



SOLVAM App

Autor: Jesús I. Mengibar Delom

Titulación: Grado Superior

Especialidad: Desarrollo de Aplicaciones Web

Índice

1. Introducción	4
1.1 Justificación	4
2. Objetivos	5
2.1 Objetivos del proyecto	5
2.2 Objetivos docentes	5
2.3 Metodología	6
2.3.1 Fases del proyecto	7
2.3.1.1 Fase de diseño	7
2.3.1.2 Selección de hardware	9
2.3.1.3 Estructura de navegación	10
2.3.1.4 Implementación	10
2.3.1.4.1 Lenguaje de programación y Tecnologías	10
2.3.1.4.1.1 HTML	10
2.3.1.4.1.2 SCSS	11
2.3.1.4.1.3 TypeScript	11
2.3.1.4.1.4 PHP	11
2.3.1.4.1.5 MySQL	12
2.3.1.5 Frameworks	12
2.3.1.5.1 Ionic	12
2.3.1.5.2 AngularJS	12
2.4 Componentes y servicios	13
2.4.1 Componentes	13
2.4.1.1 Toast	13
2.4.1.2 Loader	13
2.4.1.3 Alert	14
2.4.1.4 HTTPRequest	14
2.4.1.5 Router	15
2.4.1.6 ActivatedRoute	15
2.4.1.7 Menú	16
2.4.1.8 PapaParse	17
2.4.1.9 EmailComposer	17
2.4.2 Servicios	18
3. Análisis del contexto	19
4. Desarrollo del contenido	21
4.1 Perfiles de usuarios	21
4.1.1 Usuario administrador	21
4.1.2 Usuario no administrador	21
4.2 Documentación de las tareas	22

4.3 Usabilidad	23
5. Conclusión y valoración personal	24
5.1 Conclusión	24
5.2 Valoración personal	25
6. Webgrafía	26

Índice de figuras

Figura 1: Diseño final login	7
Figura 2: Diseño final Horarios y sesiones	7
Figura 3: Diseño final Perfil	8
Figura 4: Diseño final Mensajes	8
Figura 5: Gráfico Android vs iOS	9
Figura 6: Estructura de navegación	10
Figura 7: Ejemplo Toast cambio realizado	13
Figura 8: Ejemplo de Toast de error	13
Figura 9: Ejemplo de Loader	13
Figura 10: Ejemplo de Alert	14
Figura 11: Ejemplo de GET en HTTPRequest	14
Figura 12: Ejemplo de POST en HTTPRequest	14
Figura 13: Ejemplo de recogida de datos después de un envío POST de HTTPRequest en PHP	15
Figura 14: Ejemplo de ir a otra página con Router	15
Figura 15: Ejemplo de ir y enviar datos por URL a otra página con Router	15
Figura 16: Ejemplo de recogida de datos por URL con ActivatedRoute	15
Figura 17: Ejemplo de ion-menu	16
Figura 18: Ejemplo de PapaParse	17
Figura 19: Ejemplo de EmailComposer	17
Figura 20: Grafica población española	19

1. Introducción

En la actualidad las aplicaciones constituyen una herramienta bastante útil para la mejora de rendimiento de toda organización empresarial. Del mismo modo, no solo se trata de adquirir tecnología, sino que también es necesario saber darle el uso adecuado de acuerdo a los requisitos particulares de un determinado usuario o grupos de usuarios.

La realización de este trabajo de fin de grado, pretende plasmar los conocimientos adquiridos en los dos años de estudios en Desarrollo de Aplicaciones Web, así como tratar de explorar y añadir nuevos conocimientos.

La idea de este TFG es crear una aplicación híbrida que permita la gestión, en parte, del Centro de FP SOLVAM realizando un análisis del alumnado y profesorado. Se abordará tanto la parte teórica, sobre lo que consiste en realizar una aplicación y la realización de un estudio en profundidad sobre la misma, hasta la parte práctica, que trata sobre la gestión de profesorado y alumnado.

