

Grado en Ingeniería Informática

Trabajo Fin de Grado

Aplicación Web para el Aprendizaje de Programación

Autor

Julio Blanco Arribas

Tutores

Agustín Salgado de la Nuez

Nelson Manuel Monzón López

Las Palmas de Gran Canaria

Marzo 2021

Agradecimientos

Quiero comenzar dando las gracias a mis padres y a mi mujer que han creído en mi hasta el final, sin su apoyo no hubiera llegado hasta la meta. Aprovechar este momento para decirle a mi suegro allá donde esté que ya soy Ingeniero Informático y que su hija es ya una gran Trabajadora Social, se te echa de menos.

También quiero agradecer a mis compañeros y amigos como Yonay Cabrera López por su ayuda y apoyo durante tantos años, Eduardo Mendoza García, Carlos Rodríguez Vega, Jonatan Acosta García, y Stefan Hautz entre muchos otros por todas las experiencias que hemos compartido juntos y su apoyo.

Y por último y no menos importante, doy las gracias a mi tutor, Agustín Salgado de la Nuez por animarme durante estos años a esforzarme en el trabajo para realizarlo lo mejor posible, así como al resto de profesores que, de un modo u otro, han colaborado en formarme como Ingeniero Informático.

Muchas gracias a todos.

Resumen

Test Yourself consiste en una plataforma web en la que sus usuarios puedan adquirir un apoyo extra en el aprendizaje inicial de programación u otros ámbitos, debido a su interfaz y su manejo es aplicable a cualquier nivel de enseñanza. Este apoyo lo obtendrán a través de la realización de diferentes cursos creados por los propios usuarios de la plataforma. Estos cursos a su vez, estarán compuestos por preguntas tipo test y en algunos casos podrán tener un apoyo de manera visual mediante animaciones con el fin de que sean capaces de razonar mejor la respuesta a las preguntas planteadas.

El proyecto está dividido en dos módulos, un módulo de administradores y otro de alumnos, ambos pueden crear y realizar cursos, pero los primeros tendrán un control total sobre lo que se publica en la plataforma con el fin de mantener un orden en ella.

Índice

Capítulo 1. Introducción	9
1.1 Motivación	9
1.2 Objetivos	10
1.3 Justificación de competencias	10
Capítulo 2. Estado del arte	15
2.1 Webs de aprendizaje existentes	15
2.2 Fortalezas y debilidades frente a estas plataformas	19
Capítulo 3. Recursos utilizados	21
3.1 Software	21
3.2 Hardware	25
Capítulo 4. Presupuesto	27
4.1 Costes de Personal	27
4.2 Costes de Hardware	27
4.3 Desglose de costes	28
4.4 Planificación del proyecto	30
Capítulo 5. Desarrollo del proyecto	33
5.1 Requisitos previos	33
5.2 Metodología de desarrollo Scrum	34
5.3 Sprint 0. Primeros pasos	40
5.4 Sprint 1 y 2. Añadir inicio de sesión, preguntas y cursos	46
5.5 Sprint 3. Interfaces personalizadas y guías de uso	56
Capítulo 6. Conclusiones y trabajo futuro	59
6.1. Conclusiones	59
6.2. Trabajo futuro	60
Capítulo 7. Anexos	63
7.1 Manual de usuario	63
7.2 Anexo 2. Modelo de datos de la aplicación	73
7.3 Anexo 3. Diagramas de casos de uso de la aplicación	75
8. Bibliografía	77

Figuras

FIGURA 1 UDEMY - INICIO	16
FIGURA 2 SCRATCH – ACTIVIDAD	16
FIGURA 3 SCRATCH – REGISTRO.....	17
FIGURA 4 GOCONQR	18
FIGURA 5 MOODLE DE LA ULP GC.....	19
FIGURA 6 LOGO INTELLIJ	21
FIGURA 7 LOGO VISUAL STUDIO CODE	21
FIGURA 8 LOGO GIT.....	22
FIGURA 9 LOGO SPRING BOOT.....	22
FIGURA 10 LOGO MAVEN.....	22
FIGURA 11 LOGO VUE	23
FIGURA 12 LOGO WEBPACK.....	23
FIGURA 13 LOGO POSTGRESQL.....	23
FIGURA 14 LOGO VUETIFY.....	24
FIGURA 15 LOGO FLEXBOX.....	24
FIGURA 16 LOGO CSS	24
FIGURA 17 LOGO POSTMAN	25
FIGURA 18 PROCESO DE SCRUM.....	34
FIGURA 19 SPRINT 0 - FORMULARIO DE PRUEBA PARA CREAR UN COCHE.....	41
FIGURA 20 SPRINT 0 - CONTROLLER DE LOS COCHES	42
FIGURA 21 SPRINT 0 - DTO DEL COCHE.....	42
FIGURA 22 SPRINT 0 - SERVICE DEL COCHE	42
FIGURA 23 SPRINT 0 - INTERFAZ DEL REPOSITORIO DEL COCHE	42
FIGURA 24 SPRINT 0 - REPOSITORIO DEL COCHE.....	43
FIGURA 25 BASE DE DATOS POSTGRESQL.....	43
FIGURA 26 CROSSORIGIN	44
FIGURA 27 USUARIOS DE LA BASE DE DATOS.....	45
FIGURA 28 FORMULARIO PARA INICIO DE SESIÓN	47
FIGURA 29 VUEX.....	47
FIGURA 30 UNA VEZ FINALIZADO EL INICIO DE SESIÓN CON ÉXITO	48
FIGURA 31 TABLA DE CURSOS.....	49
FIGURA 32 CURSOS ASIGNADOS AL USUARIO	49
FIGURA 33 CURSOS DISPONIBLES	51
FIGURA 34 PETICION REST DEL CLIENTE PARA ABRIR UN CURSO.....	51
FIGURA 35 VISTA DE CURSO COMO CREADOR	52
FIGURA 36 CREACIÓN Y ELIMINACIÓN DE PREGUNTAS EN EL REPOSITORIO DEL SERVIDOR	52
FIGURA 37 ESTRUCTURA DE LOS CURSOS.....	53
FIGURA 38 VISTA DE CURSO COMO NO CREADOR	53
FIGURA 39 FOTOGRAMAS DE LA ANIMACIÓN	54
FIGURA 40 SVG DE LA ANIMACIÓN DE LA FIGURA SUPERIOR	54
FIGURA 41 VISTA DEL PERFIL DE UN USUARIO NORMAL.....	55
FIGURA 42 ESTRUCTURA DE LA PARTE VISUAL DE LA PÁGINA DEL PERFIL	55
FIGURA 43 VISTA DEL PERFIL DE UN USUARIO ADMINISTRADOR	56

FIGURA 44 MANUAL DE USUARIO - PÁGINA PRINCIPAL.....	63
FIGURA 45 MANUAL DE USUARIO - CURSOS DISPONIBLES	64
FIGURA 46 MANUAL DE USUARIO – REGISTRO	64
FIGURA 47 MANUAL DE USUARIO - INICIO DE SESIÓN	65
FIGURA 48 MANUAL DE USUARIO - INICIO DE SESIÓN - FORMATO INCORRECTO DEL EMAIL..	65
FIGURA 49 MANUAL DE USUARIO - INICIO SESIÓN - USUARIO INCORRECTO	66
FIGURA 50 MANUAL DE USUARIO - REGISTRO - FORMATO INCORRECTO DEL EMAIL.....	66
FIGURA 51 MANUAL DE USUARIO - REGISTRO - USUARIO EXISTENTE	66
FIGURA 52 MANUAL DE USUARIO - PÁGINA PRINCIPAL - SESIÓN INICIADA	67
FIGURA 53 MANUAL DE USUARIO - CURSOS DISPONIBLES - SESIÓN INICIADA.....	67
FIGURA 54 MANUAL DE USUARIO - CREAR CURSO	68
FIGURA 55 MANUAL DE USUARIO - CURSOS INSCRITOS	68
FIGURA 56 MANUAL DE USUARIO - CURSO – EDICIÓN.....	69
FIGURA 57 MANUAL DE USUARIO - CURSO - ANIMACIÓN	69
FIGURA 58 MANUAL DE USUARIO - PERFIL DE ADMINISTRADOR	70
FIGURA 59 MANUAL DE USUARIO - CAMBIO DE CONTRASEÑA	70
FIGURA 60 MANUAL DE USUARIO - CREAR ADMINISTRADOR	71
FIGURA 61 MANUAL DE USUARIO - CURSOS DISPONIBLES - FILTRO	71
FIGURA 62 MANUAL DE USUARIO - CURSOS ASIGNADOS - FILTRO.....	72
FIGURA 63 MODELO DE DATOS	73
FIGURA 64 DIAGRAMA CASOS DE USO - ALUMNO REGISTRADO	75
FIGURA 65 DIAGRAMA CASOS DE USO - ALUMNO REGISTRADO DUEÑO DEL CURSO	75
FIGURA 66 DIAGRAMA CASOS DE USO – ADMINISTRADOR	76
FIGURA 67 DIAGRAMA CASOS DE USO - USUARIO NO REGISTRADO	76

Tablas

TABLA 1 REFLEJA EL COSTE DE ELEMENTOS USADOS PARA EL DESARROLLO	28
TABLA 2 COSTES TOTALES MENSUAL Y ANUAL DE UN TRABAJADOR.....	28
TABLA 3 PLANIFICACIÓN DEL PROYECTO	30
TABLA 4 PILA DE PRODUCTO DEL SPRINT 0	36
TABLA 5 PILA DE PRODUCTO DEL SPRINT 1	38
TABLA 6 PILA DE PRODUCTO DEL SPRINT 2	39
TABLA 7 PILA DE PRODUCTO DEL SPRINT 3	39
TABLA 8 HISTORIAS DE USUARIO SPRINT 0 TEST YOURSELF.....	40
TABLA 9 HISTORIAS DE USUARIO SPRINT 1 TEST YOURSELF.....	46
TABLA 10 HISTORIAS DE USUARIO SPRINT 2 TEST YOURSELF.....	50
TABLA 11 HISTORIAS DE USUARIO SPRINT 3 TEST YOURSELF.....	57

Capítulo 1. Introducción

Actualmente un pequeño número de los alumnos de nuevo ingreso en la Escuela de Ingeniería Informática traen consigo conocimientos básicos de programación, lo cual significa que un gran número de ellos o bien no han visto código nunca o si lo han visto no entienden una lógica simple.

Por ello, es propenso que al empezar las primeras asignaturas de programación se vean superados y busquen una manera de mejorar su nivel.

Este proyecto se centra en ofrecer un refuerzo a esos alumnos a través de una plataforma web enfocada a sus necesidades, con un uso sencillo y donde los alumnos mejoren su nivel de programación.

El proyecto se dividió en dos módulos. Por un lado, tenemos el módulo de administración, donde el administrador o usuario normal podrá crear nuevos cursos compuestos de diferentes preguntas para incorporar en la plataforma, además de poder modificar y eliminar si se diese el caso.

En el módulo cliente, el usuario o alumno podrá visualizar todos los cursos y crear nuevos, ver en los que esté inscrito y con ello realizar sus respectivas preguntas. Además, dispondrá de un apartado donde podrá ver sus progresos hasta ese momento.

1.1 Motivación

La principal motivación para la realización del proyecto es el desarrollo de una aplicación web utilizando herramientas con las que no se ha trabajado durante el transcurso del grado con el fin de aprender otros tipos de desarrollo web que son bastante solicitados en el mundo laboral, en este caso se escogió Spring Boot y Vue.js.

Los principales esfuerzos se han centrado en profundizar en el uso de las herramientas y en la realización de mejor desarrollo de la aplicación web. Esto último lo considero un punto importante ya que la intención es poner en producción la aplicación web.

Por otro lado, el hecho de ofrecer una herramienta de apoyo a los alumnos que puedan necesitarlo y a quienes está dirigida esta aplicación web, supone una gran motivación para desarrollar el proyecto.

1.2 Objetivos

Los objetivos de este proyecto son los siguientes:

- Aprender a desarrollar una página web desde cero, haciendo uso de nuevas tecnologías y herramientas, partiendo de una pequeña base obtenida en el grado.
- Aprender nuevas tecnologías, ampliar aún más los conocimientos de ingeniería del software.
- Diseñar y desarrollar la interfaz gráfica de acceso y su manejo de la información, abarcando parámetros de usabilidad y la experiencia de usuario.
- Uso de un flujo de trabajo donde se apliquen técnicas de integración y entrega continua.
- Desarrollo de una plataforma online que ayude a los alumnos de nuevo ingreso en su iniciación de las materias.
- Desarrollo de una plataforma online que permita realizar cursos de distintos niveles y lenguajes a los usuarios.
- Desarrollo de una plataforma online que permita visualizar en ciertas preguntas animaciones con el fin de que comprendan la lógica como pueda ser el flujo del código en preguntas relacionadas con la programación.
- Desarrollo de una plataforma online que permita a los administradores crear nuevos cursos, editarlos o eliminarlos.

1.3 Justificación de competencias

En este apartado se citan algunas de las competencias previamente mencionadas en el documento TFT01 donde

posteriormente se justificarán cada una de ellas, además se señalan algunas otras que se han cumplido.

- ***CIIO1. Capacidad para diseñar, desarrollar, seleccionar y evaluar aplicaciones y sistemas informáticos, asegurando su fiabilidad, seguridad y calidad, conforme a principios éticos y a la legislación y normativa vigente.***

Esta competencia, se demuestra en la aplicación web diseñada, la cual se ha obtenido durante el grado. Las tareas que se han realizado se ajustan a lo exigido, dado que abarcan desde la propuesta inicial del proyecto a su ejecución, pasando por un proceso final de verificación mediante el cual se asegura el correcto funcionamiento de la aplicación.

- ***CIIO2. Capacidad para planificar, concebir, desplegar y dirigir proyectos, servicios y sistemas informáticos en todos los ámbitos, liderando su puesta en marcha y su mejora continua y valorando su impacto económico y social.***

Para cumplir esta competencia se ha hecho uso de herramientas como *Trello* donde se disponía de un *Backlog*, diferentes *Sprint* y un apartado de las tareas realizadas, todo con el fin de una planificación óptima. Durante el desarrollo del proyecto algunas historias tardaron más de lo previsto debido al tiempo de incertidumbre y que se ha desarrollado por un solo miembro.

- ***CIIO8. Capacidad para analizar, diseñar, construir y mantener aplicaciones de forma robusta, segura y eficiente, eligiendo el paradigma y los lenguajes de programación más adecuados.***

Para la realización de este proyecto se han utilizado los frameworks *Spring Boot* y *Vue*. En el caso del primero permite mediante el lenguaje de programación Java, tener el programa bien estructurado siguiendo un patrón de arquitectura de software hexagonal, lo que significa que accedas a tu lógica de negocio mediante interfaces, esto permite una mejor comprensión del código y una mejor estructura. También se hará uso de los DTO (Data Transfer Object), para no exponer toda la lógica de negocio al exterior. Con esto se consigue mantener separados los datos y la lógica de negocio de la interfaz de usuario.

En el caso del segundo, *Vue* permite que la aplicación sea una SPA (Single Page Application), esto evita que ciertos componentes que estén repetidos en las páginas se tengan que volver a cargar durante la navegación y, por tanto, desaparece la sensación de lentitud.

Como se ha mencionado anteriormente ambas partes son independientes por lo que se podría cambiar el *Front-end* posteriormente sin tener mucha dificultad.

• *IS01. Capacidad para desarrollar, mantener y evaluar servicios y sistemas software que satisfagan todos los requisitos del usuario y se comporten de forma fiable y eficiente, sean asequibles de desarrollar y mantener y cumplan normas de calidad, aplicando las teorías, principios, métodos y prácticas de la ingeniería del software.*

Al hacer uso de buenas prácticas como nombres adecuados en el código, una estructura correcta, que sea desacoplado, corto y eficiente, se consigue cumplir la competencia.

• *IS02. Capacidad para valorar las necesidades del cliente y especificar los requisitos software para satisfacer estas*

necesidades, reconciliando objetivos en conflicto mediante la búsqueda de compromisos aceptables dentro de las limitaciones derivadas del coste, del tiempo, de la existencia de sistemas ya desarrollados y de las propias organizaciones.

Hoy en día no existe ninguna página web enfocada a la EI que busque cubrir esas necesidades que puedan tener los alumnos, por lo que sería algo completamente innovador. Aunque han existido herramientas parecidas, como un Moodle de refuerzo que fue impulsado por un miembro de la delegación de la EI, al final se trataba de un mismo entorno de programación donde lo único que se encontraban eran ejercicios similares a los propuestos en clase.

• IS03. Capacidad de dar solución a problemas de integración en función de las estrategias, estándares y tecnologías disponibles.

El uso de una arquitectura hexagonal para el Back-end implica que, si un desarrollador se integra después del comienzo en el proyecto, tendría una adaptación mucho más cómoda para poder empezar a trabajar.

• IS04. Capacidad de identificar y analizar problemas y diseñar, desarrollar, implementar, verificar y documentar soluciones software sobre la base de un conocimiento adecuado de las teorías, modelos y técnicas actuales.

