



ugr | Universidad
de Granada

TRABAJO FIN DE GRADO
INGENIERÍA INFORMATICA



Autor

Antonio Marfil Sánchez

Directores

María Jesús Rodríguez Sánchez



FACULTAD DE EDUCACIÓN, ECONOMÍA Y TECNOLOGÍA DE CEUTA

Ceuta, 7 de julio de 2021

Índice general

1. Introducción	1
1.1. Estudio de aplicaciones existentes	2
1.2. Motivación e idea inicial	4
1.3. Objetivos	6
1.3.1. Objetivo principal	6
1.3.2. Objetivos secundarios	7
2. Planificación	11
2.1. Planificación inicial	11
2.2. Planificación resultante	12
2.2.1. Conclusión	12
3. Análisis	15
3.1. Requisitos Funcionales	15
3.1.1. RF1 Gestión de usuarios	16
3.1.2. RF2 Iniciar y cerrar sesión	16
3.1.3. RF3 Gestión de contenidos	16
3.1.4. RF4 Gestión de mensajes	17
3.1.5. RF5 Gestión de clases	17
3.1.6. RF6 Sistema de notificaciones	18
3.2. Requisitos no Funcionales	18
3.3. Diagramas de caso de usos	19
3.3.1. RF1 Gestión de usuarios	19
3.3.2. RF2 Iniciar y cerrar sesión	20
3.3.3. RF3 Gestión de contenidos	21
3.3.4. RF4 Gestión de mensajes	22
3.3.5. RF5 Gestión de clases	23
3.3.6. RF6 Sistema de notificaciones	26
4. Diseño	29
4.1. Diagrama E/R	30
4.2. Modelo Relacional	31
4.3. Patrón Arquitectónico	32

4.4. Diseño gráfico	34
5. Implementación	37
5.1. Marco tecnológico	37
5.1.1. GitHub	37
5.1.2. Postman	38
5.1.3. Visual Studio Code	39
5.1.4. Photoshop	39
5.1.5. DigitalOcean	40
5.1.6. Laravel Framework	40
5.1.7. Ionic Framework	41
5.2. Implementación	41
5.2.1. API	42
5.2.2. Dashboard web para administradores y profesores . .	46
5.2.3. APP Ionic	51
5.3. Futuras mejoras	56
6. Conclusión	65

Good Learn: agenda escolar online

Antonio Marfil Sánchez

Palabras clave: Sistema de Información, App móvil, agenda, colegio

Resumen

Debido a la situación pandémica que estamos atravesando actualmente, muchas de las cosas que hacíamos diariamente se han digitalizado debido a que el ser humano ha tenido que limitar sus estrechos contactos físicos. La importancia actual que existe sobre el uso de tecnologías para llevar a cabo tareas tan comunes como el asistir a clases se han visto obligadas a ser vía online.

Good Learn es una alternativa a la famosa agenda escolar donde los padres podían ver como son los avances a lo largo del curso de su hijo. El papel de la agenda escolar ha servido como canal de comunicación entre profesores y padres. Con la aparición de la Covid-19 el contacto sobre objetos físicos que puedan intercambiarse de persona a persona se ha visto suprimido por eso este sistema servirá como guía informativa para padres sobre las tareas que se deben realizar por parte del alumno, tales como las calificaciones del mismo, autorizaciones para posibles actividades extraescolares, visualizar el temario de las diferentes asignaturas o ver las faltas de asistencias del docente.

Cabe destacar la escalabilidad del sistema para que sea posible su uso en cualquier dispositivo desde ordenador hasta teléfono móvil. El incremento en el uso de este último hace que sea de vital importancia el uso del sistema en dispositivos comunes y en sistemas operativos tales como Android o IOS.

Good Learn: online school genda

Antonio Marfil Sánchez

Keywords: Information system, mobile App, agenda, school

Abstract

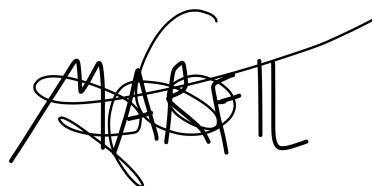
Due to the pandemic situation that we are currently going through, many of the things that we did on a daily basis have been digitized due to the fact that human beings have had to limit their close physical contacts. The current importance that exists on the use of technologies to carry out tasks as common as attending classes have been forced to be online.

Good Learn is an alternative to the famous school diary where parents could see how the progress is throughout the course of their child. The role of the school agenda has served as a communication channel between teachers and parents. With the appearance of Covid-19, contact on physical objects that can be exchanged from person to person has been abolished, so this system will serve as an informative guide for parents on the tasks that must be carried out by the student, such as grades of the same, authorizations for possible extracurricular activities, visualize the syllabus of the different subjects or see the absence of attendance of the teacher.

It should be noted the scalability of the system so that it can be used on any device from computer to mobile phone. The increase in the use of the latter makes it of vital importance to use the system on common devices and operating systems such as Android or IOS.

Yo, **Antonio Marfil Sánchez**, alumno de la titulación Ingeniería Informática de la **Facultad de Educación, Economía y Tecnología de Ceuta**, con DNI 45116837Z, autorizo la ubicación de la siguiente copia de mi Trabajo Fin de Grado en la biblioteca del centro para que pueda ser consultada por las personas que lo deseen.

Fdo: Antonio Marfil Sánchez

A handwritten signature in black ink, appearing to read "MARFIL". The signature is fluid and cursive, with the name written vertically and slightly slanted.

Granada a 7 de Julio de 2021 .

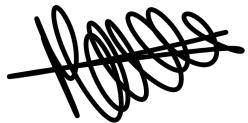
D. **María Jesús Rodríguez Sánchez**, Profesor del Área de Ciencias de la Computación e Inteligencia Artificial del Departamento Ciencias de la Computación e Inteligencia Artificial de la Universidad de Granada.

Informan:

Que el presente trabajo, titulado ***Good Learn: Agenda escolar virtual***, ha sido realizado bajo su supervisión por **Antonio Marfil Sánchez**, y autorizamos la defensa de dicho trabajo ante el tribunal que corresponda.

Y para que conste, expiden y firman el presente informe en Ceuta a 7 de julio de 2021 .

Los directores:



María Jesús Rodríguez Sánchez

Agradecimientos

Este trabajo es el reflejo de una vida de esfuerzo que han llevado a cabo mis padres para conseguir todos sus objetivos, una lucha constante contra viento y marea. Quiero darles las gracias por haberme brindado esta oportunidad y poder llegar hasta este punto de mi vida, conseguir otra meta más de las muchas que me quedan por delante. Muchas gracias Papá y Mamá por haberme enseñado lo que significa conseguir lo que te propones y por haberme inculcado todo lo que soy en la vida, sin vosotros esto no hubiera sido posible.

Capítulo 1

Introducción

El concepto "contactles" se ha instaurado hoy día con la aparición de la Covid-19 y la pandemia mundial que estamos sufriendo, es por ello que se ha experimentado un aumento del uso de la tecnología para realizar cualquier trabajo cotidiano.

El uso de dispositivos se ha convertido en algo tan común como el cepillarnos los dientes o el comer. En este proyecto vamos a elaborar una agenda escolar que servirá como medio de comunicación entre los profesores de un curso concreto con los tutores de los alumnos. La razón de este trabajo es reducir el contacto físico entre personas y la manipulación de objetos como es la agenda escolar en este caso.

Los centros educativos han vivido un salto exponencial en el uso de internet y de dispositivos electrónicos. Hemos pasado del lápiz y papel, de estar sentados en un aula a tener clases de manera online o entregas en plataformas digitales. Pero este aumento tecnológico también posee algunos problemas al ser tan drástico:

- **Formación del uso de tecnologías para docentes en centros educativos.** No todos tenemos las mismas capacidades para el uso de materiales digitales o para crear una simple videoconferencia. Un estudio de la media de edades en centros educativos [1] sitúa a España como unos de los países europeos con una media de jóvenes docentes más baja. Esto supone que el desarrollo de nuestro sistema debe ser muy intuitivo y con funciones bastante marcadas para un uso sencillo y que no pueda crear muchas confusiones.
- **Problemas de infraestructura.** Es un hecho que no todo el mundo dispone de medios para tener internet en casa o de tener un móvil u ordenador. En este gráfico [2] se puede observar mediante una encuesta realizada cuales son los principales problemas que se observan según

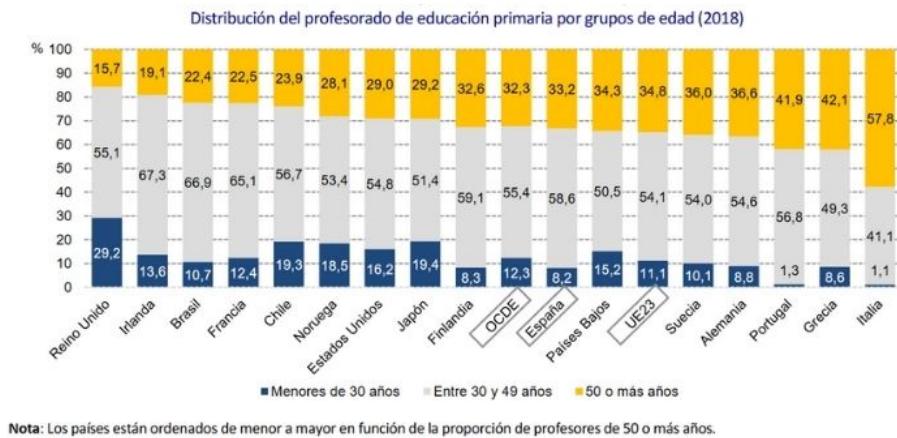


Figura 1.1: Grafica de las diferentes edades de los docentes del mundo. Fuente: Educaweb

docentes de toda España.

- **El smartphone como el dispositivo rey.** La tecnología móvil se ha convertido en el dispositivo preferido de cualquier usuario es por ello que actualmente es más común adquirir un smartphone antes que un laptop o desktop. La escalabilidad de una aplicación se ha convertido en unos de los puntos más fuertes para el desarrollo de aplicaciones. Hoy en día un dispositivo móvil se puede llegar a comparar como un ordenador en todos sus sentidos tanto que es capaz de realizar cualquier tarea que nuestro ordenador de hace 5 años atrás. En la encuesta de la Figura 1.3, se refleja qué dispositivos son más usados por edades y en qué porcentaje [3]

1.1. Estudio de aplicaciones existentes

En el mercado actual existen algunas plataformas que sirven de referencia para la implementación de nuestra aplicación. **Schooltivity** [4] es una agenda virtual que se implanta en centros educativos de primaria o guarderías. Su diseño está enfocado al uso de todas las partes, desde los docentes hasta los alumnos. Tiene un diseño ajustado a una edad de entre 3-8 años que se ajusta perfectamente a las guarderías. Como funcionalidades cabe destacar métodos de pagos dentro del centro, aulas educativas e interactivas para el alumnado, control de alimentación y sueño. Posee un horario de

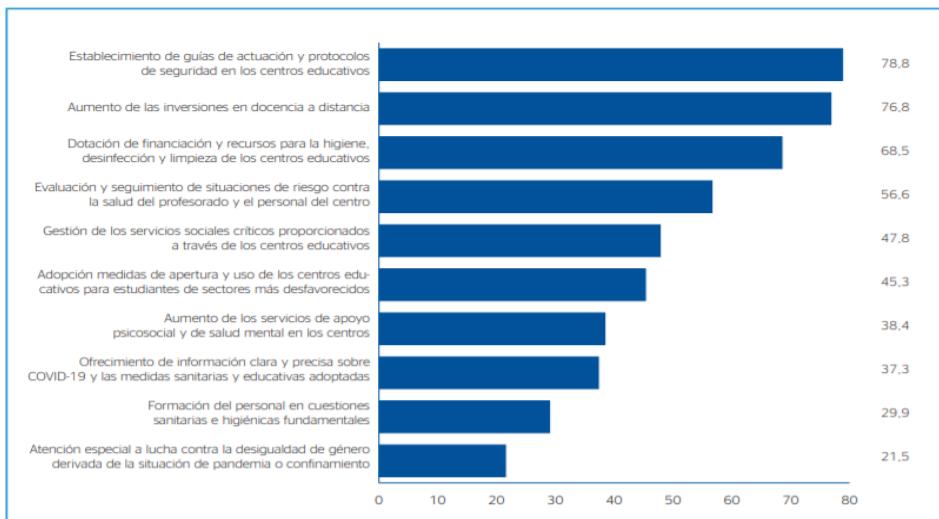


Figura 1.2: Grafica de los principales problemas que ven los docentes. Fuente: Educación conectada y BBVA

actividad de los padres para conocer las horas no laborales y para tener en cuenta las horas donde son activos para el centro educativo.

En Ceuta algunos colegios han implantado algunos sistemas de agendas escolares online como es el caso del colegio concertado La Inmaculada que ha implantado “Alexia suit classroom” [5]. En Ceuta algunos colegios han implantado algunos sistemas de agendas escolares online como es el caso del colegio concertado La Inmaculada que ha implantado **Alexia suit classroom**.

En cuanto a sus funcionalidades se observa que la app posee un feed como una red social en la que se van publicando las notas del alumno en sus diferentes asignaturas y en las que se le pueden añadir algún tipo de observaciones. Los contenidos que se trabajan en cada asignatura son subidos a la app para la revisión posterior de los padres y para que tengan constancia de lo que realmente se trabaja en cada asignatura de sus hijos. Existe una comunicación directa con los diferentes profesores lo que facilita el intercambio de mensajes entre docentes y padres.

ClassDojo [6] es un sistema comunicativo entre profesores y padres bastante intuitivo de usar y muy fácil a la vista. Este sistema tiene una estructura muy familiar que conoce en su inmensa mayoría cualquier persona que es el de red social tal como twitter o facebook. Su panel principal con

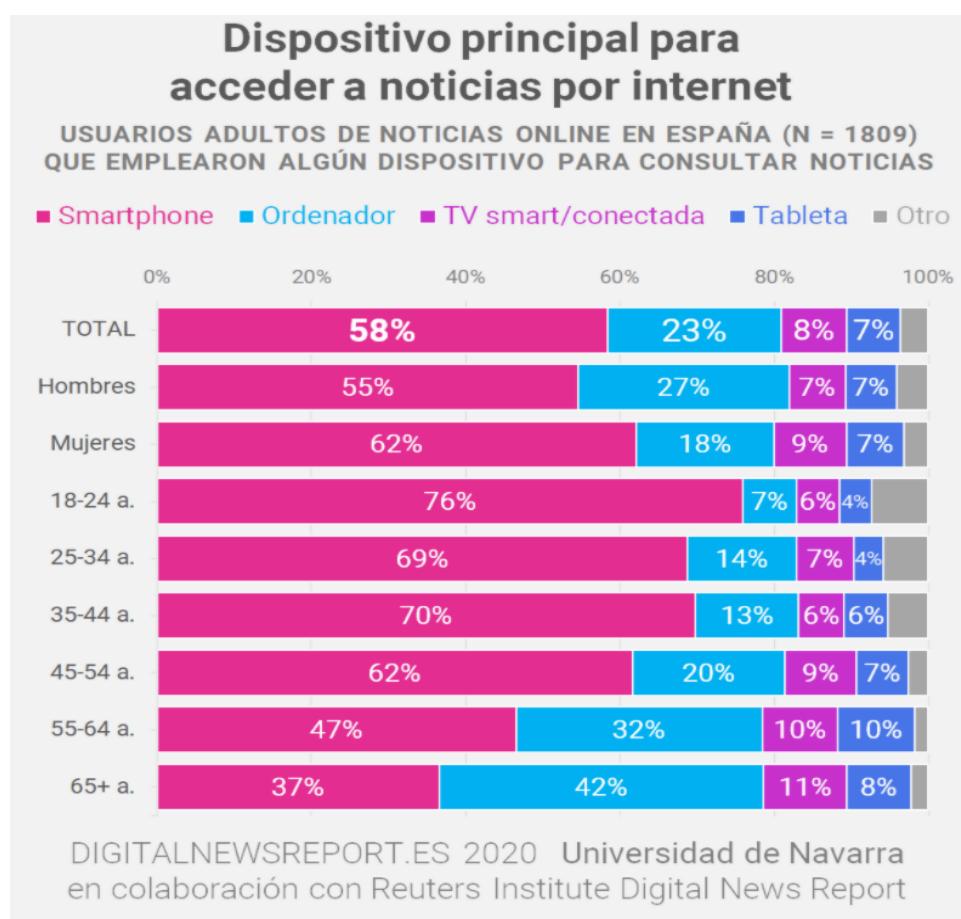


Figura 1.3: Grafica de dispositivo mas usado segun edad y sexo. Fuente: Digital News Reports

las diferentes clases que tiene el profesor en este caso. Otra pantalla con los alumnos que posee cada curso. Un feed parecido al de cualquier red social donde se van posteando los avances de sus alumnos o cualquier tipo de comunicado importante, dando la opción a poder realizar comentarios en la misma publicación. Por último la pantalla de mensajería entre padres y docentes donde se podrá abrir una nueva conversación y crear el canal de comunicación directamente con la persona que tutela a los alumnos en el centro educativo.

