de la relación, cuándo se publicó, una lista de los alumnos que la han resuelto, junto a la calificación obtenida, y un campo donde se indicará la nota media de todas las notas de los alumnos que han resuelto la relación. Además, dentro de esta ventana será podrá decidir si eliminar o no la relación pulsando el botón "Eliminar relación" (Figura 34).

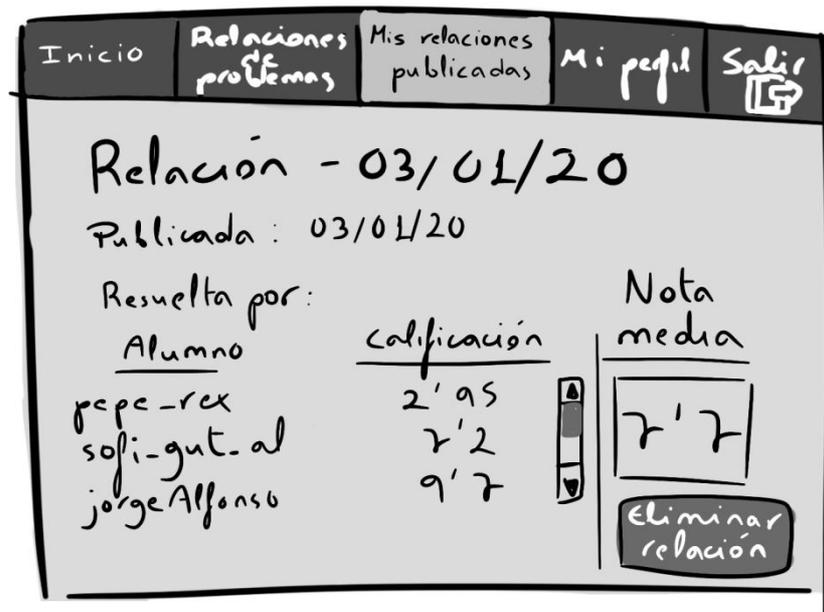


Figura 34: Ventana de detalles de una relación seleccionada.

5.2. Modelo de clases

Este apartado se centra en el modelo de clases, y en explicar la estructura del mismo (Figura 35). El modelo se estructura en dos clases principales la cuales representan los elementos que se manejan en la aplicación. Por un lugar se tiene **Usuario**. Un usuario constará de *Identificador*, *Apodo*, *Email*, *Nombre Completo*, *Rol* y *Contraseña*. Estos serán los datos principales que se guardarán en la Base de Datos. Por otro lado están las **Relaciones de Problemas**. Si bien en el modelo se refleja que se guarda *Id*, *Enunciado*, *Título*, *Metodología* y *Fecha de Publicación*, esto no es del todo completo. Debido a que los datos guardados son dinámicos, no se puede explicitar qué campos concretos como tales existirán. Bien sí se puede decir que como estructura general se tiene también *ecuaciones* que formarán el *Sistema de ecuaciones* a resolver. Como Mongo no permite guardar matrices, se propone guardar por separado los campos que componen las ecuaciones.

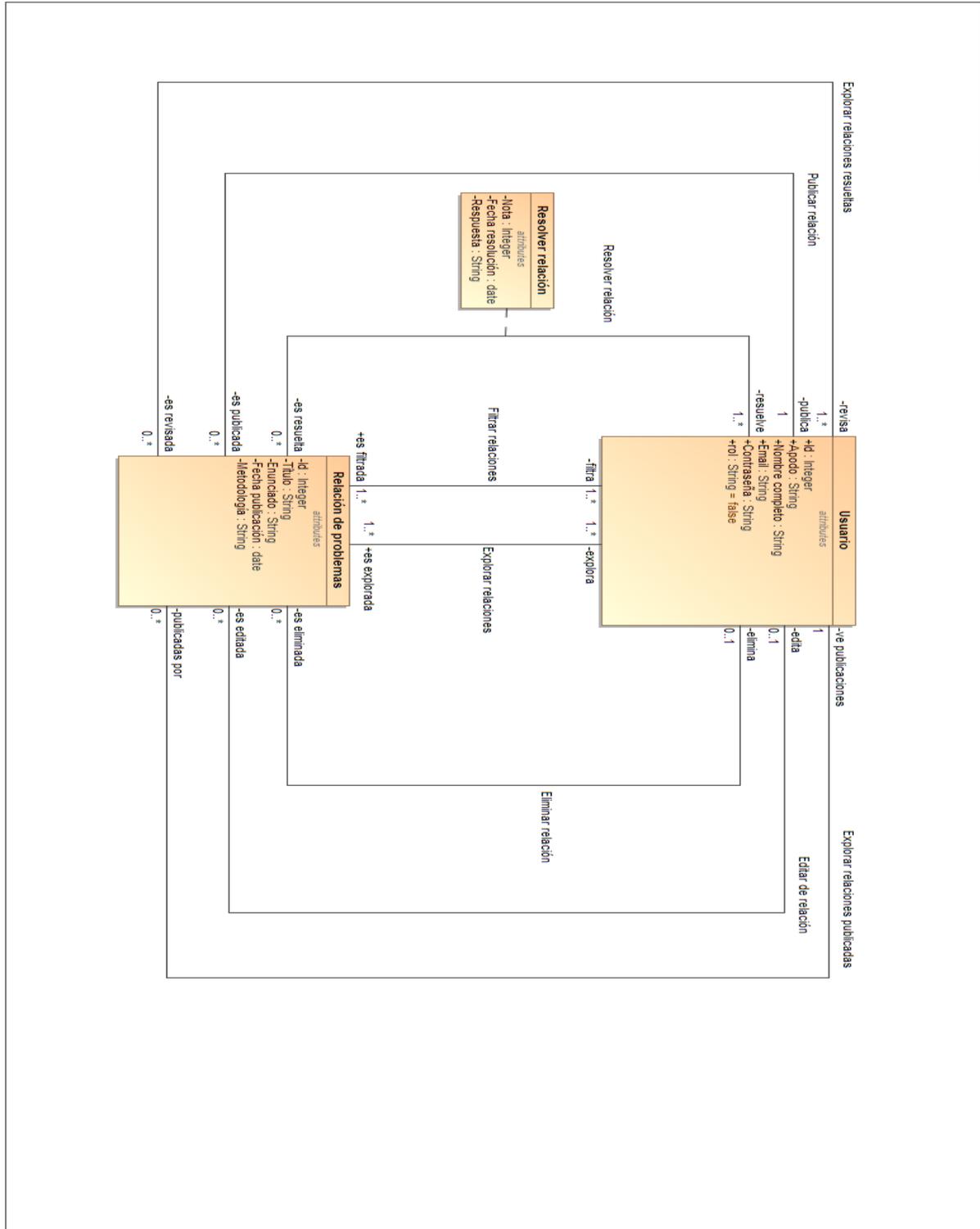


Figura 35: Modelo de la aplicación

Las relaciones implican las distintas funcionalidades y acciones que habrá en el sistema, tales como explorar problemas, publicar problemas, etcétera. Se ha tener presente la relación *Resolver relación*, la cual es una relación representada con una clase debido a que esta contiene atributos. De otra forma no se podría representar esos datos. Hay que tener presente que también

estos atributos son "dinámicos", es decir, estos datos variarán (las respuestas que dé el alumno) conforme las preguntas que existan en la relación.

5.3 Implementación de la aplicación

Este apartado explica los aspectos que componen la implementación final de la aplicación y parte de la estructuración de su código para la construcción final. Debido a la extensión del proyecto, no se puede abordar todo el código que compone la aplicación, pero sí se puede dar unas líneas generales y unas pinceladas para poder entender cómo funciona todo el proceso de la aplicación.

5.3.1 Implementación del Inicio de Sesión

La funcionalidad del inicio de sesión es prácticamente idéntica a como se ha mostrado en su diseño, y su funcionalidad es la que se describe en los casos de uso. El usuario podrá introducir su nombre de usuario y su contraseña para entrar a la aplicación (pudiendo seleccionar el mostrar su contraseña) o podrá seleccionar el botón de crear una nueva cuenta para redirigirse a la página de registro (**Figura 36**).

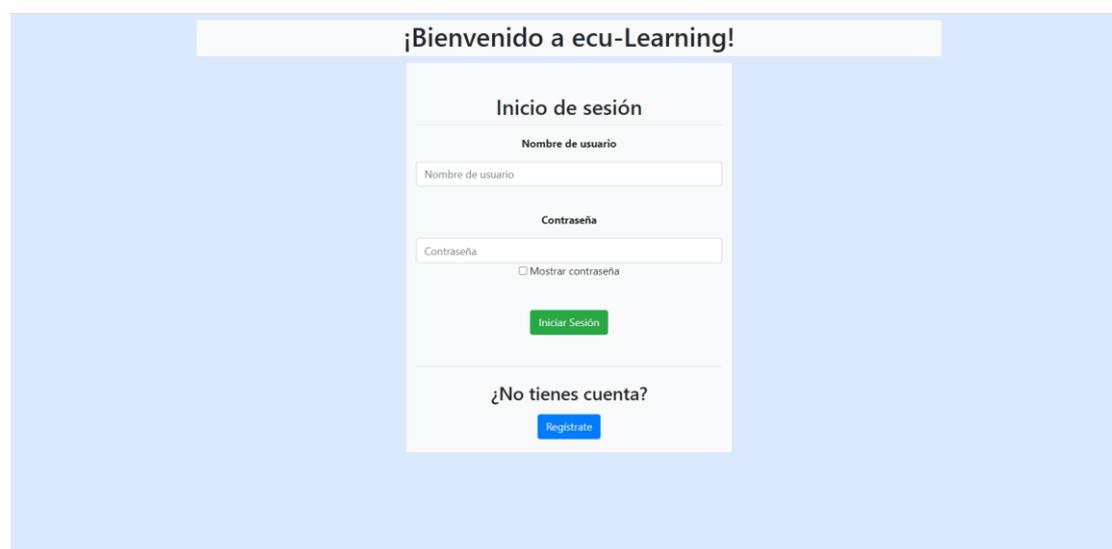


Figura 36: Ventana inicial de la aplicación (Inicio de Sesión)