En este caso, el uso de Vue.js en el Front-end hace que el navegador web no tenga que estar constantemente recargando la página y facilita el reusar componentes, por tanto, se tiene un desarrollo más eficiente y se evitan errores innecesarios, a la vez que el usuario se beneficia de una navegación más rápida.

Capítulo 2. Estado del arte

En este apartado se aborda el estado actual de webs que tengan un concepto similar o parecido al proyecto realizado.

Observando webs que puedan tener como objetivo un público similar, en este caso, personas interesadas en adquirir unos conocimientos como base o afianzar lo que ya conoce o desde el punto de vista de un programador informático aprender a programar si no lo han hecho uso nunca o se encuentran en sus comienzos.

2.1 Webs de aprendizaje existentes

A continuación, se hace mención a diferentes páginas webs que puedan tener un enfoque parecido como puede ser Udemy, Scratch, GoConqur o Typeform.

2.1.1 Udemy

Por ejemplo, se puede observar en el campo de la informática un creciente interés en aprender diversos lenguajes, podemos encontrarnos con páginas que nos ofrecen cursos concretos de lo que estemos buscando, pero en éstos suele ser un requisito fundamental el pedir el tener ciertos conocimientos previos y tratan de dar mucha materia en poco tiempo.

En la actualidad, una de las páginas más conocidas para la realización de cursos en español es Udemy.



En el ámbito de la programación, sin embargo, no es quizás uno de los mejores ejemplos a comparar pues existen otras páginas que, si bien no ofrecen cursos como tal, proponen actividades a un nivel más básico y que ayudan a entender la lógica de la programación de una manera más fácil, con una relación más cercana entre alumnos y profesores. Para este caso usaremos la página web de Scratch.



En dicha web podremos realizar animaciones añadiendo condiciones, repeticiones, eventos y comprender cómo se ejecutan desde el punto de vista de un programador. Dichas actividades pueden ser a su vez compartidas con el resto de usuarios, para que vean cómo fue realizado. En el caso de querer hacer el nuestro propio y compartirlo nos pedirá estar registrados, para ello tenemos un simple registro, que facilita mucho las cosas.

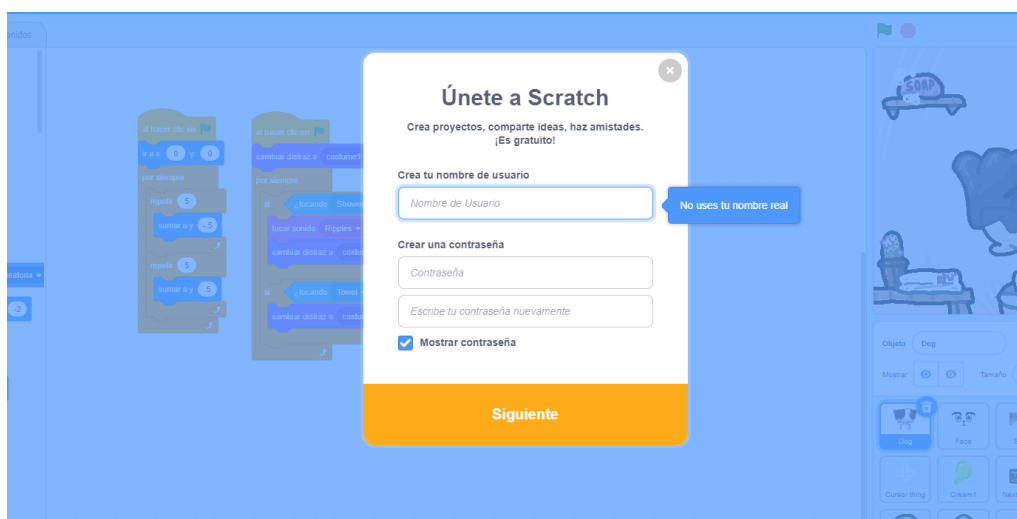


FIGURA 3 SCRATCH – REGISTRO

Una vez iniciada la sesión se permite al usuario guardar en su perfil las actividades realizadas, así como compartirlas con el resto de usuarios.

Llegados a este punto, lo más importante a destacar es que se centran en aportar una base de la lógica de programación y cómo se ejecutan los procesos, siendo una de las bases más importantes que un programador informático necesita adquirir, antes de adentrarse a programar en cualquier lenguaje y, aunque el enfoque que se le da aquí es para niños de edades entre los 5 y los 10 años, es un buen comienzo para cualquier persona interesada en aprender desde la base, por lo intuitivo y cómodo que resulta trabajar en ella.

2.1.3 GoConqr o Typeform

En cuanto a GoConqr o Typeform estos son muy parecidos entre ambos, permiten crear encuestas o test de una manera rápida y sencilla pero que nos señalan a la hora de realizarlo cuáles son las preguntas que hemos respondido mal y por tanto se pierde ahí la oportunidad de que el alumno piense y razone en que pregunta ha podido fallar. Tampoco se nos da en ninguno la posibilidad de añadir animaciones que ayuden a comprender la pregunta o razonar la respuesta.



FIGURA 4 GOCONQR

2.1.4 Moodle

En el caso del Moodle se permiten hacer cuestionarios de tipo test, eligiendo entre un banco de preguntas. Es el más usado en las universidades, en la propia Universidad de Las Palmas de Gran Canaria se usa actualmente. En estos cuestionarios los bancos de preguntas no suelen renovarse ya que dependen directamente de los profesores y a pesar de que la materia no cambia las preguntas terminan cayendo en manos de los alumnos de otros que ya los han hecho. Esto implica que los alumnos ya saben a lo que se van a enfrentar y por otro lado son preguntas que puede que no abarquen todo el temario.

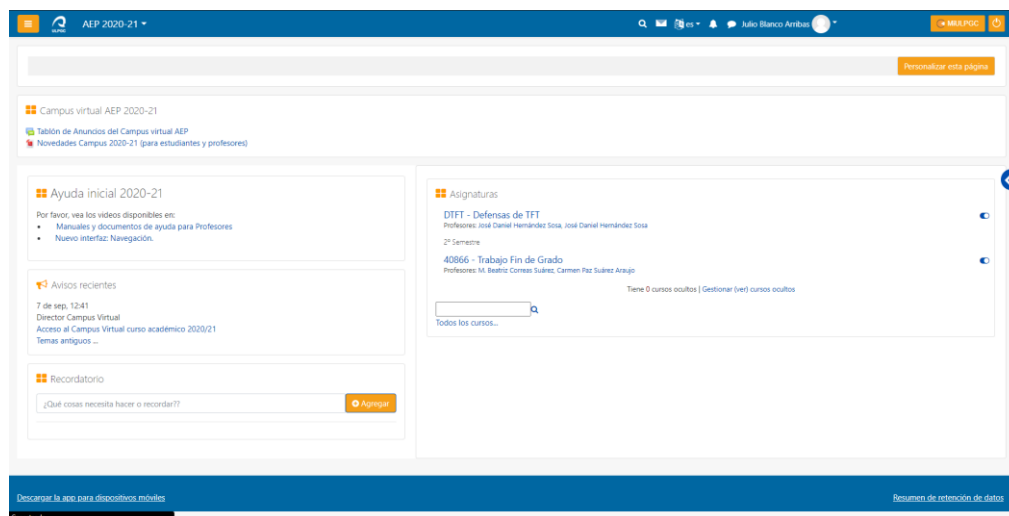


FIGURA 5 MOODLE DE LA ULPGC

2.2 Fortalezas y debilidades frente a estas plataformas

Una vez repasadas algunas de las plataformas que tienen un enfoque parecido, podemos observar que muchas ellas ya sean orientadas a la programación o abiertas a cualquier ámbito tienen una limitación en cuanto a los recursos que ofrece, ya que las preguntas de los cursos a los que se les da opción a los usuarios de matricularse suelen estar fijadas y no cambian con el paso del tiempo. GoConqr en este caso puede que sea la que mejor parada sale, ya que sí permite que sea la comunidad la que haga los test y los cursos, por tanto, la vuelve más atractiva de cara a los alumnos.

Lo que no vemos es ninguna plataforma es una combinación de estas fortalezas, que es lo que pretende hacer Test Yourself, una plataforma web o comunidad, donde los alumnos puedan crear cursos de manera autónoma, mantenerlos actualizados durante el paso del tiempo ya sean ellos o moderadores y con el atractivo de que puedan interactuar en ciertas preguntas con animaciones. Cabe mencionar por último y no menos importante que tanto el registro como la inscripción a cursos es gratuita, ya que en Udemy por ejemplo, los cursos son de pago en su mayoría.

Capítulo 3. Recursos utilizados

3.1 Software

- **IntelliJ IDEA versión 2018.3.5:** se trata de un entorno de desarrollo integrado. Dicho entorno al haber sido usado durante varios años, se ha optado por emplearlo en este proyecto con el fin de minimizar el riesgo de encontrar problemas que dificultaran el progreso. Su finalidad ha sido usarlo para el Back-end realizado con Spring Boot.



FIGURA 6 LOGO INTELLIJ

- **Visual Studio Code 1.34:** es un entorno de desarrollo integrado, el cual se ha usado para la parte de Front-end realizada con Vue.js, lo que permite poder trabajar en paralelo con el Back-end, al disponer de la API corriendo con IntelliJ.



FIGURA 7 LOGO VISUAL STUDIO CODE

- **Git:** usado para el control de las versiones de la aplicación web.



FIGURA 8 LOGO GIT

- **Spring Boot:** entre las ventajas de este Framework de Java se encuentran su condición de software libre y la documentación disponible para el desarrollo de la aplicación web.



FIGURA 9 LOGO SPRING BOOT

- **Maven:** entre otras cosas, en este proyecto se ha usado para añadir las dependencias y aprovechar la comprensión que aplica al proyecto en un .jar.



FIGURA 10 LOGO MAVEN

- **Vue:** la gran ventaja a la hora de usar este Framework de JavaScript es la facilidad con la que se aprende y la abundante documentación disponible.



FIGURA 11 LOGO VUE

- **Webpack:** la función es *transpilar* el código de Vue y ES6 a código de JavaScript para que pueda ser interpretado por los navegadores web.



FIGURA 12 LOGO WEBPACK

- **PostgreSQL 4 v4:** se usa postgresQL para crear y administrar la base de datos de la aplicación web.



FIGURA 13 LOGO POSTGRESQL

- **Vuetify:** se trata de un Framework enfocado a la interfaz gráfica del usuario, que usa los mismos componentes que contiene Vue y, a su vez, hace que estos sean configurables.



FIGURA 14 LOGO VUETIFY

- **Flexbox:** viene incluido en CSS3 y se usa para poder estructurar los componentes visuales de la aplicación web.



FIGURA 15 LOGO FLEXBOX

- **CSS-GRID:** viene incluido en CSS3 y se usa para que la web sea responsive.



FIGURA 16 LOGO CSS

- **POSTMAN:** se usa para lanzar peticiones a una API REST en nuestro caso al servidor de Spring Boot, principalmente se usa para hacer pruebas.



FIGURA 17 LOGO POSTMAN

3.2 Hardware

Para el desarrollo de la aplicación web se ha usado el siguiente material:

- Torre de montaje propio:
 - Monitores: BenQ 27" y Samsung 22".
 - Procesador: Intel Core i7 10700k.
 - RAM: 64GB DDR4.
 - Gráficos: NVIDIA RTX 3080.
 - Almacenamiento: SSD M.2 500GB, SSD 1TB, SSD 400GB.
- Portátil MacBook Pro (mediados 2012):
 - Monitor: pantalla integrada de 13".
 - Procesador: Intel Core i5 3210M.
 - RAM: 8GB DDR3.
 - Gráficos: Intel HD 4000 1512MB VRAM.
 - Almacenamiento: SSD 500GB.

Capítulo 4. Presupuesto

Para el presupuesto se muestra detalladamente cada apartado para finalmente estimar los costes de la realización de este proyecto. Dicho presupuesto se divide en dos apartados, hardware y costes del personal.

Los costes de software no se tienen en cuenta en este caso debido a que se usa todo con licencia libre o universitarias, por tanto, el coste es cero en este apartado. Finalmente, se realizará un desglose final resumiendo los costes totales, lo que permitirá tener una idea más precisa de la elaboración del proyecto.

4.1 Costes de Personal

En este punto se necesitaría un desarrollador Full Stack, lo que significa que abarque todas las capas de un proyecto. Es un tipo de puesto bastante demandado porque puede desarrollar rápidamente un prototipo inicial con el que explorar las ideas del producto para posteriormente venderlo. El sueldo medio bruto para este tipo de desarrollador en la isla de Gran Canaria ronda los 1.200€/1.400€ mensuales en 2021.

4.2 Costes de Hardware

Los costes de hardware representan todos aquellos sistemas informáticos que serían requeridos durante la etapa de desarrollo. Dichos costes los vemos reflejados en la siguiente tabla.

HARDWARE	IMPORTE	CANTIDAD	TOTAL (+IGIC)
Torre de montaje propio	1.900,00€	1	2.033,00€
Monitores BenQ 27" y Samsung 22"	315,00€	1	337,05€
Teclado y ratón Logitech G513 y G502	140,00€	1	149,80€
MacBook Pro Mid 2012	1.200,00€	1	1.284,00€
Total	3.555,00€	4	3.803,85€

TABLA 1 REFLEJA EL COSTE DE ELEMENTOS USADOS PARA EL DESARROLLO

Por tanto, tenemos un total de 3.803,85€ dedicados a hardware. Se podría haber desarrollado en un solo equipo reduciendo en más de la mitad el coste, pero hubiese supuesto más tiempo de desarrollo y complejidad. Por otro lado, hay que tener en cuenta que este producto una vez se termina el desarrollo se sigue disponiendo de él, por tanto, no es una inversión exclusiva que se pierda.

4.3 Desglose de costes

Finalmente, se muestra un coste total para un mes y un año.

	COSTE MENSUAL	COSTE ANUAL
HARDWARE	60,00€	720,00€
SOFTWARE	0,00€	0,00€
PERSONAL	1.300,00€	15.600,00€
TOTAL	1.360,00€	16.320,00€

TABLA 2 COSTES TOTALES MENSUAL Y ANUAL DE UN TRABAJADOR

Suponemos un coste mensual en gastos de luz y mantenimiento de los equipos de 60€ y un sueldo bruto de 1.300€ para un desarrollador. A ese coste anual de 16.320 € se deben sumar los 3.803,85 € del hardware como gasto inicial.

En el caso de un autónomo habría que eliminar el gasto de hardware en caso de que ya posea de dicho material, que suele ser lo normal, a no ser que se necesiten equipos en concreto de cara al desarrollo, no es el caso. Por tanto, sí debería pagarse la luz, los gastos de licencias en caso de necesitarlas y debería tener en cuenta su propio sueldo. Hablamos de un total de 19.403,85€ de gastos durante el primer año o de 19.902,85€ en el caso de no tener licencias gratuitas. A esto siempre hay que sumarle la cuota de autónomo, la cual ronda el 30% actualmente con lo que nos iríamos a un total de 25.225€ y 25.873,70€ respectivamente.

Presupuestando para este caso que el desarrollo de la aplicación llevara un año para tenerlo acabado si se valora en 30.000€ el beneficio obtenido sería de 4.500€ aproximadamente.

En una situación en la que este desarrollo se solicita a una empresa el hardware suele ser proporcionado por la misma, así como las licencias en caso de ser de pago, esto implica que únicamente se tendrá de gastos el salario del trabajador y la luz consumida.

En función del tipo de contrato que se le haga al trabajador hablamos de un gasto mínimo anual de 16.500€ más 200€ en consumo de luz aproximadamente, haciendo un total de 16.700€.

De cara a una empresa que ya paga por sus licencias y tiene el hardware disponible el coste de hacer la aplicación se reduce considerablemente y por tanto podrá competir contra otras empresas que carezcan de ello y los autónomos. Presupuestando para este caso que el desarrollo de la aplicación llevara un año para tenerlo acabado con un trabajador si se valora en 20.000€ el beneficio obtenido sería de 3.300€.

4.4 Planificación del proyecto

FASES	DURACIÓN ESTIMADA (horas)	DURACIÓN REAL (horas)	TAREAS
Creación del Product Backlog	20	15	Análisis de la situación actual. (10 horas)
			Primera toma de requisitos de usuario y priorización de necesidades. (10 horas)
Sprint Zero	40	35	Definición de requisitos del software. (10 horas)
			Diseño de la base de datos. (15 horas)
			Diseño de la interfaz de usuario. (15 horas)
Fase de desarrollo de las distintas releases	200	240	<p>Se estima el desarrollo en tres fases de implementación, la interacción con la base de datos, el acceso de distintos usuarios en cada fase e implementación iterativa de la interfaz de usuario.</p> <ul style="list-style-type: none"> • Release 1 (80 horas) • Release 2 (60 horas) • Release 3 (60 horas)
Documentación / Presentación	40	40	Documentación de la aplicación desarrollada
			Edición de la memoria del proyecto realizado
			Preparación de la defensa pública

TABLA 3 PLANIFICACIÓN DEL PROYECTO

En líneas generales, se puede decir que estuvo acertada la estimación inicial para desarrollar por completo el proyecto. Sin embargo, hubo pequeños desajustes de horas en las distintas fases, sobre todo en la fase de desarrollo como se puede apreciar en la Tabla 3. La principal causa de esa diferencia de horas fue que ciertas historias de usuario se estimaron de manera optimista a pesar de su complejidad y el desconocimiento técnico que se tenía. Debido a que las primeras fases quedaron algo sobreestimadas la duración total del proyecto no se debió mucho la planificada, quedando un exceso total de aproximadamente 30 horas, pero este exceso recayó sobre la parte importante del desarrollo y al no poder alargarse más provocó que algunas funcionalidades quedasen desarrolladas con menos dedicación.