1.2. Motivación e idea inicial

Según los estudios mostrados anteriormente, la idea para el desarrollo de esta aplicación será una aplicación web responsive que sea capaz de adap-

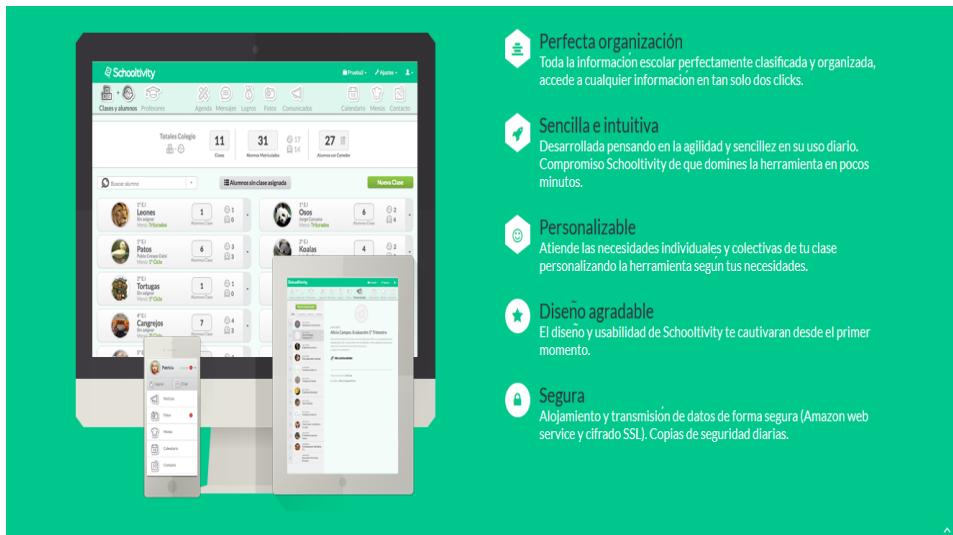


Figura 1.4: Características de Schooltivity. Fuente: Schooltivity

tarse a cualquier dispositivo ya sea ordenador, móvil o tablet. Esto nos proporcionará una gran escalabilidad en el uso para el usuario y que pueda ser útil en cualquier momento y en cualquier entorno y reduciendo las posibles limitaciones de dispositivos de uso para la aplicación.

Para comenzar nuestro sistema se llamará “**Good Learn**” y de esta formas nos referimos al sistema como tal. La idea es que sirva como intermediario en las comunicaciones diarias de los tutores de los alumnos con el colegio en cuestión eliminando el uso de mensajerías instantáneas tales como Whatsapp. Good Learn poseerá un sistema de mensajería interna para cualquier consulta entre los usuarios de la misma. El sistema se desarrollara en base a los siguientes roles:

- **Administrador**. Este rol será el encargado de la administración total del sistema para el colegio en cuestión. Podrá añadir, eliminar y modificar en cuestión las aulas, asignaturas, profesores y tutores. Todas estas modificaciones se verán reflejadas en un panel de administración que será intuitivo para poder llevar a cabo toda la gestión de la aplicación sin ningún problema.
- **Profesor**. Cada docente se le asignará una asignatura en cuestión y una clase lo que le permitirá añadir posibles tareas a entregar, fechas de exámenes, anotaciones sobre el alumno, aclaraciones sobre el material a llevar para posibles actividades o abrir una solicitud para que acepte el tutor del alumno para alguna actividad extraescolar o excursión fuera del centro. El rol de docente tendrá activo un panel de gestión

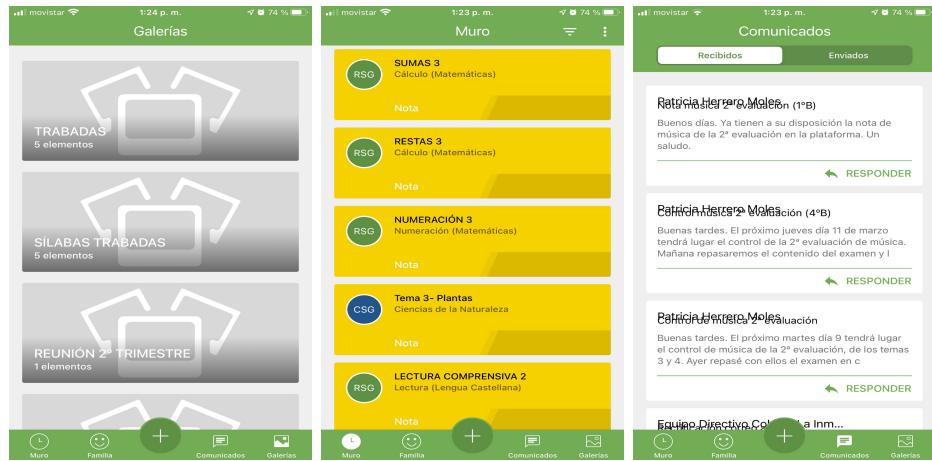


Figura 1.5: Vistas de Alexia Suit Educativa en móvil. Fuente: Alexia Suit Educativa

de clase para poder realizar todas las funciones CRUD (acrónimo de “Crear, Leer, Actualizar y Borrar” del original en inglés: Create, Read, Update and Delete).

- **Padre.** Se asigna este rol al padre o madre del alumno y podrá visualizar toda la información propia del curso de su hijo. Como funcionalidad de este rol el tutor podrá comunicarse directamente con los docentes de cada asignatura y podrá dar permisos a los docentes para que sus hijos puedan realizar actividades extraescolares y excursiones.
- **Alumno.** Será el rol identificativo en el que se le asignará a cada padre y que se encargará de tener asignado un curso con sus diferentes asignaturas. Los padres se relacionarán con este haciendo que varios padres puedan acceder a la información del alumnos en el curso.

Good Learn dispondrá de un calendario donde se refleja cualquier actividad que se va a realizar o fechas claves como exámenes o entregas de cualquier ejercicio. El sistema dispondrá de una parte donde se puedan publicar circulares o comunicados generales para todos los cursos con la idea de que el centro educativo esté al completo informado de todo.

1.3. Objetivos

1.3.1. Objetivo principal

Desarrollar un sistema de información que contenga dos aplicaciones conectadas simultáneamente, una de escritorio que además sirva para la gestión

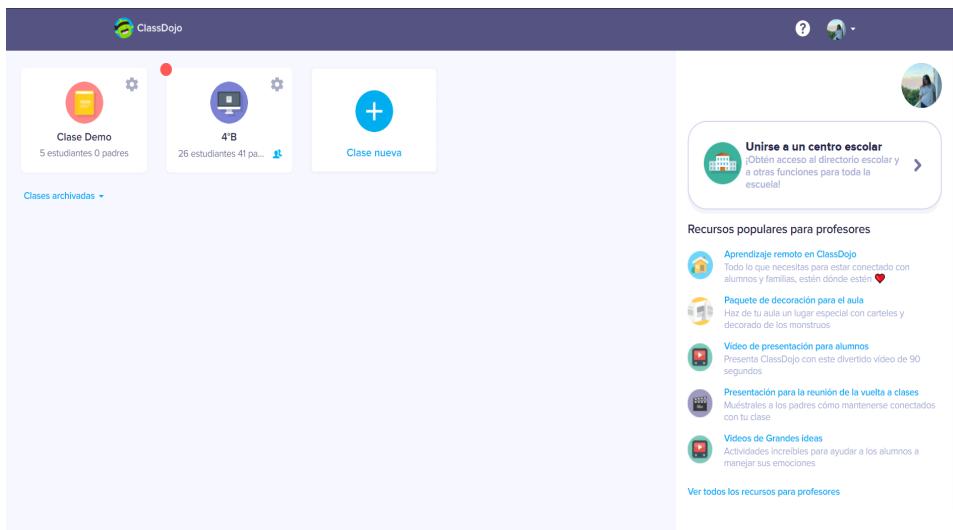


Figura 1.6: Lista de clases. Fuente: ClassDojo

de todo el personal de un colegio y otra para establecer una comunicación de los padres con los profesores de sus hijos y puedan estar informados a través de la misma de todos los avances académicos de los mismos.

1.3.2. Objetivos secundarios

- Para el desarrollo de este sistema de información se va a utilizar tecnología PHP usando un framework del tipo MVC (Model, View, Controller) llamado Laravel. El objetivo es adquirir conocimientos sobre el funcionamiento de este tipo de framework estructural para el desarrollo de nuestro proyecto con el fin de crear una herramienta lo suficientemente potente para la gestión de un centro educativo.
- Analizar los diferentes sistemas existentes en el mercado y sus funcionalidades para poder elaborar un sistema capaz de reagrupar las ventajas de uno y otro para reducir posibles debilidades entre ellos.
- Desarrollar una base de datos y una gestión de la misma con un panel en el back-end suficientemente intuitivo para docentes con conocimientos a nivel medio sobre herramientas tecnológicas.
- Crear un sistema totalmente escalable a cualquier dispositivo para su utilización sin ningún problema de donde se encuentre cualquier usuario y poder usarse en cualquier momento desde cualquier dispositivo.
- Implementar una app multiplataforma usando el framework Ionic basado en el framework MVC Vue.js. El objetivo es obtener una App

The screenshot shows the ClassDojo platform's 'Aula' (Classroom) section. At the top, it displays '4ºB' with 26 students and 41 parents. Below the header are navigation tabs: 'Aula' (which is active and highlighted in blue), 'Carpetas', 'Historia de la clase', and 'Mensajes'. To the right of these are icons for notifications (1), help, and user profile. The main area is titled 'Estudiantes Grupos' (Students Groups). It lists 26 students in three rows:

- Row 1:** Toda la clase (22), Ali Aghbal (2), Assil Boulahfa (2), Eloy Sánchez Gon... (0), Fati Ahssan (1), Firas El Kaloussi (0), Firdaus El Marouani (1), Haroun Abderrahman (0), Jawahir Mohamed Ch... (1), Jose Jaravello Ger... (1), Karim Lahsen (0), Mariam Ahardoum (1).
- Row 2:** Mariam Ghajghouj (4), Monsef Mohamed (0), Nargis Ech-Chetty (2), Nasim Mustafa (1), Nasrallah El Blai (0), Noor Ez Zaouyah (0), Renia Hassan (1), Riheb El Mahdi (2), Safa Bukachbia (0), Shirin Abdela (1), Slama Benaim (0).
- Row 3:** Tasnim Chelal Mena... (0), Tasnim Mohamed (1), Zaina Belachkar (2), and a 'Añadir alumnos' (Add Students) button.

At the bottom of the page are several utility links: Toolkit, Asistencia, Seleccionar varios, Al azar, Contador de tiempo, Grandes ideas, and Invita familias (97%).

Figura 1.7: Lista de alumnos. Fuente: ClassDojo

móvil totalmente funcional y que sirva como muestra de información del sistema.

- Tener una API para conseguir usarla en cualquier tipo de tecnología de desarrollo y utilizarla conectándola entre las diferentes interfaces y crear una conexión cliente-servidor mediante rutas de la propia API.

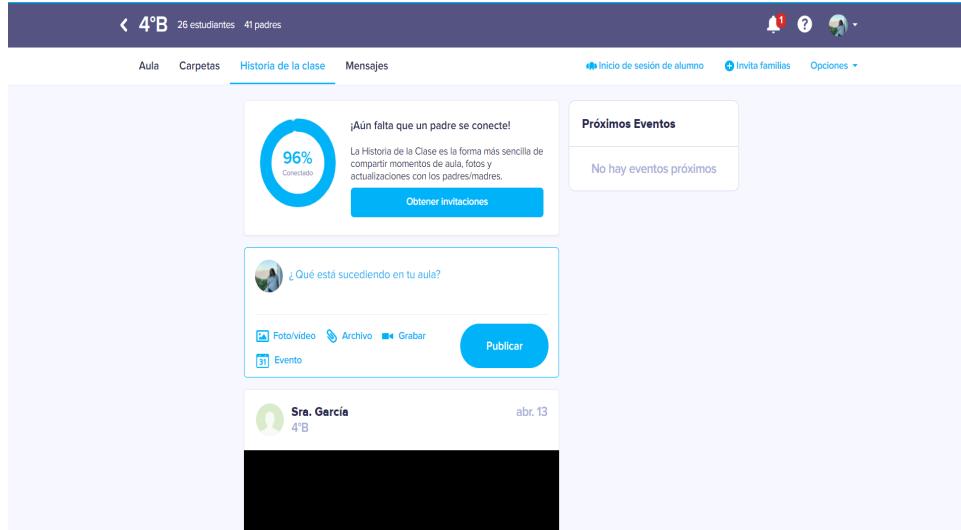


Figura 1.8: Publicaciones. Fuente: ClassDojo

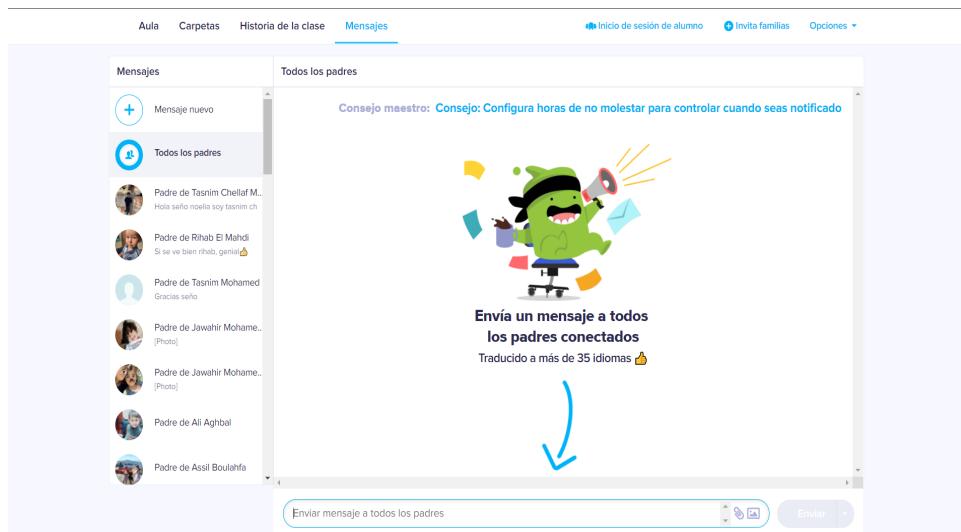


Figura 1.9: Mensajes. Fuente: ClassDojo

Capítulo 2

Planificación

2.1. Planificación inicial

El desarrollo de Good Learn se llevará a cabo según las siguientes etapas representadas en el siguiente diagrama de Gantt.

Semanas	05abr - 11abr	12abr - 18abr	19abr - 25abr	26abr - 02may	03may - 09may	10may - 16may	17may - 23may	24may - 30may	31may - 06jun	07jun - 13jun	14jun - 20jun	21jun - 26jun
Estudio previo y aprendizaje	✓											
Analisis			✓	✓								
Diseño				✓	✓							
Implementación						✓	✓	✓	✓	✓		
Pruebas											✓	✓

Figura 2.1: Diagrama de Gantt resultante. Fuente: Elaboración propia

Se plantea un flujo de trabajo en forma de cascada, esto quiere decir que no se continuará con la siguiente etapa sin terminar la anterior. Cada tarea que se refleja en el diagrama corresponde a:

- **Estudio previo y aprendizaje**, primera etapa donde se realiza el estudio previo de aplicaciones ya existentes en el mercado y se crea una evaluación de los mismos sacando algunas ideas para el análisis posterior del sistema. Mediante esta primera etapa se obtiene una idea generalizada para la creación de la aplicación.
- **Analisis**, se describirán cada funcionalidad y servicio que posee el sistema desde como el usuario inicia sesión en el sistema hasta las funciones de gestión del aula.

- **Diseño**, creación de diagramas que describen el funcionamiento interno a nivel software del sistema. Se desarrollarán diagramas E/R y creación del diseño de la base de datos.
- **Implementación**, desarrollo del sistema de información siguiendo el análisis de requisitos y diseño elaborado anteriormente con las diferentes tecnologías a usar.
- **Pruebas**, se comprueba que todos los requisitos funcionales del sistema descritos funcionan correctamente y en caso de encontrar fallos arreglarlos. Se revisa lo desarrollado hasta el momento y se intentan buscar mejoras en el sistema o posibles implementaciones que puedan servir de ayuda tales como arreglar elementos de las diferentes vistas o posibles mejoras en el rendimiento del sistema.

2.2. Planificación resultante

Semanas	05abr - 11abr	12abr - 18abr	19abr - 25abr	26abr - 02may	03may - 09may	10may - 16may	17may - 23may	24may - 30may	31may - 06jun	07jun - 13jun	14jun - 20jun	21jun - 27jun	28jun - 04jul	05jul - 07jul
Estudio previo y aprendizaje	✓													
Análisis		✓	✓											
Diseño				✓	✓	✓								
Implementación							✓	✓	✓	✓	✓	✓	✓	
Pruebas														✓

Figura 2.2: Diagrama de Gantt propuesto. Fuente: Elaboración propia

2.2.1. Conclusión

Como conclusión del tiempo que se ha dedicado a cada parte del proyecto cabe destacar que no ha sido a la perfección en los tiempos establecidos antes del proyecto. Tres de las diferentes partes se han visto afectadas en el tiempo de su desarrollo.