Sobre la estructuración de su código, puede dividirse en la actuación de la parte del cliente, y la de la parte del servidor (como se ha explicado en apartados anteriores). En la **Figura 37** se ilustra el código HTML y el código de clases y elementos de Bootstrap el cual sirve para mostrar la vista del cliente. El código Javascript utilizado aquí sirve como controlador de la parte del cliente permitiendo

tomar la recepción de datos y la creación de las peticiones de clientes (por medio de métodos HTTP) que se enviarán a la parte del servidor para procesarse en sus respectivas funciones y devolver los datos necesarios, como bien realizar las acciones que se necesiten sobre modificación de datos, creación de mismos, y demás (**Figura 38**). El código Javascript no sólo actúa como controlador de la parte del cliente, sino que además nos permite realizar la interacción con el cliente, como también con los cambios que haga falta realizar en la vista, como modificación de elementos, creación de nuevos elementos gráficos y muestra de estos elementos creados (**Figura 39**).

```

1  <!doctype html>
2  <html lang="en">
3  <head>
4
5  <meta charset="utf-8">
6  <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
7  <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/css/bootstrap.min.css" integrity="sha384-Vkoo8x4CGs03+Hhx
8  <link rel="stylesheet" href="/static/css/login-style.css">
9
10 <title>ecu-Learning - Login</title>
11 </head>
12 <body onload="checkLogin()" style="background-color: #d9e9ff;">
13 <div class="text-center">
14 <div class="container bg-light border mt-2">
15 <h1>¡Bienvenido a ecu-Learning!</h1>
16 </div>
17
18 <div class="container-fluid pt-5 center bg-light border mt-2" style="width: 500px; height: 600px;">
19 <div class="border-bottom">
20 <h2>Inicio de sesión</h2>
21 <form onreset="loginUser()">
22 </div>
23 <label class="mt-3"><b>Nombre de usuario</b></label> <br>
24 <input type="text" class="form-control mt-2" name="username" placeholder="Nombre de usuario"
25 id = "username"> <br>
26
27 <div class="mt-3">
28 <label><b>Contraseña</b></label> <br>
29 <input type="password" class="form-control mt-2" name="password" placeholder="Contraseña"
30 id = "password">
31 <input type="checkbox" onclick="showPassword()"> Mostrar contraseña <br>
32 </div>
33
34 <input type = "reset" class="btn btn-success mt-5" id="submit" value="Iniciar Sesión"/>
35 </form>
36
37 <div class = "mt-5 border-top">
38 <br>

```

Figura 37: Código HTML de la página de Inicio de Sesión.

5.3.2 Registro de un usuario en la aplicación

La ventana de registro de aplicación (**Figura 40**) tampoco varía mucho respecto a su diseño, pero se puede destacar en sus funcionalidades la comprobación de si el nombre de usuario está ya registrado en la aplicación, la comprobación de si el correo electrónico está registrado, y una comprobación de seguridad de repetición de contraseña (**Figura 41**).

```

<script>
document.getElementById("password")
.addEventListner("keyup", function(event) {
event.preventDefault();
if (event.keyCode == 13) {
document.getElementById("submit").click();
}
});

function createRequest() {
var result = null;
if (window.XMLHttpRequest) { // Firefox, Safari, etc.
result = new XMLHttpRequest();
} else if (window.ActiveXObject) { // MSIE
result = new ActiveXObject("Microsoft.XMLHTTP");
} else {
alert("Funcionalidad no soportada por el navegador");
}
return result;
}

function loginUser(){
var req = new createRequest();

req.onload = function() {
if (req.readyState == 4 && req.status == 200){
sessionStorage.setItem("username", document.getElementById('username').value);
window.location.href = '/initial_page';
} else {
alert("Usuario o contraseña erróneos, vuelve a intentarlo")
}
};
var url = "/login";
req.open("POST", url, false);
req.setRequestHeader('Content-Type', 'application/json');
var response = {
"usuario": document.getElementById('username').value,
"contraseña": document.getElementById('password').value,
}
req.send(JSON.stringify(response))
}
}

```

Figura 38: Código Javascript de la página de Inicio de Sesión

```

function showPassword(){
var typePass = document.getElementById("password");
if (typePass.type == "password"){
typePass.type = "text";
} else {
typePass.type = "password";
}
}

```

Figura 39: Código Javascript de ejemplo para modificación de elementos HTML.

Figura 40: Ventana de registro de la aplicación

Crear una nueva cuenta

El usuario indicado ya está escogido

Nombre de usuario

Nombre completo

Las contraseñas no son coincidentes

Contraseña

Repite la contraseña

El email indicado ya está registrado

Correo electrónico

Figura 41: Comprobaciones de datos de registro

5.3.3 Página de inicio de la aplicación

La página de inicio muestra las distintas páginas a las cuales los usuarios pueden acceder. Dependiendo de si el usuario es un alumno (**Figura 42**) o un docente (**Figura 43**), el usuario podrá ver unos menús y accesos u otros. Además, para evitar accesos no autorizados por diferencias de rol, se realizan dentro de las páginas comprobaciones de rol de usuario para evitar que se realicen dichos accesos (**Figura 44**).

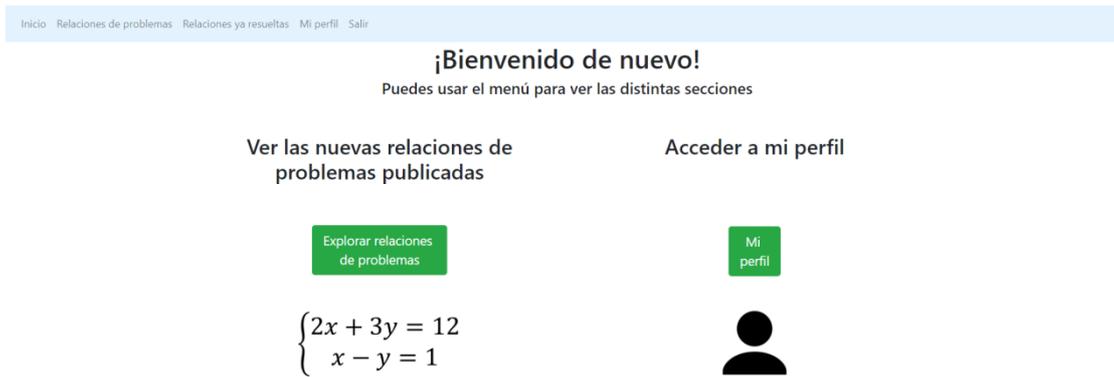


Figura 42: Página de inicio mostrada al alumno

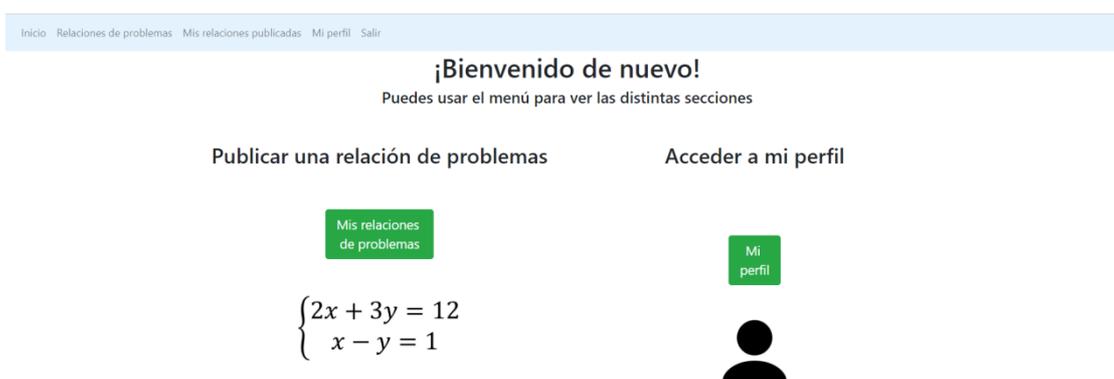


Figura 43: Página de inicio mostrada a un profesor

5.3.4 Perfil de usuario

Ambos usuarios pueden acceder a su perfil, donde se muestra su nombre de usuario, su nombre completo, correo electrónico y rol en la aplicación (**Figura 45**). El usuario podrá cambiar los datos de su perfil pulsando en el botón "Editar perfil", donde podrá cambiar su nombre de usuario, contraseña y email. Esta página también realiza comprobaciones sobre la disponibilidad del nombre de usuario y del email que se quieran escoger.

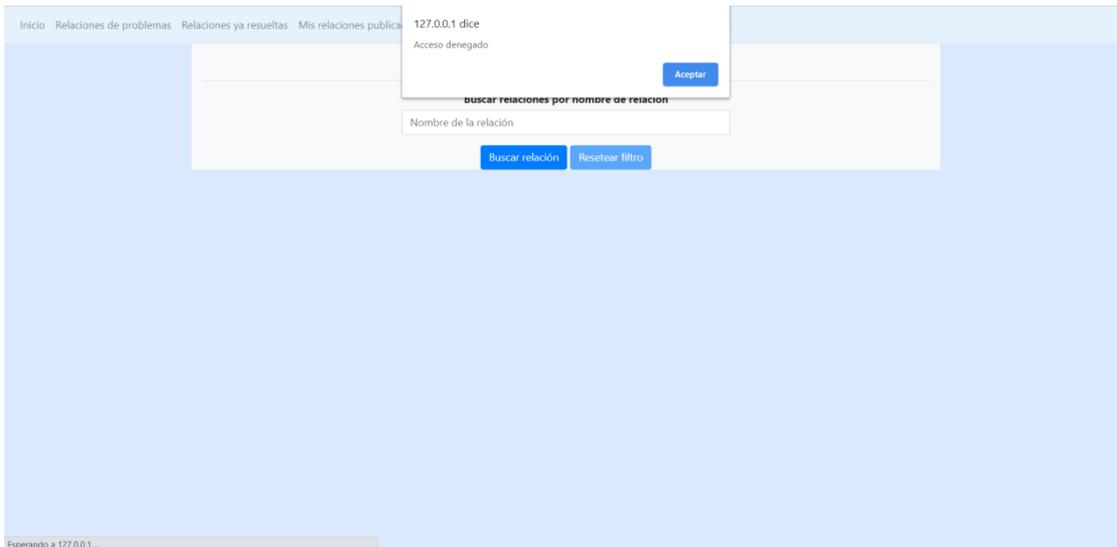


Figura 44: Ejemplo de página con acceso no autorizado por rol.