Tanto el nombre de la empresa como los datos personales y técnicos son marca registrada.

1.1 Justificación

Este trabajo surge principalmente del interés de la dirección del Centro de FP SOLVAM en mejorar la gestión del alumnado y profesorado, ya que desde su implantación, ha supuesto un reto para todas las partes implicadas. Asimismo, cuenta con el apoyo de los docentes, quienes han propuesto este trabajo, sino también del profesorado que ha participado activamente y ha sido de gran ayuda.

2. Objetivos

Este proyecto se enmarca dentro de una iniciativa del Centro de FP SOLVAM para lograr una gestión más eficiente. En este punto, se indicarán los objetivos principales que se presentan para la realización de este TFG.

2.1 Objetivos del proyecto

En este trabajo se encarga la realización de un sistema que permita gestionar los departamentos en los que se realizará la implementación de la aplicación: IT y Administración, entre otros.

Para ello, los objetivos que se basan en el presente trabajo son los siguientes:

- Realizar una aplicación híbrida para la recogida de faltas del alumnado, que además, incorpore funciones para la gestión de la misma.
- Recopilación y el estudio de investigaciones similares llevadas a cabo, así como de la bibliografía disponible sobre el tema.
- Redacción de la memoria para el uso de la aplicación.

2.2 Objetivos docentes

Con la realización de este TFG también se pretenden alcanzar unos objetivos académicos. En mi caso, estos objetivos son los siguientes:

- Afianzar mis conocimientos en el lenguaje de programación Ionic.
- Ampliar los conocimientos de JavaScript y HTML haciendo uso de hoja de estilos
- Aprender el funcionamiento y programación de Android y emplearlo para la creación de la aplicación móvil.

2.3 Metodología

Como toda investigación, el punto de partida ha sido realizar un exhaustivo análisis del centro educativo en cuestión. En este caso se ha analizado en primer lugar el diseño de la aplicación para que sea fácil de entender por los usuarios. También se han analizado la selección de hardware y software para adaptarse a las necesidades del consumidor, así como la estructura de navegación de la aplicación para que sea intuitiva. Por tanto, se ha comenzado con lo inmediatamente más cercano al centro educativo, para luego ir ampliando el abanico.

La aplicación web a desarrollar estará enfocada a las diferentes partes que intervienen en el centro educativo, en las que se distinguen **dos roles:** usuario y usuario administrador. Este último actuará como administrador del sistema. Los dos serán, por tanto, los actores que se distinguirán en el sistema.

Para poder hacer uso de la aplicación será necesario estar registrado en la misma. Para ello, existirán dos formularios: uno de registro y otro de login; el registro de los tutores correrá a cargo del usuario administrador.

Los datos con los que se trabajarán en la aplicación serán los siguientes:

- Estudiantes: NIA, nombre, apellidos, curso, teléfono móvil del alumno, teléfono móvil del padre y madre, correo electrónico del alumno y correo electrónico de los padres.
- Usuario: Nombre, contraseña, tipo, tutor, curso.

El usuario administrador podrá ser, a su vez profesor; por tanto, los datos del usuario administrador serán los mismos que los de un profesor.

2.3.1 Fases del proyecto

2.3.1.1 Fase de diseño

Esta fase es de las más complejas del proyecto debido al extenso trabajo realizado y los detalles visuales necesarios para abarcar la interacción entre elementos, también se ha de tener en cuenta la estructura lógica del propio sistema. El diseño ha sido una de las cosas que más me ha costado debido a que he tenido que tener en cuenta las necesidades de los profesores del centro, requisitos que muchas veces se han topado con las limitaciones de la lógica de la aplicación. Aún con todo lo anteriormente mencionado el boceto final creo resuelve prácticamente todas las exigencias propuestas por los usuarios del centro.