Una de ellas es el diseño del sistema, cuya parte pertenece a la elaboración del modelo relacional, diagrama de E/R y el diseño gráfico para el logotipo y paleta de colores. El tiempo que se ha dedicado a esta parte ha sido de una semana más de lo establecido debido a la carga lectiva del curso a su vez. La creación del modelo relacional y los cambios pertinentes de la primera idea que se tenía dio lugar a la creación de tablas nuevas tales como las notificaciones y por lo consiguiente la modificación de toda la estructura relacional.

La implementación ya se ve afectada con una semana de retraso debido a la semana añadida al diseño, además de que la implementación a su vez se

ha alargado dos semanas más de lo que debería haber sido. En esta fase del proyecto se ha tenido que trabajar contra problemas que han ido resultando a medida que se implementa el sistema. Se requiere un estudio previo de las diferentes tecnologías que se iban a usar para su desarrollo por lo que la lectura y formación de lenguajes, frameworks y librerías han abarcado un tiempo considerable tomando importancia en la implementación.

Por último las pruebas del sistema se han visto limitadas en el tiempo por los retrasos que han conllevado etapas anteriores afectando a esta última la que no se ha tenido un tiempo bastante prolongado para haber añadido optimizaciones en el sistema en cuanto a eficiencia.

El esquema inicial se ha visto afectado en su gran mayoría por la implementación debido a la complejidad de desarrollo que conlleva este proyecto y haciendo que las pruebas se vieran afectadas por la limitación del tiempo. En cuanto a organización se esperaba una mayor fluidez en algunos puntos de la implementación viendo retrasada por la formación en las tecnologías usadas y el conocimiento de ellas.

Capítulo 3

Análisis

En esta etapa del proyecto se describirán los requisitos funcionales que posee el sistema como sus propiedades (requisitos no funcionales).

3.1. Requisitos Funcionales

Good Learn se ha pensado para ser usada solo y exclusivamente por usuarios registrados en su base de datos y con una idea jerárquica en los roles propios de los de los usuarios de un centro educativo. Los usuarios no podrán ser creados por cualquier persona, si no que un administrador del centro educativo será el encargado de crear las diferentes cuentas ligadas a su alumnado, dando dicho acceso a los tutores de los alumnos. Esta creación de usuarios tan restrictiva se basa en la abstracción del centro educativo en cuestión para crear un sistema único para cada uno de ellos y que sea gestionado por el mismo centro.

Se creará una funcionalidad de mensajería directa entre usuarios del mismo sistema sin tener en cuenta el rol que ejerce cada uno de estos usuarios. Esto quiere decir que cualquier usuario podrá enviar un mensaje a otro sin ningún problema. Good Learn también tendrá un feed de publicaciones tal como twitter en el que el docente pueda publicar que está sucediendo en el aula con archivos de imágenes, encuestas o firma de autorización de alumnos para posibles actividades extraescolares.

La misma aula virtual tendrá un sistema de temarios donde el docente podrá subir los contenidos que los docentes están llevando al día o posibles ejercicios que tengan que entregar con una fecha límite de ellos. Se podrá ver el avance del alumnado en las diferentes asignaturas y cuales son sus notas.

3.1.1. RF1 Gestión de usuarios

- **RF1.1 Creación de usuario**, el sistema será el encargado de dar de alta al usuario y no el mismo. Esta funcionalidad la tendrá solo y exclusivamente el rol de Dirección del centro educativo. Ningún usuario por sí mismo será capaz de llenar un formulario para crear una cuenta.
- **RF1.2 Modificación de datos**, cualquier usuario dentro del sistema será capaz de modificar cualquiera de sus datos tales como nombre de usuario, contraseña o correo entre otros.
- **RF1.3 Dar de baja al usuario**, el sistema será el encargado de dar de baja a cualquier usuario que esté en la base de datos y esta funcionalidad reside solo y exclusivamente en el rol de Dirección del centro educativo.

3.1.2. RF2 Iniciar y cerrar sesión

- **RF2.1 Login**, en caso de que dicho usuario y contraseña estén registrados en la base de datos de Good Learn el usuario será capaz de entrar al resto de funcionalidades propias que se les mostrará según el rol al que pertenezca y así pudiendo usar el sistema.
- **RF2.2 Logout**, estando el usuario conectado al sistema esta funcionalidad será la encargada de destruir todas las variables de sesión creadas en login y el usuario será capaz de desconectarse del sistema.

3.1.3. RF3 Gestión de contenidos

- **RF3.1 Creación de publicaciones**, el docente podrá subir a un feed imágenes, documentos o mensajes los cuales serán visibles por los padres que tengan algún alumno en su aula. Dicho contenido será ilustrativo para saber sobre sus hij@s en el aula.
- **RF3.2 Modificación de publicaciones**, el docente será capaz de modificar cualquier tipo de información que haya podido publicar en el feed.
- **RF3.3 Eliminar publicaciones**, el rol de docente será capaz de eliminar por completo cualquier contenido publicado en el feed.

3.1.4. RF4 Gestión de mensajes

- **RF4.1 Enviar mensajes**, cualquier usuario con la sesión iniciada en el sistema podrá enviarle otro mensaje de texto a otro usuario sin ningún tipo de problema creando un canal de comunicación privado entre ambos usuarios.
- **RF4.2 Modificar mensajes**, los mensajes enviados a los diferentes usuarios podrán ser modificados sólo y únicamente su contenido.
- **RF4.3 Eliminar mensajes**, el usuario podrá decidir si quiere eliminar un mensaje de texto enviado a otro usuario.

3.1.5. RF5 Gestión de clases

- **RF5.1 Creación de notas**, el rol de docente podrá evaluar las diferentes tareas que se realicen en clase y serán mostradas solo y exclusivamente al tutor que le es debido, es decir las notas no serán públicas para todos los tutores.
- **RF5.2 Modificación de notas**, el docente será capaz de modificar cualquier tipo de datos relacionado con las notas asignadas a los distintos alumnos.
- **RF5.3 Eliminar nota**, el docente podrá eliminar cualquier calificación que se haya asignado a los alumnos.
- **RF5.4 Creación de contenidos/temarios**, el rol de docente será capaz de subir al sistema imágenes, documentos o mensajes los cuales serán visibles por los padres que tengan algún alumno en su aula. Dicho contenido será ilustrativo para saber los contenidos propios que sus hij@s se enfrentan en cada asignatura.
- **RF5.5 Modificación de contenidos/temarios**, el docente será capaz de modificar cualquier tipo de información referenciada a los contenidos de la asignatura.
- **RF5.6 Eliminar contenido/temarios**, el rol de docente será capaz de eliminar por completo cualquier contenido publicado en la asignatura.
- **RF5.7 Creación de autorizaciones**, esta funcionalidad será única y exclusivamente del docente el cual podrá subir un documento para que los padres y madres autoricen a los alumnos en cualquier actividad del centro educativo en cuestión.

- **RF5.8 Modificacion de autorizaciones**, el docente podra cambiar cualquier dato que haga referencia a las autorizaciones de su clase.
- **RF5.9 Eliminar autorización**, el docente podrá eliminar cualquier autorización publicada en la asignatura.
- **RF5.10 Sistema de asistencia**, el tutor tendrá disponible un panel con un calendario donde se pueda reflejar las faltas de asistencia que tiene cada alumno y poder modificarlas y eliminarlas según vea conveniente.

3.1.6. RF6 Sistema de notificaciones

- **RF6.1 Listado de notificaciones**, cualquier usuario con la sesión iniciada en el sistema podrá visualizar un feed con las últimas notificaciones referidas a los elementos del sistema que no haya visto y que se haga referencia al usuario en cuestión..

3.2. Requisitos no Funcionales

Los requisitos funcionales no son aquellos que guarde o modifique información, sino que son características del funcionamiento del sistema que viene implícita en el desarrollo del mismo y que podrían tratarse de atributos de calidad para el sistema.

- **RNF1 Disponibilidad**, este sistema está pensado para implantarse en centros educativos por lo que la disponibilidad del sistema será máxima. El uso de Good Learn será primordial para llevar a cabo la gestión del centro educativo y por tanto el funcionamiento del trabajo dentro de él.
- **RNF2 Usabilidad**, será la capacidad intuitiva que tendrá el sistema para su fácil entendimiento y manejo del mismo. El sistema servirá como guía para el usuario que podrá aprenderlo sin ningún problema basándose en apps muy usadas del mercado.
- **RNF3 Portabilidad**, Good Learn será multiplataforma por lo que podrá ser usado en cualquier tipo de dispositivo y por lo tanto es portable en todos los aspectos.
- **RNF4 Eficiencia**, los tiempos de carga y de respuesta del sistema deben ser lo mínimo posible siendo su uso lo más fluido posible.

- **RNF5 Confidencialidad**, el sistema será instalado en la red local y los datos además serán protegidos por lo cual todo tipo de información sensible será capada para ser imposible a su acceso.

3.3. Diagramas de caso de usos

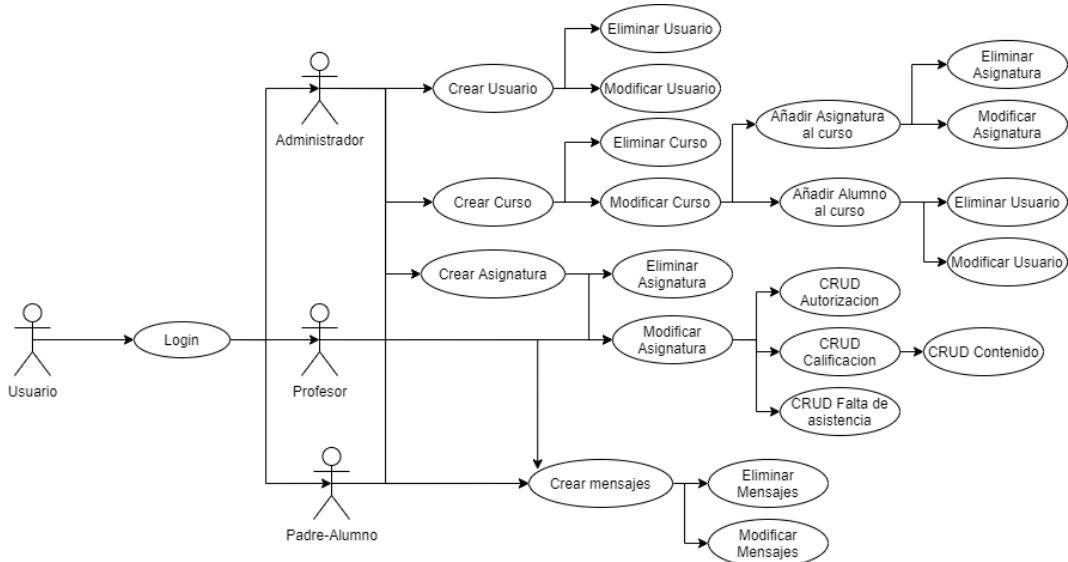


Figura 3.1: Diagrama de caso de uso. Fuente: Elaboracion Propia

Cada caso de uso está agrupado según los requisitos funcionales que se han descrito con anterioridad.

3.3.1. RF1 Gestión de usuarios

- **Caso de uso 1: Creación de usuario**

Caso de uso	Creación de usuario
Actores	Administrador
Precondición	No puede existir un usuario con el mismo D.N.I ni con el mismo correo electrónico y el administrador tiene que tener la sesión iniciada.
Postcondición	Creación de un nuevo usuario en el sistema Good Learn
Resumen	El administrador creará un nuevo usuario en el sistema que se le asignará a un alumno

Cuadro 3.1: Creación de usuario

- **Caso de uso 2: Modificación de usuario**

Caso de uso	Modificación de usuario
Actores	Usuario registrado
Precondición	El usuario tiene que estar registrado y con la sesión iniciada en el sistema.
Postcondición	Los datos del usuario en cuestión serán modificados.
Resumen	El usuario ya con la sesión iniciada podrá modificar sus datos para iniciar sesión o cualquier dato relacionado con su cuenta.

Cuadro 3.2: Modificación de usuario

- **Caso de uso 3: Dar de baja al usuario**

Caso de uso	Dar de baja al usuario
Actores	Administrador
Precondición	El usuario tiene que estar registrado, con la sesión iniciada en el sistema y con el rol de Administrador.
Postcondición	Los datos del usuario en cuestión serán modificados.
Resumen	El administrador desde su panel de usuarios podrá dar de baja al usuario que quiera y eliminarlo del sistema.

Cuadro 3.3: Dar de baja al usuario

3.3.2. RF2 Iniciar y cerrar sesión

- **Caso de uso 4: Login**

Caso de uso	Login
Actores	Usuario registrado
Precondición	Los datos del usuario deben existir en la base de datos.
Postcondición	El usuario tendrá acceso al sistema y se crearán las variables de sesión para que pueda acceder.
Resumen	Una vez ingresado los datos del usuario se comprobará si existe y en caso de que exista el usuario podrá entrar a todas las funcionalidades de GoodLearn y el sistema creará sus variables de sesión ligadas a dicho usuario.

Cuadro 3.4: Login

- Caso de uso 5: Logout

Caso de uso	Logout
Actores	Usuario logeado
Precondición	El usuario tiene que estar registrado y con la sesión iniciada en el sistema.
Postcondición	Se eliminarán las variables de sesión y el usuario cierra su sesión donde no podrá acceder a las funcionalidades del sistema.
Resumen	El usuario cierra sesión desde el sistema y este eliminará sus variables de sesión asignadas cuando se ha logeado y así restringe el acceso a las funcionalidades del sistema.

Cuadro 3.5: Logout

3.3.3. RF3 Gestión de contenidos

- Caso de uso 6: Creación de publicaciones

Caso de uso	Creación de publicaciones
Actores	Docente logeado
Precondición	Docente con una sesión iniciada.
Postcondición	El docente creará un publicación en el feed para que puedan verla todos los padres con la sesión iniciada.
Resumen	Una vez que el docente se encuentra con la sesión iniciada creará una publicación con contenido para que sea visibles en una lista de contenidos en forma de feed.

Cuadro 3.6: Creación de publicaciones

- **Caso de uso 7: Modificación de contenidos**

Caso de uso	Modificación de contenidos
Actores	Docente logeado
Precondición	Docente con una sesión iniciada y con publicación creada.
Postcondición	Se modificará los datos correspondiente al contenido de la publicación.
Resumen	El docente accederá a un formulario para poder modificar cualquier dato relacionado de la publicación que ha creado en cuestión.

Cuadro 3.7: Modificación de contenidos

- **Caso de uso 8: Eliminar contenido**

Caso de uso	Eliminar contenido
Actores	Docente logeado
Precondición	Docente con una sesión iniciada y con publicación creada.
Postcondición	Se eliminará los datos correspondiente al contenido de la publicación.
Resumen	El docente podrá desde un botón eliminar la publicación que ha creado en cuestión.

Cuadro 3.8: Eliminar contenido

3.3.4. RF4 Gestión de mensajes

- **Caso de uso 9: Enviar mensajes**

Caso de uso	Enviar mensaje
Actores	Usuario logeado
Precondición	Usuario con una sesión.
Postcondición	El usuario creará un mensaje que se envía a otro usuario.
Resumen	El usuario con la sesión iniciada podrá dirigir un mensaje a otro usuario y enviarle un texto para que se cree una conversación entre ambos.

Cuadro 3.9: Enviar mensaje

- Caso de uso 10: Modificar mensajes

Caso de uso	Modificar mensaje
Actores	Usuario logeado
Precondición	Usuario con una sesión iniciada y con mensaje creado.
Postcondición	Se modificará los datos correspondiente al mensaje enviado.
Resumen	El usuario accederá a un formulario para poder modificar el mensaje de la conversación que ha creado en cuestión.

Cuadro 3.10: Modificar mensaje

- Caso de uso 11: Eliminar mensajes

Caso de uso	Eliminar contenido
Actores	Usuario logeado
Precondición	Usuario con una sesión iniciada y con mensaje creado.
Postcondición	Se eliminará los datos correspondiente al mensaje enviado.
Resumen	El usuario podrá desde un botón eliminar el mensaje que ha creado en cuestión.

Cuadro 3.11: Eliminar contenido

3.3.5. RF5 Gestión de clases

- Caso de uso 12: Creación de notas

Caso de uso	Creación de notas
Actores	Docente logeado
Precondición	Docente con una sesión iniciada
Postcondición	El rol de docente creará una asignación de notas a los alumnos.
Resumen	Cada profesor en su aula correspondiente podrá asignar una calificación sobre algún examen o trabajo a cada alumno matriculado en la asignatura.

