



**Escuela Superior
de Ingeniería y Tecnología**
Universidad de La Laguna

Trabajo de Fin de Grado

Herramienta para la obtención de
feedback por parte del alumnado

Tool for obtaining feedback from students

Javier Alberto Martín

La Laguna, 8 de septiembre de 2021

D. **Alejandro Pérez Nava**, con N.I.F. 43821179S profesor Titular de Universidad adscrito al Departamento de Ingeniería Informática y de Sistemas de la Universidad de La Laguna, como tutor

D. **Fernando Pérez Nava**, con N.I.F. 42091420V profesor Titular de Universidad adscrito al Departamento de Ingeniería Informática y de Sistemas de la Universidad de La Laguna, como cotutor

C E R T I F I C A (N)

Que la presente memoria titulada:

"Herramienta para la obtención de feedback por parte del alumnado"

ha sido realizada bajo su dirección por D. **Javier Alberto Martín**, con N.I.F. 45733473C.

Y para que así conste, en cumplimiento de la legislación vigente y a los efectos oportunos firman la presente en La Laguna a 8 de septiembre de 2021

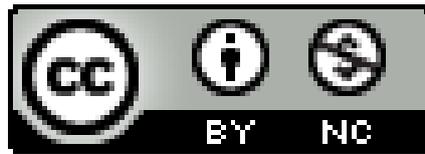
Agradecimientos

A mi toda mi familia, que siempre confió en mí, en especial a mis padres, que nunca se dieron por vencidos con mi formación y educación.

A los profesores de la Escuela Superior de Ingeniería y Tecnología de la Universidad de La Laguna con los que tuve el placer de cruzarme y que me enseñaron todo lo que ahora sé de este apasionante mundo.

Por último, a todos esos amigos y compañeros de pasión que dedicaron tanto tiempo y esfuerzo en ayudarme a superarme cada día más.

Licencia



© Esta obra está bajo una licencia de Creative Commons Reconocimiento-
NoComercial 4.0 Internacional.

Resumen

Este Trabajo de Fin de Grado ha sido realizado con el objetivo de mejorar el rendimiento académico en la formación de los estudiantes en el ámbito universitario. Debido a que las clases magistrales son grandes y la vergüenza que pueden sentir algunos alumnos, no hay 'feedback' por su parte. A causa de todo ello, se quiere resolver el problema de no saber cuando los alumnos entienden el tema que se les está explicando.

Para esto, ha sido diseñada una aplicación móvil que permite a los alumnos indicar qué temas o clases no han entendido de forma anónima. Por otro lado, se ha requerido implementar un portal web para que los profesores puedan añadir las preguntas que crean oportunas sobre un tema o clase dada. Con esto, el docente puede evaluar y gestionar en qué parte de una asignatura debe ponerse mayor empeño basándose en el porcentaje de alumnos que no entienden los diferentes temas. En este documento, se expone el estado del arte en referencia a las actuales aplicaciones existentes que harían posible afrontar este problema y las razones por las que, en este caso, es mejor crear una aplicación propia y a medida. Seguidamente, se enuncian las tecnologías seleccionadas y los motivos por las que se eligieron. Luego, se narra cómo se comenzó con el desarrollo y los requerimientos que se tuvieron en cuenta en este proceso, hasta conseguir la solución que cumple con las exigencias planteadas inicialmente.

Con este trabajo, se ha conseguido desarrollar una solución llamada "Entendiste", capaz de realizar consultas anónimas al alumnado para saber si entendieron la clase o el tema impartido.

Palabras clave: Aplicación móvil, Android, Web, PHP, Laravel

Abstract

The aim of this project is to improve students' academic performance in the university environment. Due to the large lectures and the shyness that some students may feel, it is hard to receive any feedback from them sometimes. Because of this, we pursue to solve the problem of lack of communication between educator and students.

On this basis, a mobile application that allows students to indicate anonymously which topics or classes they have not understood has been designed. Moreover, it was necessary to implement a web portal so that teachers could add the questions they deem appropriate on a given topic or class. This way, the teacher can evaluate and manage in which part of a subject more effort should be put based on the percentage of students who do not understand the different topics. In this document, the state of the art is exposed in reference to the current existing applications that would make it possible to face this problem and the reasons why, in this case, it is better to create your own custom application. The technologies selected and the reasons why they were chosen are listed below.

With this work, it has been possible to develop a solution called "Entendiste", which is capable of making anonymous queries to students to find out if they understood the class or the topic of a certain lecture.

Keywords: Mobile app, Android, Web, PHP, Laravel

Índice general

1. Introducción	1
1.1. Objetivos	1
1.1.1. Creación de un portal web para los profesores	1
1.1.2. Creación de una aplicación móvil para los alumnos	2
1.2. Planificación	2
1.2.1. Tarea 1: Prototipos	2
1.2.2. Tarea 2: Diseño de base de datos	2
1.2.3. Tarea 3: Desarrollar el portal web	2
1.2.4. Tarea 4: Desarrollar la aplicación móvil	3
1.2.5. Tarea 5: Pruebas	3
1.3. Estado del arte	3
1.3.1. Aplicaciones similares	3
1.3.2. ¿Por qué no usar una de estas aplicaciones?	3
1.4. Tecnologías utilizadas	4
1.4.1. ¿Por qué PHP?	4
1.4.2. ¿Por qué Laravel?	5
1.4.3. ¿Por qué Android?	6
2. Hitos	8
2.1. Prototipos	8
2.2. Diseño de la base de datos	13
2.3. Aplicación web	16
2.4. Aplicación móvil	20
3. Aplicación 'Entendiste'	24
3.1. Uso de la página web (profesores)	27
3.2. Uso de la aplicación móvil (alumnos)	31
4. Conclusiones y líneas futuras	37
4.1. Conclusiones	37
4.2. Líneas futuras	37
5. Summary and Conclusions	39
5.1. Conclusions	39
5.2. Future lines	39
6. Presupuesto	41

Índice de Figuras

1.1. Logo de PHP	5
1.2. Logo de Laravel 8	6
1.3. Logo de Android Studio	7
2.1. Login del prototipo web	8
2.2. Pantalla de asignaturas del prototipo web	9
2.3. Pantalla de registro de asignaturas del prototipo web	9
2.4. Pantalla de temas del prototipo web	10
2.5. Pantalla de registro de temas del prototipo web	10
2.6. Login del prototipo de la app	11
2.7. Pantalla de asignaturas del prototipo de la app	12
2.8. Pantalla de temas del prototipo de la app	13
2.9. Tabla de usuarios	14
2.10Tabla de asignaturas	14
2.11Tabla de relación entre usuarios y asignaturas	14
2.12Tabla de preguntas	15
2.13Tabla de respuestas	15
2.14Modelo de datos físico	16
2.15Lista dinámica en la pantalla de asignaturas de la app	21
2.16Pantalla para responder en la app	22
3.1. Paleta	24
3.2. Isotipo de Entendiste	25
3.3. Imagotipo de Entendiste	25
3.4. Esquema de componentes	26
3.5. Login en la web	27
3.6. Registro en la web	27
3.7. Pantalla de inicio de la web	28
3.8. Pantalla de asignaturas de la web	28
3.9. Dar de alta una asignatura	29
3.10Pantalla de temas de la web	29
3.11Añadir nuevo tema en la web	30
3.12Pantalla de inicio de la app	31
3.13Pantalla de login de la app	32
3.14Pantalla de asignaturas de la app	33
3.15Buscar una asignatura en la app	33
3.16Resultado de la búsqueda en la app	34
3.17Pantalla para introducir contraseña en la app	34
3.18Pantalla de temas en la app	35

3.19 Pantalla de respuesta en la app 36

Índice de Tablas

6.1. Presupuesto 41

Capítulo 1

Introducción

Los profesores tienen como oficio enseñar, pero es difícil saber si están transmitiendo bien estas enseñanzas si no reciben una respuesta por parte de los alumnos. Y todos sabemos que, cuando el profesor pregunta si comprenden el temario, no contesta nadie o solo lo hacen 2 o 3. Como dice la astrofísica y divulgadora Marta Seror, "Que los estudiantes no pregunten dudas no quiere decir que no las tengan. Puede que crean que, al decir en voz alta sus preguntas, están demostrando frente al resto de la clase que no son tan inteligentes"[1]. No solo entra en juego la vergüenza, a veces los alumnos están despistados o, simplemente, no les apetece interactuar con la clase.

Para solventar este problema se requirió la creación de un sistema que permitiera conocer en tiempo real el número de alumnos que entendieron los contenidos explicados frente al número de alumnos que no los entendieron. Una manera muy cómoda para los alumnos de conceder esta información a los profesores es a través de una aplicación. Los jóvenes prefieren usar su teléfono móvil para cualquier tarea, por simple que sea. Así, mirar la hora, hacer un cálculo matemático e incluso, comunicarse con otras personas, son labores que realizan a través de sus dispositivos. En un artículo[2] del periódico "El País", se estima que "para el 95,1 % de la población española, la mensajería instantánea es el canal preferido para comunicarse, por encima incluso de la comunicación en persona, que se sitúa en el 86,60 %".

1.1. Objetivos

1.1.1. Creación de un portal web para los profesores

La finalidad de crear un portal web es la de facilitar una herramienta online a los profesores para la gestión de sus asignaturas y los contenidos de cada asignatura. Se debe disponer de la posibilidad de dar de alta nuevas asignaturas, dar de alta nuevos temas, ver la cantidad de respuestas obtenidas, así como el número de alumnos que entienden o no entienden el tema. Para esto es necesario:

- Estudiar tecnologías apropiadas
- Crear un entorno local cómodo de desarrollo

- Desarrollar un backend que se ocupe de la lógica del sistema
- Implementar la apariencia de la página web
- Proporcionar una API[3] para el funcionamiento de la aplicación móvil

1.1.2. Creación de una aplicación móvil para los alumnos

La aplicación móvil proporcionará a los alumnos una herramienta para tener una lista de todas sus asignaturas y, en cada una de ellas, la manera de comunicar a los profesores qué tema no entienden, por medio de una encuesta binaria, en la que sólo tendrán que marcar 'sí' o 'no'. Para ello se debe:

- Decidir para qué dispositivos desarrollar la aplicación
- Establecer un entorno de desarrollo
- Implementar el diseño de la app
- Probar la API desarrollada en el backend
- Implementar la lógica de la aplicación

1.2. Planificación

1.2.1. Tarea 1: Prototipos

Crear prototipos del portal web y de la aplicación móvil para tener una idea general del modo de uso y así decidir mejor los requerimientos de ambas herramientas.