Capítulo 5. Desarrollo del proyecto

En este capítulo se describen las distintas fases de desarrollo por las que ha pasado el proyecto.

5.1 Requisitos previos

- Programación web:

Tener ciertos conocimientos previos de desarrollo web antes de comenzar este proyecto es imprescindible para poder realizar un proyecto de estas características en su totalidad. Una parte de estos conocimientos fueron obtenidos con la asignatura de Arquitectura del Software aunque, si bien es cierto, las prácticas de empresa aportaron la mayor parte.

- Uso de varios frameworks:

Spring Boot ha sido escogido como framework para el desarrollo de este proyecto ya que aporta facilidad a la hora de desarrollar nuevas funcionalidades.

Por otro lado, se escogió Vue en contraposición de React debido a la diferencia de aprendizaje entre uno y otro, aunque si bien es cierto que React tiene abundante documentación y pertenece a Facebook, Vue está ganando terreno y puede ser una buena apuesta.

- Desarrollo ágil de software:

Para el desarrollo de la aplicación web se ha empleado la metodología Scrum, por tanto, se necesita conocer en qué consiste un desarrollo ágil del software y cómo hacer un uso correcto del mismo. Dichos conocimientos fueron obtenidos continuamente en la rama de Ingeniería del Software.

5.2 Metodología de desarrollo Scrum

Se hace uso de la metodología ágil Scrum para facilitar la organización y el desarrollo de este proyecto. Como se aprendió en varias asignaturas del Grado de Ingeniería Informática, uno de los motivos por los que se decidió escoger esta metodología fue que es ideal para proyectos que necesitan tener resultados pronto y que acepten cambios durante el desarrollo, esto equivale a que la flexibilidad, productividad y competitividad van de la mano y son imprescindibles.

En Scrum un proyecto se ejecuta en sprints o bloques que duran normalmente poco tiempo, las iteraciones suelen tener una duración de dos semanas según la teoría. En este proyecto al existir un único desarrollador dicho tiempo se dobló haciendo que cada sprint durara entre tres y cuatro semanas, al final de cada uno se espera que se cumplan los objetivos descritos en el mismo y por tanto que el producto final esté más cerca de terminar.

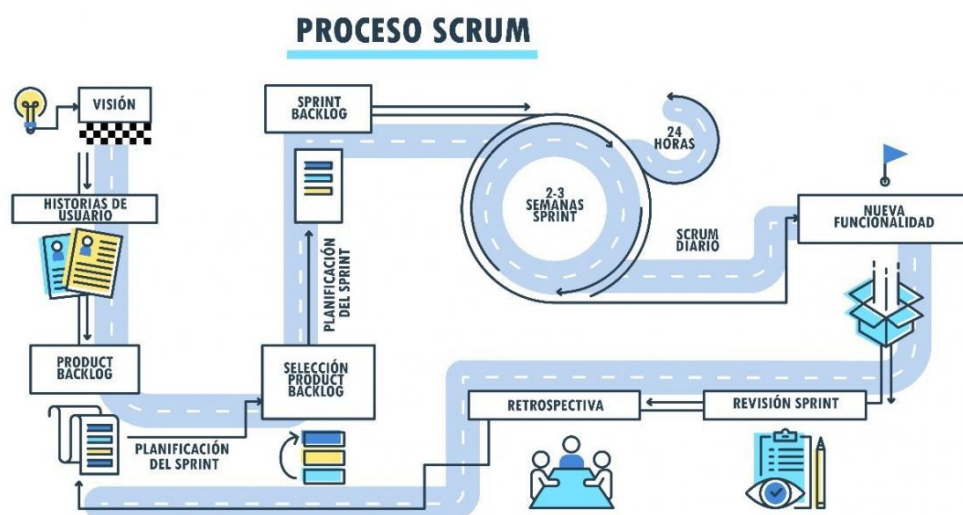


FIGURA 18 PROCESO DE SCRUM

El proceso parte de la lista de requisitos priorizados del producto que actúa como plan del proyecto. En esta lista se priorizan los objetivos balanceando el valor que le aportan respecto a su coste y quedan repartidos en iteraciones y entregas.

5.2.1 Historias de usuario

En la pila de producto, también conocido como Product Backlog, se encuentran definidas las historias de usuario que se han recopilado incluso antes de comenzar con la implementación de la aplicación web. Una vez definida la pila de producto se crearon las diferentes historias de usuario haciendo uso de la herramienta de Trello, usada durante la carrera en varias asignaturas y también por empresas. Gracias a su versatilidad se han podido definir fechas, objetivos y prioridades en cada Sprint, gracias a esto se ha podido tener en todo momento conocimiento de la situación del proyecto y de sus avances, lo cual permitió ajustar las prioridades durante el desarrollo del mismo. A continuación, se muestra la pila de producto de cada Sprint.

5.2.2 Pila de producto - Sprint 0

ID	TÍTULO	DESCRIPCIÓN	PRIORIDAD (0-10)	ESTIMACIÓN (horas)	CRITERIOS DE VALIDACIÓN
1	Instalación del software	Instalar PostgreSQL, IntelliJ, Postman, PostgreSQL, Visual Studio Code y Node.js	10	10	Tener todo instalado correctamente en el equipo
2	Curso de Spring Boot	Realizar un curso de Spring Boot de cara a la realización del Backend	7	20	Conocer cómo crear un proyecto y tenerlo operativo

3	Curso de Vue.js y Vuetify	Curso para aprender nociones básicas de Vue.js y Vuetify para el frontend	8	20	Conocer cómo crear un proyecto y tenerlo operativo
4	Crear BBDD y su estructura básica	Crear BD con PostgreSQL y su estructura básica de usuarios, cursos y preguntas	8	10	Tener la base de datos configurada correctamente
5	Crear pequeño frontend con Vue.js	Creación de un frontend de prueba con Vue.js y haciendo uso de componentes y estados	6	10	Conocer cómo hacer uso de componentes y estados en el proyecto
6	Realizar un servidor que permita ejecutar operaciones	Crear pequeño backend conectado a la BD y realizar operaciones a través de Postman	8	8	Verificar que las operaciones del servidor se realizan correctamente

TABLA 4 PILA DE PRODUCTO DEL SPRINT 0

5.2.3 Pila de producto - Sprint 1

ID	TÍTULO	DESCRIPCIÓN	PRIORIDAD (0-10)	ESTIMACIÓN (horas)	CRITERIOS DE VALIDACIÓN
1	Conectar frontend y backend	Conectar frontend y backend para realizar operaciones con la BD	10	10	Comunicación correcta entre cliente y servidor
2	Como administrador quiero poder	Administrador puede visualizar, editar,	7	20	Conocer cómo crear un proyecto y tenerlo operativo

	gestionar los usuarios de la web	eliminar y crear usuarios			
3	Como usuario quiero poder abrir sesión en el portal web de la aplicación	El usuario hace login en la aplicación	8	5	El usuario inicia sesión correctamente, creando la sesión
4	Como usuario quiero poder cerrar sesión en el portal web de la aplicación	El usuario cierra la sesión en la aplicación	8	5	El usuario cierra la sesión correctamente y por tanto ya no ve sus datos
5	Como usuario quiero poder ver mis datos personales	El usuario ve sus datos en una página de perfil	6	5	El usuario visualiza sus datos en una página de perfil
6	Como usuario quiero poder editar datos de mi perfil de la aplicación	El usuario puede cambiar algún dato como su contraseña de acceso.	8	10	El usuario puede editar sus datos y estos cambios son guardados
7	Como usuario quiero poder solicitar una cuenta en el portal web de la aplicación.	El usuario puede registrarse en la web	9	5	El usuario se registra correctamente tras cumplir los requisitos solicitados
8	Como administrador quiero poder hacer un CRUD de los cursos.	El administrador puede hacer un CRUD de cualquier curso	8	10	El administrador puede hacer una gestión completa de los cursos

9	Como usuario quiero poder ver los cursos en los que estoy apuntado	El usuario puede ver y entrar a los cursos en los que está inscrito	9	10	El usuario ve y entra a los cursos correctamente
---	--	---	---	----	--

TABLA 5 PILA DE PRODUCTO DEL SPRINT 1

5.2.3 Pila de producto - Sprint 2

ID	TÍTULO	DESCRIPCIÓN	PRIORIDAD (0-10)	ESTIMACIÓN (horas)	CRITERIOS DE VALIDACIÓN
1	Como administrador quiero poder ver preguntas y hacer CRUD de las mismas en todos los cursos	Un administrador puede hacer un CRUD de las preguntas de un curso	9	10	El administrador ve los cursos y hace el CRUD de preguntas en todos.
2	Como usuario quiero poder ver preguntas y hacer CRUD de las mismas en mi curso	Un usuario puede ver las preguntas de los cursos y si este es creado por el, hacer un CRUD de las preguntas	8	8	El usuario ve los cursos y hace el CRUD de preguntas en los suyos.
3	Como administrador quiero poder hacer un CRUD de las animaciones	Un administrador puede gestionar las animaciones de la web	8	30	El administrador gestiona las animaciones existentes
4	Como usuario quiero poder ver una animación	El usuario puede ver animaciones en las preguntas y en	8	10	El usuario ve las animaciones existentes y puede interactuar con ellas

	cuando responda a las preguntas	algunos casos interactuar con ellas			
--	------------------------------------	--	--	--	--

TABLA 6 PILA DE PRODUCTO DEL SPRINT 2

5.2.3 Pila de producto - Sprint 3

ID	TÍTULO	DESCRIPCIÓN	PRIORIDAD (0-10)	ESTIMACIÓN (horas)	CRITERIOS DE VALIDACIÓN
1	Como administrador quiero poder ver opciones diferentes	Interfaz personalizada para el administrador en varios puntos de la web	9	10	Un administrador puede ver diferentes funciones a las de un usuario normal
2	Guía de uso para el administrador	Realizar un manual de usuario que refleje las diferencias que tiene un administrador	7	10	Un administrador reconoce en el manual sus opciones
3	Guía de uso para el usuario	Realizar un manual de usuario que refleje las funciones disponibles para un usuario normal	8	5	Un usuario normal reconoce en el manual sus opciones
4	Guía de uso para vista de toda la aplicación web	Repaso de las diferentes pantallas con una explicación de los diferentes apartados	9	5	El usuario entiende el funcionamiento de la aplicación

TABLA 7 PILA DE PRODUCTO DEL SPRINT 3

5.3 Sprint 0. Primeros pasos

Al primer sprint, en este caso llamado Sprint 0 se le dedicaron un total de 78 horas, en él se pudieron identificar los principales objetivos del proyecto y por tanto comprender como serían los siguientes Sprints. De este modo se pudieron tener claras cuáles serían las expectativas y como se podrían abordar los siguientes objetivos una vez se conocían las tecnologías que se iban a usar.

Dicho de otro modo, en el Sprint 0 se obtiene todo lo necesario para poder comenzar a trabajar con un enfoque ágil.

Sprint 0 (Total horas 78h)	
Instalar PostgreSQL, IntelliJ (licencia universitaria), Postman, PostgreSQL (al no tener Windows 10 Pro), Visual Studio Code y Node.js (10h)	8 de mar. de 2019
Curso para usar SpringBoot para el backend (20h)	15 de mar. de 2019
Curso para usar Vue.js y Vuetify para el frontend (20h)	22 de mar. de 2019
Crear BD y su estructura básica (10h)	25 de mar. de 2019
Crear pequeño frontend con Node.js (10h)	28 de mar. de 2019
Crear pequeño backend conectado a la BD y realizar operaciones a través de Postman (8h)	31 de mar. de 2019
+ Añada otra tarjeta	

TABLA 8 HISTORIAS DE USUARIO SPRINT 0 TEST YOURSELF

Aunque en un inicio su duración iba a ser considerablemente menor se ha utilizado un mayor tiempo debido al desconocimiento de tecnologías como Vue o Spring Boot y la búsqueda de diversa documentación, además tal y como se tenía definido en el propio Sprint, se creó un pequeño desarrollo que sirviera como base y aprendizaje de cara al producto final.

Este pequeño desarrollo consistió en poder crear objetos de coches que podrían equivaler a los cursos, para ello, se llamaba desde el cliente al servidor pasándole una serie de parámetros y valores que eran recogidos en un formulario.

```
<template>
  <div>
    <v-text-field label="Año" v-model="year"></v-text-field>
    <v-text-field label="Extras" v-model="extras"></v-text-field>
    <v-btn color="success" @click="vacio">Enviar</v-btn>
    <span v-if="chain">{{title}}</span>
  </div>
</template>

<script>
export default {
  name: 'Formulario',
  data () {
    return {
      year: 0,
      extras: 0,
      chain: false,
      title: 'Vacío'
    }
  },
  methods: {
    vacio() {
      console.log(this.year)
      console.log(this.extras)
      this.title = 'Vacío'
      if(this.year.length>0 && this.extras.length>0){
        this.title = this.year + this.extras
      }
      this.chain = !this.chain
      const data = {
        year: this.year,
        extras: this.extras
      }
      this.$emit('send', data)
    }
  }
}
</script>
```

FIGURA 19 SPRINT 0 - FORMULARIO DE PRUEBA PARA CREAR UN COCHE

El servidor recibía la petición REST en su controlador cuyos datos debían ser los mismos que los del DTO del coche.


```
@ResponseBody
@PostMapping("/createCar")
public void createCar(@RequestBody Car car) { createBike.execute(car); }
```

FIGURA 20 SPRINT 0 - CONTROLLER DE LOS COCHES

El DTO del coche se componía del año de fabricación y el número de extras, lo cual podría equivaler al nombre del curso y temas.

```
public class Car {
    private int id;
    private int year;
    private int extras;
}
```

FIGURA 21 SPRINT 0 - DTO DEL COCHE

Una vez que los datos del coche eran recibidos en el controlador se pasaban al servicio el cual en función de la operación que se estuviese realizando llamaba al método correspondiente de la interfaz del repositorio.

```
public void execute(Car car) { bikeRepository.add(car); }
```

FIGURA 22 SPRINT 0 - SERVICE DEL COCHE

Una vez se recibía en dicha interfaz se ejecutaba la operación, en éste caso crear un nuevo coche y por tanto insertar los datos en la tabla correspondiente de la BBDD.

```
@Repository
public interface BikeRepository {
    void add(Car car);
}
```

FIGURA 23 SPRINT 0 - INTERFAZ DEL REPOSITORIO DEL COCHE

```

@Override
public void add(Car car) {
    final String query = "INSERT INTO car(year, extras)" +
        " VALUES (:year, :extras)";
    try (Connection connection = sql2o.open()) {
        connection.createQuery(query)
            .addParameter( name: "year", car.getYear())
            .addParameter( name: "extras", car.getExtras())
            .executeUpdate();
    }
}

```

FIGURA 24 SPRINT 0 - REPOSITORIO DEL COCHE

Tras esto, se conectó Spring Boot con la base de datos PostgreSQL que se gestiona y se arranca a través de un portal de gestión llamado pgAdmin 4 al cual se puede acceder en la dirección <http://127.0.0.1:52748/browser/>.

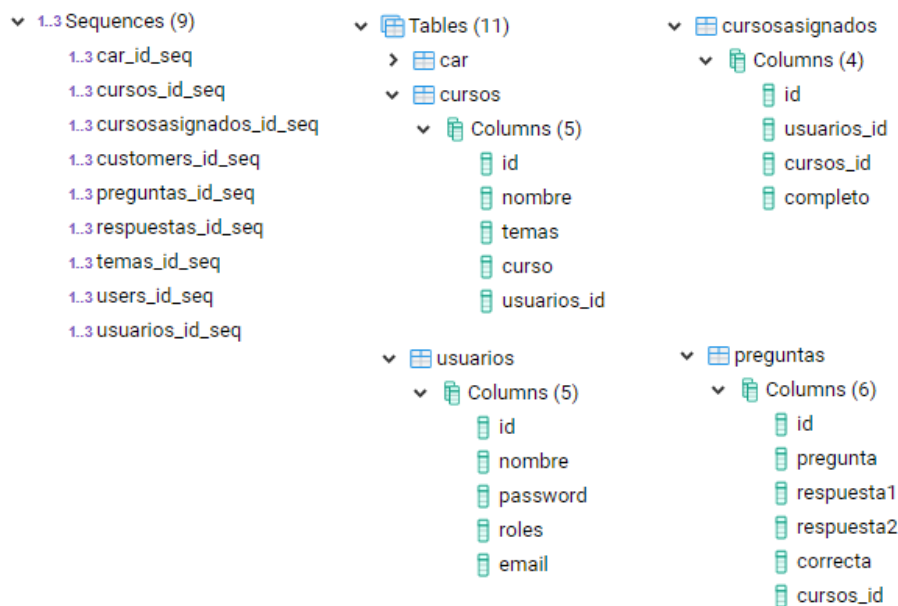


FIGURA 25 BASE DE DATOS POSTGRESQL

Para comprobar que nuestra base de datos funciona correctamente con Spring Boot se puede atacar mediante un servicio Rest, previamente creado con un programa llamado Postman, el cual nos permite hacer dichas peticiones para obtener los datos.