Figura 45: Página de edición de perfil de usuario

5.3.5 Relaciones de problemas

Esta página contiene las relaciones de problemas que los profesores han publicado, mostrándose el autor de la misma, el título, la fecha en la que se creó, y un botón para ver en detalle los datos de la relación. La página cuenta con un filtrador que permite reordenar las fechas conforme el filtro seleccionado, y un buscador para la búsqueda de relaciones específicas, ya sean por nombre de autor o por título de relación. (**Figura 46**).

El botón de "Ver problema" dirige a la página de detalles de la relación. En esta aparecen los detalles de la relación (título de los problemas de la relación, enunciado, sistemas a resolver, número de pasos

esperados), y dependiendo del rol del usuario, bien podrá aparecer:

- Si es un alumno, se mostrará un botón para resolver la relación de problemas si no está resuelta (**Figura 47**). En caso de estarlo, el botón estará inactivo, y se mostrará en cada problema los pasos que se hayan seguido (**Figura 48**).
- Si es un profesor, se mostrará qué alumnos han resuelto la relación, adjuntándose las notas que hayan obtenido, la media de todas las notas de los alumnos (**Figura 49**).

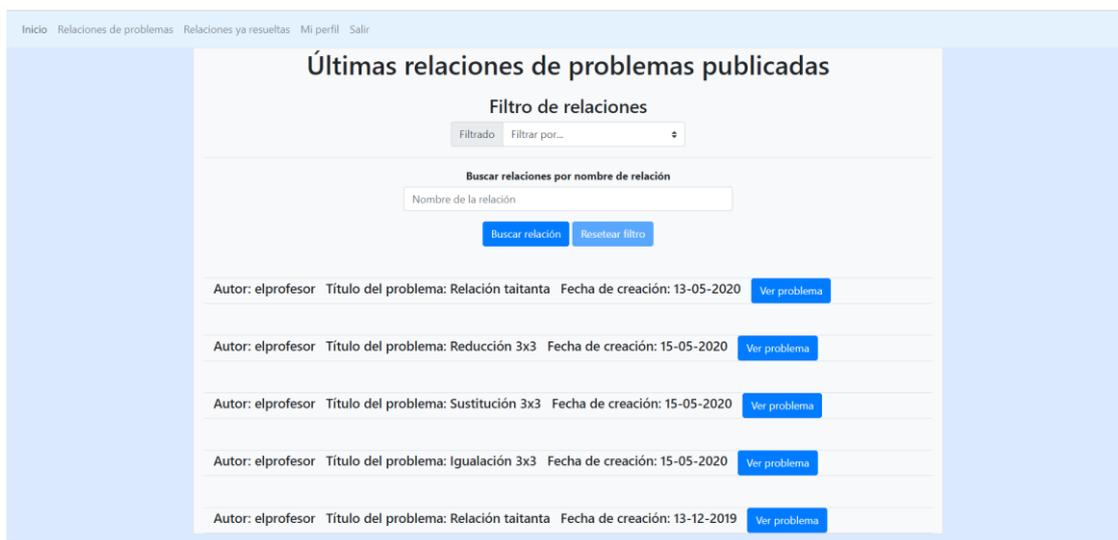


Figura 46: Página de relaciones de problemas

5.3.6 Problemas resueltos por el alumno

En esta página los alumnos pueden ver las relaciones de problemas que hayan resuelto ya, y podrán filtrarlas por medio del buscador de la página, ya sea por autor o título de la relación (**Figura 50**). Igual que en la página de exploración de relaciones, podrán ver los detalles de la relación de problemas pulsando en el botón "Ver problema", donde se les mostrarán todos los detalles antes especificados.

5.3.7 Resolución de problemas publicados

La página de resolución de problemas permite al alumno resolver el problema que haya seleccionado para resolver. En esta página podrá ir creando los pasos necesarios para indicar el proceso de resolución del problema. Dependiendo del método de resolución del problema, los pasos disponibles para el alumno serán unos u otros. Una vez finalice de indicar los pasos seguidos, podrá corregir los resultados pulsando en el botón de finalización (**Figura 51**).

Resolver problema

autor: elprofesor2
 Título: Problemas prof2
 Fecha de creación: 24-05-2020

Problema fácil

Antonio va a pescar 2 veces las lubinas que Juana piensa comprar, junto a 2 veces las lochas que Juana pensaba vender, haciendo el total de 4 pescados. Pepe quiere comprar el mismo número de lubinas que Juana iba a comprar, y dos veces el número de lochas que Juana iba a vender, quedando el total de 3 pescados. ¿Cuántas lubinas ha de pescar Antonio para que coincida con las que Pepe iba a comprar? ¿Y cuántas lochas?

Método de resolución:
Reducción

Número de pasos esperados: 6

Sistema de ecuaciones:
 $2x + 2y = 4$
 $1x + 2y = 3$

Figura 47: Página de detalles de problema no resuelto

Ya has resuelto esta relación

autor: elprofesor
 Título: Igualación 3x3
 Fecha de creación: 15-05-2020

Problema 1

Resuelve por reducción. Reduce al máximo las ecuaciones

Método de resolución:
Igualación

Número de pasos esperados: 12

Sistema de ecuaciones:
 $1x + 1y + 1z = 3$
 $2x + 1y + 1z = 4$
 $2x + 2y + 1z = 5$

Tu solución

Número de pasos dados: 12

Paso 1: Elegir pivote
 Pivote escogido: z
 Ecuación pivote escogida: Ecuación 1

Paso 2: Sustituir una incógnita por una ecuación
 Incógnita sustituida: z

Figura 48: Página de detalles de problema resuelto

autor: elprofesor
Título: Reducción 3x3
Fecha de creación: 15-05-2020

Problema 1

Resuelve por reducción. Reduce al máximo las ecuaciones

Método de resolución:
Reducción

Número de pasos esperados: 9

Sistema de ecuaciones:

$$1x + 1y + 1z = 3$$

$$2x + 1y + 1z = 4$$

$$2x + 2y + 1z = 5$$

elalumno 10
Nota media de los alumnos: 10

Figura 49: Página de detalles de un profesor

Inicio Relaciones de problemas Relaciones ya resueltas Mi perfil Salir

Mis relaciones resueltas

Buscar relaciones por nombre de relación

[Buscar relación](#) [Resetear filtro](#)

Autor: elprofesor	Título del problema: Reducción 3x3	Fecha de creación: 15-05-2020	Nota obtenida: 10
Ver problema			
Autor: elprofesor	Título del problema: Sustitución 3x3	Fecha de creación: 15-05-2020	Nota obtenida: 8.5
Ver problema			
Autor: elprofesor	Título del problema: Igualación 3x3	Fecha de creación: 16-05-2020	Nota obtenida: 10
Ver problema			
Autor: elprofesor	Título del problema: Relación taitanta	Fecha de creación: 19-05-2020	Nota obtenida: 6.666666666666667
Ver problema			

Figura 50: Página de relaciones resueltas por un alumno

5.3.8 Relaciones publicadas por el docente

Esta página es muy similar a la de los problemas resueltos por los alumnos. La principal diferencia es que el profesor desde aquí podrá realizar la publicación de problemas pulsando en el botón "Publicar un nuevo problema". Además, desde esta página podrá eliminar las relaciones de problemas que necesite. Esto hará que las calificaciones existentes de alumnos desaparezcan (**Figura 52**).

Título: Relación taitanta

Problema 1

Ayyy Imao

Método de resolución:
Sustitución

Número de pasos esperados: 6

Sistema de ecuaciones:
 $1x + 2y = 3$
 $2x + 2y = 4$

Paso 1

Escoger un pivote

Elige la incógnita que será el pivote (ten en cuenta que el pivote que se tendrá en cuenta será el último que indiques en un paso).

Incógnita x

Elige la ecuación pivote (se tiene en cuenta si has transformado esa ecuación)

Primera ecuación

Añadir paso

Figura 51: Página de resolución de problemas

Inicio Relaciones de problemas Mis relaciones publicadas Mi perfil Salir

Mis relaciones publicadas

Publicar una nueva relación

Buscar relaciones por nombre de relación

Nombre de la relación

Buscar relación

Resetear filtro

Autor: elprofesor	Título de la relación: Relación taitanta	Fecha de creación: 13-05-2020	Ver relación	Eliminar relación
Autor: elprofesor	Título de la relación: Reducción 3x3	Fecha de creación: 15-05-2020	Ver relación	Eliminar relación
Autor: elprofesor	Título de la relación: Sustitución 3x3	Fecha de creación: 15-05-2020	Ver relación	Eliminar relación
Autor: elprofesor	Título de la relación: Igualación 3x3	Fecha de creación: 15-05-2020	Ver relación	Eliminar relación
Autor: elprofesor	Título de la relación: Relación taitanta	Fecha de creación: 13-12-2019	Ver relación	Eliminar relación

Figura 52: Página de relaciones de problemas publicadas por profesor

5.3.9 Publicación de una nueva relación

Esta página está orientada a los docentes. En ella, un docente puede publicar una relación de problemas, añadiendo nuevos problemas a esta mediante el botón "Añadir un problema". Dentro de los problemas el profesor habrá de indicar distintos campos, que son el título de la relación, el número de pasos esperados por el alumno (los pasos vienen indicados mediante un panel de mensaje), el

título de cada problema, la metodología de resolución y el sistema de ecuaciones y sus dimensiones (eligiéndose primero las dimensiones, y luego creando los campos del sistema con un botón). Para poder publicar la relación, el docente tendrá que pulsar el botón que comprueba si el sistema de ecuaciones es válido. El docente podrá, además, eliminar el último problema creado de la relación mientras la edita, en caso de querer reducir el número de problemas. (Figura 53).

¡Cuidado!
La aplicación sólo acepta sistemas compatibles determinados.
[Lo pillo membrillo](#)

Título de la relación

Título del problema

Número de pasos tomados para resolver el problema (sitúa el cursor encima para saber más)

Método de resolución

Enunciado

Enunciado del problema

Dimensiones de la matriz a resolver

Dimensiones de la matriz a resolver

[Crear matriz](#)

Ecuación1:

 x + y =

Ecuación2:

 x + y =

[Comprobar si el sistema es válido](#)

[Publicar relación](#)

[Añadir problema](#)

[Eliminar último problema](#)

Figura 53: Página de publicación de problemas

6

Pruebas sobre la aplicación

En este penúltimo capítulo se abordarán las pruebas realizadas sobre el código para ver que su funcionamiento era satisfactorio para la mayoría de elementos y características que componen la aplicación.