1.2.2. Tarea 2: Diseño de base de datos

Diseñar un diagrama con todas las tablas necesarias para guardar los datos del sistema y establecer las relaciones entre las tablas.

1.2.3. Tarea 3: Desarrollar el portal web

Conseguir una web completamente funcional que permita agregar asignaturas y temas, y que a cada profesor le aparezcan las asignaturas en las que imparte clases.

1.2.4. Tarea 4: Desarrollar la aplicación móvil

Desarrollar del todo la aplicación móvil, consiguiendo que a cada alumno se le muestre su lista de asignaturas, que se le permita buscar nuevas asignaturas por código y por nombre, que puedan acceder a los temas de cada asignatura y que se les permita seleccionar la respuesta en cada tema.

1.2.5. Tarea 5: Pruebas

Testear todo el sistema en su conjunto, probando diferentes casos de uso y comprobando que todo funciona correctamente.

1.3. Estado del arte

1.3.1. Aplicaciones similares

Las aplicaciones más parecidas a esta podrían ser:

- Socrative o Poll Everywhere: Herramientas para crear diferentes tipos de encuestas por los profesores y ser contestadas por los alumnos en clase en tiempo real.
- Plickeres: Una herramienta parecida a las anteriores, pero sin la necesidad de que los alumnos tengan un dispositivo. Necesitan un código QR impreso que giran según la opción que elijan en la encuesta que hay en la pantalla del profesor.

1.3.2. ¿Por qué no usar una de estas aplicaciones?

Es totalmente posible utilizar cualquier aplicación dedicada a hacer cuestionarios para saciar las necesidades que en este proyecto se plantean, pero no tener el control de la aplicación puede tener efectos negativos. Entre los beneficios de crear una aplicación a medida se encuentra la posibilidad de modificarla para cumplir con los requisitos de un momento concreto. Poder personalizarla y mejorar la experiencia de usuario también son beneficios a tener en cuenta.

Además, una aplicación más específica es menos pesada, dado que solo contiene código para trabajar con las tareas para las que fue diseñada, o lo que es lo mismo, el contenido está más optimizado.

A todo esto hay que sumarle que no se corre el riesgo de perder datos, ya que se puede elegir donde almacenarlos y gestionar las copias de seguridad.

1.4. Tecnologías utilizadas

El proyecto se divide en dos grandes partes. Por un lado, se ha desarrollado una web haciendo uso de Laravel, un framework para backend. Laravel utiliza PHP como lenguaje de programación. Por otra parte, se ha creado una aplicación móvil para dispositivos Android con Android Studio. Java ha sido el lenguaje escogido para implementar la lógica de la app móvil.

1.4.1. ¿Por qué PHP?

Entre los lenguajes más usados para ejecutar en un servidor se encuentran JavaScript, PHP, Python o Ruby.

- JavaScript[4]

JavaScript usa plataformas como Express.js[5] y Node.js[6] pero tareas como conectar con la base de datos MySQL[7] o entender el uso de middlewares se hacen algo complejas. La libertad que ofrecen los marcos para backend de JavaScript a veces es una desventaja, al no tener un estándar claro.

- Python[8]

La ejecución de Python se vuelve lenta cuando se interrumpe, la capa de acceso a base de datos está menos desarrollada respecto a otros, se requieren pruebas exhaustivas para detectar errores y depende mucho de librerías externas.

- Ruby[9]

El tiempo de ejecución de Ruby no es tan bueno, además tiene fuentes y bibliotecas limitadas. Por si fuera poco, no es tan fácil de aprender, además de difícil de depurar.

- PHP[10]

Por otra parte, PHP ofrece una gran facilidad de uso además de versatilidad en cuanto a compatibilidad con bases de datos y distintos

sistemas operativos. Tiene una gran comunidad detrás, lo que hace que sea muy fácil resolver cualquier problema que no te deje avanzar.



Figura 1.1: Logo de PHP

1.4.2. ¿Por qué Laravel?

Se necesitaba conectar una aplicación móvil y una web para compartir datos de manera sincronizada, por lo que se tuvo la necesidad de escoger un framework backend que cumpliera con las necesidades del proyecto.

Haciendo una comparación entre los más populares, se ha escogido Laravel debido a su elegancia y simpleza. Además, la curva de aprendizaje no es tan grande como, por ejemplo, ASP.NET[11], el cual también tiene la restricción de que los proyectos desarrollados en esta tecnología deben correr en un servidor con Windows Server[12].

La documentación de Laravel es excelente, algo de agradecer cuando comienzas con herramientas nuevas.

Además, gracias al motor de plantillas Blade, se simplifica el trabajo de crear las vistas de la aplicación web, evitando así la necesidad de utilizar frameworks frontend como AngularJS[13] o React[14], al fin y al cabo, no procede crear una web muy atractiva, ni cuidar la usabilidad y experiencia de usuario para atraer y fidelizar clientes, cuando su uso se limitará a un grupo pequeño de profesores.

Eloquent, el ORM (mapeo objeto-relacional) utilizado por Laravel, brinda una manera de trabajar con la base de datos muy simple y consigue consultas complejas sin la necesidad de insertar código SQL. Las relaciones

entre modelos hacen el trabajo. En el código 1.1 puede verse la manera en la que ha sido creada la tabla de usuarios.

```
1 public function up()
2     {
3         Schema::create('users', function (Blueprint $table) {
4             $table->engine = 'InnoDB';
5
6             $table->string('idUserio');
7             $table->string('password');
8             $table->string('nombre');
9             $table->string('apellidos');
10            $table->unsignedTinyInteger('rol');
11            $table->rememberToken();
12            $table->timestamps();
13
14            $table->primary('idUserio');
15        });
16    }
```

Código 1.1: Migración

Laravel se basa en el gestor de dependencias Composer[15], un manejador de paquetes que te descarga e instala las dependencias de tu proyecto y que las mantiene actualizadas si se desea. Composer para PHP sería el equivalente de Bundler para Ruby o NPM para Node.

En definitiva, Laravel intenta facilitar el desarrollo de las tareas típicas de los proyectos web, como las rutas, la autenticación de usuarios, el manejo de bases de datos o el uso de plantillas que simplifican la creación y reutilización de vistas.



Figura 1.2: Logo de Laravel 8

1.4.3. ¿Por qué Android?

La aplicación móvil ha sido implementada para Android por ser, con diferencia, el sistema operativo para móviles más utilizado en el mundo. 9 de cada 10 usuarios móviles tienen Android como sistema operativo.

Además, Android es menos restrictivo a la hora de programar aplicaciones que su principal competidor, iOS[16]. El IDE por excelencia para programar aplicaciones para Android es Android Studio[17], que se basa en el potente IDE IntelliJ IDEA.

Android Studio dispone de muchas herramientas que facilitan la programación de aplicaciones. Proporciona un editor de interfaz de usuario que, aparte de poder modificar las vistas a base de código, permite arrastrar componentes visualmente y jugar con la disposición, tamaño, colores, etc. de dichos componentes. Esto se renderiza casi automáticamente, lo que permite ver cómo va quedando la interfaz. Además, Android Studio tiene integrado el emulador Android Virtual Device (AVD), una herramienta que gestiona todos los dispositivos virtuales que quieras, cada uno con las características que se desee: versión del sistema operativo, API, resolución de la pantalla... Y no solo móviles, los smartwatches, tablets, Android TV también pueden emularse con esta herramienta.

Android Studio permite la programación en distintos lenguajes, pero los dos lenguajes punteros para el desarrollo de apps móviles son Kotlin y Java. Java es el viejo conocido; robusto, con una gran comunidad detrás, independiente de la plataforma, orientado a objetos... La mayoría de aplicaciones móviles para Android están escritas en Java. Por otra parte está Kotlin, nombrado como lenguaje oficial para Android por Google. Interoperable con Java, fácil de aprender, se necesitan menos líneas de código para la misma tarea hecha en Java...

Ambas parecen buenas opciones, pero se ha elegido Java por la extensa comunidad y la facilidad para encontrar soluciones a cualquier problema o duda que pueda surgir.



Figura 1.3: Logo de Android Studio

Capítulo 2

Hitos

A continuación, se expondrá el conjunto de tareas realizadas y la metodología seguida durante todo el proyecto con un orden cronológico.

2.1. Prototipos

- La primera tarea ha sido diseñar el prototipo de la web dedicada a los profesores, dónde se plasma la experiencia de insertar nuevas asignaturas y nuevos temas.

Este prototipo ha sido efectuado con la herramienta proto.io[18], una plataforma para crear prototipos para diversos dispositivos y tamaños de pantalla. Este útil hace uso de una interfaz de arrastrar y soltar, lo que no precisa de codificación alguna por parte del diseñador.