Figura 1: Diseño final login



Figura 2: Diseño final Horarios y sesiones

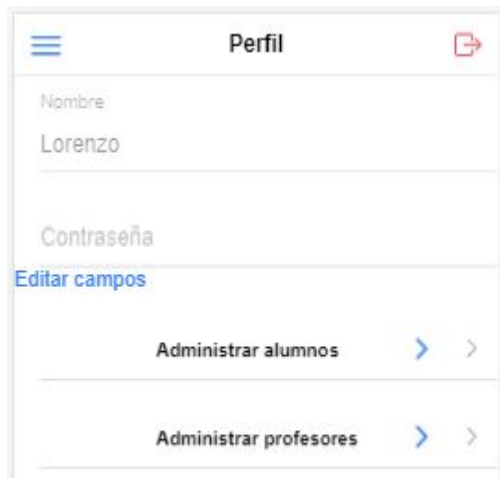


Figura 3: Diseño final Perfil



Figura 4: Diseño final Mensajes

2.3.1.2 Selección de hardware

La aplicación está destinada a trabajar para los distintos dispositivos habidos en el mercado actual, tales como: móviles, tablets, ordenadores personales, etc. debido al diseño adaptativo (Responsive Design) del sistema. Aunque su uso mayoritariamente será para teléfonos móviles, ya tengan como sistema operativo Android o iOS.

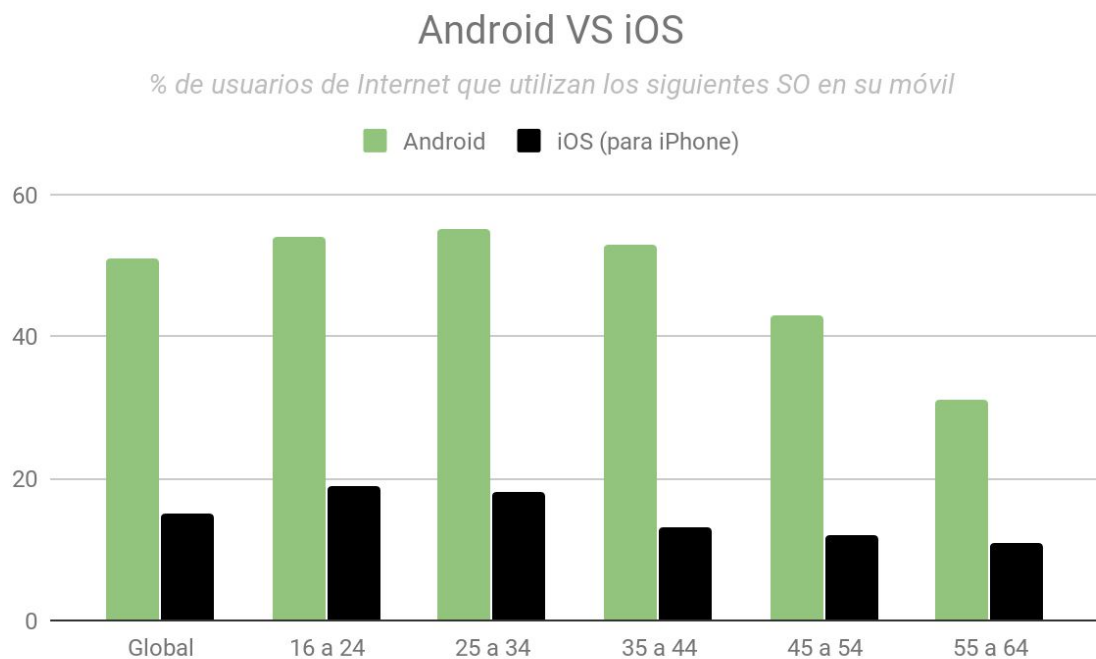


Figura 5: Gráfico Android vs iOS

Globalmente el mercado de dispositivos está liderado por el sistema operativo Android, no obstante en el Centro de FP SOLVAM la gran mayoría de usuarios de la aplicación utilizan dispositivos con iOS, y por lo tanto una de mis metas a conseguir ha sido que la implementación de la misma fuera a través de un framework como Ionic, ya que puede ser utilizado para desarrollar aplicaciones híbridas en ambos sistemas.