6.1 Tipos de prueba

Las principales pruebas realizadas sobre la aplicación han sido **pruebas unitarias**. Siguiendo la definición de Wikipedia, *"una prueba unitaria es una forma de comprobar el correcto funcionamiento de una unidad de código. Por ejemplo en diseño estructurado o en diseño funcional una función o un procedimiento, en diseño orientado a objetos una clase. Esto sirve para asegurar que cada unidad funcione correctamente y eficientemente por separado"* (Wikipedia, https://es.wikipedia.org/wiki/Prueba_unitaria).

Con las pruebas unitarias se ha querido comprobar que la funcionalidad de las rutas es correcta. Si bien pueda parecer que únicamente el uso de rutas es insuficiente, se ha de tener presente que el acceso a los métodos del servidor por medio de estas rutas. Con ello, se comprueba dos cosas en las pruebas:

- Que los métodos son accesibles correctamente, permitiendo realizar el envío y la petición de datos al servidor.
- Que se obtiene una respuesta válida o inválida esperada, según la situación, es decir, que los métodos dan una respuesta correcta (respecto a datos que deban transferir).

Con esto en mente, las pruebas permiten detectar si en algún método se ha cometido algún error.

6.2 Estructura de pruebas unitarias

La estructura general de las pruebas realizadas en cada apartado de la aplicación se puede clasificar en dos categorías:

- Pruebas de unidad con respuesta positiva: el método acepta los datos, es capaz de acceder a datos que se estén buscando o simplemente puede acceder a la ruta esperada
- Pruebas de unidad con respuesta negativa: el método no devuelve datos correctos, no encuentra los datos esperados, o bien simplemente no puede acceder a la ruta esperada

Para organizar todo esto, se ha dividido también las pruebas respecto a la funcionalidad de la aplicación que

se estuviese tratando (por ejemplo, en un fichero se contendrá las pruebas relacionadas con las acciones de un profesor con los problemas publicados, en otro las pruebas de datos de perfil, etcétera).

6.3 Usando unittest

El uso del framework requiere de un cierto conocimiento respecto a la realización de pruebas. No obstante, esta construcción es sencilla si ya se conoce otros frameworks como JUnit, debido a que guarda muchos aspectos similares respecto a construcción de pruebas. Lo más importante es el uso de **asertos**, en este caso, pues permite comparar las salidas y los datos esperados (en caso de que hubiese).

Como aspectos adicionales, se destacan los siguientes:

- Para el uso de las pruebas **no se ha usado un mock para bases de datos** (los mocks son objetos preprogramados que esperan un comportamiento concreto para la prueba). Si bien se ha hecho uso de la clase **Client** que trae Flask para la utilización de pruebas para métodos, no se ha construido mock alguno para mongo. Esto se debe principalmente a cómo se ha estructurado la aplicación, y cómo se ha realizado la conexión a la base de datos. En su lugar, **se ha hecho uso de una base de datos de pruebas real**. A efectos prácticos, se ha ahorrado el hecho de tener que construir mock alguno. No obstante, esto tiene ciertas repercusiones, asociadas básicamente a los costes reales que tiene la lectura y escritura de datos en una base no relacional. Se ha de puntualizar, además, que un mock habría de crearse para evitar también problemas de posibles errores y escrituras indeseadas.
- Los ficheros Python y las pruebas que se quieran construir para ejecutar **han de comenzar con el nombre test_**. Si no, la consola de Python no podrá detectar las pruebas.

6.4 Ejemplo de pruebas unitarias

En este apartado se expone cómo se realiza la ejecución de una prueba de unittest, mediante un ejemplo específico.

Se va a realizar la prueba del inicio de sesión de un usuario registrado en la aplicación. En la **Figura 54** se expone el código de la prueba. Se crea una estructura con los datos que el método espera recibir (en un formato de diccionario que luego se transforma a formato json), y se envía por el método de "client.post(...)", asociado a un método POST que se tenga disponible. Este método de la prueba genera una respuesta, la cual contendrá datos como

los de respuesta devueltos (si los hay), el código de respuesta obtenido, entre otros. Para finalizar la prueba, se realiza un aserto, que comprueba que el valor obtenido del código de respuesta es correcto (en este caso, un código 200). Para poder ver el comportamiento esperado, en la **Figura 55** se adjunta el código del método, para ver que efectivamente es la respuesta que se quiere. En la **Figura 56** se puede comprobar que la prueba se ejecuta correctamente y es superada, como se esperaba.

```
20 # Prueba de un inicio de sesión con unos credenciales
21 # válidos
22 def test_correct_login(self):
23     client = app.test_client(self)
24     sent_data = {
25         'usuario': 'elprofesor',
26         'contraseña': 'contraseña'
27     }
28     response = client.post('/login', data=json.dumps(sent_data), content_type='application/json')
29     self.assertEqual(response.status_code, 200)
```

Figura 54: Prueba para un Inicio de sesión correcto

```
@app.route('/login', methods = ['POST'])
def login():
    """
    Realiza el login del usuario con los datos proporcionados.
    Si no encuentra el usuario, devuelve un error 404.
    """
    if request.method == 'POST':
        resp = request.json
        encoded_pass = sha3.sha3_224(resp['contraseña'].encode('utf-8')).hexdigest()
        try:
            usuario = users.find_one({'usuario' : resp['usuario'],
                                     'contraseña' : encoded_pass})
            if (usuario != None) :
                return('', 200)
            else:
                return('', 404)
        except Exception:
            return ('', 401)
```

Figura 55: Método de inicio de sesión, marcándose la respuesta esperada

```
PS C:\Users\Dan\Desktop\Universidad\TFG\tfg\testing> python test_login.py FlaskTest.test_correct_login -v
test_correct_login (__main__.FlaskTest) ... 15e567494dd3fd5c166fba4274faded4cd263570b4e33aa1fe901345
ok

-----
Ran 1 test in 1.314s

OK
PS C:\Users\Dan\Desktop\Universidad\TFG\tfg\testing> |
```

Figura 56: Resultado correcto de la ejecución de la prueba

7

Conclusiones y líneas futuras

En este último capítulo del proyecto se discute acerca de los resultados obtenidos con respecto al desarrollo, los problemas encontrados, qué objetivos se han alcanzado y demás.

7.1 Objetivos y resultados

En este proyecto se había planteado como objetivo principal a cumplir, sí o sí, el funcionamiento válido del corrector de sistemas de ecuaciones. Era el mínimo básico impuesto. Si bien era, valga la redundancia, mínimo, se ha podido alcanzar satisfactoriamente. Todos los requisitos que se han propuesto han podido ser plasmados en el proyecto de una forma u otra. También ha habido una evolución gradual del proyecto y de estos requisitos, que si en un principio estaban especificados de una manera, (y en la gran mayoría se ha intentado mantener así), se ha tenido que adaptar otros. Esto ha sido bien porque la complejidad de la especificación no permitía la ejecución exacta, bien fuera porque se podía añadir una mejora que fuera más allá de lo especificado, o bien simplemente porque las necesidades cambiaban una vez se diseñaba e implementaba el requisito, y se replanteaba las necesidades actuales, junto a la situación del proyecto. Esta problemática encaja de forma adecuada con la aplicación de metodologías ágiles.

Trabajar con Flask ha sido una experiencia satisfactoria. Si bien el framework es minimalista, para este proyecto no se necesitaba una complejidad excesiva. Es más, el hecho de que Flask sea minimalista ha proporcionado grandes ventajas respecto a otros frameworks. Aunque se conocía de antemano y ya se había trabajado en proyectos anteriores con él, no se puede dejar pasar el hecho de que su curva de aprendizaje es de lo más sencilla. Gracias a esto el tiempo necesario a invertir en su investigación ha sido, o en el supuesto de no haberlo usado, sería mínimo. No sólo nos se usa de apoyo su curva de aprendizaje mínima o su simpleza inicial, Flask tiene una gran versatilidad, y permite realizar múltiples modificaciones en su base (bien sea por librerías o por código propio), permitiendo adaptarse a las necesidades del momento. Por poner un ejemplo, se podría adaptar para que funcionase con un modelo MVC (Modelo-Vista-Controlador).

7.2 Problemas encontrados

Cierto es que si bien la experiencia general y el flujo de trabajo han sido bastante satisfactorios, no todo ha sido ni sencillo ni directo. Han surgido múltiples problemas respecto a temas de implementación y planteamiento de

diseño del proyecto. Aspectos como la decisión de comprobación de datos en el corrector por pasos, la gestión de algunas entradas de datos, la comprobación de que los métodos recogían bien los datos y los podían tratar, sin producir excepciones inesperadas han supuesto un cierto reto. Pero la parte más problemática de todas ha sido el propio corrector de problemas y la disposición de los datos guardados. Sobre el corrector de problemas, debido a la cantidad de posibles pasos que había, a los planteamientos sobre cómo guardar las transformaciones de datos que el usuario realizaba (por ejemplo, al principio se guardaba la última ecuación transformada, no siendo posible acceder a las ecuaciones transformadas de pasos anteriores para realizar operaciones), cómo acceder a los pasos que el usuario realizaba, etc. han sido problemas a cuya solución se ha tenido que dedicar un tiempo considerable. Otros aspectos que pueden destacarse están relacionados con mejoras del propio corrector y de aspectos de la aplicación, que se especificará en el apartado de *Posibles mejoras y futuro del proyecto*.

Sobre el tratamiento de qué datos mostrar al usuario en una página, el problema ha sido propiamente la cantidad de datos que se han tenido que procesar. No se contaba con un conjunto predefinido, por lo que no se podía crear un código HTML sencillo que permitiese plasmarlo todo de forma esquemática. Se ha tenido que depender de código Javascript para poder plasmar todos estos datos. Esto ha supuesto dedicar bastante tiempo a escribir una gran cantidad de código Javascript, y a replantear cómo desarrollarlo sin caer en malas prácticas de programación (que aún así se propondrá en el punto de *Posibles mejoras [...]*).

7.3 Posibles mejoras y futuro del proyecto

Se expone en este apartado ciertas mejoras que se cree que podrían realizarse en el proyecto, así como también mencionar ciertos aspectos sobre el futuro del mismo.