Tras realizar correctamente el primer servicio Rest de prueba se inicia el desarrollo del resto de funcionalidades. Para hacer las llamadas desde el cliente o frontend en Vue hay que tener en cuenta que, al estar el servidor o backend de Spring Boot en el mismo ordenador, nos encontramos un problema de uso compartido de recursos, cors. Esto se soluciona añadiendo una serie de anotaciones en Spring Boot que indican donde se harán las peticiones y por tanto se evitarán las confusiones causadas.

```
2
3  import org.springframework.beans.factory.annotati
4  import org.springframework.web.bind.annotation.*;
5  import backendApi.domain.*;
6  import backendApi.services.user.*;
7
8
9  @CrossOrigin(origins = "*")
10 @RestController
11 public class usersController {
12
13     private DeleteUser deleteUser;
14     private CreateUser createUser;
15     private UpdateUsers updateUsers;
16     private LoginUser loginUser;
17     private GetUsers getUsers;
```

FIGURA 26 CROSSORIGIN

Llegados a este punto se pudieron obtener en el cliente los datos de los usuarios, que habían sido previamente insertados en la base de datos y, por tanto, se implementó un control de sesión de los usuarios en la página web.

Como se puede ver en la imagen inferior atacando a la dirección “http://localhost:8090/users/22” de nuestro servidor recuperamos los datos del usuario con el id 22, en este punto todavía no se aplica un cifrado a las contraseñas, el rol 1 nos indica que es un usuario normal, ya que el 0 queda reservado para los administradores.

```
1 {  
2   "id": 22,  
3   "nombre": "anbecur@outlook.com",  
4   "password": "1234",  
5   "roles": 1  
6 }
```

FIGURA 27 USUARIOS DE LA BASE DE DATOS

Para el registro de usuarios se añade posteriormente un nuevo campo llamado email y el de nombre se queda como identificador en la plataforma. Ambos campos son únicos y no pueden repetirse. Si un usuario intenta registrarse con un nombre de usuario o email que ya esté en uso se impedirá el registro del mismo, ya que no se pueden duplicar dichas claves de base de datos. Además se aprovecha para implementar un cifrado MD5 en las contraseñas, añadiendo por tanto una capa de seguridad a los usuarios.

De cara al frontend este cambio implica que el campo de la contraseña no debe mostrar los caracteres que se estén escribiendo, se ha de validar mediante regex que el formato del email es válido y se añade el nuevo campo del email, que si bien ahora mismo no permite recuperar la contraseña al carecer de un servidor de correos electrónicos en un futuro facilitaría la recuperación de cuentas o el envío de datos al usuario.

Este registro se hace mediante una petición POST al servidor con un objeto de usuario, el servidor se encarga de verificar si los datos son correctos y de que no existen duplicidades a la hora de crearse en la base de datos de tal manera que responderá hacia el cliente correctamente en caso de haber creado el usuario o con un mensaje de error que posteriormente se mostrará en el frontend.

```

1 {
2   "id": 1,
3   "nombre": "jjblanco1993",
4   "password": "81dc9bdb52d04dc20036dbd8313ed055",
5   "roles": 1,
6   "email": "julioblanco93@hotmail.com"
7 }

```

FIGURA 28 USUARIO DE LA BASE DE DATOS CON NUEVOS CAMPOS

5.3.1 Conclusiones del Sprint 0

En este primer Sprint pudimos encontrarnos con algunos errores durante la instalación o incompatibilidades que fueron solucionadas sin mayores inconvenientes. Los pequeños proyectos realizados fueron realizados correctamente y entraron dentro de las estimaciones gracias a la base que otorgaron los cursos realizados del frontend y backend. Este Sprint fue realizado entre el 8 de marzo de 2019 y el 31 de marzo de 2019, pudiéndolo completar en 5 horas menos de las estimadas.

5.4 Sprint 1 y 2. Añadir inicio de sesión, preguntas y cursos

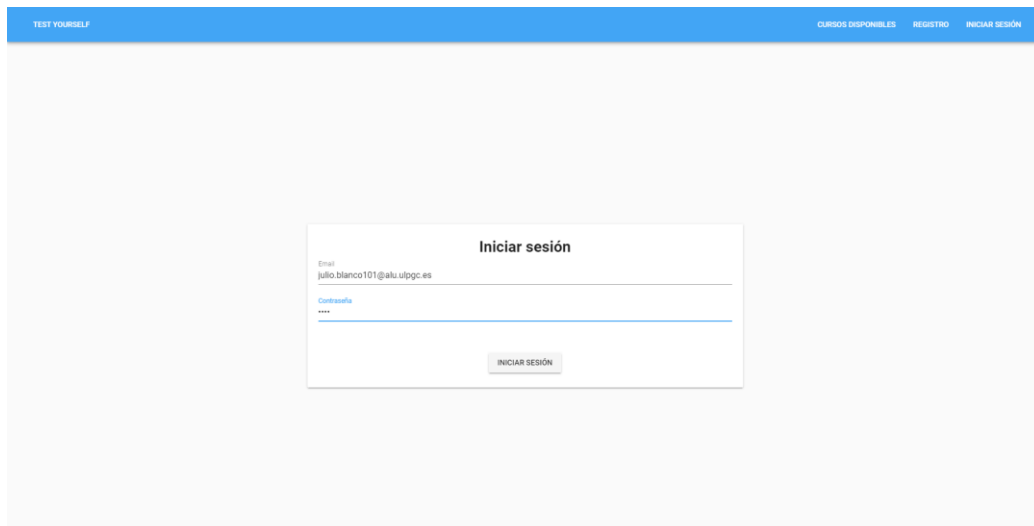
5.4.1 Inicio de sesión

En el Sprint 1 se comenzó añadiendo el inicio de sesión para posteriormente desarrollar los cursos y sus respectivas preguntas.

Sprint 1 (Total horas 80h)		
Conectar frontend y backend para realizar operaciones con la BD (10h)	🕒 4 de abr. de 2019	📧 1/1
Administrador puede visualizar, editar, eliminar y crear usuarios (CRUD) desde el frontend (20h)	🕒 7 de abr. de 2019	📧 1/1
Log in (5h)	🕒 14 de abr. de 2019	📧 1/1
Log Out (5h)	🕒 14 de abr. de 2019	📧 1/1
Ver perfil (5h)	🕒 17 de abr. de 2019	📧 1/1
Editar perfil (10h)	🕒 19 de abr. de 2019	📧 0/1
Registro (5h)	🕒 21 de abr. de 2019	📧 0/1
Cursos (administrador) (10h)	🕒 27 de abr. de 2019	📧 0/1
Cursos (usuario) (10h)	🕒 28 de abr. de 2019	📧 0/1
+ Añada otra tarjeta		

TABLA 9 HISTORIAS DE USUARIO SPRINT 1 TEST YOURSELF

Para ello hizo falta aprender qué era y cómo funcionaba Vuex, lo cual permite manejar variables de manera global.



The screenshot shows a web application interface with a blue header bar containing links: 'TEST YOURSELF', 'CURSOS DISPONIBLES', 'REGISTRO', and 'INICIAR SESIÓN'. The main content area is white and features a login form titled 'Iniciar sesión'. The form has two input fields: 'Email' with the placeholder text 'jullo.blanco101@alu.ulpgc.es' and 'Contraseña' with masked characters '****'. Below the fields is a button labeled 'INICIAR SESIÓN'.

FIGURA 28 FORMULARIO PARA INICIO DE SESIÓN

Como se observa en la figura inferior, Vuex es una arquitectura enfocada en el desarrollo de aplicaciones Vue. Esto nos permite declarar variables y/o funciones globales que puedan ser accedidas desde cualquier punto de la aplicación, como por ejemplo los datos del usuario actual.

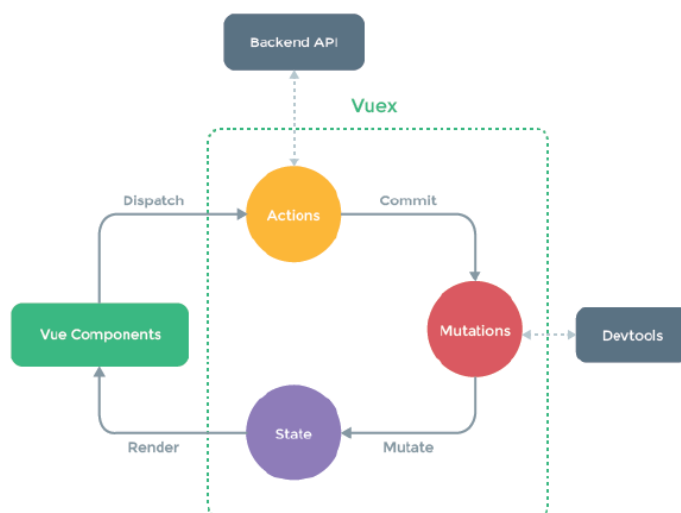


FIGURA 29 VUEX

Para poder realizar un inicio de sesión al servidor se manda mediante una petición POST, en este caso a “http://localhost:8090/login”, un objeto del usuario con los datos introducidos, los cuales se comprueban con la base de datos y se verifican si son correctos o no, en caso de que no sean correctos los datos no se dará feedback al usuario de qué campo introducido es el incorrecto ya que de esa manera se puede averiguar si un email está registrado y por tanto que traten de averiguar la contraseña, se devolverá un mensaje genérico en el frontend indicando de que uno de los dos campos debe de estar mal. Por el contrario, si los datos son correctos, se creará la sesión del usuario con sus datos y se redirigirá la página a la de inicio.

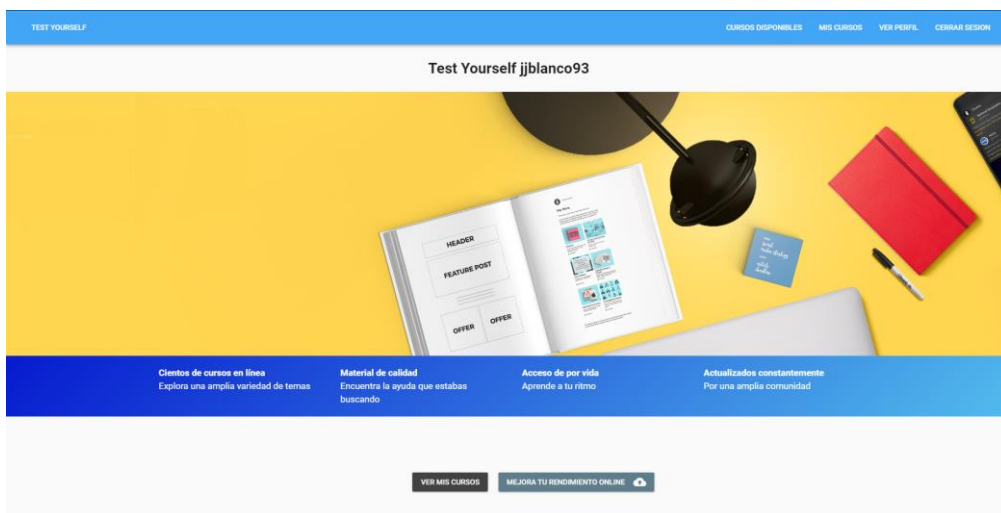


FIGURA 30 UNA VEZ FINALIZADO EL INICIO DE SESIÓN CON ÉXITO

5.4.2 Cursos

Una vez llegados a este punto se insertan cursos manualmente en la BBDD. Se verifica que el usuario es capaz de verlos en una nueva página creada y, por tanto, se añade la posibilidad de que pueda inscribirse a ellos lo cual implica nuevas llamadas REST al servidor con nuevas operaciones y crear una relación entre los cursos y los usuarios a través de una tabla intermedia.

```

<template>
  <div>
    <div class="container">
      <h1 align="center">Cursos disponibles</h1>
      <div class="container">
        <v-app>
          <v-card-title>
            <v-text-field
              v-model="search"
              label="Buscar"
              single-line
              hide-details
            ></v-text-field>
          </v-card-title>
          <v-data-table
            :headers="headers"
            :items="courses"
            class="elevation-1"
            :search="search"
            hide-default-header
            hide-default-footer: true
            :rows-per-page-items="[10, 15, 30, 50]"
            :pagination.sync="pagination"
            rows-per-page-text= "Cursos por página"
          >

            <template v-slot:items="props">
              <td class="text-xs-center">{{ props.item.curso }}</td>
              <td class="text-xs-left">{{ props.item.nombre }}</td>
              <td class="text-xs-left">{{ props.item.temas }}</td>
            </template>
            <template slot="no-data">
              <v-alert :value="true" color="error" icon="warning">
                <p align="center">No tienes cursos asignados</p>
              </v-alert>
            </template>
          </v-data-table>
        </v-app>
      </div>
    </div>
  </div>
</template>

```

FIGURA 31 TABLA DE CURSOS

TEST YOURSELF

CURSOS DISPONIBLES MIS CURSOS VER PERFIL CERRAR SESIÓN

Cursos de jjblanco93

Buscar

Curso	Nombre	Temas	Progreso	
1	Introducción de computadoras	Puertas lógicas, señales	Completado	Asignar curso
1	Inglés	Conversaciones	Completado	Asignar curso

Cursos por página 10 12 of 2

FIGURA 32 CURSOS ASIGNADOS AL USUARIO

Para el registro de un usuario en un curso se llama al servidor desde el cliente con un POST, lo cual crea una relación entre el usuario y el curso. Una vez está creada la relación, se crea una página donde se pueda visualizar, mediante un GET al servidor, los cursos a los que un usuario está inscrito, recuperando aquellos cursos que tengan asociados ese usuario.

Sprint 2 (Total horas 63h)		
Preguntas (administrador) (10h)		
🕒 5 de may. de 2019	☰	✉ 0/1
Preguntas (usuario) (8h)		✎
🕒 6 de may. de 2019	☰	✉ 0/1
Animaciones (administrador) (30h)		
🕒 12 de may. de 2019	☰	✉ 0/1
Animaciones (usuario) (15h)		
🕒 14 de may. de 2019	☰	✉ 0/1
+ Añada otra tarjeta		

TABLA 10 HISTORIAS DE USUARIO SPRINT 2 TEST YOURSELF

Comprobado que los cursos se asignan correctamente a los usuarios, se implementa la creación de cursos desde la propia aplicación, para ello se añade un botón en la vista de cursos disponibles el cual carga un componente con un formulario para crear el curso.

Cursos disponibles			
<div> <div>TEST YOURSELF</div> <div>CURSOS DISPONIBLES</div> <div>MIS CURSOS</div> <div>VER PERFIL</div> <div>CERRAR SESION</div> </div>			
<div> <div>Buscar</div> <div></div> </div>			
Curso ↑	Nombre	Temas	
1	Introducción de computadores	Puertas lógicas, señales	INSCRIBIRSE
1	Introducción a la informática	Lógica, Java, JavaScript, CSS	INSCRIBIRSE
1	Python Intro	Operadores	INSCRIBIRSE
1	Python Intro 2	Bucles	INSCRIBIRSE
1	Python Intro 3	Arrays	INSCRIBIRSE
1	Inglés	Conversaciones	INSCRIBIRSE
1	Introducción a Aulacity	Como se mezclan varias pistas	INSCRIBIRSE
1	Redes	Instalación en servidores	INSCRIBIRSE
1	Trabajo Social	Personas mayores	INSCRIBIRSE
2	Estructura de computadores	Circuitos, Caché	INSCRIBIRSE
<div> <div>Crear nuevo curso</div> <div> <div>Cursos por página</div> <div>10</div> <div>1/10 of 11</div> <div><</div> <div>></div> </div> </div>			

FIGURA 33 CURSOS DISPONIBLES

Una vez que se han rellenado todos los campos por parte del usuario se manda una petición POST al servidor que inserte en la base de datos los datos del nuevo curso. Posteriormente, el usuario puede inscribirse en el curso.

```

abrirCurso (idCurso, nombre) {
  console.log(idCurso + " " + localStorage.getItem('id'))
  axios.post('http://localhost:8090/inscribircurso/' + localStorage.getItem('id') + '/' + idCurso)
    .then((user) => {
      localStorage.setItem('cursoActual', nombre)
      localStorage.setItem('idCursoActual', idCurso)
      router.push({ path: 'preguntas' })
    })
    .catch(() => console.log('error'))
}

```

FIGURA 34 PETICION REST DEL CLIENTE PARA ABRIR UN CURSO

5.4.3 Preguntas

Una vez inscrito, al abrir el curso y ser el creador o un administrador se da la posibilidad de añadir las preguntas al curso hasta un máximo de 5, también se da la posibilidad de eliminar mediante una petición DELETE al servidor el curso o preguntas en concreto si así lo desea, recargando tras la eliminación los componentes para que pueda visualizar los cambios realizados.