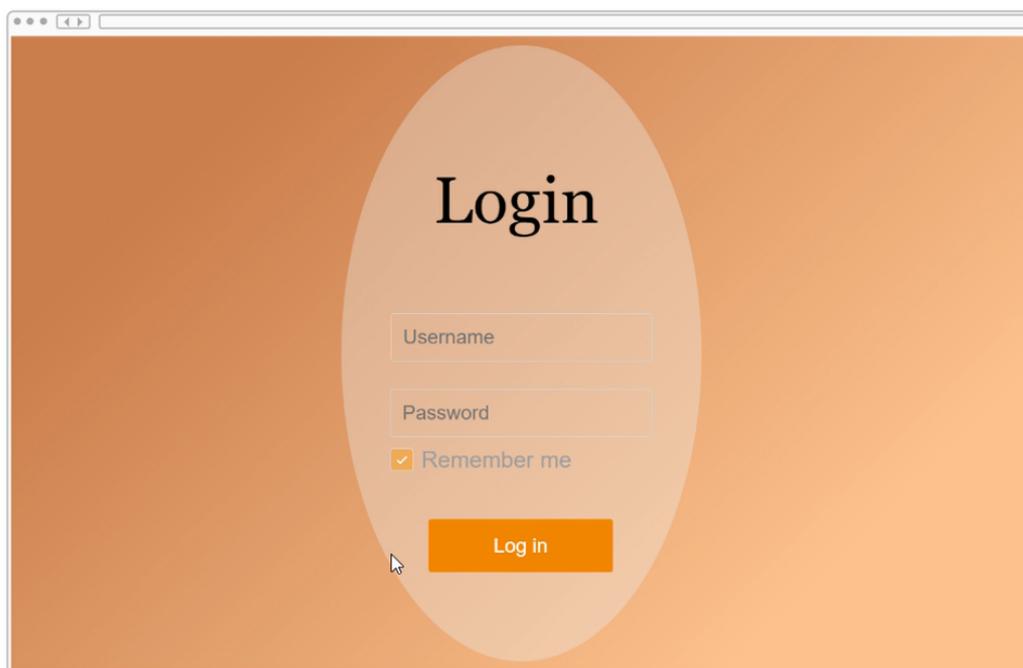


Figura 2.1: Login del prototipo web



Figura 2.2: Pantalla de asignaturas del prototipo web



Figura 2.3: Pantalla de registro de asignaturas del prototipo web



Figura 2.4: Pantalla de temas del prototipo web



Figura 2.5: Pantalla de registro de temas del prototipo web

- Seguidamente, se diseñó el prototipo de la aplicación móvil. Se mostró cómo acceder a una asignatura y, posteriormente, a un tema de dicha asignatura para terminar marcando la respuesta a la pregunta “¿Has entendido este tema?”.

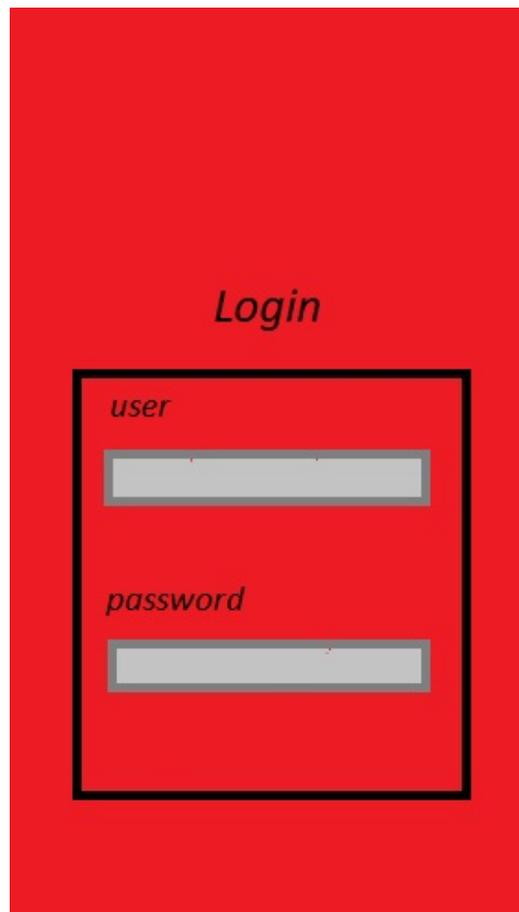


Figura 2.6: Login del prototipo de la app

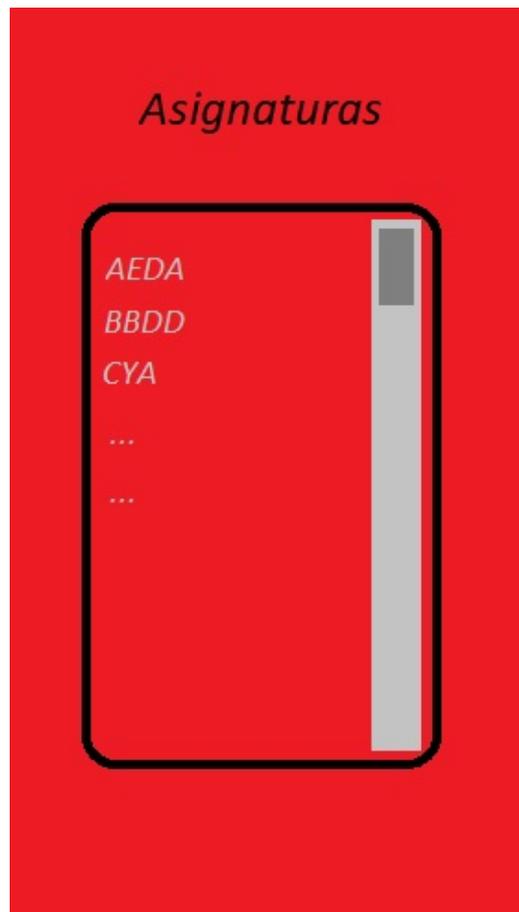


Figura 2.7: Pantalla de asignaturas del prototipo de la app



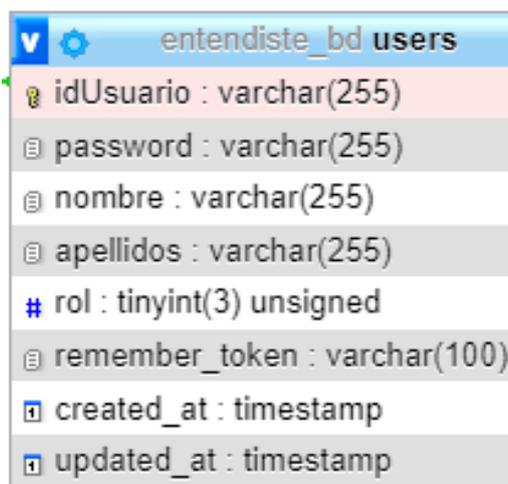
Figura 2.8: Pantalla de temas del prototipo de la app

2.2. Diseño de la base de datos

Analizando las necesidades del proyecto se han creado las siguientes tablas En un primer vistazo al futuro del proyecto, surge la necesidad de crear las siguientes tablas:

- Usuarios

Esta tabla contendrá la información de cada usuario, así como el rol (alumno o profesor), con un identificador único, que en el caso de los alumnos coincide con el usado durante todo su proceso académico: prefijo "alu" seguido del Número de Identificación Universitario (NIU), por ejemplo, alu0100836400.

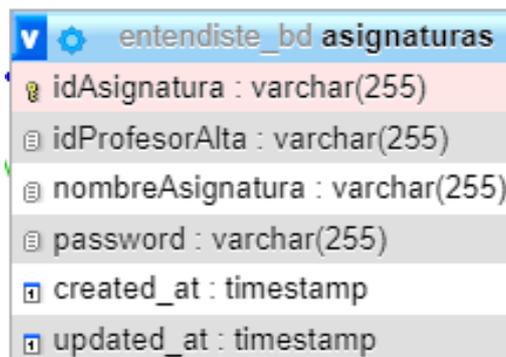


entendiste_bd users	
idUsuario	varchar(255)
password	varchar(255)
nombre	varchar(255)
apellidos	varchar(255)
rol	tinyint(3) unsigned
remember_token	varchar(100)
created_at	timestamp
updated_at	timestamp

Figura 2.9: Tabla de usuarios

- Asignaturas

En esta tabla se refleja la información de cada asignatura: un identificador que coincide con el código de asignatura usado en la Universidad de La Laguna, el id del profesor que le dio de alta a la asignatura, la contraseña necesaria para acceder a dicha asignatura...



entendiste_bd asignaturas	
idAsignatura	varchar(255)
idProfesorAlta	varchar(255)
nombreAsignatura	varchar(255)
password	varchar(255)
created_at	timestamp
updated_at	timestamp

Figura 2.10: Tabla de asignaturas

- Pertenencia Asignaturas

En esta tabla se establece la relación entre la tabla “usuarios” y la tabla “asignaturas”.



entendiste_bd perteneceasignaturas	
idAsignatura	varchar(255)
idUsuario	varchar(255)
created_at	timestamp
updated_at	timestamp

Figura 2.11: Tabla de relación entre usuarios y asignaturas

- Preguntas

Esta tabla contiene la información de cada pregunta o tema del que se quiere obtener los datos por parte del profesor. Se almacenará la asignatura a la que pertenece la pregunta, el profesor que la publicó y la fecha en la que se creó el registro.



entendiste_bd preguntas	
idPregunta	bigint(20) unsigned
idAsignatura	varchar(255)
idProfesor	varchar(255)
pregunta	text
created_at	timestamp
updated_at	timestamp

Figura 2.12: Tabla de preguntas

- Respuestas

En esta tabla se insertan las respuestas de cada alumno a cada tema.



entendiste_bd respuestas	
idPregunta	bigint(20) unsigned
idAlumno	varchar(255)
respuesta	tinyint(3) unsigned
created_at	timestamp
updated_at	timestamp

Figura 2.13: Tabla de respuestas

Se creó un diagrama con todo detalle haciendo uso de la herramienta online Visual Paradigm[19], añadiendo las relaciones entre tablas, claves foráneas y claves primarias, así como el tipo de cada campo en todas las tablas. Dicho diagrama puede verse en la figura 2.14