7.3.1 Posibles mejoras

- **Mejoras en la página de resolución de problemas:** Comprobaciones de pasos dados por el alumno, y la corrección de unos fallos cuando se crean nuevos pasos se podrían corregir.
- **Optimización de la cantidad de código Javascript por parte del cliente:** Si bien se ha intentado optimizar el código Javascript al justo y necesario, no se ha podido evitar que la cantidad de código que hay creado sea sustancial. Seguramente sea posible

encontrar una forma de reducir el código existente, y hacer una implementación más eficiente.

- **Evitar el uso de sessionStorage:** sessionStorage no es una herramienta que realmente haya dado muchos problemas, pero el proyecto depende bastante de ello. Esto tendría consecuencias graves si el usuario decidiera desactivar los datos de guardado de sesión.
- **Indicación visual de los fallos del alumno:** Algo que se podría implementar como mejora sería que se indicasen los pasos donde el alumno se ha equivocado, y la respuesta esperada a tal paso.
- **Sistemas de ecuaciones más complejos:** Con algo más de tiempo se podrían introducir sistemas de ecuaciones con mayores dimensiones, más posibles métodos de resolución para los sistemas, e incluso otro tipo de sistemas de ecuaciones, tales como compatibles indeterminados, indeterminados, entre otros, ampliando el rango de usuarios que podrían usar la aplicación.
- **Pruebas en más exploradores web:** Pruebas realizadas en otros exploradores además de Google Chrome hubieran servido para comprobar que la aplicación es funcional en todos los dispositivos modernos.

7.3.2 Futuro del proyecto

Echando un vistazo al resultado del proyecto, y comparando con la situación actual de los sistemas correctores (los cuales rara vez permiten al usuario introducir las respuestas paso a paso), se podría decir que el proyecto podría ser usable en su estado actual para el propósito que está creado. La herramienta propone la creación de sistemas de ecuaciones compatibles determinados sencillos, que podría ser perfectamente útil para alumnos que estuviesen introduciéndose en tal temática, y para profesores que necesiten de un método automatizado que corrija las relaciones, teniendo presente el proceso de los alumnos, más que el propio resultado. Mejoras futuras podrían ampliar el rango de usuarios objetivo, permitiendo el uso de sistemas de ecuaciones más elaborados, más métodos de resolución para corregir los problemas, más dimensiones para sistemas de ecuaciones, y no solo eso, sino quizá más funcionalidades dentro de la aplicación (ideas como un chat para los alumnos, foros... una especie de red o campus virtual).

Referencias

Wikipedia, Prueba Unitaria, primer párrafo
https://es.wikipedia.org/wiki/Prueba_unitaria

Documentación Python, unittest - Unit Testing Framework
<https://docs.python.org/3/library/unittest.html>

Documentación de Flask
<https://flask.palletsprojects.com/en/1.1.x/>

Hipertextual, "Flask: minimalismo para el desarrollo web en Python"
<https://hipertextual.com/archivo/2014/08/flask-python/>

Documentación de Bootstrap
<https://getbootstrap.com/docs/4.5/getting-started/introduction/>

Apéndice A

Manual de usuario

A.1 Introducción:

Este manual de usuario se destina a ver el uso de la herramienta y de sus funcionalidades en profundidad.

A.2 Organización:

- o Inicio de sesión
- o Registrar un nuevo usuario
- o Página de inicio
- o Exploración del perfil personal
- o Exploración de relaciones
- o Exploración de mis relaciones publicadas (docente)
- o Publicar una relación (docente)
- o Ver detalles de una relación de problemas
- o Resolver una relación de problemas (alumno)
- o Exploración de relaciones resueltas (alumno)

A.2.1 Inicio de sesión:

Para realizar el inicio de sesión en la aplicación, hay que indicar un nombre de usuario que haya registrado en la misma, y una contraseña (**Figura 57 - 1**). En caso de que se quiera mostrar lo escrito en el campo contraseña, hay que hacer clic en la casilla "Mostrar contraseña" (**Figura 57 - 2**). Si se necesita un usuario para acceder, se puede crear uno nuevo pulsando el botón "Regístrate" (**Figura 57 - 3**).

A.2.2 Registrar un nuevo usuario:

Registrar un nuevo usuario requiere que se rellenen los campos explícitos del formulario del Registro (**Figura 58**). El formulario le irá indicando conforme avance si el nombre de usuario que intenta registrar existe ya (**Figura 59 - 1**), si la contraseña se ha introducido correctamente (**Figura 59 - 2**), y si el correo electrónico que intenta registrar no ha sido escogido (**Figura 59 - 3**). Para confirmar el registro, pulsa "Regístrate".

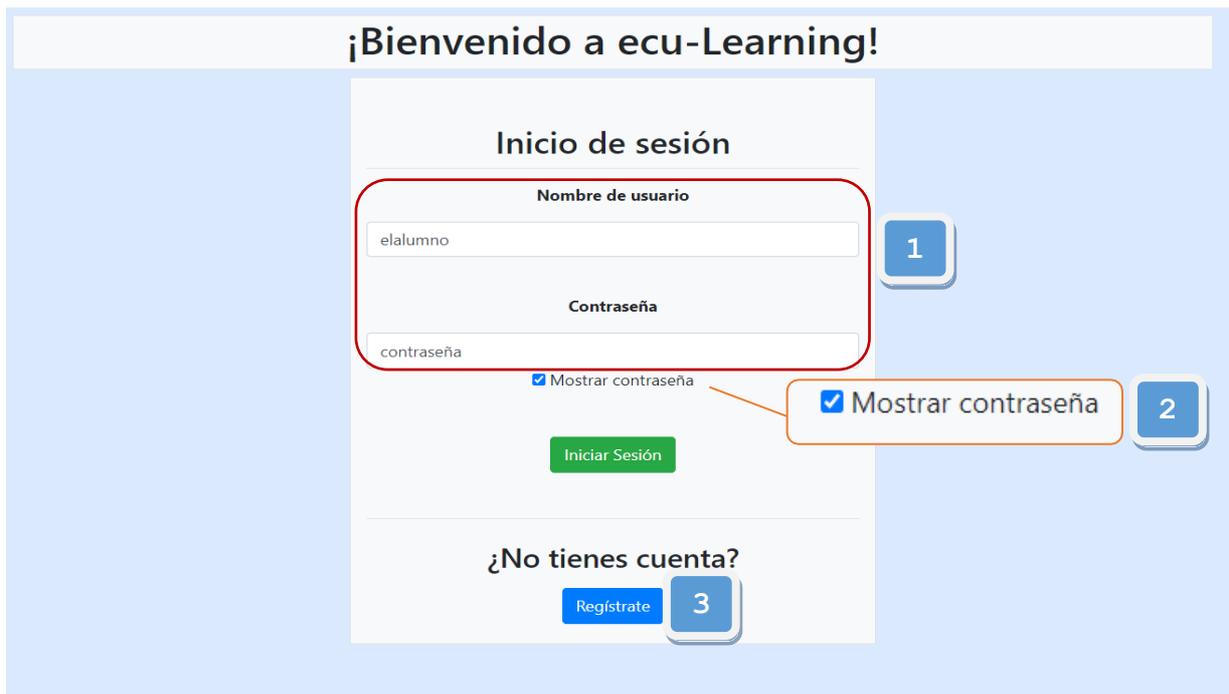


Figura 57: Inicio de sesión

Crear una nueva cuenta

Nombre de usuario
Nombre de usuario

Nombre completo
Nombre completo

Contraseña
Contraseña

Repite la contraseña
Repite la contraseña

Correo electrónico
Correo electrónico

Mi rol es...

Alumno

Profesor

Regístrate

Figura 58: Registro en la aplicación

Crear una nueva cuenta

El usuario indicado ya está escogido 1

Nombre de usuario

Nombre completo

Las contraseñas no son coincidentes 2

Contraseña

Repite la contraseña

El email indicado ya está registrado 3

Correo electrónico

Mi rol es...

Alumno
 Profesor

Figura 59: Comprobación de datos en el registro

A.2.3: Página de inicio

Dependiendo del tipo de usuario con el que se haya iniciado sesión o registrado, se mostrará un menú concreto u otro. En caso de que el perfil sea de alumno, se mostrarán las relaciones de problemas a resolver (**Figura 60**) y las relaciones resueltas. En el caso del docente, se mostrará en su lugar las relaciones publicadas por (**Figura 61**). El resto de menús es compartido, y redirigen a las mismas páginas. El menú superior redirige a las distintas partes de la aplicación, donde "Inicio" dirige a esta página, "Relaciones de problemas" muestra las relaciones publicadas para explorar, "Relaciones ya resueltas" muestra las relaciones que hayas resuelto (sólo alumnos), "Mis relaciones publicadas" muestra qué relaciones has publicado (sólo docentes), y "Mi perfil" muestran los datos de tu perfil.

A.2.4 Exploración del perfil personal

Para acceder a los datos del perfil, selecciona el botón de "Mi Perfil" en el menú superior del menú de inicio. Se te redirigirá a una página con los datos de tu perfil (**Figura 62**). En caso de querer cambiar los datos de tu perfil, pulsa el botón "Editar perfil". En él se podrá

editar el nombre de usuario por uno nuevo no escogido, la contraseña, y la dirección de email registrada por otra no registrada (**Figura 63**). Para confirmar los cambios, hay que pulsar "Actualizar datos". Si todo fue correcto, aparecerá una ventana emergente confirmando que los cambios se han realizado. Las comprobaciones de los datos se realizan aquí, al igual que en la página de registro.