TEST YOURSELF

CURSOS DISPONIBLES MIS CURSOS VER PERFIL CERRAR SESIÓN

Introducción de computadores - Puertas lógicas, señales

Pregunta	Opción 1	Opción 2	Respuesta	Guardar
¿Cuál es el operando de un OR en programación?	0	65	Respuesta	GUARDAR RESPUESTA ELIMINAR PREGUNTA
¿Cuál es la salida de una puerta lógica OR si una de las entradas es 1 y la otra 0?	1	0	Respuesta	GUARDAR RESPUESTA ELIMINAR PREGUNTA

Preguntas por página: 5 1 of 2

ELIMINAR CURSO FINALIZAR CURSO

Preguntas actuales de tu curso: 2 de 5

Pregunta

Opción 1

Opción 2

Respuesta correcta

AÑADIR PREGUNTA

FIGURA 35 VISTA DE CURSO COMO CREADOR

```

@Override
public void crearPregunta(Preguntas pregunta) {
    System.out.println(pregunta.getCursos_id());
    final String query = "INSERT INTO preguntas(pregunta, respuesta1, respuesta2, correcta, cursos_id) +
        " VALUES (:pregunta, :respuesta1, :respuesta2, :correcta, :cursos_id)";
    try (Connection connection = sql2o.open()) {
        connection.createQuery(query)
            .addParameter( name: "pregunta", pregunta.getPregunta())
            .addParameter( name: "respuesta1", pregunta.getRespuesta1())
            .addParameter( name: "respuesta2", pregunta.getRespuesta2())
            .addParameter( name: "correcta", pregunta.getCorrecta())
            .addParameter( name: "cursos_id", pregunta.getCursos_id())
            .executeUpdate();
    }
}

@Override
public void eliminarPregunta(int preguntaId) {
    final String query = "DELETE FROM preguntas WHERE id=" + preguntaId;
    try (Connection connection = sql2o.open()) {
        connection.createQuery(query).executeUpdate();
    }
}

```

FIGURA 36 CREACIÓN Y ELIMINACIÓN DE PREGUNTAS EN EL REPOSITORIO DEL SERVIDOR

Si por el contrario el usuario abre un curso en el cual responde todas las preguntas correctamente, se considera que ha adquirido el nivel suficiente como para dar el curso por completo. En caso de que alguna respuesta no sea la correcta se le indicará que existe una o más respuestas incorrectas, de modo que tendrá que revisar todas las opciones.

Esto, aunque puede llegar a ser frustrante en un inicio, asegura que al finalizar el curso el usuario ha sido capaz de resolver correctamente todas las preguntas y por tanto tiene el nivel suficiente, se trata de pensar y

superar retos, no de indicarle el camino al usuario mostrándole cuáles están correctas y cuáles no. De ahí la decisión de limitar a un máximo de 5 preguntas los cursos, ya que dentro de una misma asignatura se les da la posibilidad de crear cursos sin límites y abarcando los temas que deseen.

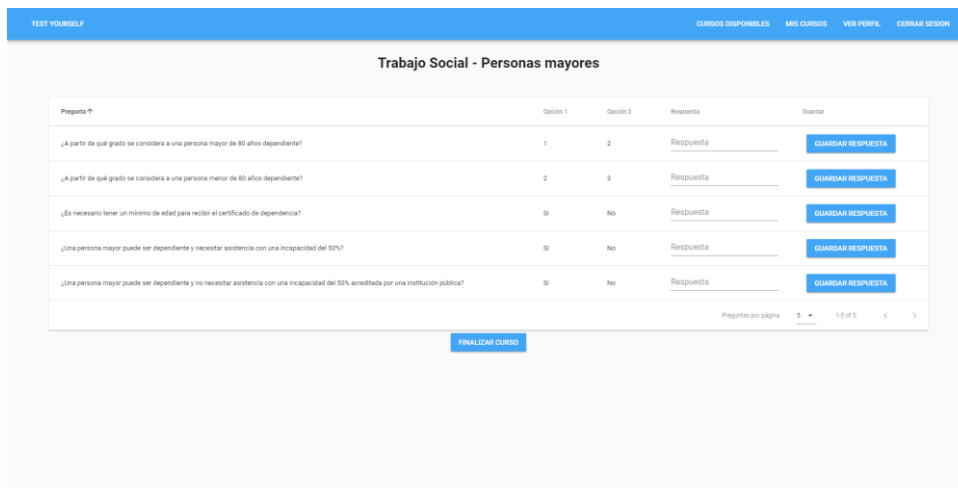
```

1  [
2
3      {
4          "id": 1,
5          "nombre": "Introducción a la informática",
6          "temas": "Lógica, Java, JavaScript, CSS",
7          "curso": "1"
8      },
9
10     {
11         "id": 2,
12         "nombre": "Introducción de computadores",
13         "temas": "Puertas lógicas, señales",
14         "curso": "1"
15     },
16
17     {
18         "id": 4,
19         "nombre": "Estructura de computadores",
20         "temas": "Circuitos, Caché",
21         "curso": "2"
22     }
23 ]
  
```

FIGURA 37 ESTRUCTURA DE LOS CURSOS

Estos cursos se componen de un nombre, temas que abarcan y curso en el que se indica el nivel.

En caso de que el usuario no sea el creador del curso únicamente podrá responder, preguntar y finalizar el curso en caso de responder a todas las preguntas correctamente.



Pregunta	Opción 1	Opción 2	Respuesta	Guardar
¿A partir de qué grado se considera a una persona mayor de 60 años dependiente?	1	2	Respuesta	GUARDAR RESPUESTA
¿A partir de qué grado se considera a una persona menor de 60 años dependiente?	2	3	Respuesta	GUARDAR RESPUESTA
¿Es necesario tener un mínimo de edad para recibir el certificado de dependencia?	Si	No	Respuesta	GUARDAR RESPUESTA
¿Una persona mayor puede ser dependiente y necesitar asistencia con una incapacidad del 50%?	Si	No	Respuesta	GUARDAR RESPUESTA
¿Una persona mayor puede ser dependiente y no necesitar asistencia con una incapacidad del 50% acreditada por una institución pública?	Si	No	Respuesta	GUARDAR RESPUESTA

Finalizar curso

FIGURA 38 VISTA DE CURSO COMO NO CREADOR

5.4.4 Animaciones

Si en el curso existen preguntas que tengan animaciones asociadas se podrán visualizar en la parte inferior y reproducirlas las veces que se desee. Estas animaciones son generadas a través de imágenes SVG y haciendo uso de la librería AnimeJS se les da el movimiento deseado. Para esto se puede hacer uso de editores visuales gratuitos como Boxy-SVG el cual te facilita el código HTML del SVG diseñado.

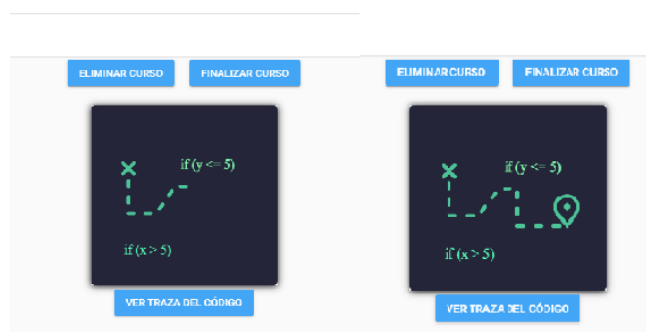


FIGURA 39 FOTOGRAMAS DE LA ANIMACIÓN

Las animaciones pueden suponer un extra de cara al usuario ya que le pueden ayudar a razonar la respuesta de una manera lógica. En el caso de la programación puede visualizar de manera simbólica la traza de ejecución del código y deducir cuando llega a ciertos puntos qué condiciones tiene que cumplir para llegar al destino.

```

<div align="center">
<svg viewBox="172.418 133.885 100 100" width="100" height="100">
<rect width="100" height="100" rx="2" fill="#242539" x="205.331" y="171.942" transform="matrix(1, 0, 0, 1, -32.912983, -38.056732)"/>
<g fill="none" stroke="#4DC498" stroke-width="2" stroke-linecap="round" stroke-linejoin="round" transform="matrix(1, 0, 0, 1, 172.418121, 133.884918)">
<g transform="translate(20 35)">
<path d="M 0 0 v 25 h 15 l 10 -15 h 10 v 20 h 25" stroke-dasharray="3 6" stroke-dashoffset="2"/>
<path class="in-between" stroke="#242539" stroke-width="2.5" d="M 60 30 h -25 v -20 h -10 l -10 15 h -15 v -25"/>
</g>
<g transform="translate(20 35)">
<path class="start" d="M -3 -3 l 6 6"/>
<path class="start" d="M 3 -3 l -6 6"/>
</g>
<g transform="translate(80 65)">
<g transform="translate(40 20)">
<g class="end">
<g transform="translate(-40 -20)">
<path d="M 0 0 q 6 -6 10 a 6 6 0 0 0 -12 0 q 0 4 6 10"/>
<circle cy="-10" r="2" stroke="none" fill="#4DC498"/>
</g>
</g>
</g>
</g>
</div>
<text transform="matrix(0.250914, 0.000137, 0.002266, 0.279954, 143.751022, 132.521842)" style="fill: rgb(90, 255, 184); font-family: "Roboto Slab">
<text transform="matrix(0.250914, 0.000137, 0.002266, 0.279954, 173.959, 85.113358)" style="fill: rgb(131, 255, 174); font-family: "Roboto Slab">
</div>
<v-btn color="primary" @click="anime2">Ver traza del código</v-btn>
</div>

```

FIGURA 40 SVG DE LA ANIMACIÓN DE LA FIGURA SUPERIOR

Por último, se crea la página del perfil del usuario en la cual se pueden visualizar los datos del usuario actual. Estos datos abarcan el nombre de usuario en la plataforma, el correo electrónico asociado a su cuenta y el número de cursos que ha completado satisfactoriamente en la plataforma. También se le da la posibilidad de modificar la contraseña en caso de que lo desee.

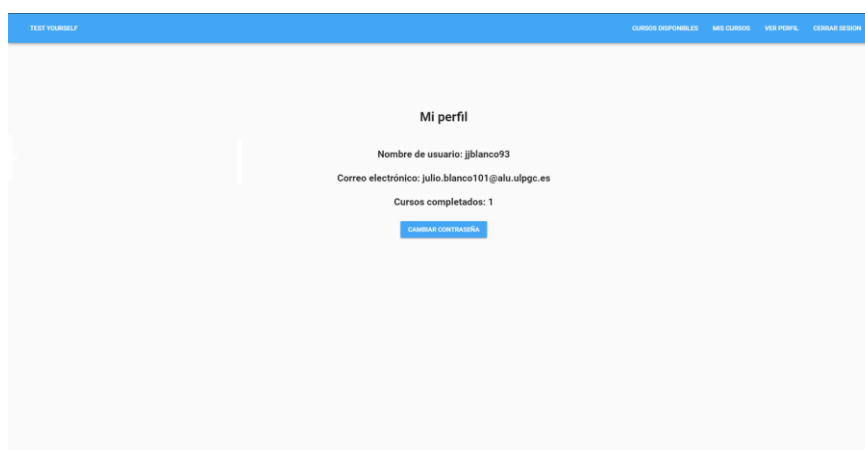


FIGURA 41 VISTA DEL PERFIL DE UN USUARIO NORMAL

```

<div>
  <div class="container"><h1 align="center"> Mi perfil</h1></div>
  <br>
  <div align="center">
    <h2 v-if="this.rol == 0">ADMINISTRADOR</h2>
    <br>
    <h2>Nombre de usuario: {{this.nameOfUser}}</h2>
    <br>
    <h2>Correo electrónico: {{this.email}}</h2>
    <br>
    <h2>Cursos completados: {{this.cursosCompletados}}</h2>
    <br>
    <v-btn color="blue darken-1" flat @click="nuevoAdministrador" v-if="this.rol == 0">Nuevo administrador</v-btn>
    <v-layout row justify-center>
      <v-dialog v-model="dialog" max-width="600px">
        <template !v-if="dialog">
          <v-btn color="primary" @click="dialogChange">Cambiar contraseña</v-btn>
        </template>
        <br>
        <v-card v-if="dialog">
          <v-card-title>
            <span class="headline">Cambio de contraseña</span>
          </v-card-title>
          <v-card-text>
            <v-container grid-list-md>
              <v-layout wrap>
                <v-flex xs12>
                  <v-text-field label="Contraseña actual" type="password" required></v-text-field>
                  <v-text-field label="Nueva contraseña" type="password" required></v-text-field>
                </v-flex>
              </v-layout>
            </v-container>
          </v-card-text>
          <v-card-actions>
            <v-spacer></v-spacer>
            <v-btn color="blue darken-1" flat @click="dialog = false">Cancelar</v-btn>
            <v-btn color="blue darken-1" flat @click="dialog = false">Cambiar</v-btn>
          </v-card-actions>
        </v-card>
      </v-dialog>
    </v-layout>
  </div>
</div>

```

FIGURA 42 ESTRUCTURA DE LA PARTE VISUAL DE LA PÁGINA DEL PERFIL

5.4.5 Conclusiones del Sprint 1 y 2

Estos dos Sprint suponían la parte importante del desarrollo y eran los que más horas tenían asignadas ya que aquí se concentraban las tareas más largas y prioritarias. A pesar de que se obtuvieron conocimientos del backend y el frontend el conseguir integrar las animaciones fue una de las partes más complicadas, ya que la librería de AnimeJS daba problemas de compatibilidad y por otro lado la documentación y los ejemplos no eran muy abundantes. Esto quizás fue lo que provocó el mayor retraso en estos Sprint y lo que supuso un aumento de 30 horas con respecto a la estimación total de los Sprint 1 y. Este Sprint fue realizado entre el 31 de marzo de 2019 y el 15 de febrero de 2021, las fechas abarcan un gran espacio de tiempo ya que por ciertos motivos se tuvo que paralizar el desarrollo del proyecto, por tanto, se terminó completando en 30 horas más de trabajo con respecto a las estimadas.

5.5 Sprint 3. Interfaces personalizadas y guías de uso

En este último Sprint se da la posibilidad a los administradores de eliminar preguntas y cursos, aun no siendo los creadores del mismo, de manera que se pueda mantener un orden y control en la aplicación. Las vistas serán las mismas que tendría el usuario creador del curso, de manera que se reutilizan los componentes ya existentes. Además, el administrador podrá crear otros con el fin de que la plataforma se pueda moderar a medida que los usuarios de la plataforma vayan en aumento.

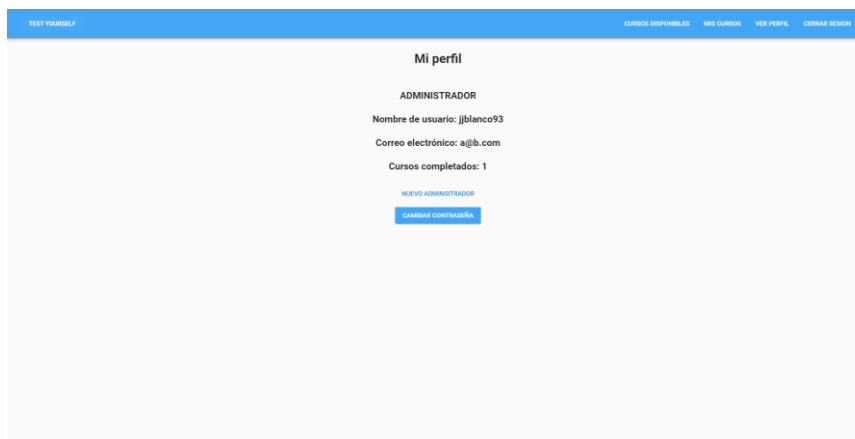


FIGURA 43 VISTA DEL PERFIL DE UN USUARIO ADMINISTRADOR

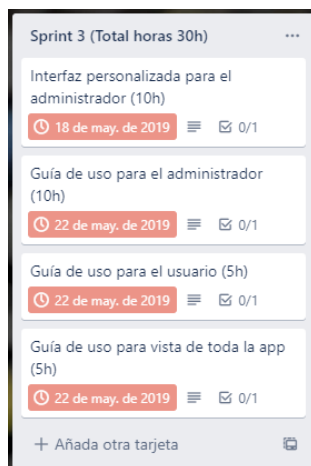


TABLA 11 HISTORIAS DE USUARIO SPRINT 3 TEST YOURSELF

Por otro lado, se crean unas guías de uso de cara a facilitar a los usuarios, en función de los roles que tengan, como hacer un uso correcto de la aplicación mediante la explicación de las diferentes pantallas y las opciones que pueden encontrarse en ellas ya que, aunque aparentemente un administrador puede hacer lo mismo que un usuario normal siempre y cuando a la hora de visualizar un curso el usuario sea el creador del mismo, también puede por ejemplo crear nuevos administradores que gestionen la web.

5.5.1 Conclusiones del Sprint 3

En este último Sprint, se trató de dar a los administradores de la aplicación ciertas funciones que no podía tener un usuario normal. Además de esto, se realizaron diagramas, se crearon los manuales de uso y se finalizó la memoria del proyecto integrando una documentación lo más actualizada posible. Este Sprint fue realizado entre el 15 de febrero de 2021 y el 05 de marzo de 2021, pudiéndolo completar en las estimadas.