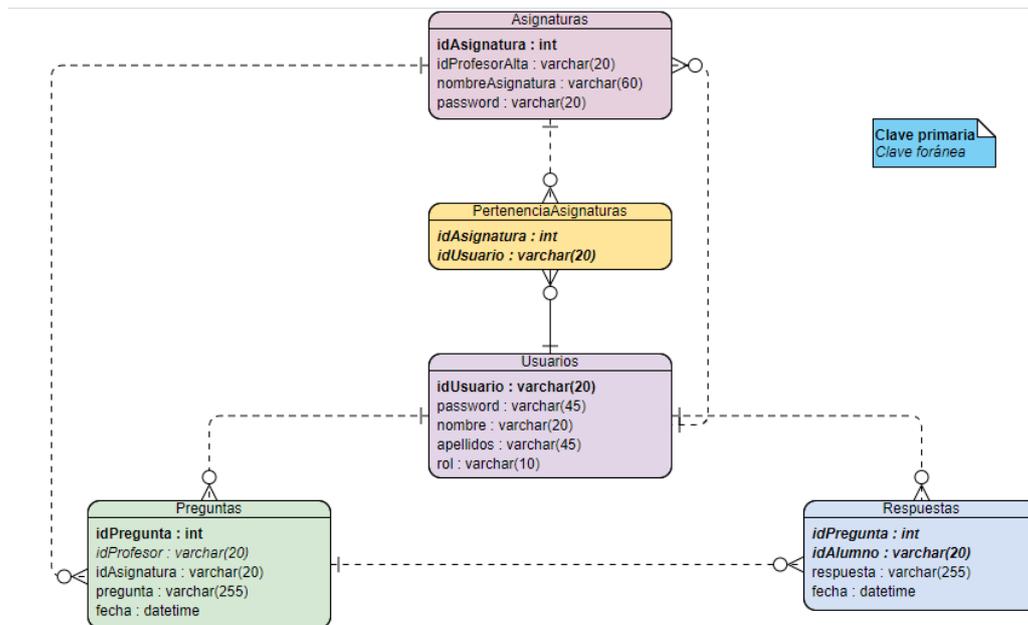


Figura 2.14: Modelo de datos físico

2.3. Aplicación web

Lo primero ha sido conseguir un sistema de login completamente funcional, dejando acceder solo a los usuarios con rol de profesor. El sistema ha sido probado insertando usuarios ficticios con la librería Faker[20] de PHP. Para usar esta librería solo es necesario incluirla en el fichero que la vayamos a usar, crear un objeto Faker y llamar a cada propiedad del objeto. Cada llamada genera un valor distinto. En el código 2.1 se puede ver un ejemplo sencillo con esta librería.

```

1 $faker = Faker\Factory::create();
2
3 echo $faker->name;           //esto devuelve un nombre aleatorio -> 'Lucy Cechtelar'
4 echo $faker->address;       //dirección falsa -> '8888 Cummings Vista Apt. 101, Susanbury,
5 echo $faker->text;          //texto -> "And yet I wish you could manage it?) 'And what are
                             they made of?' Alice asked in a shrill, passionate voice. 'Would YOU like cats if you
                             were never even spoke to Time!' 'Perhaps not,' Alice replied."

```

Código 2.1: Uso de la librería Faker

Una vez conseguido esto, se implementó la pantalla de las asignaturas. De nuevo, con datos ficticios, pero esta vez, introducidos a mano. También se insertaron datos en la tabla de “PertenenciaAsignaturas” para comprobar que solo aparecían las asignaturas a las que el profesor estaba inscrito.

Todas las pantallas, menos el login, heredan de una plantilla que contiene la barra superior con el menú y los estilos de la página. Esto se ha realizado con el motor de plantillas Blade. Como se ve en los códigos 2.3 y 2.2, la directiva ‘@content’ abre un bloque que permite insertar código para luego mostrarlo en otra página con la directiva ‘@yield’.

```

1 @extends('layouts.app')
2
3 @section('title', 'Inicio') <!-- directiva section -->
4 @section('head') <!-- directiva section -->
5     <link rel="stylesheet" href="{{asset('css/inicio.css')}}">
6 @endsection
7
8 @section('content') <!-- directiva section -->
9 <p>
10     <span>
11         Esta web forma parte del Trabajo de Fin de Grado <br><br><label class="italic">"
12         Herramienta para la obtenci n de
13         feedback por parte del alumnado".</label> <br><br>Escuela Superior de Ingenier a
14         y Tecnolog a.
15         <br>Universidad de La Laguna
16     </span>
17     &mdash; Javier Alberto Mart n &mdash;
18 </p>
19 @endsection

```

Código 2.2: Directivas content

```

1 <!DOCTYPE html>
2 <head>
3     <meta charset="UTF-8">
4     <title>Entendiste | @yield('title')</title>
5     <link rel="stylesheet" href="{{asset('css/menu-bar.css')}}">
6     @yield('head') <!-- directiva yield -->
7 </head>
8 <body>
9     <?php
10         if(!isset($_COOKIE["user"])) {
11             echo "<script> window.location.replace('".route('login')."') </script>";
12             $user = "";
13         }
14         else $user = $_COOKIE['user'];
15     ?>
16     <div class="dropdown">
17         <nav>
18             
19             <ul>
20                 <li><a href="{{route('inicio')}}">Inicio</a></li>
21                 <li>
22                     <a href="{{route('asignaturas')}}">Asignaturas <i class="fas fa-caret-down"></i>
23                 ></a>
24                 <li><a href="{{route('login')}}">Salir</a></li>
25             </ul>
26         </nav>
27     </div>
28     <div class="container">
29         @yield('content') <!-- directiva yield -->
30     </div>
31     @yield('modal') <!-- directiva yield -->
32 </body>

```

Código 2.3: Directivas yield

Lo siguiente ha sido desarrollar la parte en la que se muestran los temas de cada asignatura. Para cada tema se muestra un gráfico circular reflejando el porcentaje de alumnos que han respondido sí frente a los que han respondido no.

A continuación, fue implementada la opción de añadir nueva asignatura y la de añadir nuevo tema mediante modales. Este modal es incluido en ambas páginas para no repetir código. El código HTML del modal es bastante sencillo, como puede verse en el código 2.4. Este elemento contiene mucho código Javascript para conseguir el efecto que se crea al pulsar el botón. En el código 2.5 se ve, por ejemplo, la función que mueve el elemento.

```

1 <link rel="stylesheet" href="{{asset('css/add-modal.css')}}">
2 <!-- Modal -->
3 <div id="modal" class="modal modal__bg" role="dialog" aria-hidden="true">
4   <div class="modal__dialog">
5     <div class="modal__content">
6       <h1>{{ $titulo }}</h1>
7       <!-- hay que pasarle al modal la ruta directamente -->
8       <form id='nuevoItem' action={{ $ruta }} method="post" class="modal-form">
9         @csrf
10        @foreach ($campos as $campo)
11          <label class="modal-form">{{ $campo }}: <input class="modal-label" name
12 ={{ $campo }} /></label>
13        @endforeach
14        <button onclick='form_submit()' id='btnNew' class='confirmarBtn' type="
15 submit" name="btnNuevo">Confirmar</button>
16      </form>
17
18      <!-- modal close button -->
19      <a href="" class="modal__close demo-close">
20        <svg class="" viewBox="0 0 24 24"><path d="M19 6.41l-1.41-1.41-5.59
21 5.59-5.59-5.59-1.41 1.41 5.59 5.59-5.59 5.59 1.41 1.41 5.59-5.59 5.59
22 1.41-1.41-5.59-5.59z"/><path d="M0 0h24v24z" fill="none"/></svg>
23      </a>
24
25    </div>
26  </div>
27 </div>

```

Código 2.4: Modal

```

1 var moveTrig = function(trig, modal, div) {
2     var trigProps = trig.getBoundingClientRect();
3     var m = modal;
4     var mProps = m.querySelector('.modal__content').getBoundingClientRect();
5     var transX, transY, scaleX, scaleY;
6     var xc = w.innerWidth / 2;
7     var yc = w.innerHeight / 2;
8
9     // this class increases z-index value so the button goes overtop the other
10    buttons
11    trig.classList.add('modal__trigger--active');
12
13    // these values are used for scale the temporary div to the same size as the
14    modal
15    scaleX = mProps.width / trigProps.width;
16    scaleY = mProps.height / trigProps.height;
17
18    scaleX = scaleX.toFixed(3); // round to 3 decimal places
19    scaleY = scaleY.toFixed(3);
20
21    // these values are used to move the button to the center of the window
22    transX = Math.round(xc - trigProps.left - trigProps.width / 2);
23    transY = Math.round(yc - trigProps.top - trigProps.height / 2);
24
25    // if the modal is aligned to the top then move the button to the center-
26    y of the modal instead of the window
27    if (m.classList.contains('modal--align-top')) {
28        transY = Math.round(mProps.height / 2 + mProps.top - trigProps.top -
29        trigProps.height / 2);
30    }
31
32    // translate button to center of screen
33    trig.style.transform = 'translate(' + transX + 'px, ' + transY + 'px)';
34    trig.style.webkitTransform = 'translate(' + transX + 'px, ' + transY + 'px)';
35    // expand temporary div to the same size as the modal
36    div.style.transform = 'scale(' + scaleX + ', ' + scaleY + ')';
37    div.style.webkitTransform = 'scale(' + scaleX + ', ' + scaleY + ')';
38
39    window.setTimeout(function() {
40        window.requestAnimationFrame(function() {
41            open(m, div);
42        });
43    }, contentDelay);
44 };

```

Código 2.5: Javascript del modal

2.4. Aplicación móvil

Al igual que en la parte web, lo primero ha sido implementar el sistema de login, haciendo la respectiva llamada a la API del backend y comprobar que la comunicación era correcta.