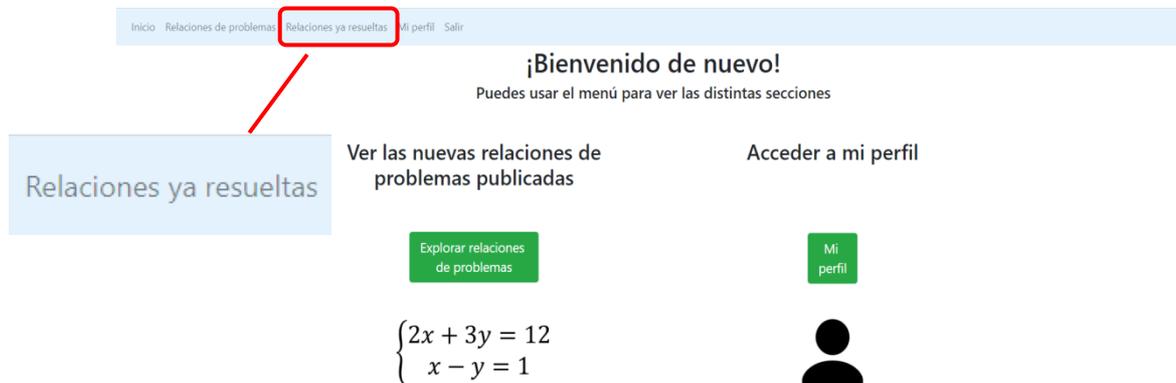


Figura 60: Página de inicio de un alumno

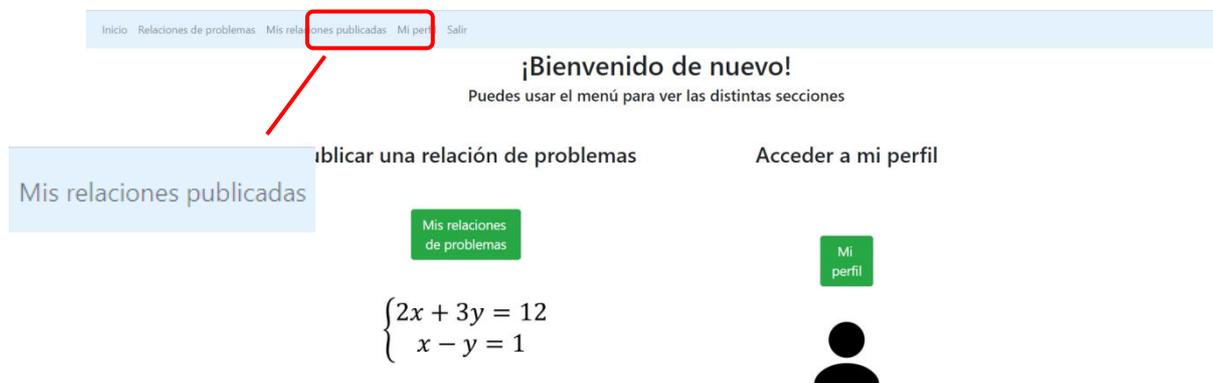


Figura 61: Página de inicio de un profesor

A.2.5. Exploración de relaciones de problemas

Un usuario, ya sea alumno o docente, podrá explorar las relaciones publicadas. Para ello, desde la página de inicio, habrá que pulsar el botón "Relaciones de problemas", donde se redirigirá a una página con las relaciones publicadas por el momento (incluidas las del propio usuario si es docente).

En la página se mostrarán dos filtros de problemas. El primero de ellos permitirá ordenar los problemas por los filtros predeterminados que se indican (**Figura 64**). El segundo es un buscador, que permitirá buscar las relaciones de problemas bien sea por título de relación, o bien por nombre de autor (ambos a la vez no) (**Figura 65**). Los filtros son excluyentes, por lo que si se filtra con uno de los del menú desplegable, no se podrá filtrar a la vez con el del buscador. En caso de querer eliminar cualquier filtro aplicado, simplemente hay que pulsar "Resetear filtro"

Dentro de la página se podrán ver las relaciones más recientes, mostrando su título, su autor, y su fecha de creación. Para ver los detalles de la relación, simplemente pulsa "Ver relación" (**Figura 66**).



Figura 62: Página de perfil personal

Figura 63: Página de edición de perfil

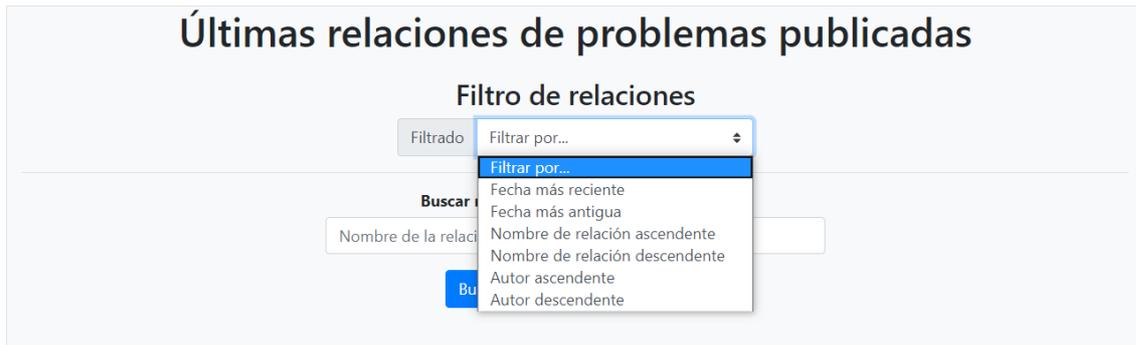


Figura 64: Filtrado de relaciones por selector

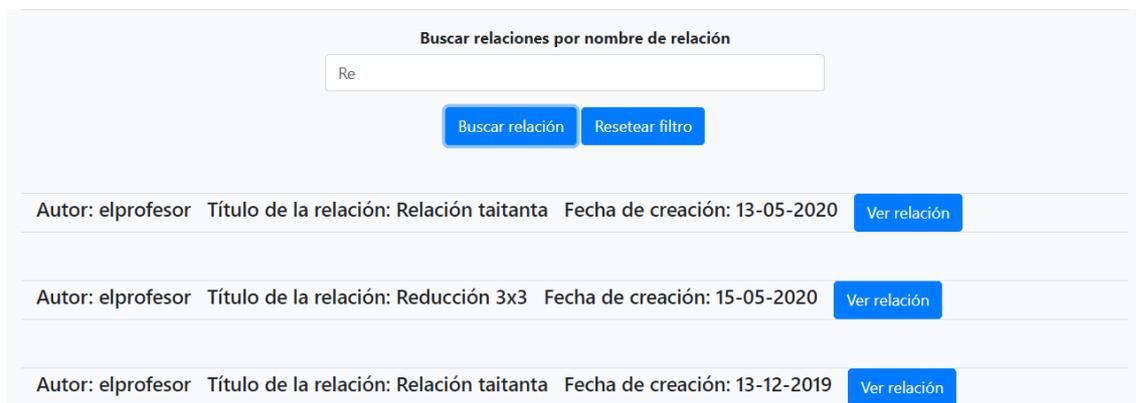


Figura 65: Filtrado de relaciones por buscador



Figura 66: Relaciones de problemas

A.2.6 Exploración de mis relaciones publicadas (docente)

Para ver este menú, es necesario que el rol del usuario sea de "profesor". Esta página es idéntica a la de exploración de problemas, solo que aquí se muestran las relaciones de problemas que se hayan publicado. Podrás filtrarlas con el buscador de la página (**Figura 67 - 1**). En caso de que quieras eliminarla alguna relación, pulsa en el botón "Eliminar relación" de la relación específica (**Figura 67 - 2**). Una ventana emergente aparecerá para indicarte si deseas eliminar la relación (**Figura 68**). Al hacer clic en "aceptar" eliminarás la relación definitivamente.

The screenshot shows the 'Mis relaciones publicadas' page. At the top, there is a blue button 'Publicar una nueva relación'. Below it is a search section titled 'Buscar relaciones por nombre de relación' with a text input field containing 'Nombre de la relación', a blue 'Buscar relación' button, and a blue 'Resetear filtro' button. A blue callout box with the number '1' points to the search input field. Below the search section is a list of three relationships. Each row contains the author 'Autor: elprofesor', the title 'Título de la relación', and the creation date 'Fecha de creación'. The first row has 'Relación taitanta' and '13-05-2020'. The second row has 'Reducción 3x3' and '15-05-2020'. The third row has 'Sustitución 3x3' and '15-05-2020'. Each row has a blue 'Ver relación' button and a red 'Eliminar relación' button. A blue callout box with the number '2' points to the 'Eliminar relación' button of the second row.

Figura 67: Filtro y botón eliminar de "Mis relaciones publicadas"

This screenshot shows the same 'Mis relaciones publicadas' page as Figure 67, but with a different set of relationships. The search section is identical. The list now contains five relationships. The first row is 'Relación taitanta' (13-05-2020). The second row is 'Reducción 3x3' (15-05-2020). The third row is 'Sustitución 3x3' (15-05-2020). The fourth row is 'Igualación 3x3' (15-05-2020). The fifth row is 'Relación taitanta' (13-12-2019). Each row has a blue 'Ver relación' button and a red 'Eliminar relación' button.

Figura 68: Página de "Mis relaciones publicadas"

A.2.7 Publicar una relación (docente)

Para publicar un problema, es necesario haber accedido a la página de "Mis relaciones publicadas". Una vez ahí, hay que pulsar el botón de "Publicar una nueva relación", que redirigirá a la página de publicación de relaciones (**Figura 69**). En ella se podrá crear la relación de problemas que se quiera publicar. Para poder publicar una relación, es necesario crear al menos un problema. Para crearlo, primero se pulsa el botón "Añadir problema". Aparecerán nuevos campos a rellenar, entre ellos el título del problema (**Figura 70 - 1**), el número de pasos esperados que dé el alumno (**Figura 70 - 2**) (para ver la información sobre cuáles son los pasos aceptados, se deberá colocar el ratón encima del texto para ver los detalles [**Figura 71**]), el método de resolución esperado (**Figura 70 - 3**), el enunciado del problema (**Figura 70 - 4**), y las dimensiones del sistema de ecuaciones a resolver (**Figura 70 - 5**).



Figura 69: Página de publicación de relaciones

A la hora de crear un sistema de ecuaciones, es necesario primero seleccionar las dimensiones del mismo. Una vez hecho esto, habrá que pulsar el botón de "Crear matriz" (**Figura 72 - 1**), se creará unos campos correspondientes a los valores de las incógnitas (**Figura 72 - 2**). Las dimensiones permitidas por el sistema son de 2x2 y 3x3.

Título del problema

Título del problema 1

Número de pasos tomados para resolver el problema (sitúa el cursor encima para saber más)

1 2

Método de resolución

Sustitución 3

Enunciado

Enunciado del problema 4

Dimensiones de la matriz a resolver

2x2 5

Crear matriz

Publicar relación

Añadir problema

Eliminar último problema

Figura 70: Datos a rellenar en la relación a publicar

Número de pasos tomados para resolver el problema (sitúa el cursor encima para saber más)

1

Método de resolución

Sustitución

Enunciado

Los pasos esperados son, según el método:

- Para el método de Reducción: Escoger un pivote, transformar una ecuación (multiplicar/ dividir), operar dos ecuaciones (restar), sustituir una incógnita en una ecuación y resolver una incógnita.
- Para el método de Sustitución: Escoger un pivote, transformar una ecuación (multiplicar/ dividir), sustituir una incógnita por una ecuación, sustituir una incógnita en una ecuación y resolver una incógnita.
- Para el método de Igualación: Escoger un pivote, transformar una ecuación (multiplicar/ dividir), sustituir una incógnita por una ecuación, sustituir una incógnita en una ecuación, resolver una incógnita e igualar dos ecuaciones

Figura 71: Información sobre los pasos

Para finalmente poder publicar la relación, una vez estén rellenos todos los campos, habrá de pulsarse el botón "Comprobar si el sistema es válido", para ver si el sistema de ecuaciones introducido es procesable o no en el sistema. (Figura 72 - 3). En caso de no serlo, la aplicación lo indicará en pantalla, y se habrá de introducir un sistema de ecuaciones válido.