Capítulo 6. Conclusiones y trabajo futuro

Finalmente, se exponen a continuación las conclusiones obtenidas del desarrollo de este proyecto y las posibles mejoras que pueden ser aplicadas en un futuro.

La aplicación web podrá aportar un gran valor en varios aspectos sociales. Esta sería la primera página web de refuerzo en la EII que trataría de formar una comunidad y de dar una base a los alumnos, los cuales en muchos casos no son capaces de comprender o razonar en momentos iniciales los problemas que se les plantean.

6.1. Conclusiones

La principal limitación con la que me he podido encontrar a la hora de desarrollar este proyecto ha sido el poco conocimiento que tenía con el uso de Vue y del lenguaje JavaScript que, aunque tiene mucho parecido con Java termina resultando bastante diferente. Al empezar de cero a utilizar dicho framework, ha sido necesario buscar bastante documentación en la parte inicial del desarrollo. Principalmente acerca de cómo hacer un uso correcto de Spring Boot, Vue.js, Vuetify, CSS Grid y Flexbox entre otros, esto supuso la necesidad de hacer un pequeño proyecto previo para aprender el manejo de las tecnologías anteriormente mencionadas.

Por ello, es posible que no se hiciera una buena estimación en algunos sprint ya que algunos de ellos, al comienzo, han durado más de lo previsto al tratarse de tecnologías nuevas, que no se habían visto en la carrera. Esto ha supuesto sacrificar parte del desarrollo en pro de sacar un producto estable y que se pueda dar por finalizado.

Pero esto es un camino necesario que he podido ver, tanto en las prácticas de empresa como en el trabajo personal, sin una adecuada formación previa, a la larga el rendimiento no es el esperado y por tanto se vuelve contraproducente para el producto final.

Hace 8 años, al comenzar el grado de Ingeniería Informática, me encontré asignaturas que pensé que no tenían sentido y que no me aportaban nada. Con el paso del tiempo te das cuenta de que en el día a

día, ya sea a la hora de realizar este proyecto o en el trabajo, hacemos un uso diario de eso que no enseñaron, que no es más que “aprender a pararse y pensar en cómo hacer las cosas de la mejor forma posible”.

Gracias a este proyecto no sólo he podido aprender nuevas tecnologías, me permitió acceder al mundo laboral y por tanto un cambio de vida enorme. Por supuesto, mi tutor ha tenido mucho que ver aquí, me ha dado gran libertad para el desarrollo de este proyecto y me ha ayudado en momentos puntuales con ideas que fueron dando forma a lo que es ahora el proyecto, por ello y por su atención por el correo cuando lo he necesitado, le doy las gracias.

6.2. Trabajo futuro

El Trabajo de Fin de Grado tiene un uso simple de cara al usuario gracias, en parte, a las tecnologías usadas. La interfaz es sencilla y rápida, por lo que cualquier tipo de usuario puede hacer uno de la plataforma. Esto abre un gran abanico de público por tanto los siguientes pasos que se podrían dar son algunos de los siguientes:

- Siendo una aplicación para el aprendizaje y observando la evolución de la aplicación durante el desarrollo podría integrarse en los centros educativos que están promoviendo el uso de las nuevas tecnologías, se podría facilitar su conocimiento para la utilización por parte del profesorado y alumnado, no sólo a nivel universitario sino en otras etapas como en colegios e institutos.
- Se puede adaptar a dispositivos móviles, lo cual no sería muy complejo gracias al uso de CSS Grid y Flex.
- Buscar un diseñador para introducirlo al equipo de desarrollo, el cual se dedique a mejorar el diseño de la interfaz y la experiencia del usuario, además de ampliar el catálogo de animaciones, ya que al estar basadas en imágenes SVG, se hace casi imprescindible tener este puesto cubierto.

- De cara a atraer a nuevos usuarios y mantener los existentes se pueden ir añadiendo progresivamente diferentes funcionalidades, como crear diferentes categorías, de cursos y temas, rankings para ver los alumnos con mejores progresiones, hojas de rendimiento personales para que el usuario pueda ver su rendimiento de cara a saber en qué temas debe mejorar, o incluso la posibilidad de que sea el propio usuario el que añada las animaciones a la plataforma.
- También se hace necesario una vez salga a producción el tener un servidor de correo electrónico de manera que se puedan mandar comunicaciones a los usuarios de la plataforma y estos también puedan recuperar su cuenta a través del mismo si olvidan su contraseña.
- Con el fin de facilitar al usuario la creación de preguntas se puede crear un banco de preguntas y que puedan elegir de ese banco las que más se aproximen a lo buscado.

Capítulo 7. Anexos

7.1 Manual de usuario

7.1.1 Página principal

En esta sección se muestra la página principal con el menú de navegación en la parte superior de la web sin haber iniciado sesión. Dicha barra de navegación se compone de los siguientes botones:

- Botón de inicio: redirige a la página de inicio.
- Cursos disponibles: permite ver en una nueva página los cursos disponibles en la plataforma.
- Registro: permite hacer un registro en la plataforma.
- Iniciar sesión: permite a un usuario registrado iniciar sesión en la plataforma.

Por otro lado, se muestra información genérica de lo que aporta la plataforma al usuario.

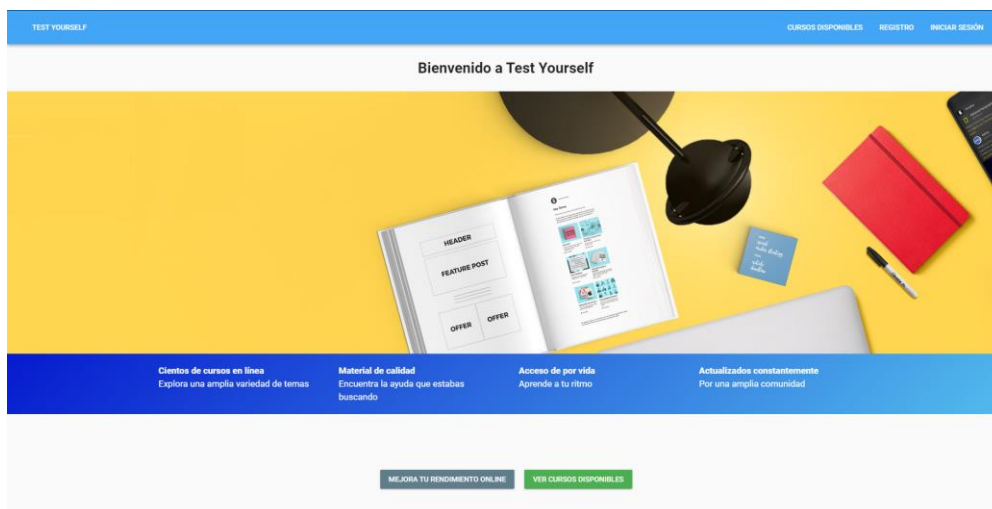
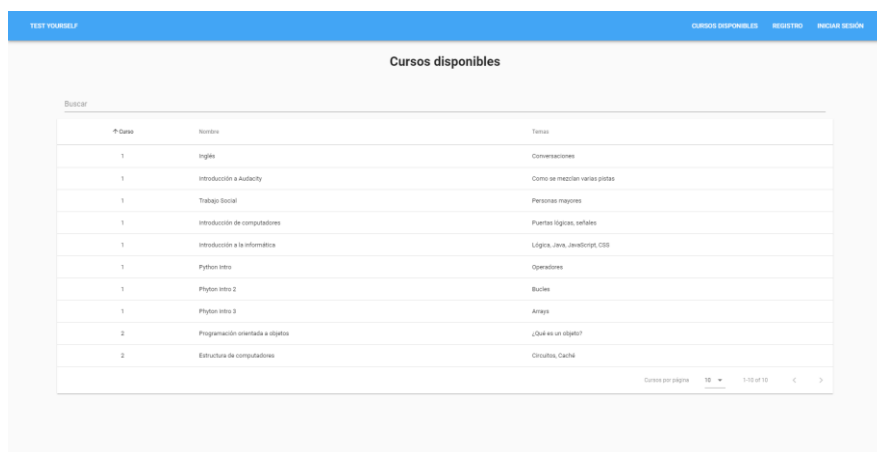


FIGURA 44 MANUAL DE USUARIO - PÁGINA PRINCIPAL

7.1.2 Cursos disponibles

En esta página el usuario puede visualizar los cursos creados en la plataforma y su información, al no tener la sesión iniciada no se da la opción de inscribirse a los mismos. Se da la opción de filtrar dichos cursos con la barra de búsqueda encontrada en la parte superior de la tabla, la cual filtra tanto por curso, como por nombres y temas.

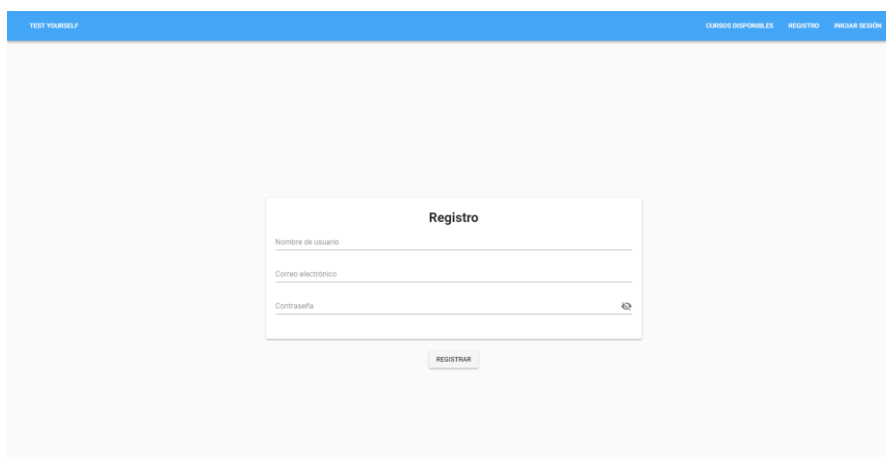


Curso	Nombre	Temas
1	Inglés	Conversaciones
1	Introducción a Audacity	Como se mezclan varias pistas
1	Trabajo Social	Personas mayores
1	Introducción de computadores	Puertas lógicas, señales
1	Introducción a la informática	Lógica, Java, JavaScript, CSS
1	Python Intro	Operadores
1	Python Intro 2	Bucles
1	Python Intro 3	Arrays
2	Programación orientada a objetos	¿Qué es un objeto?
2	Estructura de computadores	Circuitos, Cache

FIGURA 45 MANUAL DE USUARIO - CURSOS DISPONIBLES

7.1.3 Registro

En esta página se permite a un usuario registrarse en la plataforma. Los campos solicitados son un nombre de usuario, un email y una contraseña.



Registro

Nombre de usuario

Correo electrónico

Contraseña

REGISTRAR

FIGURA 46 MANUAL DE USUARIO – REGISTRO

7.1.4 Inicio de sesión

En esta página se permite a un usuario iniciar sesión en la plataforma. Los campos solicitados son el email con el que se registró el usuario y la contraseña.

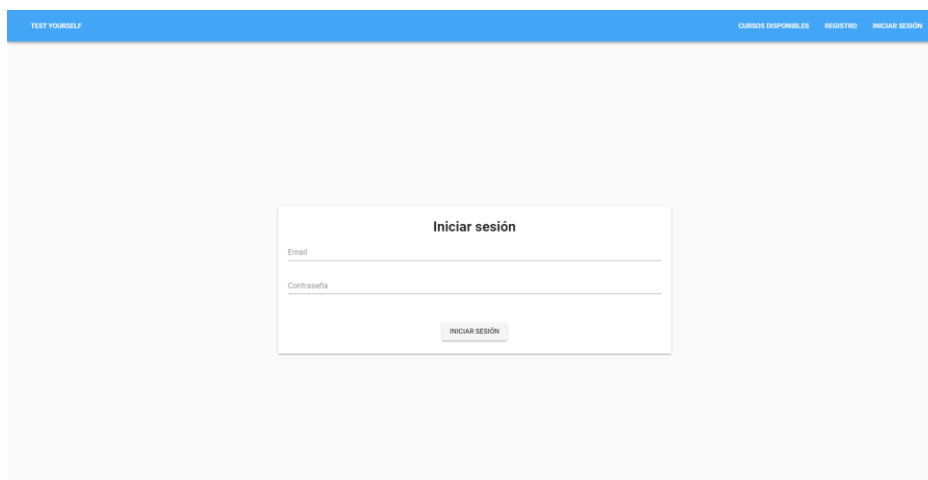


FIGURA 47 MANUAL DE USUARIO - INICIO DE SESIÓN

7.1.5 Validaciones y verificaciones

Tanto a la hora de registrarse como a la de iniciar sesión se valida que el formato del email sea correcto, además de ello, cuando esto se cumple se verifica que los datos sean correctos. En el caso de que se esté registrando el usuario se verifica que no exista ya un usuario con ese email o nombre de usuario. Si se está iniciando sesión se verifica que exista un usuario con esos datos.

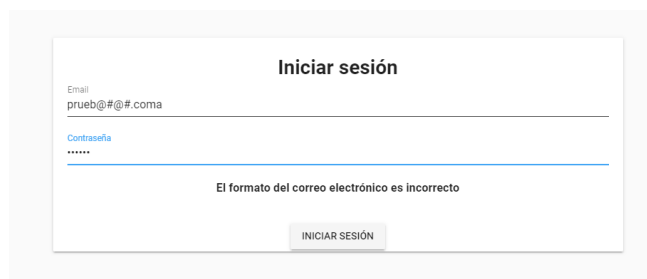
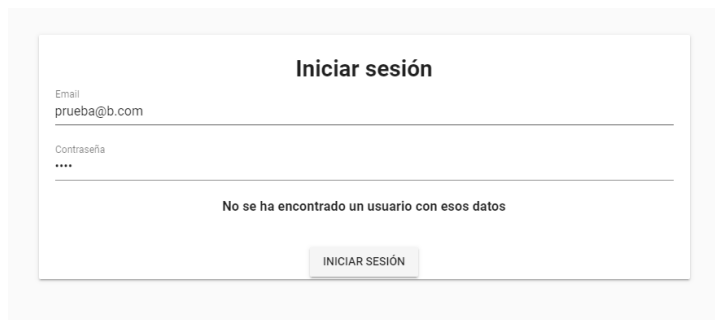


FIGURA 48 MANUAL DE USUARIO - INICIO DE SESIÓN - FORMATO INCORRECTO DEL EMAIL



Iniciar sesión

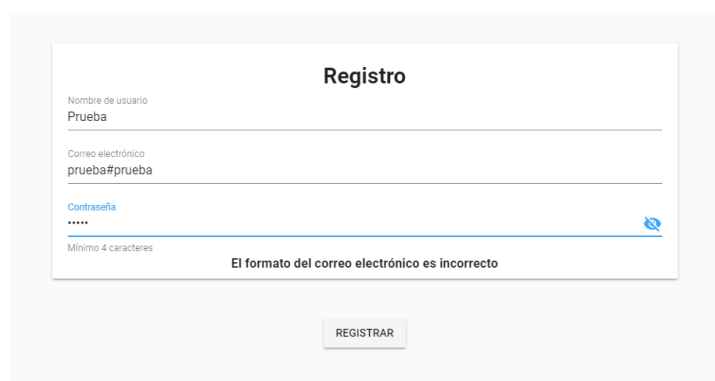
Email
prueba@b.com

Contraseña

No se ha encontrado un usuario con esos datos

INICIAR SESIÓN

FIGURA 49 MANUAL DE USUARIO - INICIO SESIÓN - USUARIO INCORRECTO



Registro

Nombre de usuario
Prueba

Correo electrónico
prueba#prueba

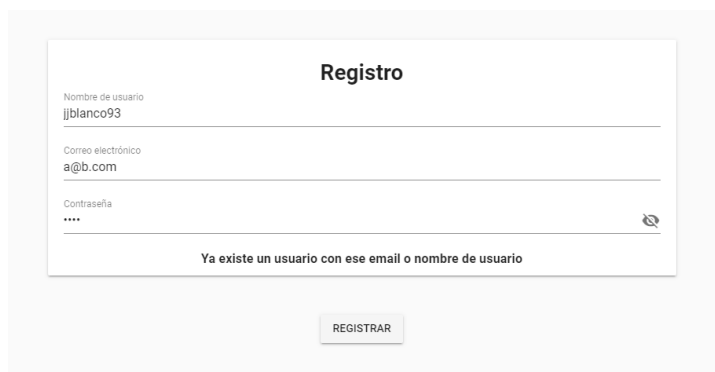
Contraseña

Mínimo 4 caracteres

El formato del correo electrónico es incorrecto

REGISTRAR

FIGURA 50 MANUAL DE USUARIO - REGISTRO - FORMATO INCORRECTO DEL EMAIL



Registro

Nombre de usuario
jjblanco93

Correo electrónico
a@b.com

Contraseña

Ya existe un usuario con ese email o nombre de usuario

REGISTRAR

FIGURA 51 MANUAL DE USUARIO - REGISTRO - USUARIO EXISTENTE

7.1.6 Página principal - Sesión iniciada

Cuando el usuario ha iniciado sesión en la barra superior podrá ver un nuevo botón dedicado a su perfil, un botón para ver sus cursos y el

botón de iniciar sesión desaparece dando lugar al de cerrar sesión, el cual como indica finalizará la sesión del usuario.