Como bien se ha dicho, Laravel brinda la facilidad de especificar rutas, no solo para la web sino también para aplicaciones externas de una forma cómoda y limpia. En el código 2.6 se puede ver la lista de rutas usadas por la aplicación móvil.

```
1 Route::get('/user', [UserController::class, 'index']);
2 Route::get('/login', [UserController::class, 'validateLogin']);
3 Route::post('/register', [LoginController::class, 'registerStudent']);
4 Route::get('/asignaturas', [AsignaturasController::class, 'index']);
5 Route::get('/asignaturas-buscar', [AsignaturasController::class, 'search']);
6 Route::get('/asignaturas-ingresar', [AsignaturasController::class, 'access']);
7 Route::get('/preguntas', [AsignaturaController::class, 'index']);
8 Route::get('/estadisticas', [RespuestasController::class, 'statistics']);
9 Route::get('/respuesta', [RespuestasController::class, 'index']);
10 Route::post('/respuesta', [RespuestasController::class, 'update']);
```

Código 2.6: Rutas para la API definidas en el backend

En el código 2.7 se observa cómo se declara cada servicio que se quiere consumir desde la app, especificando qué argumentos se le debe pasar.

```
1 public interface ApiService {
2
3     @GET("login")
4     Call<LoginResponse> getLogin(
5         @Query("username") String username, //argumentos que le pasas a la api
6         @Query("password") String password
7     );
8
9     @POST("register")
10    Call<StandardResponse> register(
11        @Query("email") String email,
12        @Query("password") String password
13    );
14
15    @GET("asignaturas")
16    Call<List<AsignaturasResponse>> getAsignaturas(
17        @Query("user") String user
18    );
19
20    @GET("preguntas")
21    Call<List<TemasResponse>> getTemas(
22        @Query("idAsignatura") String idAsignatura
23    );
24
25    @GET("estadisticas")
26    Call<EstadisticasResponse> getEstadisticas(
27        @Query("idPregunta") int idPregunta
28    );
29
30    @GET("respuesta")
31    Call<RespuestaResponse> getRespuesta(
32        @Query("idPregunta") int idPregunta,
```

```

33     @Query("user") String user
34 );
35
36 @FormUrlEncoded
37 @POST("respuesta")
38 Call<RespuestaResponse> setRespuesta(
39     @Field("idPregunta") int idPregunta,
40     @Field("user") String user,
41     @Field("respuesta") boolean respuesta
42 );
43
44 @GET("asignaturas-buscar")
45 Call<ArrayList<AsignaturasResponse>> getAsignaturasBuscar(
46     @Query("asignatura") String asignatura,
47     @Query("modo") int modo,
48     @Query("user") String user
49 );
50
51 @GET("asignaturas-ingresar")
52 Call<AsignaturasResponse> asignaturaIngresar(
53     @Query("asignatura") String asignatura,
54     @Query("user") String user
55 );
56 }

```

Código 2.7: Api Service en la app

Seguidamente, se implementaron las pantallas de asignaturas y temas, las cuales son muy parecidas. En ambas se debe hacer uso de un elemento llamado 'recyclerView', es un tipo de lista que se rellena dinámicamente según los elementos a mostrar. En la figura 2.15 se observa que esta vista está compuesta por el 'recyclerView' antes mencionado, un 'textView' que tiene oculto el texto "Lista vacía" y que se muestra en el caso de no haber elementos y el botón con la imagen de una lupa que sirve para buscar nuevas asignaturas.

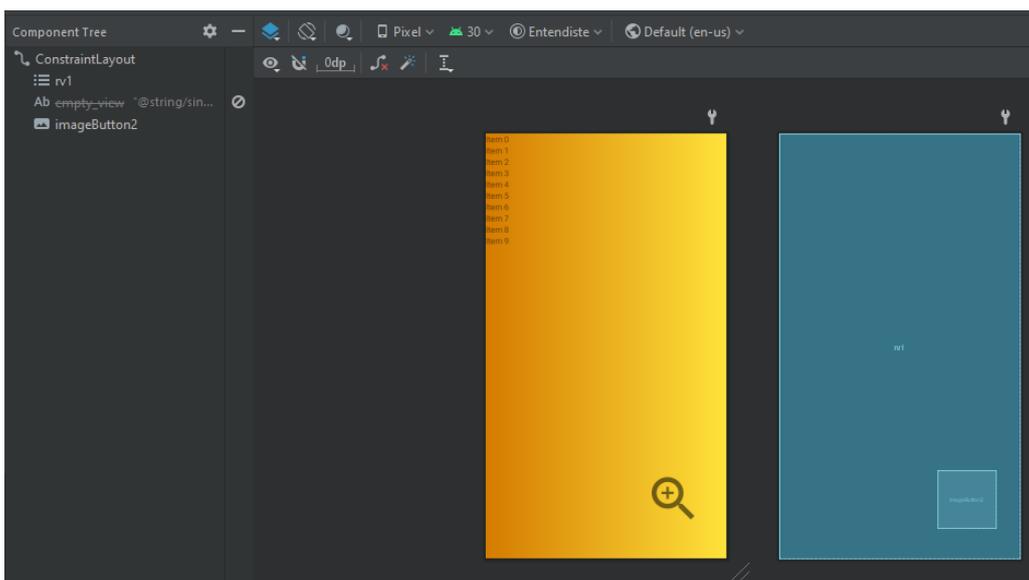


Figura 2.15: Lista dinámica en la pantalla de asignaturas de la app

En el código 2.8 puede verse la función que se utiliza para cerrar sesión en las opciones de la barra superior, accesible desde cualquier pantalla menos el login. Con la interfaz 'SharedPreferences' se gestionan las preferencias guardadas en el dispositivo. Puede apreciarse que, al cerrar sesión, se borran los datos de autenticación del usuario actual.

```
1 public void salir() {
2     SharedPreferences userpref = getSharedPreferences("datos", Context.MODE_PRIVATE);
3     SharedPreferences.Editor useredit = userpref.edit();
4     useredit.remove("user");
5     useredit.commit();
6
7     Intent principal = new Intent(this, MainActivity.class).addFlags(Intent.
8     FLAG_ACTIVITY_CLEAR_TOP | Intent.FLAG_ACTIVITY_SINGLE_TOP);
9     startActivity(principal);
10
11     Toast.makeText(this, "Hasta pronto", Toast.LENGTH_SHORT).show();
12     finish();
13 }
```

Código 2.8: Función para cerrar la sesión en la app

Y, por último, se realizó la pantalla de respuesta en cada tema.

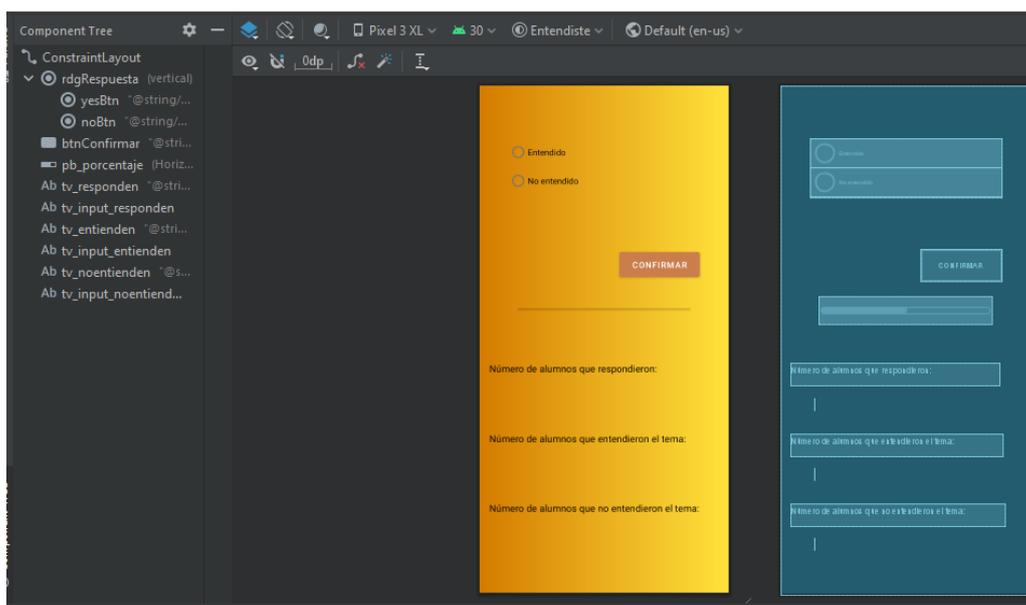


Figura 2.16: Pantalla para responder en la app

Esta pantalla está compuesta de varios 'textView' para mostrar cómo van los resultados hasta el momento, además de un 'progressBar', que sirve para mostrar el progreso de una forma más gráfica. En la parte superior, se encuentra el 'radioGroup', que sirve para marcar la respuesta y, más abajo, el botón para confirmar la respuesta seleccionada.

Un ejemplo de llamada a la API se puede observar en el código 2.9, donde se obtiene la respuesta seleccionada anteriormente, si la hubiera.

```

1 Call<RespuestaResponse> call = ApiAdapter.getApiService().getRespuesta(Integer.parseInt(
  idPregunta), user);
2     call.enqueue(new Callback<RespuestaResponse>() {
3         @Override
4         public void onResponse(Call<RespuestaResponse> call, Response<
  RespuestaResponse> response) {
5             RespuestaResponse respuesta = response.body();
6             if(!respuesta.getEmpty()) {
7                 if(respuesta.getRespuesta()) rdgRespuesta.check(R.id.yesBtn);
8                 else rdgRespuesta.check(R.id.noBtn);
9             }
10        }
11
12        @Override
13        public void onFailure(Call<RespuestaResponse> call, Throwable t) {
14            Toast.makeText(getApplicationContext(), "Fallo buscando la respuesta"
  , Toast.LENGTH_SHORT).show();
15        }
16    });

```

Código 2.9: Llamada a la API desde la app

Capítulo 3

Aplicación 'Entendiste'

- Gama de colores:

Los colores de la aplicación y de la web han sido escogidos en base a los colores de la Escuela Superior de Ingeniería y Tecnología (ESIT). Se ha seleccionado una paleta pública de la web colors.co[21] con colores que combinan entre sí[22], seleccionando solo los 4 primeros colores.



Figura 3.1: Paleta

- Logo:

El logo ha sido diseñado con la herramienta online Tailor Brands[23], un creador de logotipos en línea impulsado por IA. En concreto, se utilizan el imagotipo y el isotipo[24].