2.3.1.3 Estructura de navegación

Al estar la aplicación programada conjuntamente con los frameworks Ionic y AngularJS la estructura de la misma tiene un tipo de organización de navegación entre las páginas de tipo árbol o de tipo jerarquía, en la que el usuario siempre está ubicado y tiene la facilidad de moverse libremente todo gracias a que la mayoría de las páginas secundarias permiten regresar a la página de inicio.



Figura 6: Estructura de navegación

2.3.1.4 Implementación

Durante la fase de implementación, la aplicación multiplataforma ha sido desarrollada utilizando las tecnologías apropiadas y necesarias teniendo en cuenta el diseño.

2.3.1.4.1 Lenguaje de programación y Tecnologías

2.3.1.4.1.1 HTML

HyperText Markup Language (HTML), es un lenguaje de marcación de elementos amplio utilizado para la creación de elementos. Es altamente soportado por los navegadores, y es utilizado para la creación y presentación visual del contenido de una página web. La función principal es describir la estructura de la página, pero no se encarga ni de la apariencia ni de su funcionalidad.

HTML define los elementos que compondrán una página web, como por ejemplo imágenes, listas, vídeos, enlaces, formularios, mediante el uso de etiquetas o marcas (tags).

2.3.1.4.1.2 SCSS

SCSS es utilizado para renderizar estilos en la aplicación, le da un estilo más personal que las hojas de estilo convencionales, conocidas como CSS. Convierte todos los estilos descritos en el archivo SCSS a tecnología CSS que es la que actualmente entienden casi todos los navegadores, para convertir este tipo de archivos se hace uso de SASS que es conocido por ser el preprocesador de archivos SCSS.

2.3.1.4.1.3 TypeScript

TypeScript es un lenguaje de programación parecido a JavaScript que presenta un fuerte tipado en la manera de programarlo.

El tipado de una variable es la manera que tiene el lenguaje de determinar si es de un tipo u otro, o para determinar el tipo de una clase.

2.3.1.4.1.4 PHP

Hypertext Preprocessor (PHP) es un lenguaje de programación de código abierto muy popular especialmente adecuado para el desarrollo web y que puede ser incrustado en HTML, es utilizado generalmente en el lado de servidor, para el desarrollo de una página web dinámica.

El código es interpretado en el servidor, generando HTML y enviándolo al cliente. El cliente recibirá el resultado de ejecutar el script, aunque no se sabrá el código subyacente que era.

Permite un acceso simple a las bases de datos mediante sus librerías propias para manipular los datos, con lo cual, es muy adecuada para la aplicación a desarrollar.

2.3.1.4.1.5 MySQL

MySQL es el sistema de gestión de bases de datos relacional más extendido en la actualidad al estar basada en código abierto. Sirve para almacenar toda la información que se desee en bases de datos relacionales, como también para administrar todos estos datos sin apenas complicaciones gracias a su interfaz visual y a todas las opciones y herramientas de las que dispone. Es algo esencial, sobre todo en webs que cuentan con la opción de registrar usuarios para que inicien sesión.

2.3.1.5 Frameworks

En esta sección, se menciona los frameworks utilizados para facilitar el desarrollo de la aplicación.

2.3.1.5.1 Ionic

Ionic es una estructura tecnológica de código abierto que es utilizada en el desarrollo de aplicaciones móviles híbridas, combina HTML5, CSS y JavaScript dando como resultado aplicaciones con una interfaz amigable e intuitiva para el usuario que luego se comercializan o descargan en plataformas como Android o iOS.

2.3.1.5.2 AngularJS

AngularJS es un framework de código abierto desarrollado por Google, se utiliza popularmente durante los últimos años para crear aplicaciones web de una sola página (SPA).

Este framework es modular, por lo tanto es fácil dividir una aplicación en diferentes módulos y gestionar las relaciones entre las vistas, controladores y modelo.