Cuadro 3.12: Creación de notas

▪ **Caso de uso 13: Modificación de notas**

Caso de uso	Modificación de notas
Actores	Docente logeado
Precondición	Docente con una sesión iniciada y con notas creadas
Postcondición	Se modificará los datos correspondiente a las notas asignadas.
Resumen	El usuario accederá a un formulario para poder modificar las notas asignadas a sus alumnos.

Cuadro 3.13: Modificación de notas

▪ **Caso de uso 14: RF5.3 Eliminar notas**

Caso de uso	Eliminar notas
Actores	Docente logeado
Precondición	Docente con una sesión iniciada y con notas creadas
Postcondición	Se eliminará los datos correspondiente a las notas asignadas.
Resumen	El usuario podrá desde un botón eliminar las notas asignadas a sus alumnos que ha creado en cuestión.

Cuadro 3.14: Eliminar notas

▪ **Caso de uso 15: Creación de contenidos/temarios**

Caso de uso	Creación de contenidos/temarios
Actores	Docente logeado
Precondición	Docente con una sesión iniciada
Postcondición	El rol de docente publicará unos contenidos o temarios de clase para los alumnos.
Resumen	Cada profesor en su aula correspondiente podrá crear unos contenidos o temarios sobre la asignatura.

Cuadro 3.15: Creación de contenidos/temarios

■ Caso de uso 16: Modificación de contenidos/temarios

Caso de uso	Modificación de contenidos/temarios
Actores	Docente logeado
Precondición	Docente con una sesión iniciada y con contenidos/temarios creados
Postcondición	Se modificará los datos correspondiente a los contenidos o temarios de clase.
Resumen	El usuario accederá a un formulario para poder modificar los contenidos o temarios creados en clase para sus alumnos.

Cuadro 3.16: Modificación de contenidos/temarios

■ Caso de uso 17: Eliminar contenidos/temarios

Caso de uso	Eliminar contenidos/temarios
Actores	Docente logeado
Precondición	Docente con una sesión iniciada y con contenidos/temarios creados
Postcondición	Se eliminará los datos correspondiente a los contenidos o temarios de clase.
Resumen	El usuario podrá desde un botón eliminar los contenidos o temarios de clase.

Cuadro 3.17: Eliminar contenidos/temarios

■ Caso de uso 18: Creación de autorizaciones

Caso de uso	Creación de autorizaciones
Actores	Docente logeado
Precondición	Docente con una sesión iniciada
Postcondición	El rol de docente publicará un formulario asignado al aula para aceptar o rechazar una autorización.
Resumen	Cada profesor en su aula correspondiente podrá crear autorizaciones asignadas al aula para que los padres mediante un formulario puedan aceptar o rechazar.

Cuadro 3.18: Creación de autorizaciones

- **Caso de uso 19: Modificación de autorizaciones**

Caso de uso	Modificación de autorizaciones
Actores	Docente logeado
Precondición	Docente con una sesión iniciada y con autorizaciones creadas
Postcondición	Se modificará los datos correspondiente a las autorizaciones creadas para la clase.
Resumen	El docente accederá a un formulario para poder modificar las autorizaciones creadas en clase para sus alumnos.

Cuadro 3.19: Modificación de autorizaciones

- **Caso de uso 20: Eliminar autorizaciones**

Caso de uso	Eliminar autorizaciones
Actores	Docente logeado
Precondición	Docente con una sesión iniciada y con contenidos/temarios creados
Postcondición	Se eliminará los datos correspondiente a las autorizaciones de clase.
Resumen	El profesor podrá desde un botón eliminar las autorizaciones abiertas de clase.

Cuadro 3.20: Eliminar autorizaciones

- **Caso de uso 21: Sistema de asistencia**

Caso de uso	Sistema de asistencia
Actores	Docente logeado
Precondición	Docente con una sesión iniciada.
Postcondición	Quedar reflejado en un calendario las faltas de asistencias de los alumnos.
Resumen	El docente será capaz de asignar en un calendario de manera individual las diferentes faltas de asistencia por parte de cada alumno de su clase.

Cuadro 3.21: Sistema de asistencia

3.3.6. RF6 Sistema de notificaciones

- **Caso de uso 22: Listado de notificaciones**

Caso de uso	Listado de notificaciones
Actores	Usuario logeado
Precondición	Usuario con una sesión iniciada.
Postcondición	Quedar reflejado en un feed las novedades.
Resumen	El usuario tendrá un listado de los cambios en el sistema en los que el mismo intervenga o tenga que ver con él, tales como mensajes, autorizaciones o nuevos temarios entre otros.

Cuadro 3.22: Listado de notificaciones

Capítulo 4

Diseño

En este capítulo vamos a ver como diseñar nuestro sistema y como se van a organizar y relacionar los diferentes datos que usará para su funcionamiento completo. Una vez que hemos analizado las diferentes funcionalidades del sistema y hemos hecho un estudio previo tenemos que diseñar cómo serán las tablas de la base de datos en cuestión para el sistema.

Como elemento principal de GoodLearn y que ya se ha mencionado con anterioridad están los usuarios a los que cada uno le pertenece un rol diferente con el que podrán tener acceso a unas características u otras. Toda la información de las diferentes asignaturas será visualizada por cualquiera de estos usuarios pero su modificación sólo será exclusiva de los administradores o de los docentes.

Los elementos que integran un curso en el sistema son tanto asignaturas como los alumnos del mismo curso. En esta relación se refleja los datos de los usuarios que pertenecen a cada curso al completo en el caso de alumnos y los padres asignados a ellos y para la parte de los diferentes profesores solo el curso en el que impartirán las clases.

Por cada asignatura se tienen diferentes elementos que se relacionan a la asignatura, estos elementos son los contenidos, autorizaciones, calificaciones y faltas de asistencia. Todos ellos se verán reflejados en los diferentes alumnos y los padres podrán visualizarlos a su vez. Elementos como calificaciones, faltas de asistencias y las autorizaciones serán exclusivos a cada alumno en específico al ser personales a diferencia de los contenidos que son para toda la clase.

Como punto final se almacenarán las notificaciones asignadas a los diferentes usuarios y además los mensajes enviados entre usuarios del sistema que tendrá un atributo valioso como es el de estado para controlar si han sido o no leídos.

4.1. Diagrama E/R

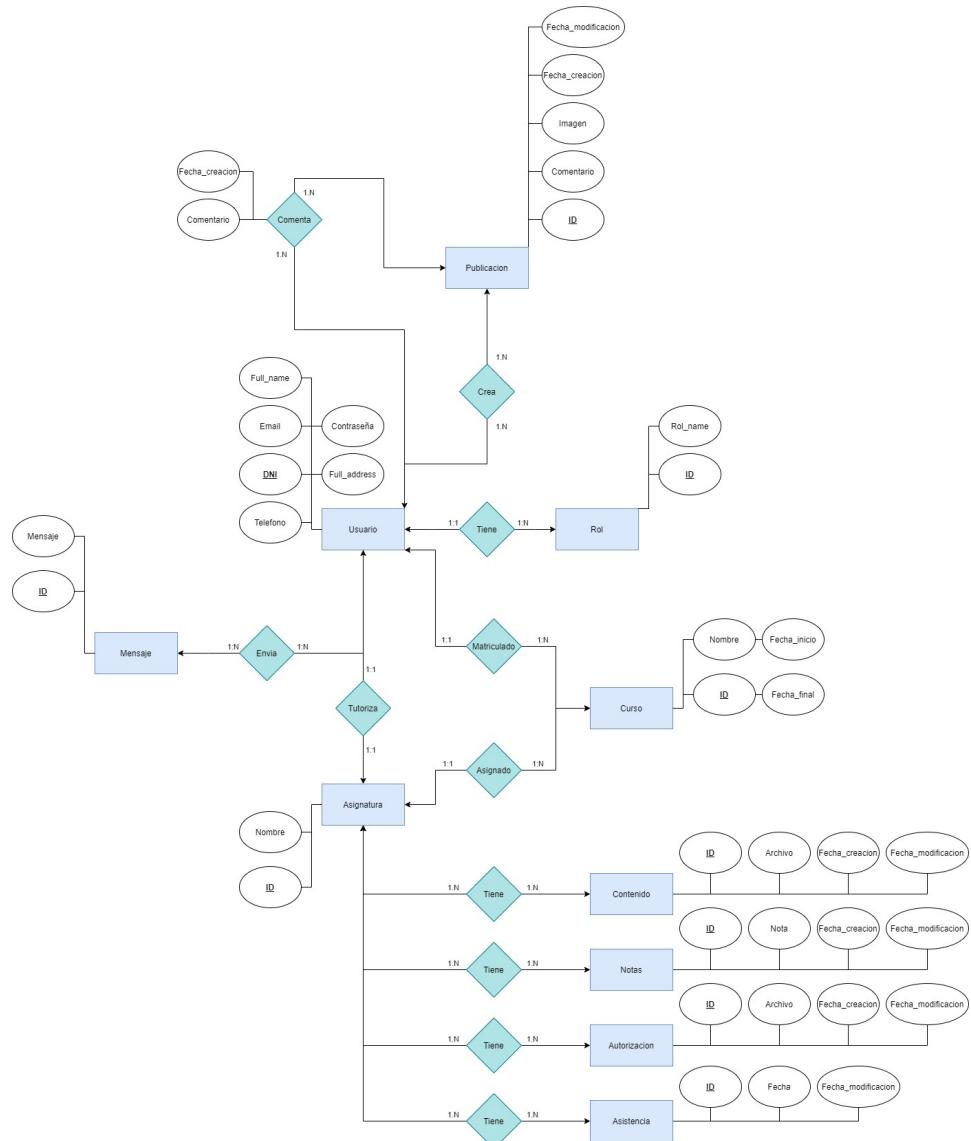


Figura 4.1: Diagrama E/R. Fuente: Elaboración propia

4.2. Modelo Relacional

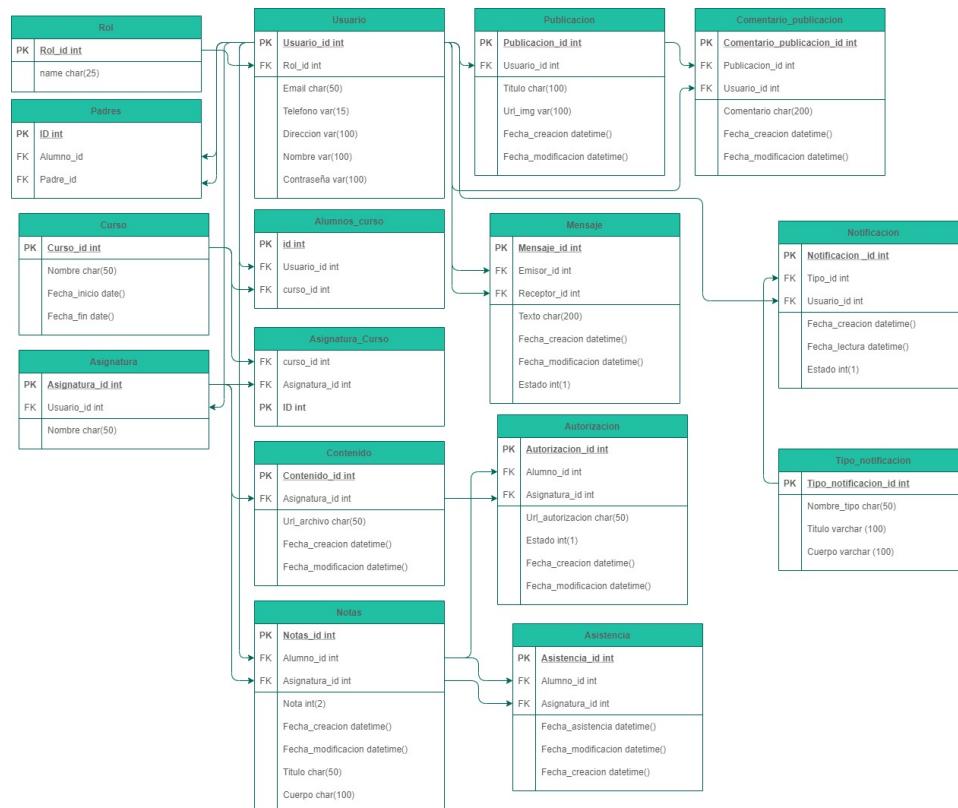


Figura 4.2: Modelo Relacional. Fuente: Elaboración propia

4.3. Patrón Arquitectónico

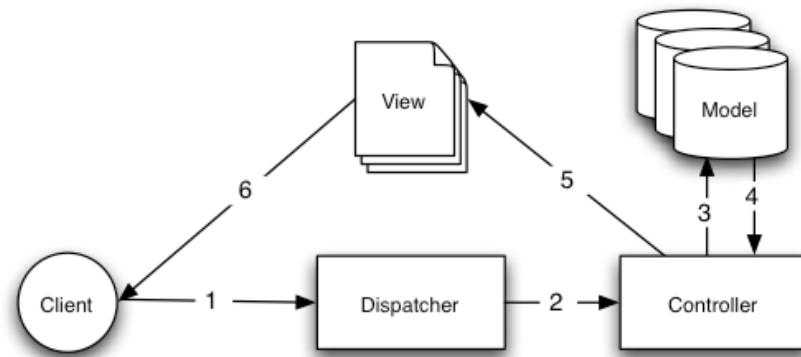


Figura 4.3: Esquema de la arquitectura MVC. Fuente: Universidad de Alicante[7]

GoodLearn está implementado en tecnologías tales como Laravel o Ionic que son frameworks con arquitectura MVC (Model-View-Controller). Este diseño arquitectónico está basado en tres componentes bien diferenciados con los que cada uno de ellos tendrá sus funciones específicas en el sistema en cuestión a desarrollar. Estos componentes son:

- **Modelo**, es el encargado del diseño de los datos que maneja el sistema. Se encarga de la creación de los objetos y sus atributos (Tablas de la base de datos), el almacenamiento de los datos en el sistema y las tareas de gestión de los mismos tales como obtener datos, eliminarlos o modificarlos.
- **Vista**, son todos los elementos que se les muestra al usuario final del sistema. En él se muestran los componentes visibles y los datos que consume del modelo para que sean visibles a nivel cliente. Elementos tales como HTML son los encargados en GoodLearn de mostrar la información al usuario.
- **Controlador**, es la capa intermedia encargada de unir el modelo con la vista. Funciones implementadas en Laravel que se acceden mediante requests a las urls de las api son elaboradas en este componente. En el controlador se incluyen todas las operaciones con los datos obtenidos por el modelo y el que se encarga de pasar esta información a la vista para que sea mostrada al usuario.

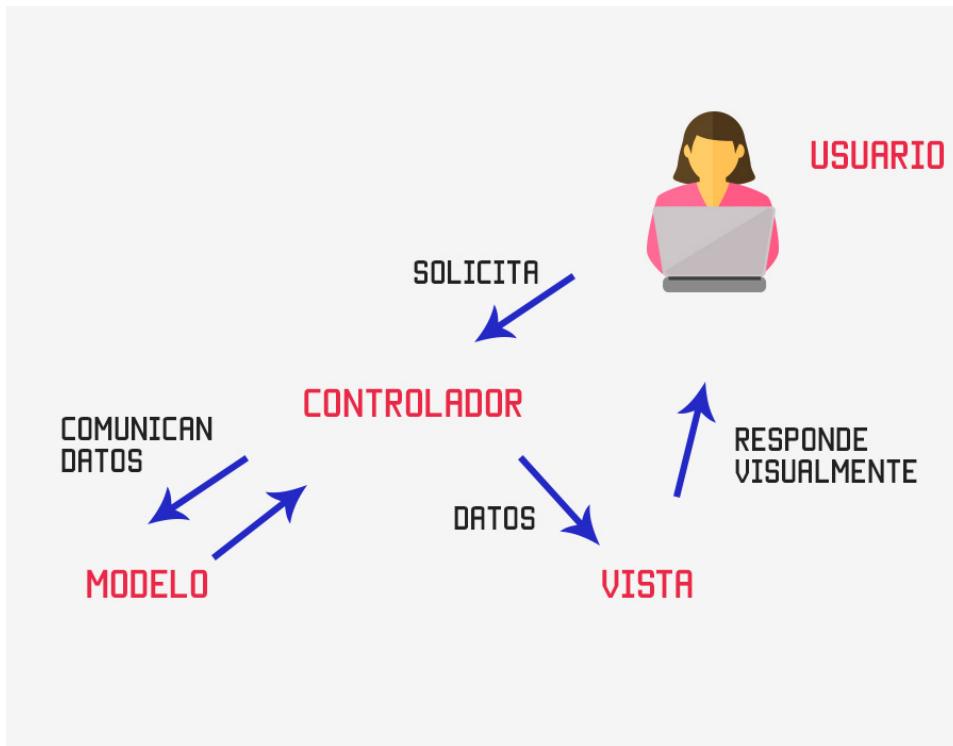


Figura 4.4: Conexiones de la arquitectura MVC. Fuente: Codigofacilito[8]

La comunicación que se establece en la arquitectura MVC es sencilla de entender y es como se puede apreciar en la imagen anterior. Esta arquitectura facilita el separar los diferentes componentes y tener una mayor visibilidad en grandes proyectos en que parte de nuestro código se encuentra en concreto lo que necesitamos modificar.