Figura 3.2: Isotipo de Entendiste



Figura 3.3: Imagotipo de Entendiste

- Esquema del sistema:

Este es un esquema que muestra la comunicación entre las distintas partes que conforman el sistema.

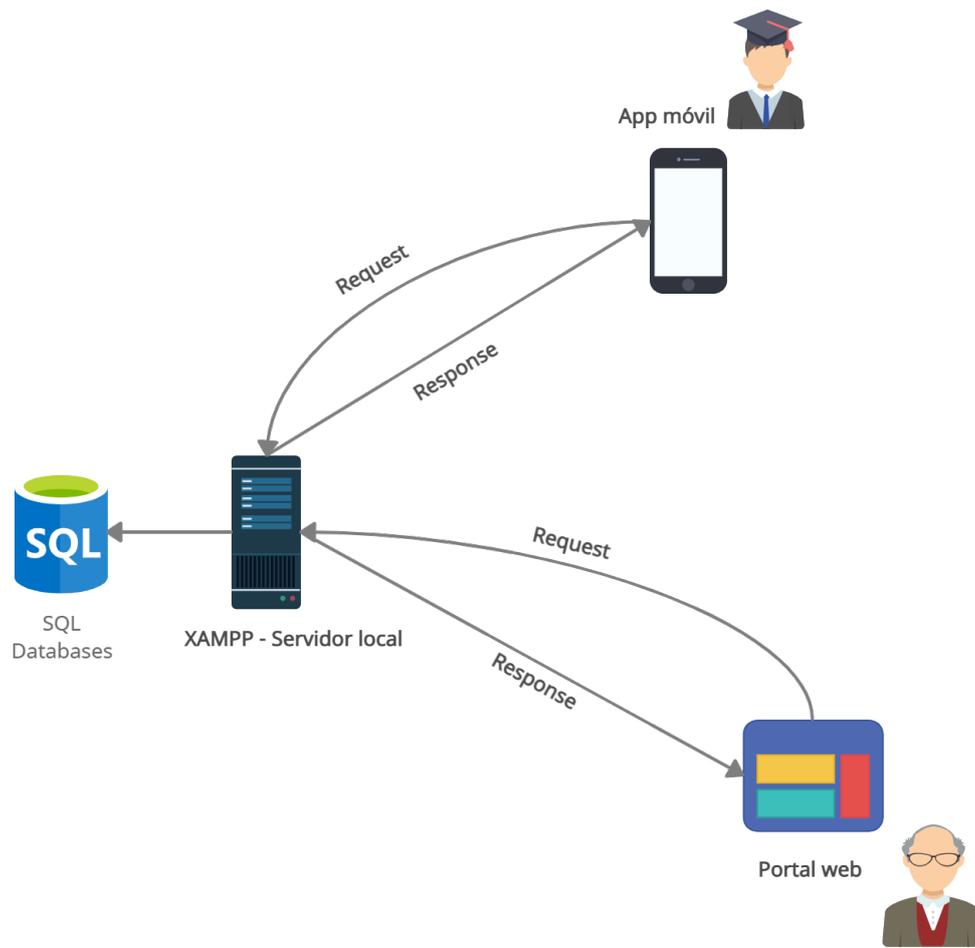
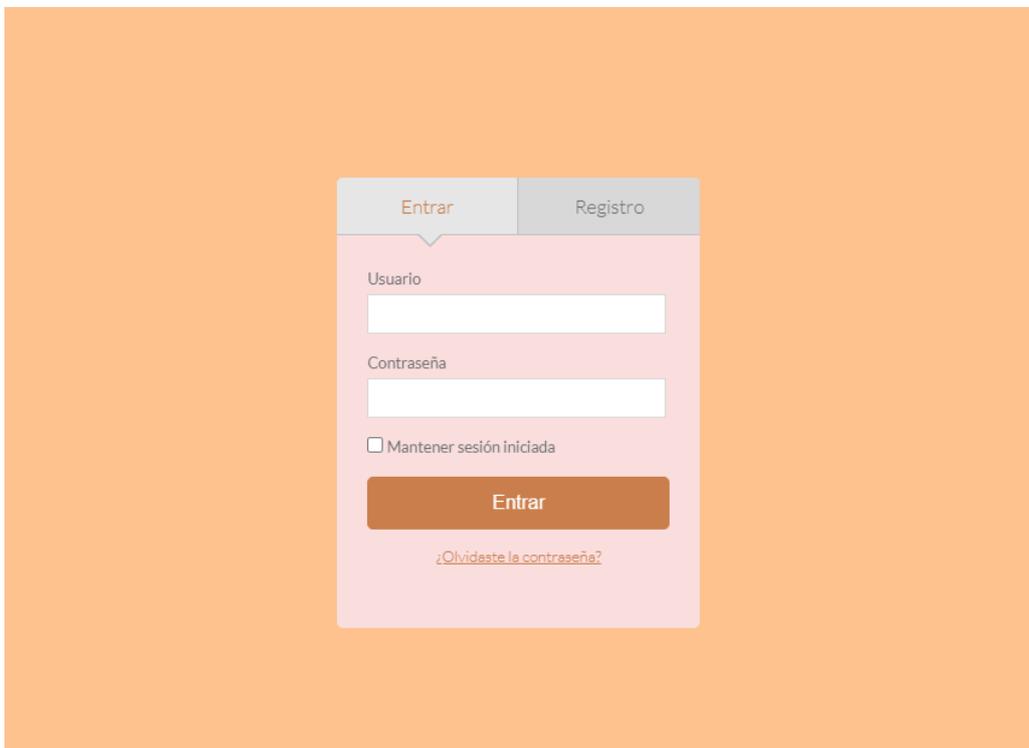


Figura 3.4: Esquema de componentes

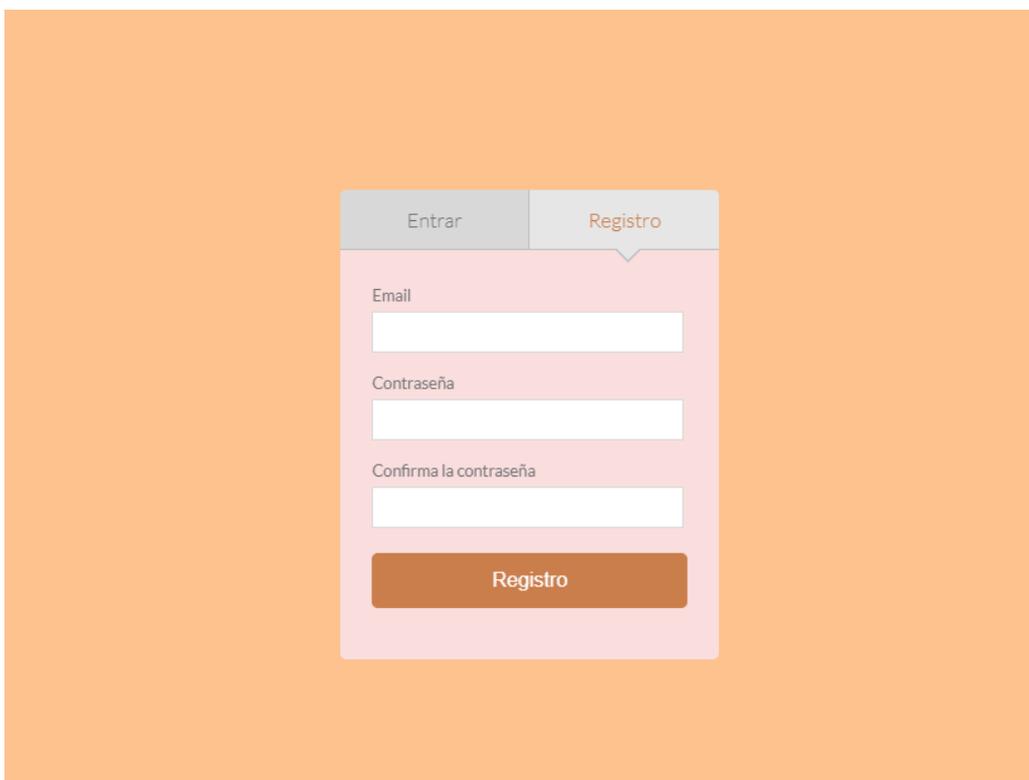
3.1. Uso de la página web (profesores)

La web cuenta con un sistema de login y con la posibilidad de registrarse usando un correo electrónico.



The image shows a login form on a website. At the top, there are two tabs: 'Entrar' (Login) and 'Registro' (Registration). The 'Entrar' tab is selected. Below the tabs, there are two input fields: 'Usuario' (Username) and 'Contraseña' (Password). Below the password field, there is a checkbox labeled 'Mantener sesión iniciada' (Keep session started). At the bottom of the form, there is a large orange button labeled 'Entrar' and a link that says '¿Olvidaste la contraseña?' (Forgot your password?).

Figura 3.5: Login en la web



The image shows a registration form on a website. At the top, there are two tabs: 'Entrar' (Login) and 'Registro' (Registration). The 'Registro' tab is selected. Below the tabs, there are three input fields: 'Email', 'Contraseña' (Password), and 'Confirma la contraseña' (Confirm password). At the bottom of the form, there is a large orange button labeled 'Registro'.

Figura 3.6: Registro en la web

El registro se completa cuando el usuario ha pulsado el enlace que previamente se le envía a su correo electrónico. Para ello, se crea una clave que se guarda en la base de datos, enviando la misma clave al email del usuario para compararlas una vez haya pulsado en el enlace.

Después de que el profesor se haya logeado, encontrará la página inicial con un mensaje informativo y un menú en la barra superior. En la parte superior de la figura 3.7 se puede observar que el submenú de asignaturas está desplegado. Se tiene la posibilidad de acceder a la lista de asignaturas o acceder a una asignatura concreta directamente. La lista de asignaturas que aparece en este submenú está ordenada en base a la fecha de modificación.



Figura 3.7: Pantalla de inicio de la web

Una vez situados en la pantalla de asignaturas, se tiene la opción de acceder a una asignatura o pulsar el icono con un '+' y dar de alta una nueva asignatura.