2.4 Componentes y servicios

Para la gestión del alumnado, profesorado y mensajería del centro se han usado los siguientes componentes:

2.4.1 Componentes

2.4.1.1 Toast

Toast avisa al usuario de cualquier cambio u error realizado en la parte inferior del dispositivo; si el cambio ha sido realizado se mostrará un toast de color verde y si ha habido un error el toast se mostrará de color rojo.

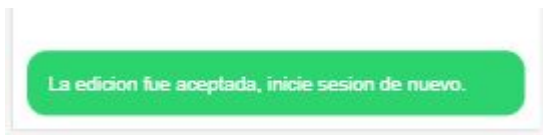


Figura 7: Ejemplo Toast cambio realizado



Figura 8: Ejemplo de Toast de error

2.4.1.2 Loader

Loader sirve para cargar los cambios realizados.

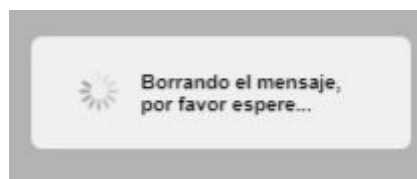


Figura 9: Ejemplo de Loader

2.4.1.3 Alert

Alert envía un aviso si el usuario va a realizar un cambio bastante grande, antes de hacer el cambio le pedirá al usuario la confirmación del mismo.

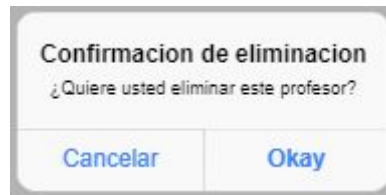


Figura 10: Ejemplo de Alert

2.4.1.4 HTTPRequest

HTTPRequest es un componente encargado del envío y de la recogida de datos vía JSON, por medio de ficheros PHP.

```
this.http.get('https://www.solvam.es/app/php/db_select_alumnos.php', {observe: 'body'})
.subscribe((res: Response) => {
  localStorage.setItem("Alumnos", JSON.stringify(res));
}, (err)=>{
  console.log("Error!");
});
```

Figura 11: Ejemplo de GET en HTTPRequest

```
this.http.post('https://www.solvam.es/app/php/db_select_alumnos_curso.php', JSON.stringify(json), {observe: 'body'})
.subscribe((res: Response) => {
  localStorage.setItem("Alumnos", JSON.stringify(res));
}, (err)=>{
  console.log("Error!");
});
```

Figura 12: Ejemplo de POST en HTTPRequest

```
$recogidaDatos = file_get_contents("php://input");  
$request = json_decode($recogidaDatos);
```

Figura 13: Ejemplo de recogida de datos después de un envío POST de HTTPRequest en PHP

2.4.1.5 Router

Router es utilizado para ir de una página a otra de la aplicación o para pasar los datos entre las páginas.

```
this.router.navigateByUrl('/mensajes');
```

Figura 14: Ejemplo de ir a otra página con Router

```
this.route.navigate(['/admin-profesor', { userType: this.userType }]);
```

Figura 15: Ejemplo de ir y enviar datos por URL a otra página con Router

2.4.1.6 ActivatedRoute

ActivatedRoute es utilizado sobretodo para la recogida de variables mediante direcciones URL entre las páginas de la aplicación.

```
this.userType = this.activeRoute.snapshot.paramMap.get('userType');
```

Figura 16: Ejemplo de recogida de datos por URL con ActivatedRoute

2.4.1.7 Menú

El componente Menú es un cajón de navegación que se desplaza desde el lado izquierdo de la vista actual; por defecto, se desplaza desde el lado izquierdo, pero el desplazamiento ya sea del lado izquierdo o derecho puede ser configurado.

El menú se mostrará de forma diferente según el tipo, sin embargo, la manera de visualización puede cambiarse a cualquiera de los de menú disponibles que se encuentran en la página de [ion-menu](#). Un mismo menú puede contener un número ilimitado de menús, estos se pueden manejar desde las plantillas o mediante **“MenuController”**.