Como elemento principal tenemos un cliente que será el encargado de realizar las llamadas al controlador como primera instancia a la arquitectura. En el controlador se almacenan todas las funciones y procesos del sistema tales como clases o métodos. Según el tipo de llamada que solicite el usuario el controlador tiene un rol importante porque sirve como puente de comunicación entre la vista y el modelo. Imaginemos que el usuario quiere ver la información de un usuario, pues éste llamará al controlador y a la función `showUsuario()`. Esta función tendrá una conexión al modelo que es donde están los registros de tipo usuario y obtendrá de la conexión con el modelo estos datos. Una vez conseguidos el controlador enviará dichos datos a la vista. El proceso es sencillo y se realizará para todo lo que requiera de modelos y vista actuando como cerebro del sistema.

Por último la vista que es el componente visual donde se adquieren los datos obtenidos por el controlador de los modelos y donde se mostrará dicha

información. En el ejemplo anterior de mostrar información del usuario, la vista equivale al HTML donde se listara a todos los usuarios que ha devuelto el modelo al controlador y a su vez esté enviado a la vista.

4.4. Diseño gráfico

En cuanto al diseño gráfico de GoodLearn cabe resaltar el dilema social que existe en redes sociales entre los jóvenes de cuál es el color que representa a las asignaturas de Matemáticas o Lenguajes[9]. Pues bien en cuanto a la gama de colores se ha escogido uno de los colores de este debate que es el azul, frente al color rojo. El motivo de esta elección es que personalmente el color rojo me molesta más a la vista y lo veo como un color que significa peligro o algo erróneo.

Después de esta elección cromática se crea una paleta de colores que serán los que se implementaran en el sistema en casi todo momento creando un patrón en la elección de colores cuando se crean los diferentes elementos que se mostraran al cliente.

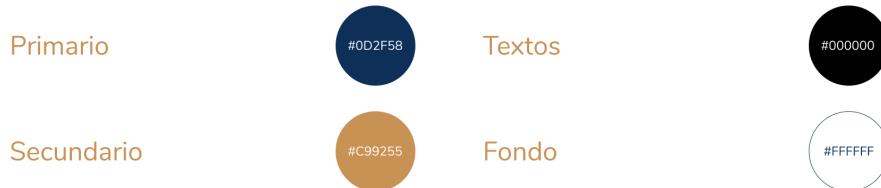


Figura 4.5: Paleta de colores. Fuente: Elaboracion propia

Como marca de empresa el logotipo tiene un elemento que es un libro de color dorado que sirve como marco para contener las iniciales de GoodLearn. El logotipo está diseñado de dos formas, una en la que solo se puede ver el elemento del libro que corresponderá al icono del sistema y al favicon.png de la web, y otro que se le añadirá el nombre del sistema para elemento tales como navbars de la app y de la web.



Figura 4.6: Favicon GoodLearn. Fuente: Elaboración propia



Figura 4.7: Logo con texto de GoodLearnFuente: Elaboracion propia

Capítulo 5

Implementación

En este capítulo se va a redactar sobre las tareas de implementación llevadas a cabo para desarrollar el sistema GoodLearn. Se va a mostrar las diferentes tecnologías y herramientas usadas a lo largo del proyecto y cómo se comunican cada una de ellas para llevar a cabo las diferentes tareas que se han elaborado a lo largo del proyecto.

5.1. Marco tecnológico

El marco tecnológico servirá como punto clave del desarrollo del sistema de información propuesto ya que serán las diferentes herramientas que se han usado para crearlo e indagar un poco más en ellas para tener claro qué papel tienen en GoodLearn.

5.1.1. GitHub



Figura 5.1: Logo de Github. Fuente: Github [10]

GitHub es una herramienta de control de versiones Git donde se pueden almacenar proyectos en repositorios asignados a tu perfil. Esta herramienta ha sido utilizada para ir asignando al repositorios diferentes versiones del

proyecto para poder controlar las versiones estables que se han ido implementando del sistema.

Además GitHub trae incorporado en el repositorio un canvas donde mediante tableros y tarjetas se han ido añadiendo diferentes tareas y donde se puede controlar en qué punto del proyecto nos encontramos además de abrir debates sobre los diferentes problemas que han podido surgir. El control de versiones con git es un pilar fundamental en el desarrollo de cualquier proyecto y sobre todo cuando se trabaja en equipo. Esta herramienta ha servido como guía en todo momento de los diferentes estados de GoodLearn y como pauta de control del proyecto.

Para visualizar el repositorio de GitHub [Pincha aquí.](#)

5.1.2. Postman



Figura 5.2: Logo de Postman. Fuente: Postman [11]

Esta plataforma brinda la oportunidad de colaborar en proyectos de desarrollo de API's. Para este proyecto ha servido como herramienta de control de las diferentes request que se han necesitado para el control de los datos del sistema.

Desde esta herramienta se han podido establecer las conexiones pertinentes para obtener, crear o modificar los datos de la base de datos. Además proporciona todas las variables para los headers de una request e incluso el método que se usa para la conexión y los datos que se pasan mediante JSON. Al igual que GitHub es colaborativa por lo que el control del equipo sobre las direcciones de la API es común para el entorno de desarrollo. Si se quiere trabajar en equipo es una plataforma adecuada para el desarrollo de todo tipo de API's.

Para ver la documentación de la api de GoodLearn [Pincha aquí.](#)

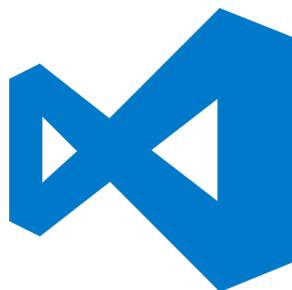


Figura 5.3: Logo de VS Code. Fuente: VS Code [12]

5.1.3. Visual Studio Code

VS Code es un IDE(Integrated Development Environment) creado por Microsoft y es una herramienta que proporciona facilidades a la hora de implementar cualquier tipo de código. Se caracteriza por una amplia gama de plugins que se pueden instalar en el mismo para crear algunos atajos que ahoran tiempo en la escritura de código.

Este entorno de desarrollo ha servido para la ayuda de algunas funciones escritas y la previsualización de las mismas antes de escribir así es más difícil equivocarse en alguna letra de la llamada a alguna clase o método. Se han incorporado plugins de Laravel y Vue.js para tener un entorno más completo a la hora de la implementación.

5.1.4. Photoshop



Figura 5.4: Logo de Photoshop. Fuente: Photoshop [13]

Esta famosa herramienta proporciona la oportunidad de crear y modificar imágenes a nivel de capas. Todo lo relacionado con el diseño gráfico del sistema tales como logotipo y paleta de colores está sacado de esta plataforma de adobe. Photoshop proporciona una amplia paleta de colores donde se extraen los códigos rgb de los diferentes colores usados en los elementos del sistema de GoodLearn.

5.1.5. DigitalOcean



Figura 5.5: Logo de DigitalOcean. Fuente: DigitalOcean [14]

Esta plataforma provee servicios de hosting personalizados con servidores ya establecidos o dedicados. DigitalOcean posee las facilidades necesarias para montar tu propia aplicación web además de que es capaz de recoger tus proyectos directamente de servicios como GitHub GitLab para montar y conectar tu aplicación a cualquier repositorio.

GoodLearn está montado en este servicio y conectado con el repositorio a la rama principal, por lo que cualquier cambio que se haga en esta rama será montado en el hosting automáticamente facilitando la labor de cambio de versiones en el sistema.

Para ir a la web de GoodLearn [Pincha aquí](#).

5.1.6. Laravel Framework



Figura 5.6: Logo de Laravel framework. Fuente: Laravel Framework [15]

Laravel es un framework basado en la arquitectura MVC y en lenguaje PHP. Esta tecnología brinda la oportunidad de diseñar aplicaciones complejas de una manera fácil menos costosa por la gran cantidad de herramientas que se incluyen en el framework.

Con este framework se ha construido toda la parte del servidor con los controladores a los que llama la API implementada para GoodLearn. Además la vista que proporciona la web está desarrollada en con las rutas y las vistas que se han incluido en el proyecto.

La documentación que proporciona Laravel es bastante completa y con ella se puede ver que se puede hacer en cada componente de la arquitectura.

5.1.7. Ionic Framework



Figura 5.7: Logo de Ionic framework. Fuente: Ionic Framework [16]

Con este framework se puede construir aplicaciones móviles multi plataforma basadas en javascript. La documentación que nos muestran en su página web se puede ver que ionic está basado en typescript para las llamadas a los diferentes tipos de frameworks de javascript que soporta como Angular, React y Vue.

GoodLearn utiliza Vue.js como pilar fundamental en la obtención, modificación y eliminación de los datos que nos provee la API de la mano de Laravel.

5.2. Implementación

Esta sección se dividirá en los tres elementos fundamentales en la implementación de GoodLearn y en la explicación de cómo ha sido a lo largo del proyecto su desarrollo valorando la dificultad y las adversidades que se han ido planteando en su programación.

Cabe destacar la importancia que posee la idea de crear una API propia con la que poder controlar todo el sistema y poder utilizar tecnologías totalmente diferentes para alcanzar el mismo objetivo principal que es el de una agenda escolar online.

Hay que tener en cuenta que es necesario la instalación de todas las tecnologías que se ven reflejadas en el marco tecnológico para poder implementar este proyecto. En [10] viene una guia de como es la instalación mediante los comandos proporcionados por cada una de las tecnologías usadas.

5.2.1. API

La implementación de la API es el elemento funcional más importante para dar paso al resto del sistema, sin ella no sería posible la obtención de los datos de la base de datos e incluso cualquier acción sobre ella que se quiera hacer. Esta interfaz de programación será desarrollada con las herramientas que nos proporciona Laravel.

En primer lugar y teniendo claro el modelo relacional del sistema se comienza a crear los diferentes modelos que se han analizado con Laravel. Personalmente, yo he creado a la vez diferentes modelos con migraciones y controladores con el siguiente comando.

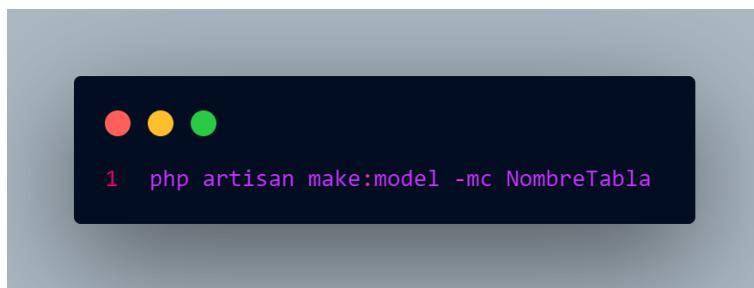


Figura 5.8: Comando para crear modelo, controlador y migración. Fuente: Elaboracion Propia

Las migraciones son ficheros que se encarga mediante llamadas a laravel de exportar y crear en la base de datos las diferentes tablas que serán referenciadas al modelo. El modelo facilita la creación de objetos y Laravel los reconocerá como tales por lo que cuando queramos hacer alguna consulta a la base de datos no será necesario escribirla con sintaxis SQL o parecida.

La estructura de los ficheros de migraciones en Laravel es una función que viene predeterminada al utilizar el comando anterior y en el que mediante la documentación se puede obtener la información para crear la estructura de las diferentes tablas que va a tener nuestra base de datos. Aquí un ejemplo de la tabla de Usuarios en la que se puede observar en su mayoría como poder relacionarla con otras tablas o como se crean los diferentes tipos de atributos además de añadir el dropdown en caso de que queramos eliminar cualquier referencia de las tablas.

Como ayuda para la API se ha implementado una librería que se llama **prettyPrint.php** y servirá como ayuda para devolver los datos en JSON de manera más detallada y elegante. Esta librería sustituye los id 's de las claves foráneas por los objetos enteros lo que proporcionará mayor información cuando se llama a un controlador vía URL. Esta librería ha servido de ayuda porque cuando se quiera acceder a la información completa o algún atributo

```

1  public function up()
2  {
3      Schema::create('usuario', function (Blueprint $table) {
4          $table->id();
5          $table->unsignedBigInteger('rol_id')->default('3');
6          // $table->foreign('rol_id')->references('id')->on('rol')->onDelete('cascade');
7          $table->foreign('rol_id')->references('id')->on('rol')->onDelete('cascade')->onUpdate('cascade');
8          $table->string('name', 50);
9          $table->string('email', 50)->unique();
10         $table->string('telefono', 15)->nullable();
11         $table->string('direccion', 100)->nullable();
12         $table->string('password');
13     });
14 }
15 /**
16  * Reverse the migrations.
17  *
18  * @return void
19  */
20 public function down()
21 {
22     Schema::dropIfExists('usuario');
23 }

```

Figura 5.9: Estructura migración. Fuente: Elaboracion Propia

en específico de una relación esta librería hace que esto sea posible y se pueda acceder de una misma consulta a cualquier tabla sin necesidad de hacer varios request a la API.

Una vez implementada la estructura de todas las tablas de la base de datos con las migraciones el siguiente elemento es desarrollar los modelos que posee el sistema para la utilización mediante objetos las tablas y poder aplicar las consultas con Eloquent que es un mapeador a objetos de las tablas y es el que usa Laravel para que sea posible el uso de los diferentes modelos. La estructura que utilizan los modelos es una clase que se extiende a los modelos que provee Eloquent. Aquí un ejemplo de cómo es la estructura del modelo para la tabla de usuarios que usa GoodLearn.

Laravel va a conectar las diferentes rutas que se le asignen a una función de los controladores creando la conexión de una url como si fuera la llamada a la función en cuestión. Esto es una de las cualidades por la que nuestra API es el centro neurálgico del sistema, podemos acceder a las diferentes operaciones que se implementaran en funciones de los controladores exclusivamente por una URL que se asignará en el fichero **api.php** localizado en el directorio **routes** de nuestro directorio del proyecto con Laravel. Aquí un ejemplo de cómo sería la estructura de rutas de la API de GoodLearn para la tabla Usuarios.

Como se puede observar en la imagen anterior las diferentes rutas son llamadas con un tipo de método diferente para que sea posible su funcionamiento. Las llamadas pueden ser de tipo GET, si no se le pasa ninguna

```

1  class Usuario extends Model
2  {
3      use HasFactory;
4      protected $fillable = [
5          'name',
6          'email',
7          'telefono',
8          'direccion',
9          'password',
10     ];
11
12     public function rol(){
13         return $this->belongsTo('App\Models\Rol', 'rol_id', 'id');
14     }
15     protected $table = 'usuario';
16     public $timestamps = false;
17 }

```

Figura 5.10: Estructura de modelo. Fuente: Elaboracion Propia

información y es para obtenerla únicamente y POST, PUT o DELETE, que en estos casos tienen en común que si se le pasa información para poder crear, modificar o borrar, respectivamente a cada método.

El siguiente paso es saber que función llamar del controlador para cada url y cual es la finalidad que tienen cada una de ellas. La estructura de los controladores es común para todas ellas ya que se realizan las mismas operaciones para cada una de las diferentes tablas. Como funciones en los controladores tenemos:

- **index()**, esta función será la encargada de llamar al modelo de Eloquent y obtener todos los registro de la tabla a la que se haga referencia en el controlador además de pasar esta información que se devuelve por la función prettyPrint para que sea devuelta de la manera más completa. La url que apunte a esta función debe ser llamada con el método GET.
- **byIndex(Object)**, Eloquent obtendrá de la URL a la que se llame esta función un id con el que se devolverá un objeto en cuestión identificado por un valor numérico y además se le pasará la consulta a prettyPrint para devolverla correctamente y la llamada a la URL deberá ser con GET.



```
● ● ●
1 //Lista todo los usuarios.
2 Route::get('usuarios', 'UsuarioController@index');
3 //Devuelve por ID el usuario.
4 Route::get('usuarios/{usuario}', 'UsuarioController@byIndex');
5 //Hace una busqueda con una variable GET llamada text.
6 Route::get('usuarios/show', 'UsuarioController@show');
7 //Crear un usuario.
8 Route::post('usuario', 'UsuarioController@store');
9 //Modifica un usuario
10 Route::put('usuario/{usuario}', 'UsuarioController@update');
11 //Elimina un usuario
12 Route::delete('usuario/{usuario}', 'UsuarioController@destroy');
13 //Check usuario
14 Route::post('usuario/credentials', 'UsuarioController@checkCredentials');
```

Figura 5.11: Modelo de url para la API. Fuente: Elaboracion Propia

- **show()**, mediante una llamada de tipo GET a esta url y pasándole una variable llamada text con algún dato de la tabla en cuestión que se quiera obtener los datos, Eloquent hará una consulta a la tabla buscando en cada uno de sus atributos la información pasada por url y obtendrá todos los objetos que tengan en común dicho dato. Es decir hará un SELECT WHERE.
- **store()**, está llamada deberá de ser de tipo POST y en ella se le pasaran los datos necesarios para crear un nuevo registro en la tabla en cuestión a la que haga referencia la url llamada. Se han creado filtros para los datos introducidos para que no se corrompa la tabla de la base de datos con elementos creados con Laravel llamados request. Estos componentes request obtienen los datos que se han introducidos y los pasa por un filtro que se han implementado tales como que lo introducido sea de tipo text o que sea requerido.
- **update(Object)**, el método de esta llamada es de tipo PUT y como contenido tiene los datos a editar del registro que se le pasa por parámetro en la url. Al igual que la función store() se ha creado un componente request que será el encargado de filtrar los datos introducidos y que todos ellos estén correctamente como necesita la tabla en cuestión.
- **delete(Object)**, a diferencia de los dos anteriores esta llamada no necesita ningún dato y es con el método DELETE. Como su traducción al inglés, significa eliminar y se elimina el objeto que se le pasa por

parámetro a la función que es ingresado en la url al llamar a esta función.