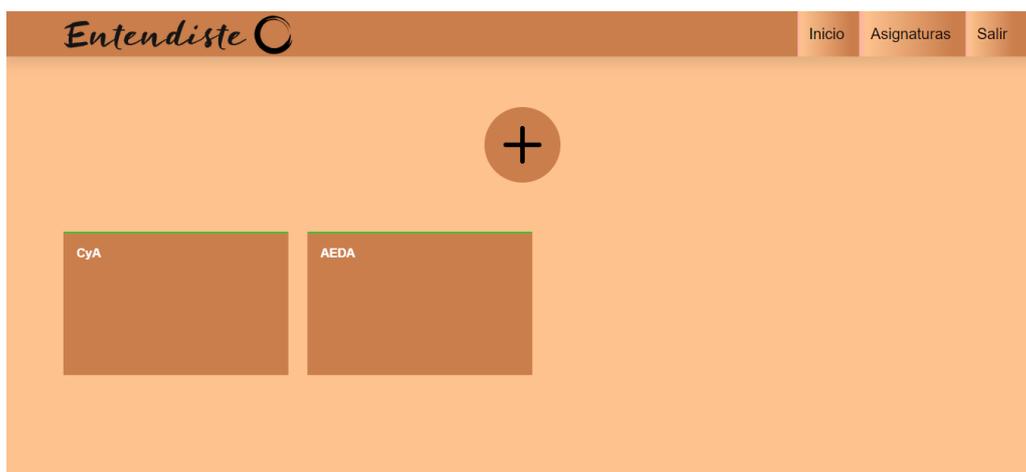


Figura 3.8: Pantalla de asignaturas de la web

En la figura 3.9 se puede ver el cuadro de dialogo que se abre al pulsar en el botón de añadir asignatura. En este formulario se deberá ingresar el código de la asignatura, el

nombre de la asignatura y la contraseña necesaria para que los alumnos puedan acceder a dicha asignatura.

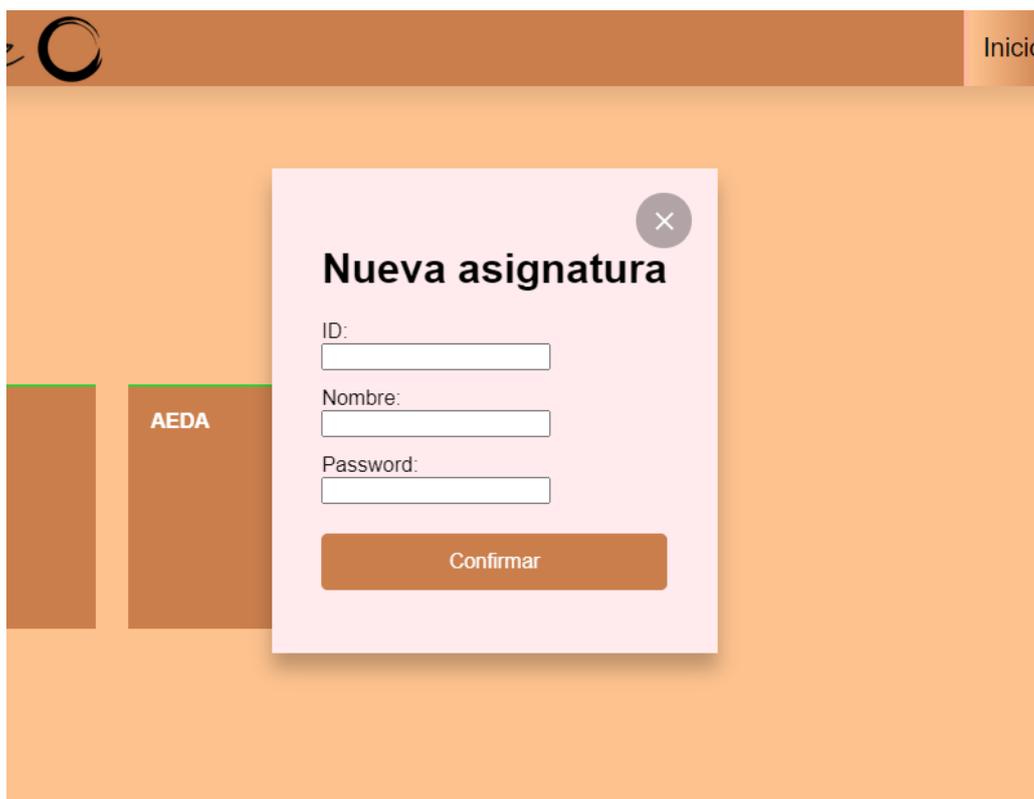
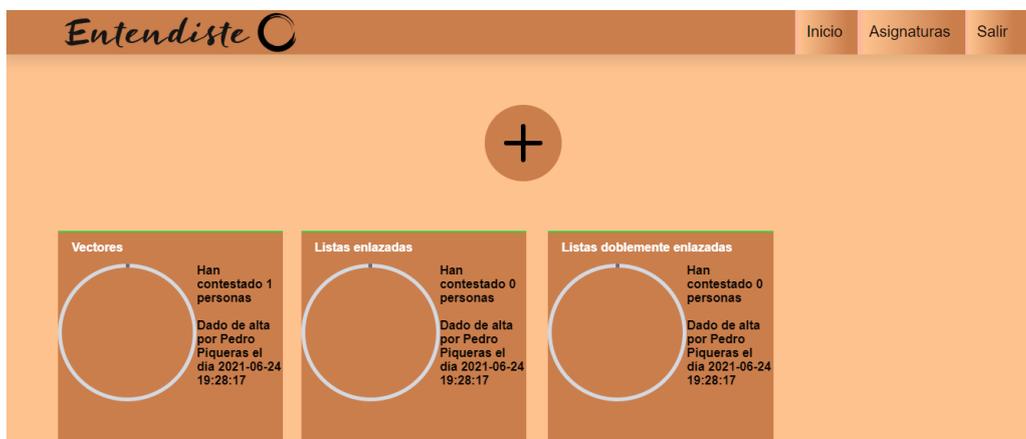


Figura 3.9: Dar de alta una asignatura

Cuando se accede a una asignatura, se despliega la lista de temas que hayan sido añadidos por los profesores de esa asignatura, como puede verse en la figura 3.10. Cada tema contiene información sobre las respuestas de los alumnos: un gráfico circular que muestra el porcentaje de alumnos que entendieron el tema, el número de alumnos que han contestado y el nombre del profesor que añadió ese tema junto a la fecha.



Topic	Answered	Created by	Date
Vectores	1 person	Pedro Piqueras	2021-06-24 19:28:17
Listas enlazadas	0 people	Pedro Piqueras	2021-06-24 19:28:17
Listas doblemente enlazadas	0 people	Pedro Piqueras	2021-06-24 19:28:17

Figura 3.10: Pantalla de temas de la web

Por último, se puede pulsar en el botón '+' para añadir un nuevo tema. Como puede verse en la figura 3.11, solo haría falta rellenar el título del tema o bloque de contenidos.

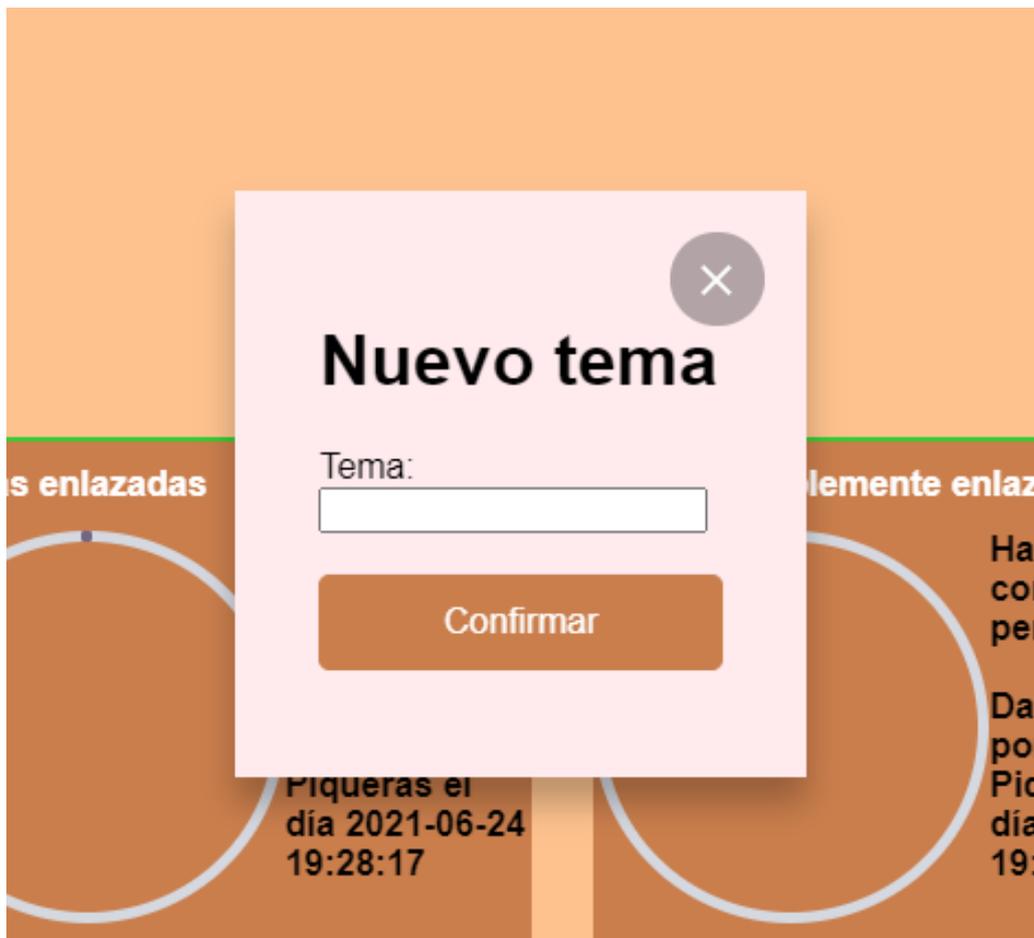


Figura 3.11: Añadir nuevo tema en la web

3.2. Uso de la aplicación móvil (alumnos)

En la aplicación móvil se encuentra una primera pantalla para acceder al login o directamente a la lista de asignaturas, en el caso de ya estar autenticado.