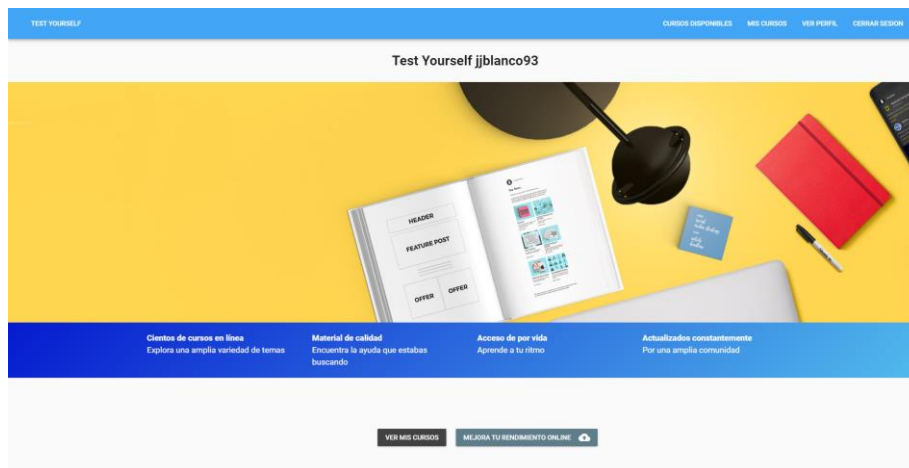


FIGURA 52 MANUAL DE USUARIO - PÁGINA PRINCIPAL - SESIÓN INICIADA

7.1.7 Cursos disponibles - Sesión iniciada

En caso de que el usuario entre a esta página con la sesión iniciada, se le da la posibilidad de inscribirse a los cursos existentes y además se le da la opción de crear cursos nuevos y de poder filtrar los cursos ya existentes.

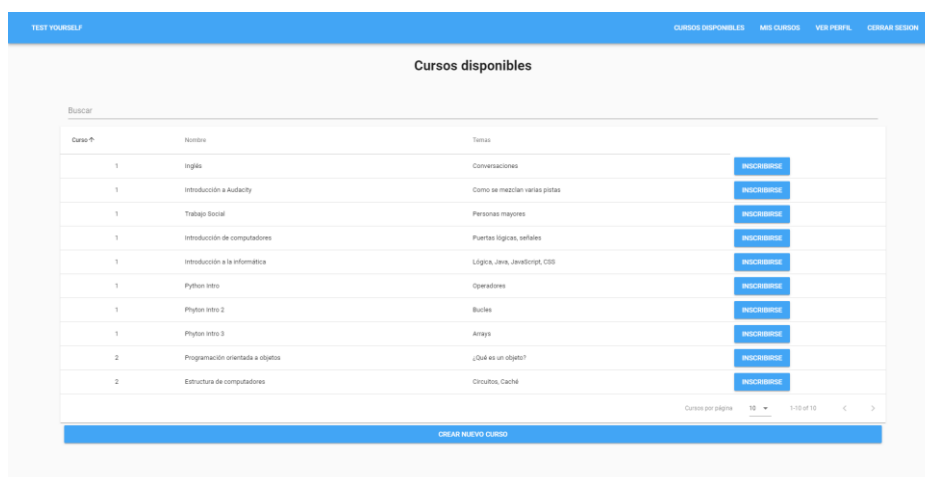


FIGURA 53 MANUAL DE USUARIO - CURSOS DISPONIBLES - SESIÓN INICIADA

7.1.8 Crear curso - Sesión iniciada

Si un usuario accede a la página de creación de un curso, podrá rellenar los datos iniciales, el curso o nivel, nombre del curso y los temas que abarca.

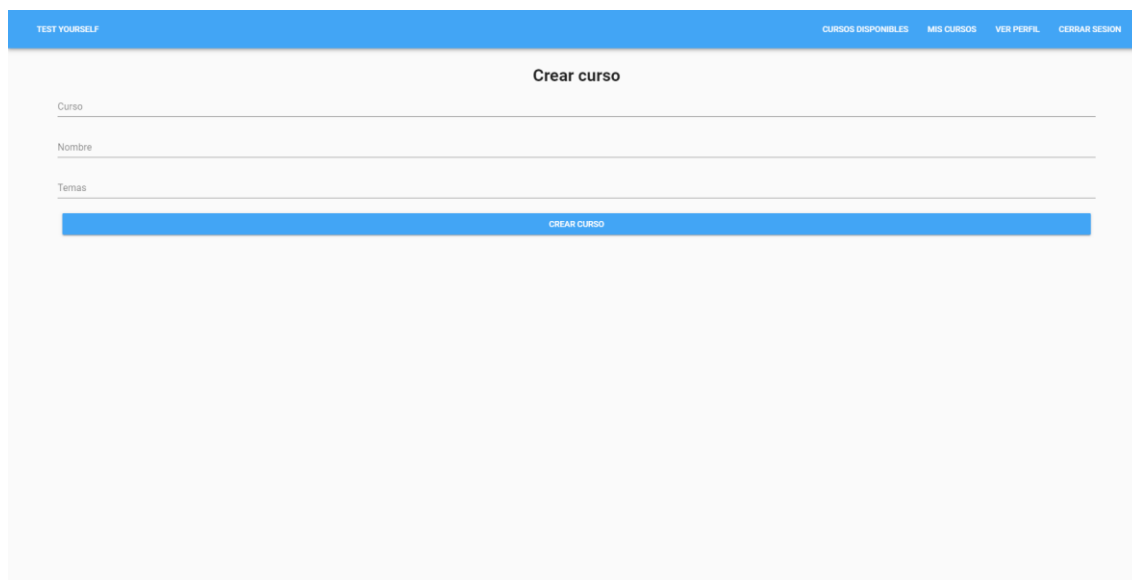
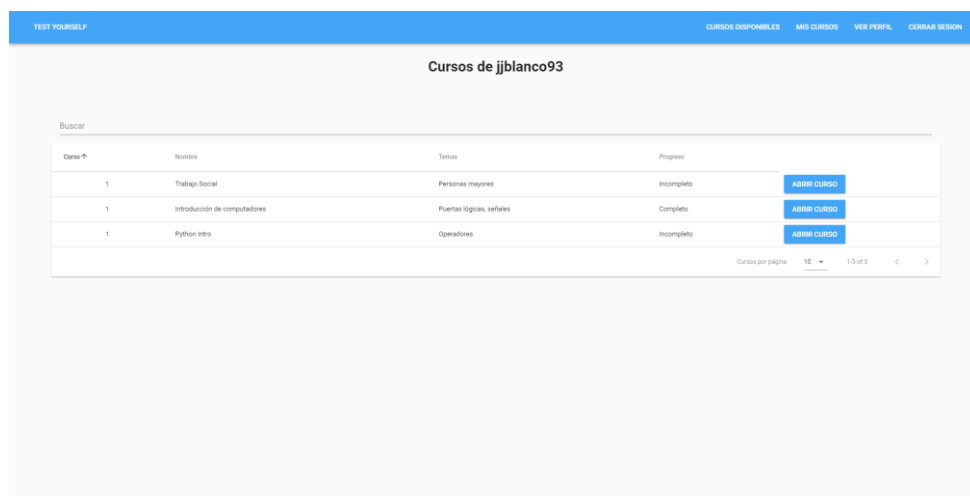


FIGURA 54 MANUAL DE USUARIO - CREAR CURSO

7.1.9 Cursos asignados - Sesión iniciada

Aquí el usuario podrá abrir los cursos en los que está inscrito y filtrar por ellos, además de poder ver los que ya ha completado y los que no.



Curso	Nombre	Temas	Progreso	
1	Trabajo Social	Personas mayores	Incompleto	Abrir curso
1	Introducción de computadores	Puertas lógicas, señales	Completo	Abrir curso
1	Python intro	Operadores	Incompleto	Abrir curso

Cursos por página: 10 1 of 3

FIGURA 55 MANUAL DE USUARIO - CURSOS INSCRITOS

7.1.10 Edición de un curso - Sesión iniciada

En el caso de que un administrador o un usuario abra un curso del cual sea el creador podrá añadir preguntas al curso hasta un máximo de 5, eliminar preguntas o eliminar el curso.

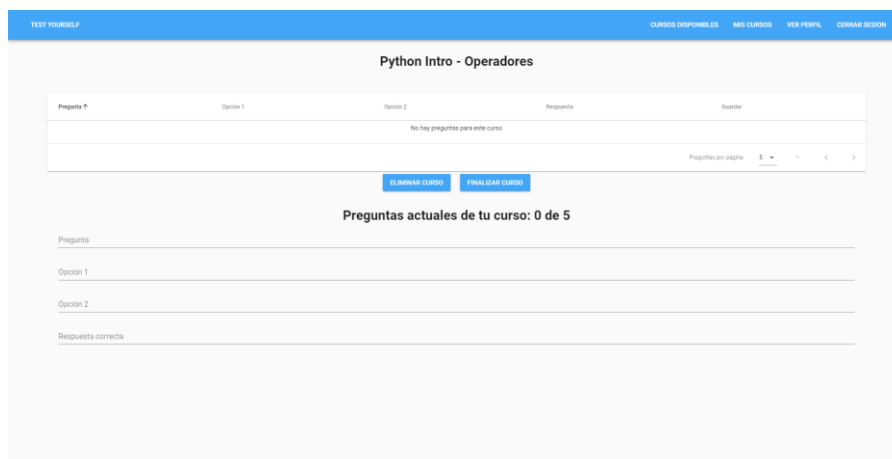


FIGURA 56 MANUAL DE USUARIO - CURSO – EDICIÓN

7.1.11 Animaciones en los cursos - Sesión iniciada

En ciertos cursos se podrán añadir animaciones que irán directamente relacionadas con una o más preguntas del curso. Estas animaciones estarán reproduciéndose desde que se abra el curso, en algunas se dará opción a repetir la animación.

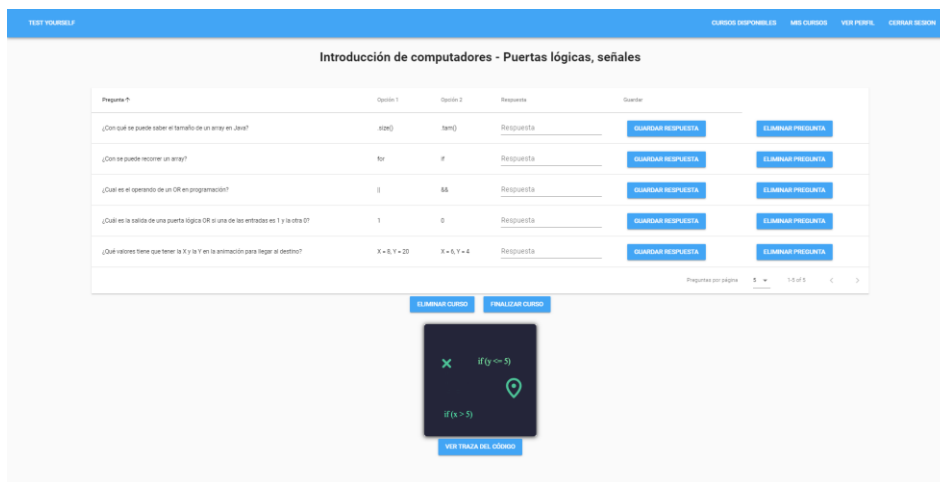


FIGURA 57 MANUAL DE USUARIO - CURSO - ANIMACIÓN

7.1.12 Perfil - Sesión iniciada

En la pantalla del perfil se podrá visualizar si el usuario es administrador, lo cual le permitirá crear más administradores, el nombre de usuario, el correo electrónico y el número de cursos completados. Además, tendrá la opción de cambiar la contraseña por una nueva.

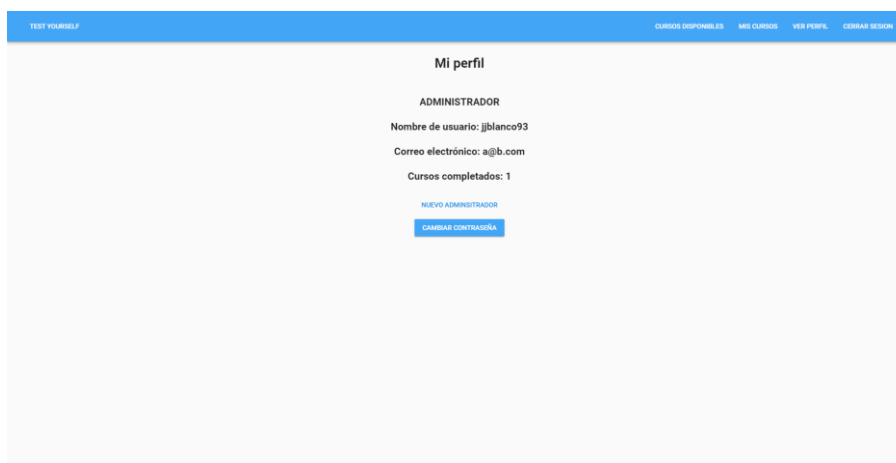


FIGURA 58 MANUAL DE USUARIO - PERFIL DE ADMINISTRADOR

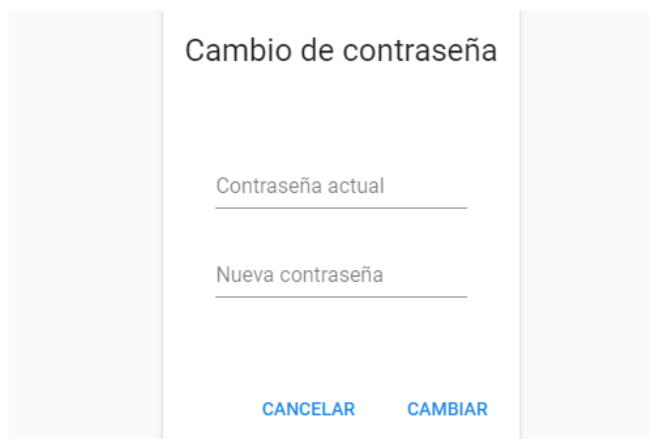


FIGURA 59 MANUAL DE USUARIO - CAMBIO DE CONTRASEÑA

7.1.13 Crear nuevo administrador

Si un administrador desde su perfil procede a crear uno nuevo es redirigido a esta pantalla, donde se le piden los datos del nuevo usuario que tendrá el rol de administrador, si cumple todas las verificaciones y validaciones.

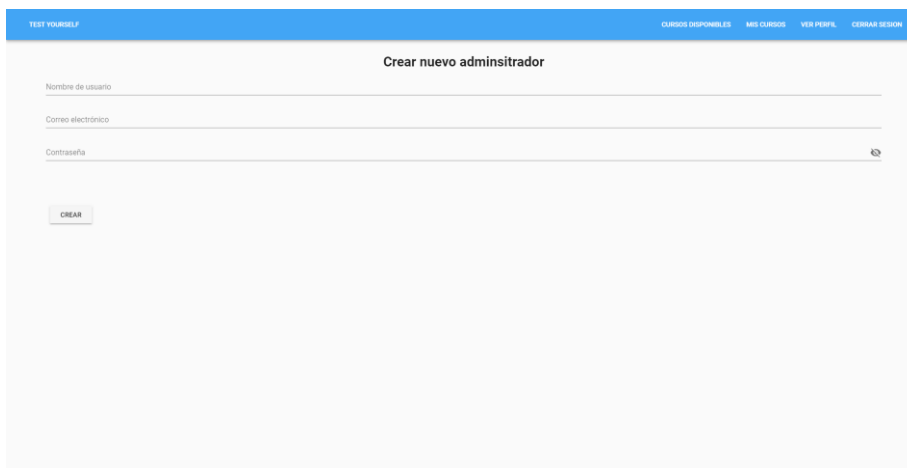


FIGURA 60 MANUAL DE USUARIO - CREAR ADMINISTRADOR

7.1.14 Filtros de cursos

Tanto en la pantalla de cursos disponibles como en la de cursos asignados, si hay un usuario con la sesión iniciada, se pueden filtrar los cursos haciendo uso del buscador que se encuentra situado encima de la tabla.