Este proceso de asignar las diferentes url's a las funciones de los diferentes controladores de las tablas se debe hacer para cada uno de ellos y además implementar las diferentes funciones descritas arriba. Se debe desarrollar como puntos más importantes los almacenamientos en las tablas, las modificaciones de registros y las eliminaciones de estos.

Como dificultad de esta parte cabe destacar la implementación de una de las tablas para saber que tipo de comprobaciones se deben hacer o como Eloquent se comunica para crear los diferentes registro, una vez completado una de las tabla el trabajo es largo pero muy parecido en todas y cada una de las tablas.

También se ha utilizado Postman como herramienta de comprobación y realización de llamadas a cada una de las urls creadas en Laravel, viendo los mensajes obtenidos cuando se realizan las funciones y observando que tipo de respuesta JSON se obtiene.

Otro de los factores de complejidad es la comprobación de los diferentes datos pasados a la API con los componentes request al poseer cada tabla información de tipos y contenido diferente.

Si se quiere observar cualquiera de las llamadas posibles que nos provee GoodLearn les aconsejo entrar en la documentación elaborada en [11] para la API y donde se puede ver y observar los diferentes nombre que tienen que tener los datos para su correcto funcionamiento.

5.2.2. Dashboard web para administradores y profesores

Teniendo la API totalmente implementada y con su correcto funcionamiento toca implementar la vista web para usar las funciones implementadas en los diferentes controladores y llamarlos desde la vista. Como se ha descrito anteriormente Laravel está compuesto de tres componentes de los cuales ya se han usado dos de ellos. La API ha sido la encargada de utilizar los modelos para crear los diferentes registros y los controladores para crear la comunicación entre modelos y rutas, ahora toca dar paso a la vista para que el cliente pueda establecer la comunicación con los controladores y que además sea posible mostrar la información almacenada.

En la parte de la vista se implementara código HTML, CSS y JS para los componentes visuales que el cliente consumirá y además donde se reflejaran todas las funcionalidades del sistema haciendo uso de los controladores.

Como primer punto importante en el sistema es tener una gestión de sesiones donde el usuario pueda logearse al sistema y poder utilizar las fun-

cionalidades. Cabe destacar que el dashboard implementado en Laravel es sólo y exclusivamente para los roles que puedan crear y subir contenidos al sistema por lo tanto solo será accesible para el administrador y los profesores. A continuación se verán unas imágenes de las primeras vistas que se obtienen y del login al sistema esto es común para todos.

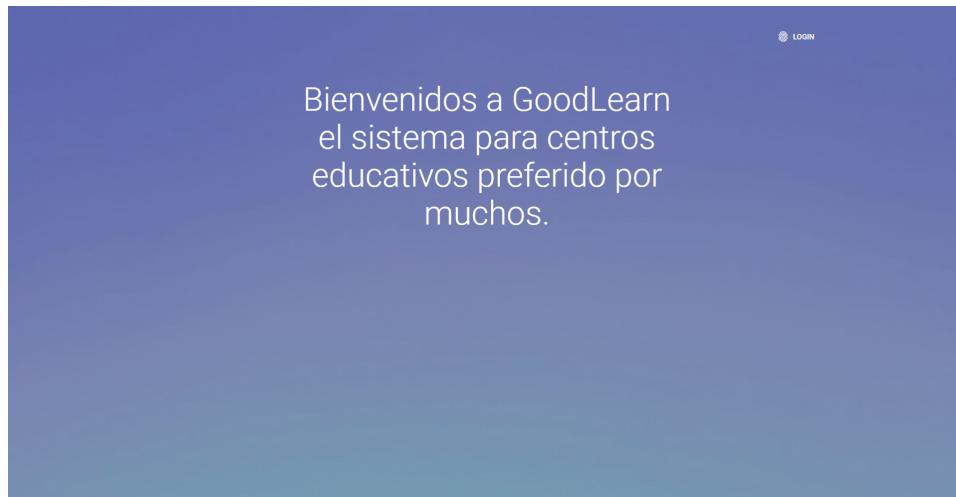


Figura 5.12: Home. Fuente: Elaboracion Propia

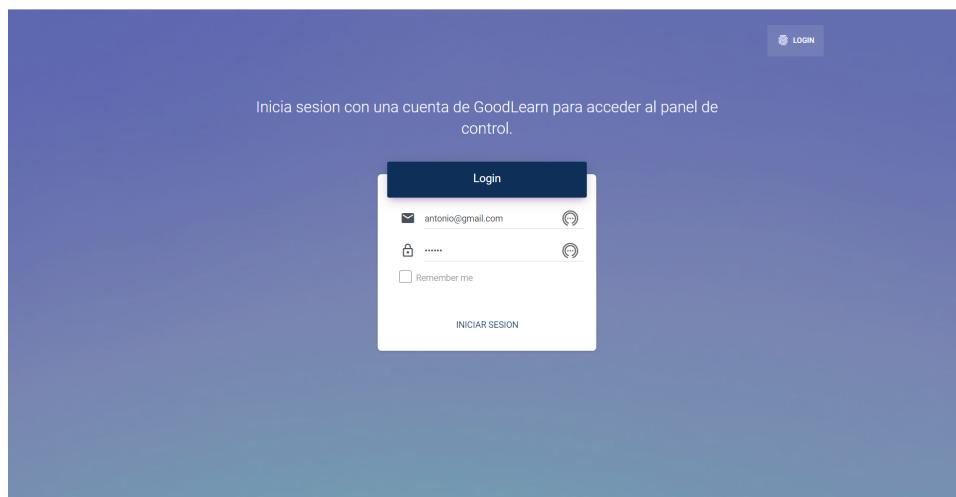


Figura 5.13: Login. Fuente: Elaboracion Propia

El sistema de logueo al sistema es único para estas vistas en web por lo que en el controlador de usuarios se creará una función `login()` que recogerá los datos pasados desde el formulario de login y se comprueba si existe en la base de datos con Eloquent y que además sea un usuario con rol administrador o profesor. Después de que estas comprobaciones sean correctas

se crean las diferentes variables de sesión para su uso en el sistema y que sirvan como filtro para saber que debe mostrar y que no según el evento que necesite cada usuario. El sistema provee mensajes de alertas donde se envían mensajes sobre el estado de los datos ingresados en cada formulario o las interacciones sobre errores obtenidos y sucesos aceptados.

Una vez iniciada la sesión y creada las variables que referencian al usuario que se ha logueado tenemos dos tipos de vistas diferentes según que tipo de usuario es el que ha iniciado sesión. La vista del administrador es la más compleja y en ella se puede observar el CRUD de todas las tablas del sistema haciendo posible el acceso al sistema al completo. Sin embargo el rol de profesor tendrá acceso solo y exclusivo a la creación de publicaciones y a la gestión de la asignatura a la que le ha sido proporcionado el acceso por parte del administrador.

Cabe resaltar que a ningún usuario se le brinda la posibilidad de registrarse al sistema ya que el objetivo de GoodLearn es proporcionar un sistema personalizado a cada centro educativo dándole la posibilidad al administrador de crear los diferentes usuarios que componen todo la estructura del centro educativo en cuestión.

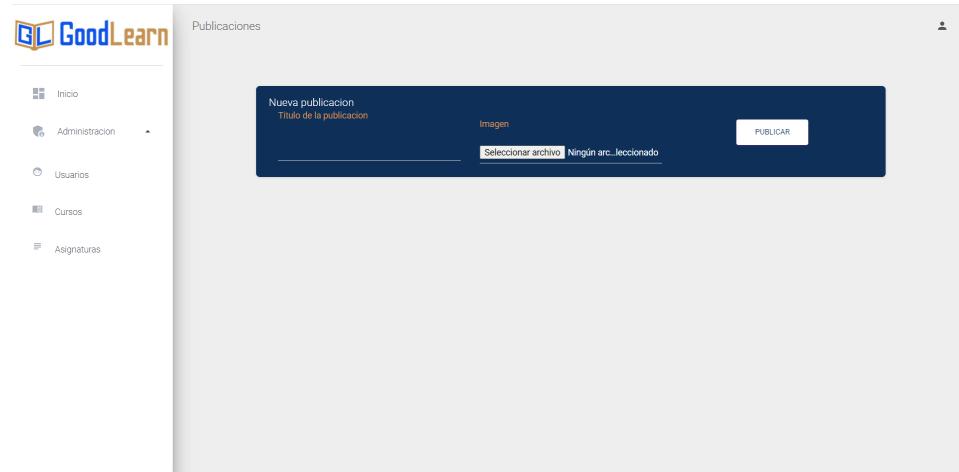


Figura 5.14: Home para administradores. Fuente: Elaboracion Propia

Comencemos con el profesor y enseñar las diferentes vistas que se le provee a este rol en el sistema. Todas las vistas que se van a describir a continuación también son aplicables al rol de administrador del sistema.

En primer lugar tal y como se inicia sesión en el sistema se puede observar un formulario donde se puede crear una publicación. Este formulario está compuesto por la subida de un fichero de imagen y una título para la publicación como se puede observar en las dos fotos anteriores. Una vez que se añade una publicación este sería el resultado.

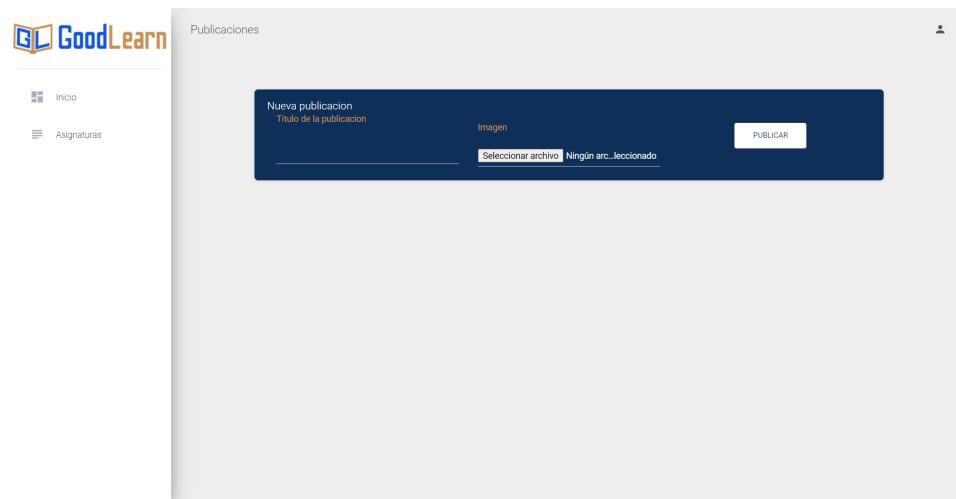


Figura 5.15: Home para profesores. Fuente: Elaboracion Propia

Se tiene un elemento de menú dropeable para acceder a la eliminación de la publicación o la modificación de la misma mediante un formulario. Además se tiene un botón a los comentarios donde hay un formulario similar al de la publicación donde se crean comentarios desde el dashboard.

La siguiente vista sería la de la asignatura a la que pertenece el profesor y será una lista mostrada en una tabla con un botón donde al pinchar se abre el contenido de la asignatura.

Dentro del contenido de la asignatura tenemos un menú con eventos en los que cambiará la información según los datos que se quieran mostrar, crear o modificar de la asignatura tales como faltas de asistencias, contenidos, autorizaciones o notas estas serían las diferentes vistas de lo que contiene una asignatura.

En los apartados de autorizaciones y contenidos el formulario de creación es el mismo porque es una subida de ficheros de formato pdf que luego será mostrado si se pincha en el nombre del fichero para que pueda ser visualizado en un frame.

Las autorizaciones poseen un menú parecido al de los contenidos de las asignaturas pero se diferencia en que también se muestra la información sobre alumnos que han aceptado o rechazado la autorización creada. Además el frame de los temarios es igual que el de autorizaciones pero eliminando la lista de alumnos ya que no son asignados a ningún usuario en cuestión si no a la asignatura al completo.

Ya mostrado las diferentes vistas que posee el rol de profesor vamos con algunos de los elementos que son solo y exclusivos del administrador tales como la gestión de usuarios, cursos y creación de las asignaturas.

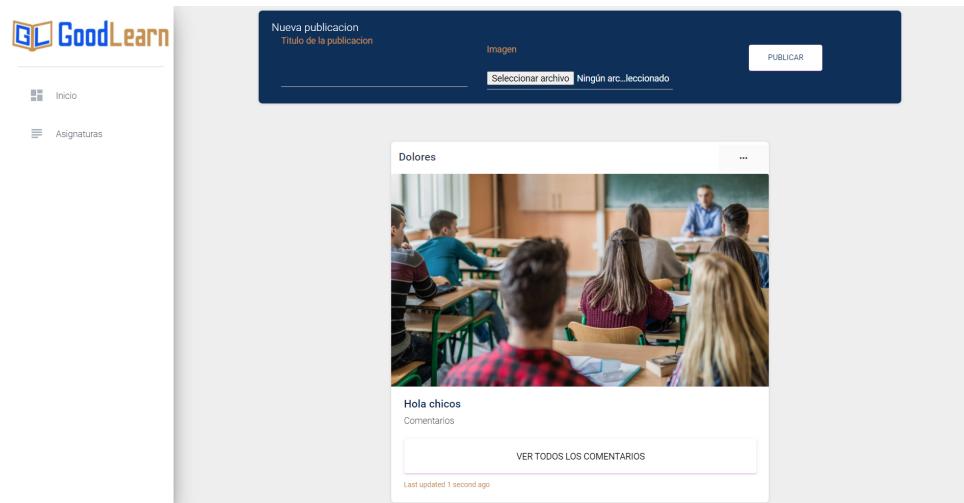


Figura 5.16: Lista de publicaciones. Fuente: Elaboracion Propia

Dentro de la lista de contenidos tenemos un elemento en las opciones que es el de visualizar las diferentes asignaciones que posee el curso tales como teníamos en las asignaturas donde se podían visualizar las calificaciones, autorizaciones etc. El sistema de vista de los cursos es el mismo pero se muestran los alumnos y las asignaturas que posee cada curso. Dentro de esta vista se podrán asignar las relaciones desde un formulario prototípico para la creación y modificación de cualquier tabla del sistema.

Otro elemento que es modificado para el administrador es la cabecera de la lista de asignatura en la cual tiene la opción de crear un nueva asignatura y asignarla a un profesor. Esta funcionalidad está capada para el rol de profesor por lo que si nos fijamos en la lista de asignaturas para los profesores no se puede crear siendo ese rol.

La vista de cualquier formulario se representa con este elemento el cual varía según los datos que se necesitan para crear los registros propios de cada tabla que tiene GoodLearn

La vista de usuarios se compone de un listado con todos los usuarios registrados en el sistema y con los datos que se le han asignado a cada atributo, además de varias opciones que aparecen en todos los diferentes objetos del sistema que son modificar y eliminar. El rol de padre posee un botón que redirecciona a otra vista donde se le asigna al padre los hijos que tienen en el colegio dando la posibilidad de que cada padre pueda ver los contenidos de las asignaturas en las que cursan sus hijos y todos los datos relacionados con el usuario hijo asignado.