Dimensiones de la matriz a resolver

2x2 ▾

Crear matriz 1

Ecuación1:

x + y =

Ecuación2:

x + y =

Comprobar si el sistema es válido 3

Figura 72: Creación del sistema de ecuaciones en la relación

A.2.8 Ver detalles de una relación de problemas

Cualquier tipo de usuario puede ver los detalles de una relación de problemas. Para ello, hay que estar en una de las páginas donde se muestren relaciones de problemas (ya sea en la página de exploración de relaciones), en la página de relaciones publicadas por un docente, o en la página de problema. Para ver los detalles de un problema, hay que pulsar el botón de "Ver detalles". Dependiendo del rol de usuario que se tenga, se verán unos datos u otros. Para los profesores se mostrará los detalles de la relación de problemas, quiénes han resuelto la relación, las notas obtenidas y la media de estas notas (**Figura 73**).

En el caso de los alumnos se mostrará un botón que estará activo si el alumno no ha resuelto la relación (**Figura 74**). En caso de haber resuelto la relación, el botón estará inactivo, y se mostrará los pasos que dio en cada problema para resolver la relación de problemas (**Figura 75**)

A.2.9 Resolver una relación de problemas (alumno)

Para resolver una relación de problemas, hay que estar en la página de detalles de la relación de problemas. Una vez dentro, se pulsa el botón de "Resolver relación" (en caso de que no la hayas resuelto). La aplicación redirigirá a una página donde se mostrará un editor de respuestas para los problemas de la relación (**Figura 76**). En el editor se podrá seleccionar un tipo de paso de los múltiples a escoger (que variarán según el método de resolución del problema [figuras 77, 78 y 79]), y dependiendo del paso escogido, se habrá de rellenar unos datos u otros. Para añadir un nuevo paso, hay que pulsar el botón "Añadir Paso". Una vez añadido, hay que seleccionar el tipo de paso que se quiera escoger y rellenar los datos que indique el paso. Se adjunta como apéndice final de este capítulo la guía de cómo rellenar estos pasos.

autor: elprofesor
Título: Relación taitanta
Fecha de creación: 13-05-2020

Problema 1

Ayyy lmao

Método de resolución:
Sustitución

Número de pasos esperados: 6

Sistema de ecuaciones:
 $1x + 2y = 3$
 $2x + 2y = 4$

elalumno 6.666666666666667
Nota media de los alumnos: 6.666666666666667

Figura 73: Página de detalles de relación de un profesor

Resolver relación de problemas

autor: elprofesor2
Título: Problemas prof2
Fecha de creación: 24-05-2020

Problema fácil

Antonio va a pescar 2 veces las lubinas que Juana piensa comprar, junto a 2 veces las lochas que Juana pensaba vender, haciendo el total de 4 pescados. Pepe quiere comprar el mismo número de lubinas que Juana iba a comprar, y dos veces el número de lochas que Juana iba a vender, quedando el total de 3 pescados. ¿Cuántas lubinas ha de pescar Antonio para que coincida con las que Pepe iba a comprar? ¿Y cuántas lochas?

Método de resolución:
Reducción

Número de pasos esperados: 6

Sistema de ecuaciones:
 $2x + 2y = 4$
 $1x + 2y = 3$

Figura 74: Página de detalles de alumno (relación no resuelta)

A.2.10 Exploración de relaciones resueltas (alumno)

La exploración de relaciones del alumno es idéntica a la de relaciones publicadas por un profesor. Como elemento diferenciador, simplemente en esta página se muestra también la nota del alumno. Por lo demás (la exploración de detalles de problemas, el buscador de problemas) son iguales a los apartados anteriores (**Figura 80**).

Ya has resuelto esta relación

autor: elprofesor

Título: Reducción 3x3

Fecha de creación: 15-05-2020

Problema 1

Resuelve por reducción. Reduce al máximo las ecuaciones

Método de resolución:

Reducción

Número de pasos esperados: 9

Sistema de ecuaciones:

$$1x + 1y + 1z = 3$$

$$2x + 1y + 1z = 4$$

$$2x + 2y + 1z = 5$$

Tu solución

Tu solución

Número de pasos dados: 8

Paso 1: Elegir pivote

Pivote escogido: z

Ecuación pivote escogida: Ecuación 1

Paso 2: Operar dos ecuaciones

Primera ecuación escogida: equation_1

Segunda ecuación escogida: equation_2

$$-1x + 0y + 0z = -1$$

Paso 3: Resolver una incógnita

$$x = 1$$

Paso 4: Operar dos ecuaciones

Primera ecuación escogida: equation_2

Segunda ecuación escogida: equation_3

$$0x + -1y + 0z = -1$$

Paso 5: Resolver una incógnita

$$y = 1$$

Paso 6: Sustituir una incógnita en una ecuación

Incógnita sustituida: x

Ecuación escogida: Ecuación 1

$$0x + 1y + 1z = 2$$

Paso 7: Sustituir una incógnita en una ecuación

Incógnita sustituida: y

Ecuación escogida: Ecuación del paso 6

$$0x + 0y + 1z = 1$$

Paso 8: Resolver una incógnita

Figura 75: Página de detalles de alumno (relación resuelta)



Figura 76: Página de resolución de relación de problemas

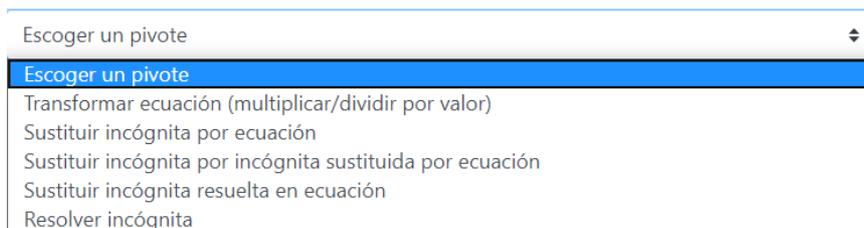


Figura 77: Pasos posibles para método de sustitución

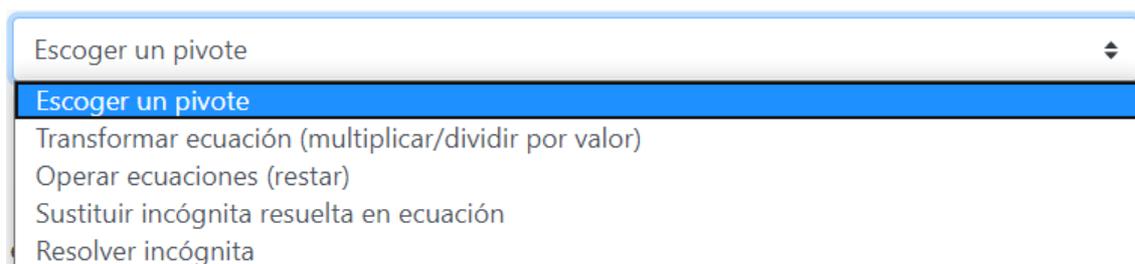


Figura 78: Pasos posibles para método de reducción

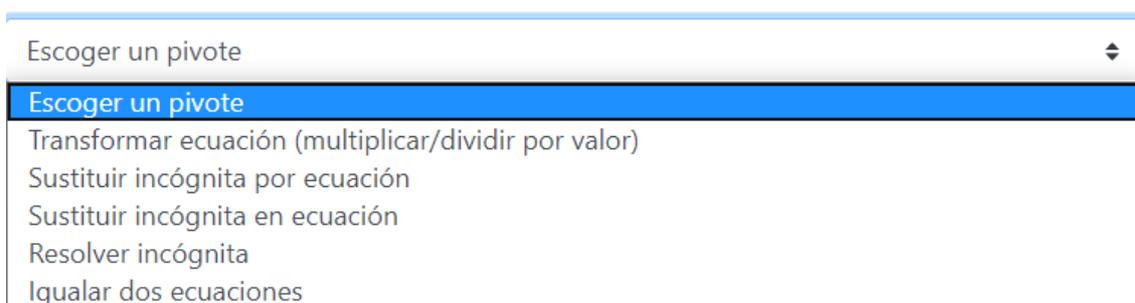


Figura 79: Pasos posibles para método de igualación



Figura 80: Página de problemas resueltos por alumno

Apéndice: Pasos de resolución de relaciones y guía de uso
Crear un pivote

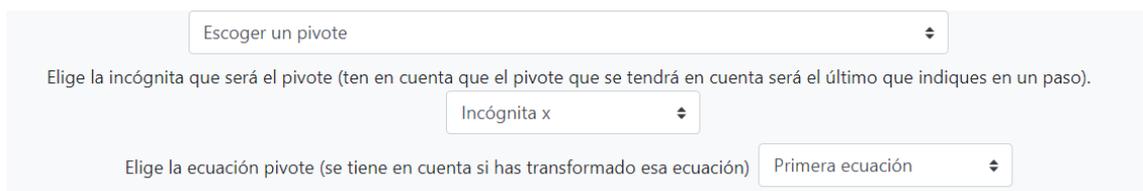


Figura 81: Crear pivote

Para crear un pivote, se debe escoger la incógnita (x o y (o z si se tiene un sistema 3x3) que hará de pivote y la ecuación que será pivote. No obstante, el dato más relevante es la propia incógnita escogida. Para que aparezcan estos selectores, se ha de pulsar en el selector del paso.