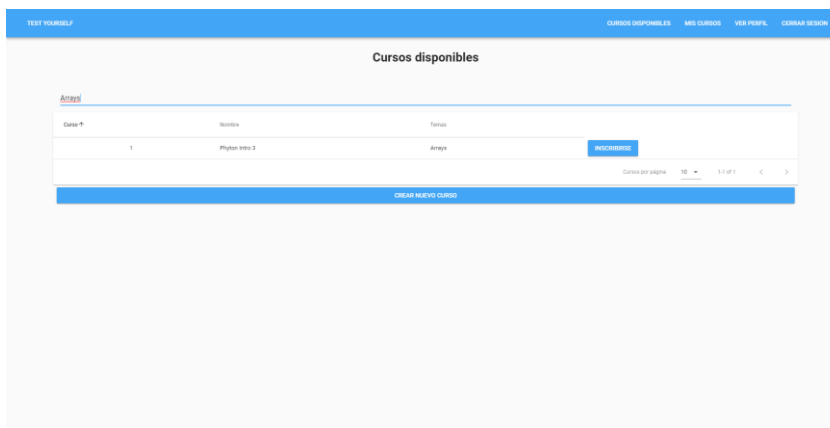


FIGURA 61 MANUAL DE USUARIO - CURSOS DISPONIBLES - FILTRO

TEST USER@UP

CURSOS DISPONIBLES MIS CURSOS MIS PROYECTOS ESTADÍSTICAS

Cursos de jblanco93

Operar

Curso	Nombre	Temas	Progreso	
1	Python Intro	Operaciones	Incompleto	Asignar Curso

Cursos por página: 10 20 30 40 50 60 70 80 90 100

FIGURA 62 MANUAL DE USUARIO - CURSOS ASIGNADOS - FILTRO

7.2 Anexo 2. Modelo de datos de la aplicación

En nuestro modelo de datos podemos observar como un usuario puede ser o bien un alumno o un administrador, en el caso del último se le da la opción de crear más administradores. Los usuarios pueden estar matriculados en muchos cursos o ninguno al igual que los cursos pueden estar asociados al usuario creador o muchos usuarios que estén inscritos al mismo. A su vez a un curso se le pueden agregar preguntas y estas preguntas quedan asociadas a un curso.

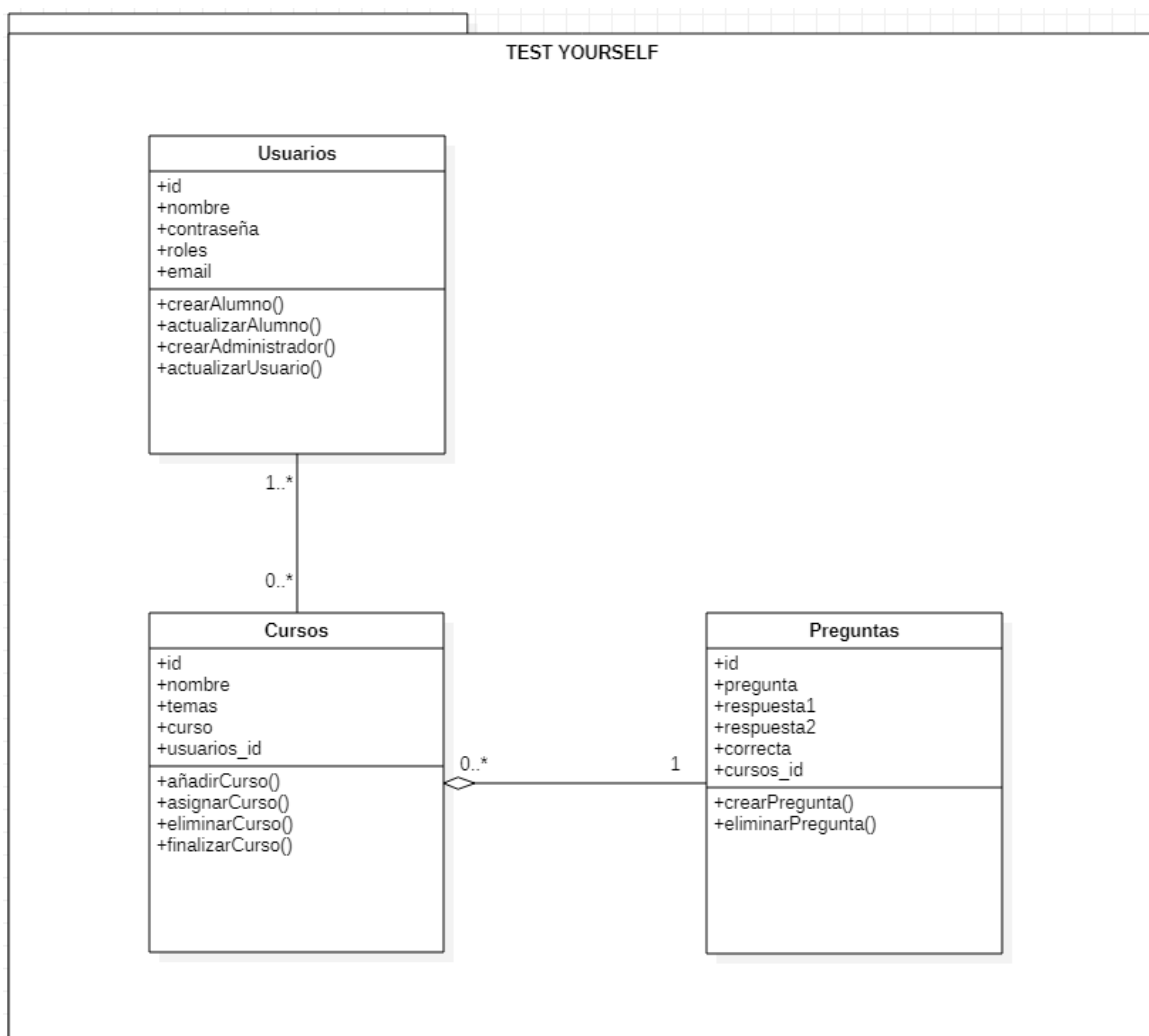


FIGURA 63 MODELO DE DATOS

7.3 Anexo 3. Diagramas de casos de uso de la aplicación

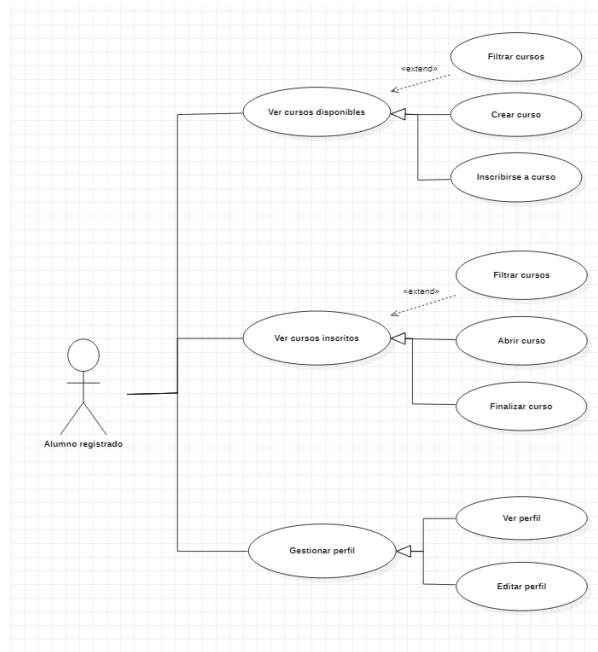


FIGURA 64 DIAGRAMA CASOS DE USO - ALUMNO REGISTRADO

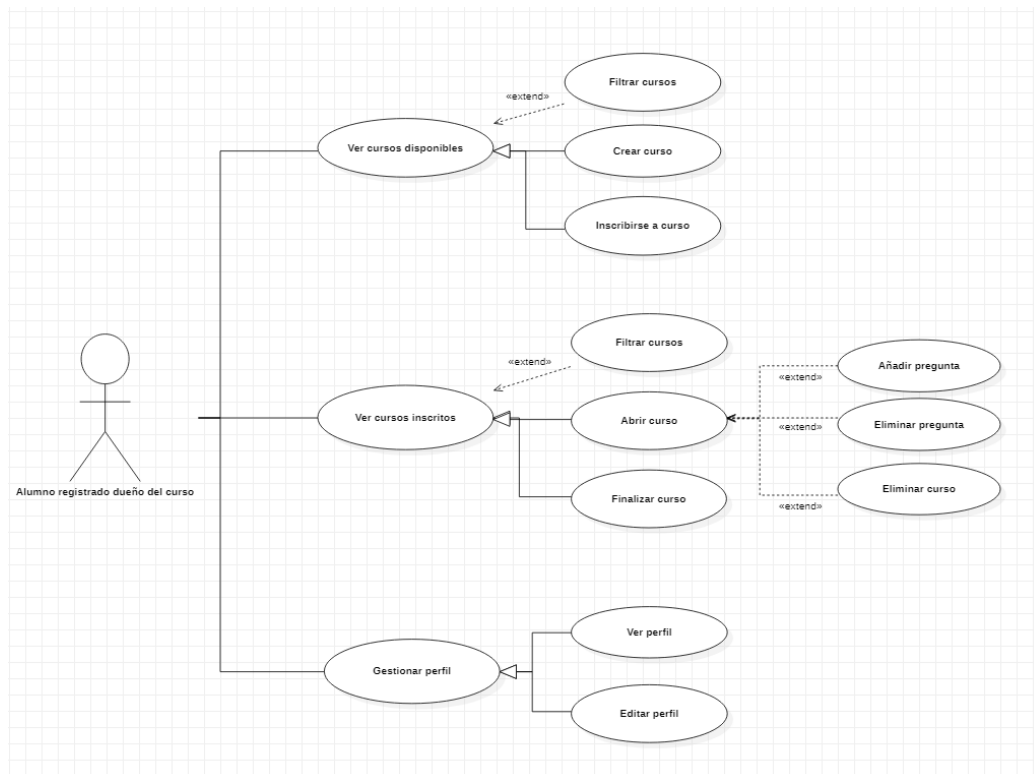


FIGURA 65 DIAGRAMA CASOS DE USO - ALUMNO REGISTRADO DUEÑO DEL CURSO

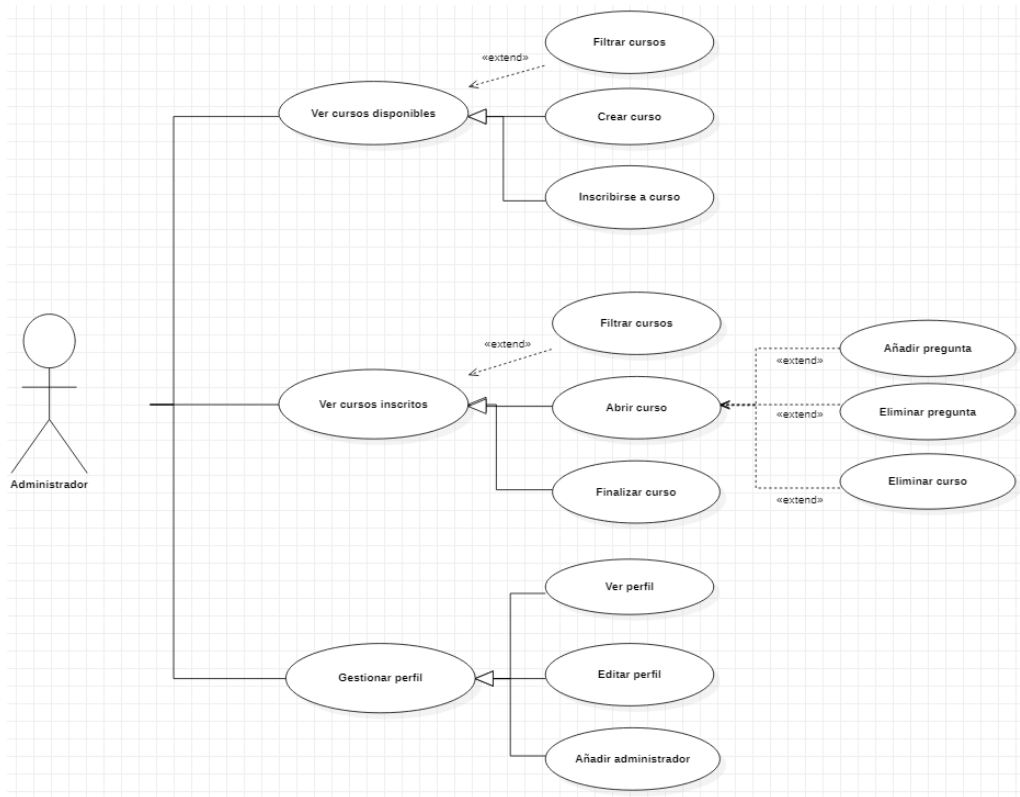


FIGURA 66 DIAGRAMA CASOS DE USO – ADMINISTRADOR

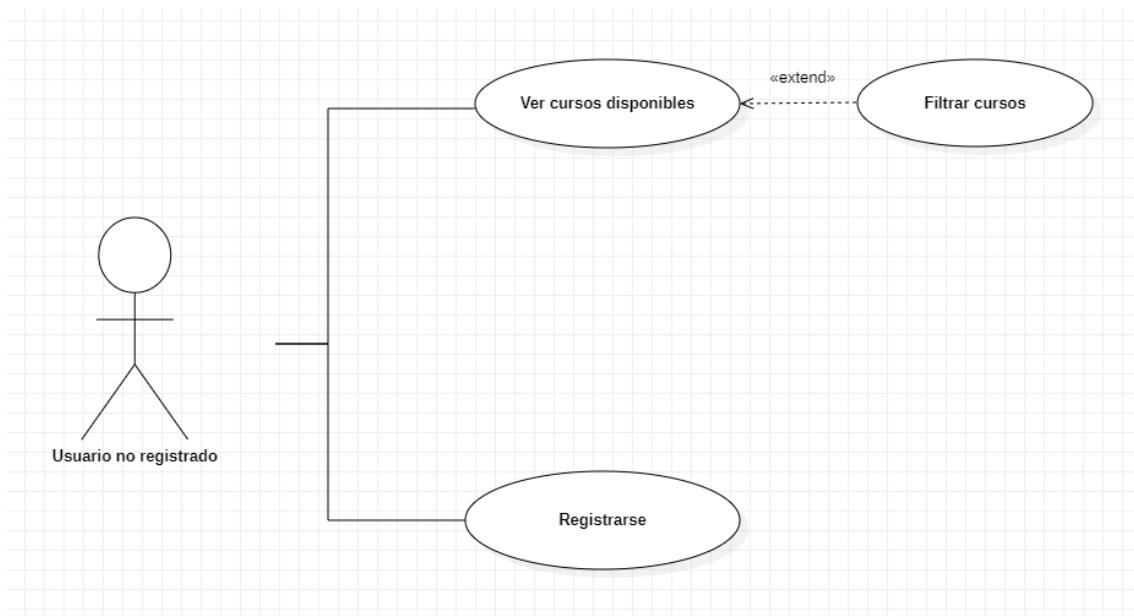


FIGURA 67 DIAGRAMA CASOS DE USO - USUARIO NO REGISTRADO

8. Bibliografía

- [1] Vue, Documentación de Vue. (2014). Recuperado de <https://vuejs.org/v2/guide/#Getting-Started>
- [2] Documentación de Vuex, What is Vuex?. (2018). Recuperado de <https://vuex.vuejs.org/>
- [3] Vuetify, Documentación de vuetify. (2016). Recuperado de <https://vuetifyjs.com/en/getting-started/quick-start>
- [4] CSS-GRID, Documentación de CSS-GRID. (2018). Recuperado de https://developer.mozilla.org/es/docs/Web/CSS/CSS_Grid_Layout
- [5] React, Documentación de React. (2018). Recuperado de <https://reactjs.org/docs/hello-world.html>
- [6] Design Patterns, Patrón model view controller. (2018). Recuperado de https://www.tutorialspoint.com/design_pattern/mvc_pattern.htm
- [7] Sébastien Deleuze, documentación para solventar el cors. (2015). Recuperado de <https://spring.io/blog/2015/06/08/cors-support-in-spring-framework>
- [8] Spring boot, Documentación de spring boot. (2018). Recuperado de <https://spring.io/guides/gs/spring-boot/>
- [9] Cecilio Álvarez Caules, Información sobre los DTO. (2017). Recuperado de <https://www.arquitecturajava.com/jpa-dto-data-transfer-object-y-jpql/>
- [10] FlexBox, Documentación de flexbox. (2018). Recuperado de <https://css-tricks.com/snippets/css/a-guide-to-flexbox/>
- [11] Proyectos agiles, Qué es SCRUM. (2008). Recuperado de <https://proyectosagiles.org/%20que-es-scrum/>
- [12] Webpack, Documentación sobre webpack. (2018). Recuperado de <https://webpack.js.org/concepts/>
- [13] Accessing Data with JPA, Documentación de JPA (2018). Recuperado de <https://spring.io/guides/gs/accessing-data-jpa/>

- [14] Node, How to install local packages?. (2017). Recuperado de <https://docs.npmjs.com/getting-started/installing-npm-packages-locally>
- [15] Axios, Documentación de Axios. (2017). Recuperado de <https://github.com/axios/axios>
- [16] Postman, Documentación de Postman. (2018). Recuperado de <https://www.getpostman.com/docs/v6/>
- [17] Y cómo no, nuestro querido Stackoverflow (2021).
Recuperado de <https://stackoverflow.com/>
- [18] Gobierno de Canarias, Programación en 4º ESO (2015).
Recuperado de https://www.gobiernodecanarias.org/cmsweb/export/sites/educacion/web/galerias/descargas/bachillerato/curriculo/nuevo_curriculo/nuevas_julio_2015/especificas/59_tecnologias_informacion_comunic_eso.PDF
- [19] Gobierno de Canarias, Programación en Bachillerato (2015).
Recuperado de https://www.gobiernodecanarias.org/cmsweb/export/sites/educacion/web/galerias/descargas/bachillerato/curriculo/nuevo_curriculo/nuevas_julio_2015/especificas/60_tecnologias_informacion_comunic_bach.pdf