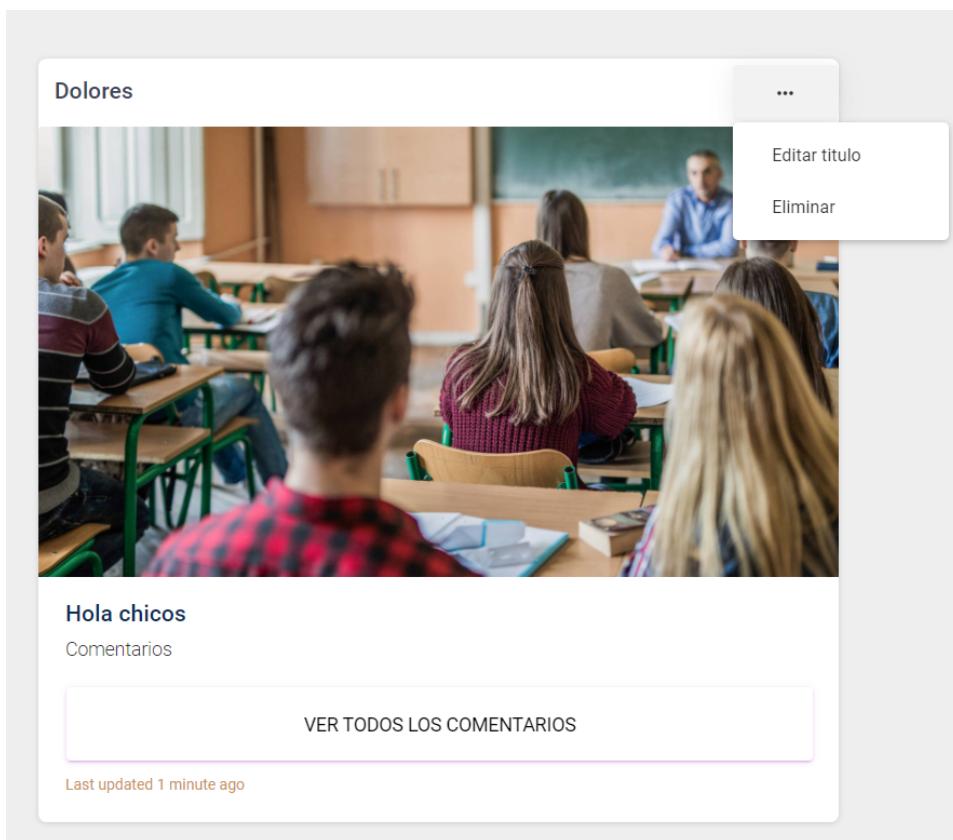


Figura 5.17: Elemento de publicación. Fuente: Elaboracion Propia

5.2.3. APP Ionic

La siguiente implementación es la APP multiplataforma con el framework Ionic. Antes de crear el proyecto con el framework te pregunta en qué framework de JavaScript va estar basado el MVC de ionic. En el caso de la app de GoodLearn está basado en Vue.js que su implementación está basada en el desarrollo de componentes haciendo posible la escalada de los mismo desde ser componentes simples hasta convertirse en complejos.

Cada fichero de la app se compone de dos partes bien diferenciadas, una que es el template la parte visual que consumirá el cliente y un componente script basado en TypeScript donde se desarrollara las llamadas a la api y el tratamiento de los datos como parte del backend. El template de las app está basado en componentes ionic que facilitan la posición en dispositivos móviles y que además se pueden intercalar con elementos HTML. Esta característica de ionic hace que desarrolladores web puedan tener la oportunidad de desarrollar aplicaciones sin tener que saber el lenguaje nativo para android o ios. El desarrollo en ionic está por una parte limitado porque si queremos

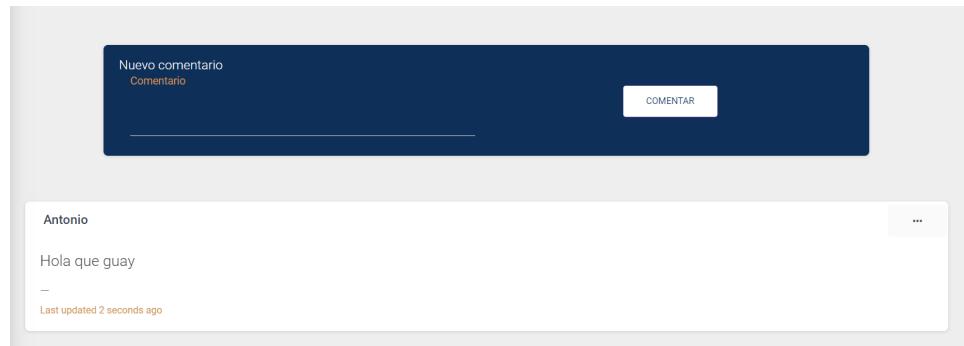


Figura 5.18: Lista de comentarios. Fuente: Elaboracion Propia

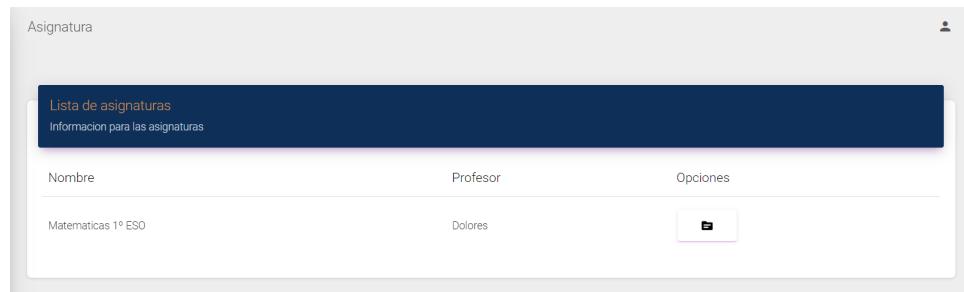


Figura 5.19: Lista de asignaturas. Fuente: Elaboracion Propia

acceder a la cámara del dispositivo hace falta llamar a funcionalidades nativas que son prestadas por cordova pero que esté a su vez no es usado con el Vue.js por lo que segun en que framework se base el backend podremos tener disponibles unas características u otras.

En la parte del backend tenemos una estructura de componente en Vue con los siguientes elementos que caben destacar:

- **components**, en él se incluirán todos los elementos que usamos de ionic. En la documentación del framework viene cuales son los componentes que se pueden incluir y como, por ejemplo si queremos añadir un componente ionic-button se tendría que añadir dentro de este objeto IonButton.
- **mounted()**, todos los procesos que se pongan en esta función serán ejecutados después de la función created() y de que la app sea renderizada con todos sus elementos del template.
- **created()**, los procesos que se incluyen en esta función serán los primeros en ser ejecutados y llevar a cabo antes de cualquier otro.
- **methods:**, en él se añadirán las diferentes funciones locales del com-

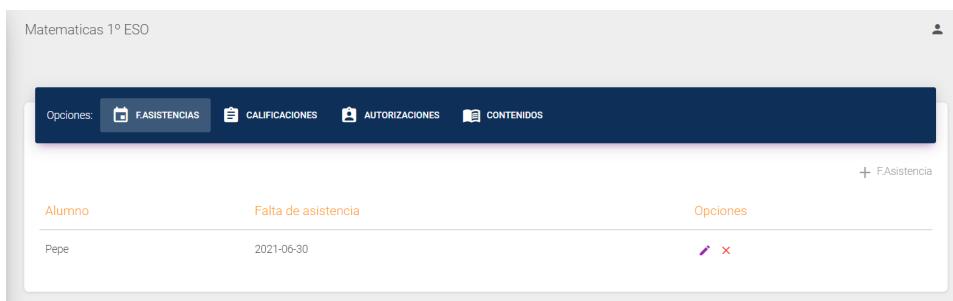


Figura 5.20: Lista de faltas de asistencia. Fuente: Elaboracion Propria

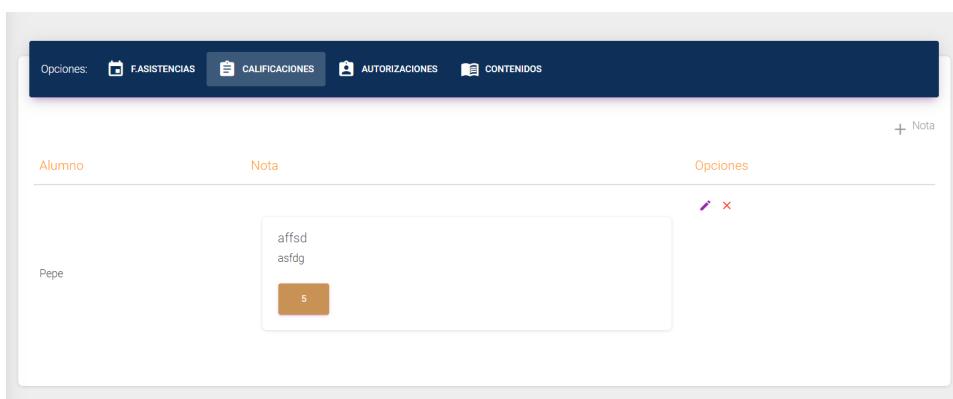


Figura 5.21: Lista de calificaciones. Fuente: Elaboracion Propria

ponente como cualquier método de alguna clase. Serán usados por las otras funciones de created o mounted llamandolos con this.method.

- **data**, se incluirán todas las variables que contendrá el componente de Vue y las que se podrán mostrar en el template de la app.

Se utiliza axios como librería de JavaScript para llevar a cabo las conexiones a la API de GoodLearn y obtener los diferentes datos que serán usados en la app. Esta librería nos brinda la oportunidad de crear request bastante fácil para obtener y enviar datos a la API además de poder modificar las diferentes cabeceras según lo que necesitemos. Aquí un ejemplo usado para obtener la información de las publicaciones llamando a la api de GoodLearn.

Para el Login de la app se ha creado una url de la api en la que se le manda el formulario ingresado por el usuario y se comprueba si existe su email y además se comprueba la contraseña ya que está al crearse se cifra con una función Hash que provee Laravel la que consigue un cifrado único para dicha contraseña y solo y exclusivamente comprobada por la autenticación

Autorizacion	Fecha de creacion	Opciones
1625522400MODELO_AUTORIZACIONPIC.PDF	2021-07-06	

Figura 5.22: List de autorizaciones. Fuente: Elaboracion Propia

Contenido	Fecha de creacion	Opciones
1625522400TEMA1.PDF	2021-07-06	

Figura 5.23: Lista para contenidos. Fuente: Elaboracion Propia

de Laravel. Está llamada a la api de volver un valor true o false en caso de que coincida o que no lo que dará paso a Vue a crear las variables de session pertinentes para cada usuario que desee loguearse.

Una vez logueado todos los usuarios tendrán diferentes pantallas en común donde lo único que cambiaría son los datos de los diferentes contactos a los que tienen acceso y a los cursos y asignaturas. Estas páginas son la pestaña de mensajes donde se muestra una agenda con los contactos a los que se puede enviar mensajes, la pestaña de la información del usuario logueado y por último los diferentes cursos a los que el usuario tiene acceso.

La pestaña del home es diferente para los roles de administrador-profesor y alumno-padre ya que los primeros tienen la posibilidad de crear publicaciones subiendo una foto y un título cosa que los padres y alumnos no pueden pero si comentar en ellas.

Otra de las diferencias notables en el sistema es la posibilidad que tienen los padres y alumnos en aceptar desde la app las diferentes autorizaciones creadas en las asignaturas y dar permiso a su hijo a realizar alguna actividad extraescolar.

Las notificaciones en la app es otra característica importante como evento a las diferentes modificaciones del sistema y reflejadas en los usuarios que se ven involucrados en estos cambios en el sistema. Las notificaciones aparecerán si existen en tal caso y solo aparecen en la pestaña home de cada

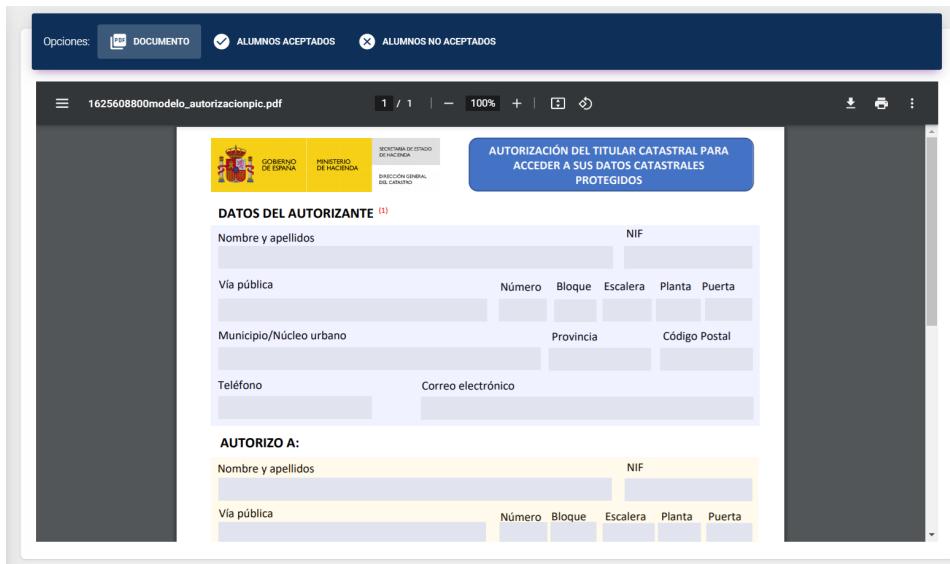


Figura 5.24: Frame de autorizaciones. Fuente: Elaboracion Propia

Lista de cursos				Añadir Curso
Nombre	Comienzo	Finalización	Opciones	
2º Primaria	2021-06-29	2021-07-29		
1º ESO	2021-06-28	2021-07-05		

Figura 5.25: Lista de cursos. Fuente: Elaboracion Propia

usuario apareciendo un botón flotante que si se pincha en él aparecerá los diferentes eventos para el usuario tales como aparece en la siguiente foto.

La dificultad del desarrollo de la app se ha encontrado en aprender el framework de Vue.js el cual no se tenía ningún conocimiento sobre él ni sobre el lenguaje en el que se basa qué es JavaScript. Una vez realizado un aprendizaje básico sobre el manejo de urls con axios y como se obtienen los datos con esta funcionalidad se tenía que almacenar los datos resultantes en la estructura del componente Vue para que fuese mostrado en el template. La implementación ha sido más fluida cuando se tenía el conocimiento perfecto de la obtención de datos y almacenamiento de ellos, luego el trabajo ha sido muy parecido para cada una de las vistas.

Nombre	Profesor	Opciones
Matematicas 2º Primaria	Lorena	

Figura 5.26: Lista de asignaturas del curso. Fuente: Elaboracion Propia

Nombre	Email	Opciones
Cristian	cristian@correo.es	

Figura 5.27: Lista de alumnos del curso. Fuente: Elaboracion Propia

5.3. Futuras mejoras

Como todo sistema de información existen varias versiones de desarrollo y GoodLearn se encuentra en su primera versión lo que da la posibilidad a que en un futuro se implementen funcionalidades nuevas o se mejoren algunas de las que ya se tienen debido a motivos de eficiencia o simplemente innovación en los diferentes entornos en los que están desarrollados.

Una posible mejora futura es el acceso al sistema de ficheros que posee cada sistema operativo en dispositivos móviles como Android e IOS. El acceso a este sistema se hace con lenguaje nativo por lo que una mejora sería añadir estos elementos para que sea posible el acceso a los ficheros o a la galería del dispositivo. Esto lo que hará es dar la posibilidad de poder subir o almacenar archivos de la aplicación directamente al disco del dispositivo.

Figura 5.28: Cabecera de asignatura para administradores. Fuente: Elaboracion Propia

Nombre	pepito
Email	pepito@hotmail.com
Telefono	673453987
Direccion	calle Marina, 13
Contraseña	
Repite la Contraseña	
Rol	administrador

SAVE

Figura 5.29: Vista de formulario Alumno y prototipo de todos. Fuente: Elaboracion Propia

móvil. Como primera versión del sistema el acceso para subir un fichero al sistema se realiza desde un formulario con input de tipo file que hará la subida del fichero pero el acceso a estos ficheros se ve limitado al usar vue y ionic como tecnología y no elementos nativos que brindan la eficiencia para el uso del sistema de ficheros.

Los elementos de notificaciones típicos que aparecen en los móviles o tablets tales como las notificaciones emergentes o las burbujas que aparecen en los iconos de la app son elementos como el sistema de ficheros, son nativos. La app de GoodLearn tiene un elemento de notificación interno en la app pero no elementos emergentes en el sistema. Esta mejora podría verse reflejada en el futuro con la implementación de dichos elementos para cada uno de los sistemas móviles.

Para la creación de publicaciones además del acceso al sistema de ficheros podría ser una buena idea el acceso a la cámara del dispositivo haciendo posible la creación de la imagen al momento desde la cámara y añadida como publicación directamente. El acceso a este elemento viene brindado por el lenguaje propio del entorno en el que se quiera desarrollar la app, por lo que es diferente si se quiere elaborar en Android o en IOS.

Por último una mejora que podría implementarse es crear un sistema de eventos para las conversaciones de los chats donde se puedan controlar de manera instantánea qué mensaje son los no leídos y estos se modifiquen en la vista e incluso la mejora del flujo en el que aparecen dichos mensajes como en redes tales como Whatsapp.