Transformar una ecuación

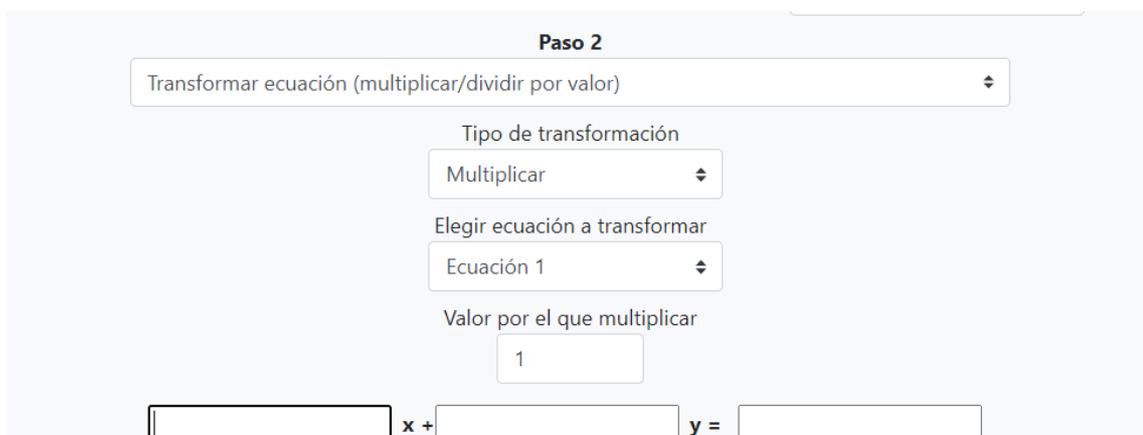


Figura 82: Transformar ecuación

Para transformar una ecuación, hay que seleccionar la operación a realizar (Multiplicar o Dividir), la Ecuación

a transformar, y el valor por el que operar. Una vez especificado esto, se indica también el resultado de la ecuación final. Además de las ecuaciones iniciales, se podrá escoger otras ecuaciones resultado de otros pasos que se quieran transformar.

Sustituir una incógnita por una ecuación

Figura 83: Sustitución incógnita por ecuación

Para sustituir una incógnita por una ecuación, se selecciona la incógnita a sustituir, y la ecuación en la que sustituir (incluidas ecuaciones transformadas de otros pasos). Finalmente, se indica el resultado de la ecuación resultante (donde se habrá de poner 0 en la casilla de la incógnita por la que se ha sustituido).

Sustituir una incógnita ya sustituida por una ecuación dentro de otra ecuación

Figura 84: Fallo de paso de sustitución de incógnita sustituida por ecuación

Figura 85: Sustitución de incógnita por incógnita ya sustituida por una ecuación

Para poder realizar este paso, es necesario previamente haber sustituido una incógnita por una ecuación. Una vez realizado tal paso, se selecciona la incógnita que se haya

sustituido por ecuación (se indica el paso al cual se refiere en el selector), la ecuación en la que sustituir (transformadas también aceptadas) y el resultado que se espera.

Sustituir una incógnita resuelta en una ecuación

Sustituir incógnita resuelta en ecuación

Elige la incógnita a sustituir en la ecuación

Elige la ecuación en la que sustituir y el resultado de la misma.
(las incógnitas que pongas con valor 0 asociado significarán que no aparecen en la ecuación)

Ecuación 1

x + y =

Añadir paso

Figura 86: Sustitución de incógnita resuelta en una ecuación

Para poder realizar este paso, es necesario haber hecho anteriormente el paso de "Resolver incógnita" (explicado a continuación). En este paso se selecciona la incógnita a sustituir, la ecuación en la que sustituir, y el resultado esperado.

Resolver una incógnita

Resolver incógnita

Elige la incógnita a resolver

x =

Figura 87: Resolución de incógnita

En este paso se selecciona la incógnita a resolver (se ha de pulsar el selector de la incógnita para que aparezca el campo de relleno) y el resultado esperado.

Operar dos ecuaciones (resta)

Paso 1

Operar ecuaciones (restar)

¿Qué ecuaciones restar? Ecuación 1 Ecuación 1

Resultado:

x + y =

Figura 88: Operación de dos ecuaciones

Este paso es exclusivo para problemas de reducción. Se debe seleccionar qué dos ecuaciones restar, y el resultado esperado.

Igualar dos ecuaciones

¿Qué ecuaciones igualar? Ten presente que se igualarán conforme al pivote que hayas indicado. Si no has indicado ninguno, se te restará la puntuación de este paso

Indica la ecuación resultante de igualar las ecuaciones (pon a 0 las incógnitas no usadas)

x + y + z =

Figura 89. Igualación de dos ecuaciones

Este paso es exclusivo en para problemas de igualación. Se escogen las dos incógnitas que se quieran igualar (incluidas transformadas) y se indica el resultado esperado. Si no se ha escogido un pivote en pasos anteriores, el paso se considerará directamente erróneo, restando puntuación.

Apéndice B

Manual de Instalación

Requerimientos:

- Dispositivo con Python 3.8
- Librerías Python:
 - Flask
 - unittest (ya instalada con Python)
 - numpy
 - bson
 - Fernet (criptografía)
 - pymongo
 - sha3
- Un explorador web (recomendado Google Chrome)
- Ficheros del proyecto

Primero se extraen los archivos en el directorio que se quiera trabajar (fichero **ecu-learning.zip**). Una vez extraídos, se abre la clase **main.py** en un editor de texto. Hay que dirigirse a las líneas indicadas en la **Figura 90**.

```
20
21 # Conexión a la BD
22 # Creado por: Daniel Cruzado Poveda
23
24 cluster = MongoClient('mongodb+srv://root:' + os.environ['PASS_MONGO'] +
25 | | | | | '@cluster0-2gbdv.mongodb.net/test?retryWrites=true&w=majority')
26
27 DB = cluster['ecu-learning']
28 users = DB['usuario']
29 problems_relationships = DB['relacion-problemas']
30 students_relationships_solved = DB['relacion-alumno']
31
32 # Direcciones de la web
```

Figura 90: Conexión base de datos

Estas líneas indican la configuración de la base de datos. Para ello, será necesario tener una base de datos en MongoDB, donde habrá que crear estas estructuras:

- Una base de datos que se llame *ecu-learning*
- Tres colecciones de datos, *usuario*, *relacion-problemas* y *relacion-alumno*

Además de esto, se necesita configurar la conexión a Mongo, la cual se puede obtener en la propia página de MongoDB (se selecciona en el Clúster creado "CONNECT" [Figura 91], "Connect your application" [Figura 92]). Una vez hecho esto, habrá que seleccionar el controlador y su versión (en nuestro caso, Python versión 3.6 o posterior). Con esto hecho se habrá creado nuestro String de conexión (Figura 93). Hay que tener presente que debe existir un usuario para poder usar la conexión (se habrá de crear uno). Además, por motivos de seguridad, la contraseña se ha guardado en una variable de entorno. Si se va a utilizar este proyecto únicamente en local, se puede saltar este paso. No obstante **se recomienda que la contraseña no se muestre directamente en el código.**

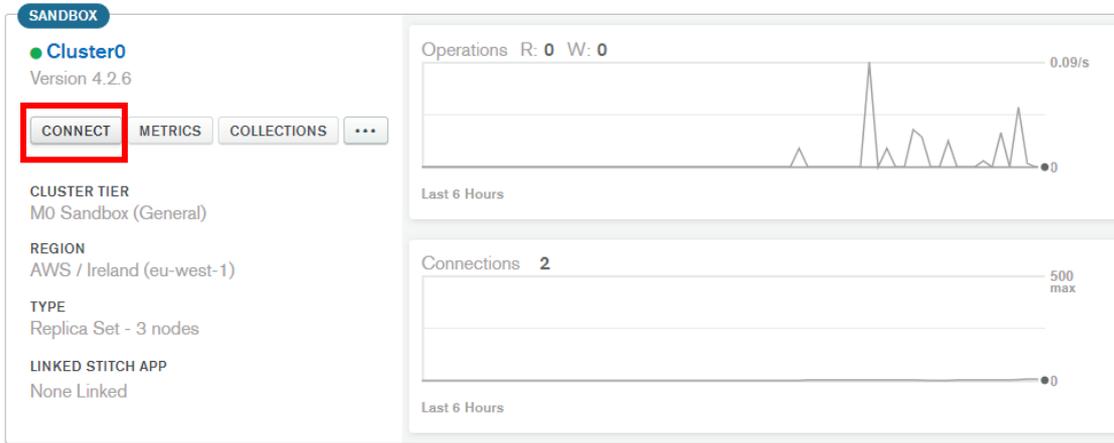


Figura 91: Conectar clúster en la aplicación

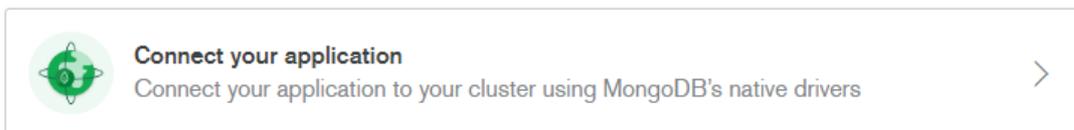


Figura 92: Conectar a nuestra aplicación

Una vez creadas las colecciones y configurado el string de conexión, se podrá ejecutar la aplicación ejecutando con python el script **main.py** (Figura 95) desde una consola de comandos. Se recomienda Visual Studio para mayor comodidad.

Finalmente, si las bases de datos están vacías, se habrá de crear los usuarios desde la aplicación.

1 Select your driver and version

DRIVER	VERSION
Python	3.6 or later

2 Add your connection string into your application code

Include full driver code example

```
mongodb+srv://root:<password>@cluster0-2gbdv.mongodb.net/test?retryWr
```

Copy

Replace `<password>` with the password for the user, `root`, and ensure all special characters are [URL encoded](#).

Figura 94: Selección de controlador y strin de conexión creado.

```
PS C:\Users\Dan\Desktop\Universidad\TFG\tfg> python main.py
* Serving Flask app "main" (lazy loading)
* Environment: production
  WARNING: This is a development server. Do not use it in a production deployment.
  Use a production WSGI server instead.
* Debug mode: on
* Restarting with stat
* Debugger is active!
* Debugger PIN: 118-068-749
* Running on http://127.0.0.1:80/ (Press CTRL+C to quit)
```

Figura 95: Ejecución del código con el comando python.



UNIVERSIDAD
DE MÁLAGA

| uma.es

E.T.S. DE INGENIERÍA
INFORMÁTICA

E.T.S de Ingeniería Informática
Bulevar Louis Pasteur, 35
Campus de Teatinos
29071 Málaga