Figura 3.12: Pantalla de inicio de la app

La pantalla de login ofrece la opción de autenticación, usando el correo electrónico o registro.

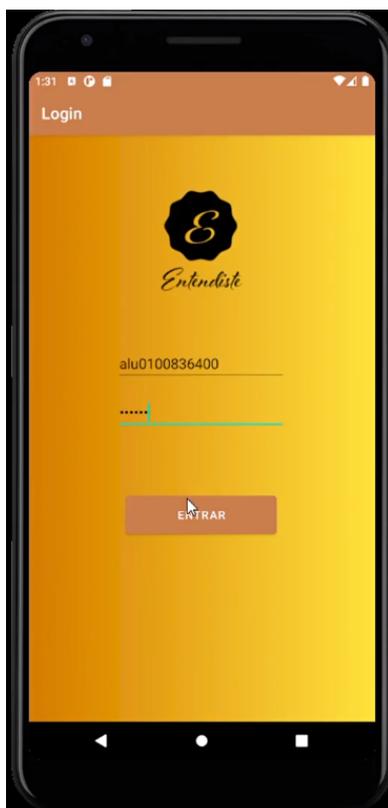


Figura 3.13: Pantalla de login de la app

En la pantalla de asignaturas se encuentra la lista de asignaturas a las que tenemos acceso y una lupa para poder buscar nuevas asignaturas. En la figura 3.14 se observa esta pantalla.



Figura 3.14: Pantalla de asignaturas de la app

La búsqueda de asignatura admite búsqueda por nombre y por código, no siendo necesario completar el nombre o el código.



Figura 3.15: Buscar una asignatura en la app



Figura 3.16: Resultado de la búsqueda en la app

Una vez encontrada la asignatura, se deberá ingresar la contraseña facilitada por el profesor.

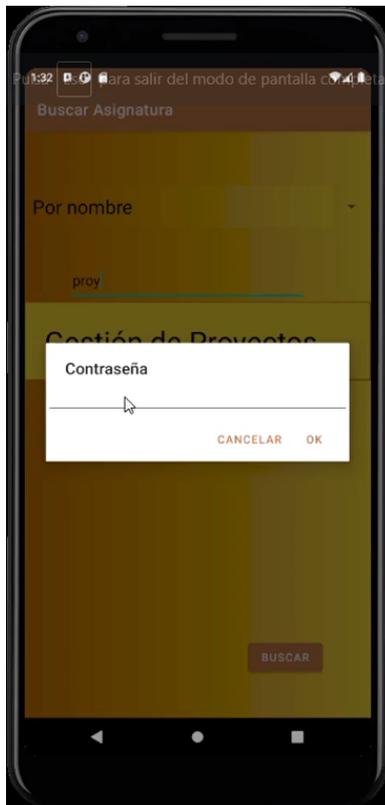


Figura 3.17: Pantalla para introducir contraseña en la app

Cuando se accede a una asignatura, se muestra la lista de temas, tal como se muestra en la figura 3.18.



Figura 3.18: Pantalla de temas en la app

Por último, accediendo a un tema, se abre la pantalla para seleccionar si se entiende o no el tema. Además, se muestran las estadísticas de ese tema en la actualidad.



Figura 3.19: Pantalla de respuesta en la app

Capítulo 4

Conclusiones y líneas futuras

4.1. Conclusiones

Con este trabajo se ha conseguido crear una aplicación que podría ayudar a mejorar la comunicación entre profesorado y alumnado y que puede ser útil para intentar garantizar un progreso en el rendimiento académico en las aulas.

Este Trabajo de Fin de Grado también ha servido para aprender sobre tecnologías web actuales, además de reforzar los conocimientos en distintos lenguajes, así como de la creación de aplicaciones móviles, dos ámbitos en auge que, sin duda, lo seguirán siendo por mucho tiempo.

Se espera que este producto siga creciendo y mejorando, y sea llevado, no solo a las aulas de la Universidad de La Laguna, sino que pueda promoverse su uso por diferentes centros educativos españoles.

4.2. Líneas futuras

Una vez concluido el proyecto, cabe mirar hacia el futuro y preguntarse cómo mejorarlo y ampliar la ambición por obtener un producto superior.

Podría ser interesante poder ahorrarle el trabajo de registro a todos los usuarios importando los datos de acceso de las cuentas de la Universidad de La Laguna, teniendo en cuenta que son los usuarios finales del sistema. Otra forma de cumplir con esto sería ceder el trabajo de autenticación a la página de la Universidad de La Laguna por medio de una redirección.

Al igual que con la autenticación, se podría importar la lista de asignaturas de la universidad, incluyendo los códigos de asignatura y la misma contraseña utilizada para acceder a dicha asignatura en el Moodle[25].

Se podría implementar un ordenamiento de asignaturas y temas siguiendo diferentes criterios: fecha de creación, fecha de última modificación, más o menos tasa de alumnos que entendieron, alfabéticamente...

Poder ingresar temas con un estado de "no publicado" y ajustando una fecha concreta para su publicación automática.

Envío automático de correos electrónicos a los alumnos que no entendieron un tema con actividades o contenido de refuerzo.

Implementar la versión de la aplicación móvil en iOS, para así cubrir prácticamente el 100 % de los dispositivos móviles actuales.

Por último, si se quiere llegar más lejos y ampliar el alcance del sistema, cabría incorporar distintos tipos de encuestas, añadir niveles de entendimiento en vez de limitarlo a un sí o un no, mejorar el apartado de estadísticas y facilitar comparativas entre alumnos, entre turnos de mañana y tarde o entre un año y otro.

Capítulo 5

Summary and Conclusions

5.1. Conclusions

With this work, it has been possible to create an application that could help to improve communication between teachers and students while also ensuring students' progress regarding academic performance in the classroom.

This End-of-Degree Project has also contributed to learn about current web technologies, in addition to reinforcing knowledge in different languages, as well as the creation of mobile applications. These two areas will undoubtedly continue their outstanding development in the forthcoming decades.

It is expected that this product will be further developed and optimized, and it will not only be implemented in the classrooms of the University of La Laguna, but also in different Spanish educational centres.

5.2. Future lines

Once the project is completed, it would be inspiring to look to the future and ask ourselves how to improve it and expand our ambition for a superior product.

It could be interesting to be able to save the registration work for all users by importing the access data of the accounts of the University of La Laguna, considering that they are the end-users of the system. Another way to achieve this would be to assign the authentication work to the page of the University of La Laguna by means of a redirect.

As with authentication, the list of university subjects could be imported, including the subject codes and the same password used to access the subject in Moodle[25].

An ordering of subjects and topics could be implemented following different criteria: date of creation, date of last modification, rate of students who understood, alphabetical order. . .

Besides, it is also possible to insert topics with a status of "not published" and set a specific date for their automatic publication or the automatic sending of emails to students who did not understand a topic with activities or reinforcement content.

To promote its use and acceptance, implementing the version of the mobile application in iOS would be a suitable option since this would help to cover practically 100 % of current mobile devices.

Last but not least, in order to enhance the system and its range, it's possible to add different types of surveys. Not only enabling the option "yes" or "no" but also allowing to add elaborated answers, which improve the heuristic aspect and facilitate comparisons between the students from the morning and evening class, or from different years.

Capítulo 6

Presupuesto

Descripción	Cantidad	Precio unitario	Total
Mano de obra			
Programador junior	150h	15€/h	2250€
Software			
Visual Studio Code	N/A	0€	0€
Android Studio	N/A	0€	0€
Hardware			
Alquiler PC	1 mes	45€/mes	45€
Total			2295€

Tabla 6.1: Presupuesto

Bibliografía

- [1] Marta Seror. Vergüenza en las aulas. <https://www.mheducation.es/blog/verguenza-en-el-aula>, 2018. McGraw Hill Education.
- [2] Israel Rubio. La "generación muda". https://elpais.com/tecnologia/2019/04/02/actualidad/1554220116_828140.html, 2019. El País.
- [3] APIs. <https://www.redhat.com/es/topics/api/what-are-application-programming-inte>
- [4] Javascript. <https://www.javascript.com/>.
- [5] Express.js. <https://expressjs.com/es/>.
- [6] Node.js. <https://nodejs.org/es/>.
- [7] MySQL. <https://www.mysql.com/>.
- [8] Python. <https://www.python.org/>.
- [9] Ruby. <https://www.ruby-lang.org/es/>.
- [10] PHP. <https://www.php.net/>.
- [11] ASP.NET. <https://dotnet.microsoft.com/apps/aspnet>.
- [12] Windows Server. <https://www.microsoft.com/es-es/windows-server>.
- [13] Angular. <https://angular.io/>.
- [14] React. <https://es.reactjs.org/>.
- [15] Composer. <https://getcomposer.org/>.
- [16] iOS. <https://www.apple.com/es/ios/ios-14/>.
- [17] Android Studio. <https://developer.android.com/studio>.
- [18] Proto.io. <https://proto.io/>.
- [19] Visual Paradigm. <https://www.visual-paradigm.com/>.
- [20] Faker. <https://github.com/fzaninotto/Faker>.
- [21] colors.co. <https://colors.co>.
- [22] Mi paleta de colors.co. <https://colors.co/ca7e4c-fdc28e-fadede-feb6a0-a6c8eb-71678>
- [23] Tailor Brands. <https://www.tailorbrands.com/>.

[24] Diferencia entre logotipo, isotipo, imagotipo e isologo. <https://baetica.com/logotipo-isotipo-imagotipo-e-isologo-se-diferencian>.

[25] Moodle. <https://moodle.org/>.