Lista de usuarios					
Informacion para los usuarios del sistema					
Nombre	Email	Telefono	Direccion	Rol	Opciones
Antonio	antonio@gmail.com	8235945	alto de las heras	administrador	 
Bibiana	bibi@correo.es	67298595	Alto De Las Heras	padre	  
Cristian	cristian@correo.es			alumno	 
Dolores	lola@correo.es			profesor	 
Lorena	lore@correo.es			profesor	 
Pepe	pepe@correo.es			alumno	 

Figura 5.30: Lista de los usuarios. Fuente: Elaboracion Propia

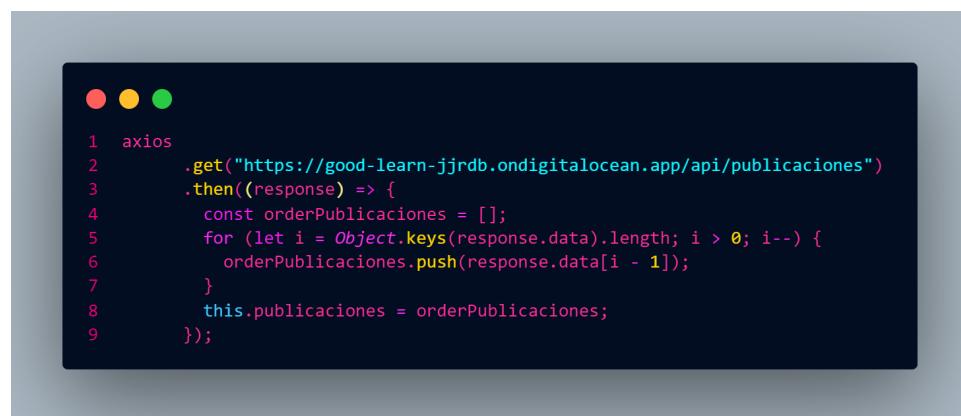
Lista de hijos					
Informacion para los alumnos del sistema					
Nombre	Email	Telefono	Direccion	Opciones	
Pepe	pepe@correo.es			 	
Cristian	cristian@correo.es			 	

Figura 5.31: Lista de los hijos asignados al padre. Fuente: Elaboracion Propia



```
● ● ●
1 export default defineComponent({
2   name: '',
3   components: {
4     ...
5   },
6   ...
7   mounted() {
8     ...
9   },
10  created() {
11    ...
12  },
13  ...
14  methods: {
15    ...
16  },
17  ...
18  data() {
19    return {
20      ...
21    };
22  },
23  ...
24  setup() {
25    ...
26  },
27});
```

Figura 5.32: Estructura Vue.js. Fuente: Elaboracion Propia



```
● ● ●
1 axios
2   .get("https://good-learn-jjrdb.ondigitalocean.app/api/publicaciones")
3   .then((response) => {
4     const orderPublicaciones = [];
5     for (let i = Object.keys(response.data).length; i > 0; i--) {
6       orderPublicaciones.push(response.data[i - 1]);
7     }
8     this.publicaciones = orderPublicaciones;
9   });

```

Figura 5.33: Ejemplo del uso de axios. Fuente: Elaboracion Propia

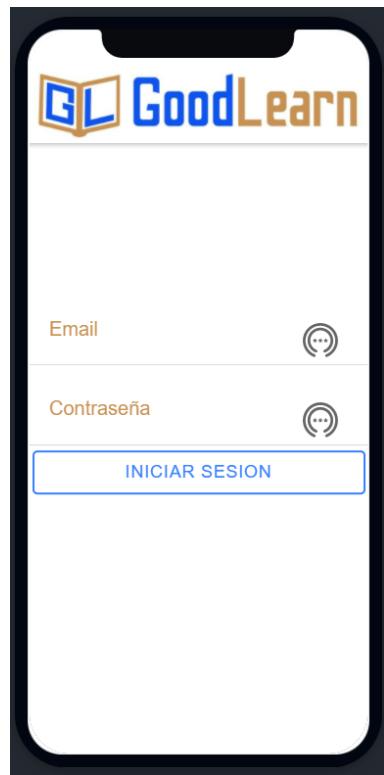


Figura 5.34: Ejemplo del uso de axios. Fuente: Elaboracion Propia

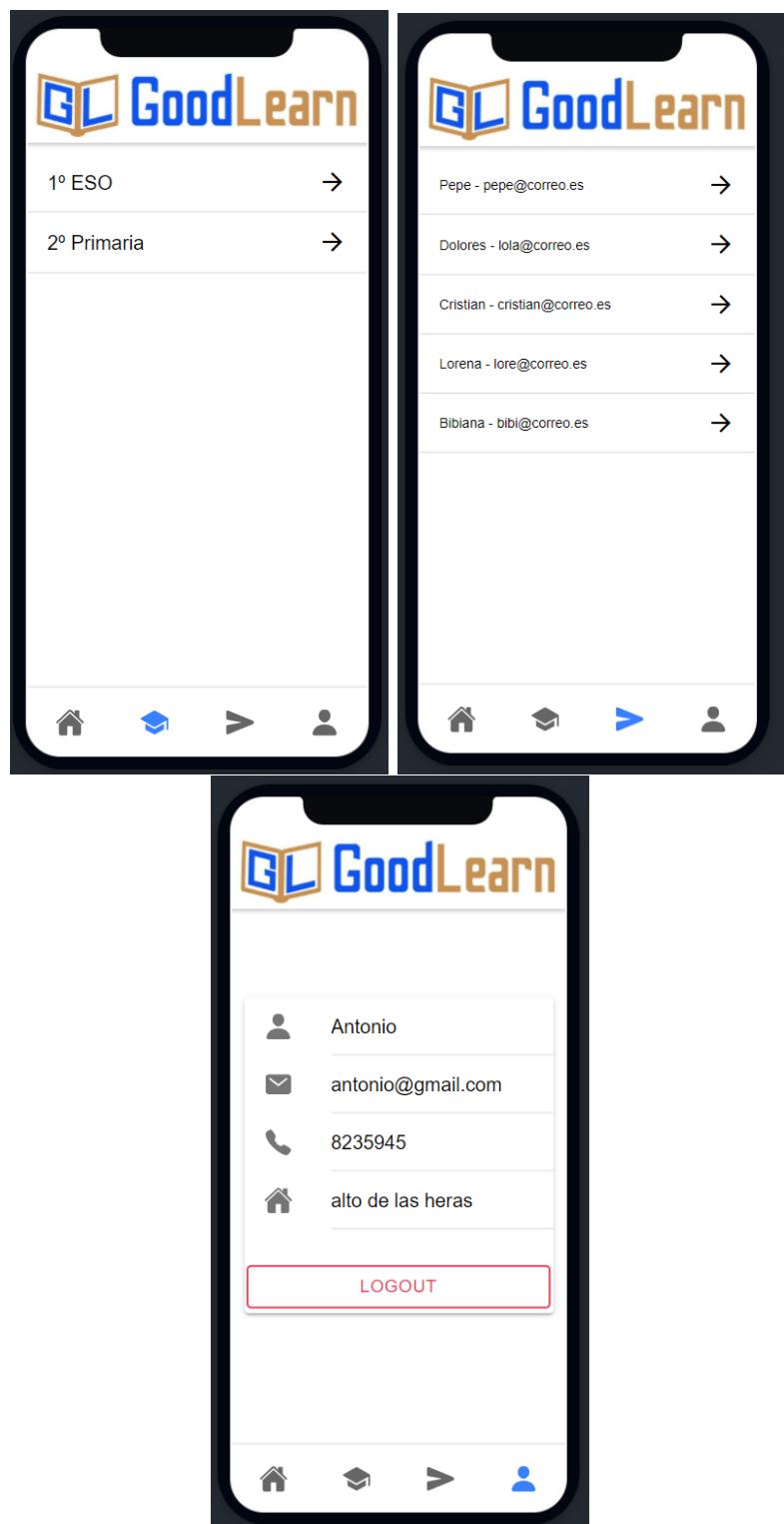


Figura 5.35: Pantallas comunes a los usuarios. Fuente: Elaboracion Propia

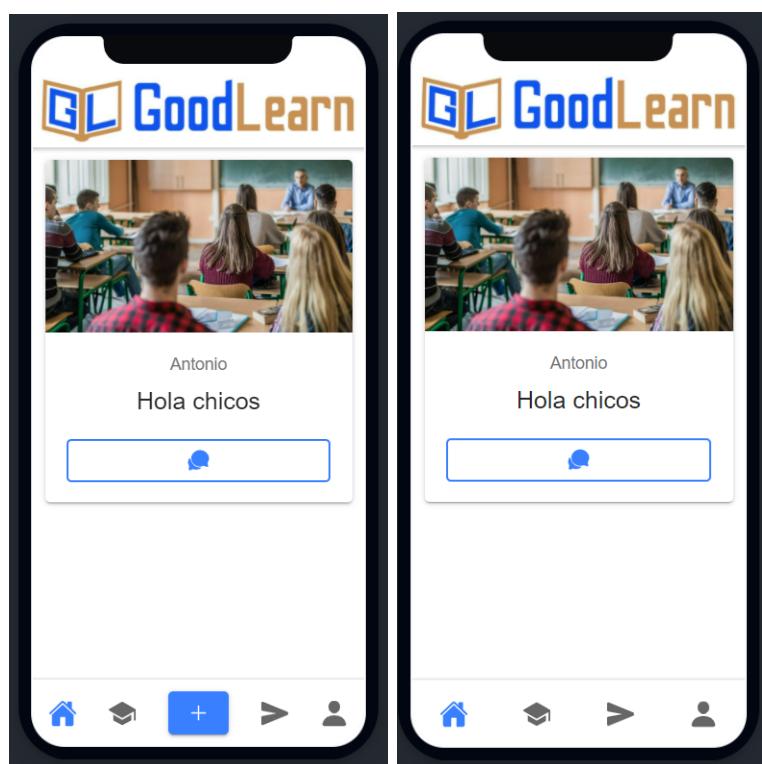


Figura 5.36: Diferencia en el home. Fuente: Elaboracion Propia

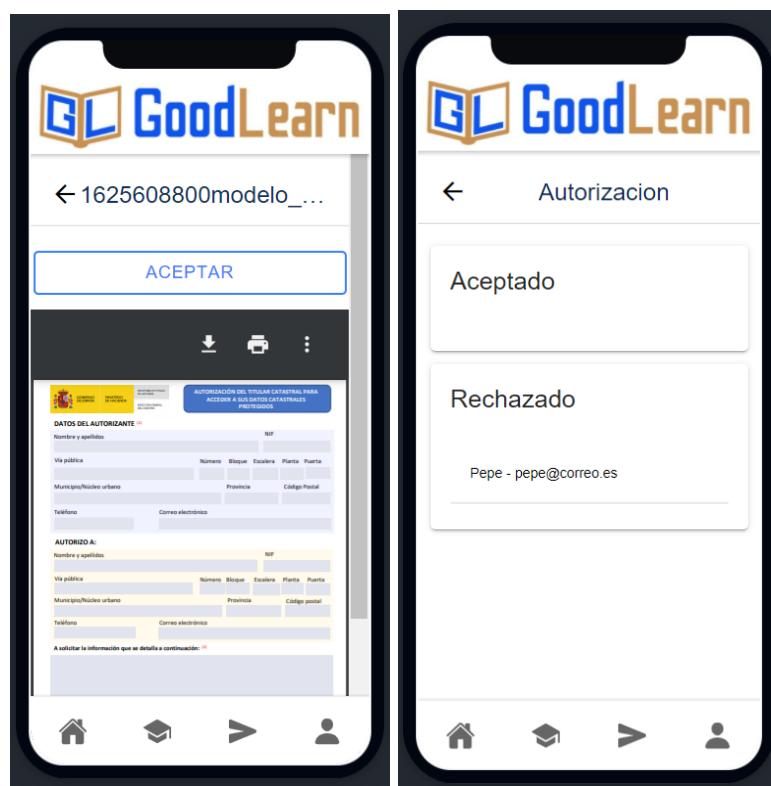


Figura 5.37: Diferencia en las autorizaciones. Fuente: Elaboracion Propia

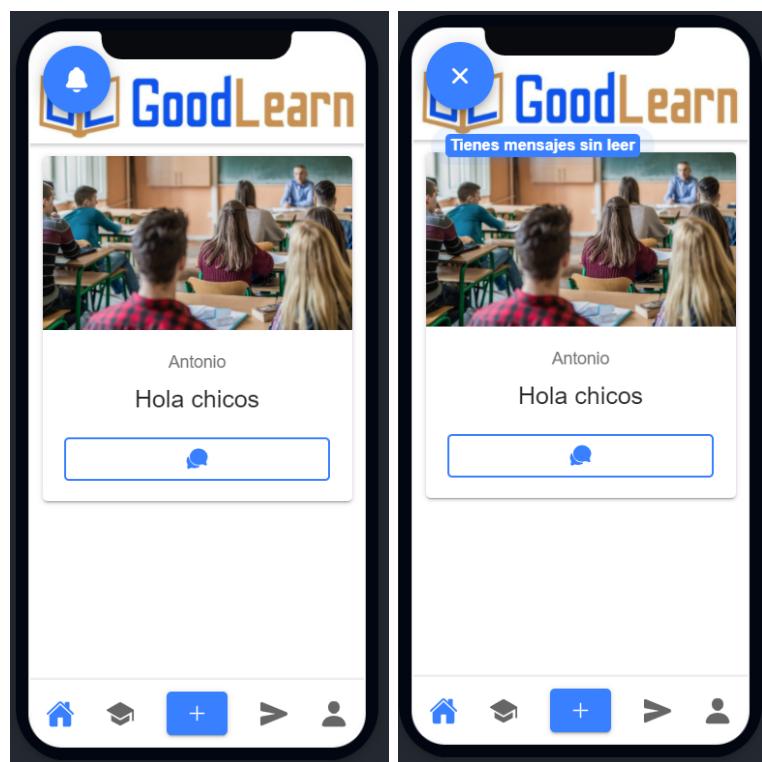


Figura 5.38: Notificaciones. Fuente: Elaboracion Propia

Capítulo 6

Conclusión

GoodLearn ha sido el proyecto más grande al que me he desafiado personalmente y el más complejo de los que he realizado. Se ha creado con las tecnologías como Github y Postman un entorno de trabajo y canal de comunicación entre alumno-tutor que ha servido como guía de los avances y con el que se ha controlado el tiempo.

La comunicación creada entre diferentes tecnologías mediante la API ha sido el pilar fundamental del proyecto y el objetivo principal para que todo funcione. Personalmente nunca había desarrollado una API propia para un sistema y me ha parecido en un principio algo complejo debido a que hay que tener en cuenta muchos tipos de métodos y urls. El tiempo invertido en la implementación de este elemento ha sido el más corto debido a que ha sido muy repetitivo su implementación pero por lo general ha sido de las implementaciones personalmente la más fácil.

En cuanto al dashboard de profesores y administrador no me ha resultado difícil puesto que me encanta el desarrollo web y php por lo que su implementación en Laravel ha sido bastante sencilla y sobre todo teniendo ya el trabajo desarrollado en la API por lo que estas llamadas a controladores ya estaban meditadas antes de implementar la vista de la web.

En conclusión con Laravel me ha parecido un framework no de los más fáciles como Django de python pero me parece más complejo que este ya que te brinda muchas librerías ya predeterminadas y un documentación extensa donde se puede encontrar cualquier información, además de una gran comunidad que desarrolla en este Framework o grupos como telegram donde se puede obtener información.

Personalmente la mayor dificultad encontrada en el proyecto ha sido el desarrollo de la app con ionic y más en específico Vue.js ya que el lenguaje de JavaScript siempre me ha costado mucho su implementación. Los elementos ionic son bastante fáciles puesto que tenemos la documentación que te faci-

lita el trabajo y además la sintaxis es muy parecida a HTML y se pueden incluir elementos de los anteriores sin ningún problema. La dificultad añadida es el aprendizaje del framework Vue e indagar más en los conocimientos de JavaScript. En comparación con Laravel el lenguaje php lo tengo muy familiarizado y por eso me resultó más fácil. Tener que aprender la estructura de los componentes vue y como ejecutar los flujos de obtención de datos con axios y almacenamientos me resultaron bastante difíciles.

Como conclusión final si alguien me dice que en unos años sería capaz de crear GoodLearn no me lo creería y actualmente me veo capaz de afrontar cualquier problema que se me plantee o al menos intentarlo hasta conseguir la solución como el caso de Vue.js y crear una app y una API, cosas que nunca había hecho.

Bibliografía

- [1] Educaweb. La educación en españa ante el coronavirus.
- [2] FAD Educacion conectada and BBVA. Panorama de la educación en España tras la pandemia de covid-19: La opinión de la comunidad educativa. (Figura 6):21, 2020.
- [3] Digitalnewsreport. Tres de cada cuatro internautas usan el ‘smartphone’...
- [4] Schooltivity. Schooltivity.
- [5] Alexia Suit Educativa. Alexia suit educativa.
- [6] ClassDojo. Classdojo.
- [7] Universidad de Alicante. Universidad de alicante.
- [8] Código Facilito. Código facilito.
- [9] La Vanguardia. La Vanguardia.
- [10] GitHub. Github.
- [11] Postman. Postman.
- [12] Visual Studio Code. Visual studio code.
- [13] Photoshop. Photoshop.
- [14] DigitalOcean. Digitalocean.
- [15] Laravel. Laravel framework.
- [16] Ionic Framework. Ionic framework.