Figura 17: Ejemplo de ion-menu

2.4.1.8 PapaParse

PapaParse es un componente que realiza la subida de archivos a la aplicación y los convierte a un array de JSON para que más tarde pueda ser leído.

```
this.papa.parse(csvData, {  
  complete: parsedData => {  
    this.headerRow = parsedData.data.splice(0, 1)[0];  
    this.csvData = parsedData.data;  
  }  
});
```

Figura 18: Ejemplo de PapaParse

2.4.1.9 EmailComposer

EmailComposer es un componente que permite enviar un correo electrónico, con formato HTML, a un cliente de correo para ser enviado.

```
this.emailComposer.addAlias('gmail', 'com.google.android.gm');  
let email = {  
  app: 'gmail',  
  to: this.alumno.CorreoA,  
  cc: '',  
  bcc: [],  
  attachments: [],  
  subject: 'Faltas de asistencia',  
  body: this.textMess,  
  isHtml: true  
}  
this.emailComposer.open(email);
```

Figura 19: Ejemplo de EmailComposer

2.4.2 Servicios

Los servicios que he utilizado en este proyecto son los siguientes: archivos TypeScript junto con archivos PHP los cuales son manejados mediante el servidor de SOLVAM debido a que dichos archivos necesitan de un servidor para ser funcionales.

Estos archivos PHP son usados para la comunicación entre la aplicación y la BBDD mediante el uso del componente HTTP, descrito en la página anterior, de Ionic, framework posteriormente descrito, el cual es usado para la recogida o el envío de datos vía URL.

La mayoría de estos archivos PHP contienen las consultas a la BBDD para recoger, insertar, actualizar y eliminar datos.

3. Análisis del contexto

En los tiempos en los que nos encontramos, con tanta población joven, se acrecienta la necesidad de encontrar un centro educativo que se adecue a las exigencias, tanto de los padres como de los alumnos y una vez escogido son los departamentos correspondientes los que a día de hoy siguen aceptando al alumnado y profesorado, que son los que comprueban que el alumno asiste a las clases, esta acción se realiza pasando lista del mismo en formato físico, papel; para resolver este problema se ha realizado una aplicación que controla la gestión de los estudiantes y profesores para el Centro de FP SOLVAM.

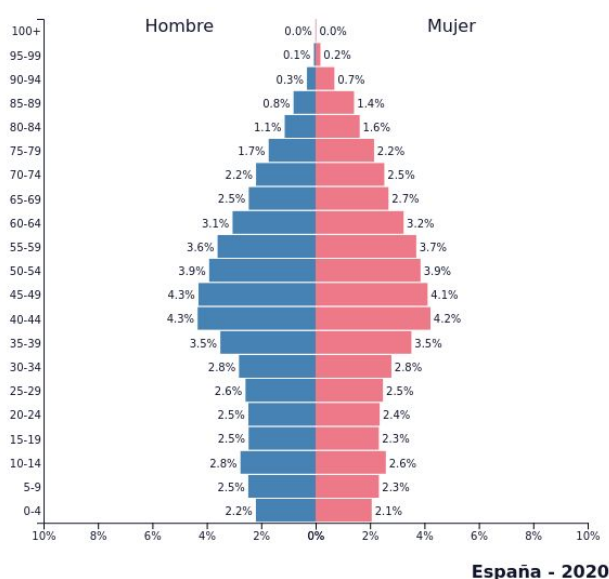


Figura 20: Grafica población española

Para poder comprender mejor este proyecto se describen a continuación las diferentes fases que se han implementado:

- **Inserción y matriculación del alumnado.** Al principio de curso se realiza la matriculación del alumnado en las asignaturas, si este tiene alguna por recuperar, o cursos, si va a realizar el curso completo, también tendrá ocasión de hacer la matrícula aquel estudiante que llegue a mitad de curso.

La aplicación permite que la administración del centro lleve a cabo la inscripción de los estudiantes en su propia base de datos.

- **Pasar lista.** Por causas de fuerza mayor los estudiantes retrasan su asistencia a las clases, se abstienen de las mismas, o en caso contrario si estos asisten y no tienen un comportamiento adecuado se les expulsa de la clase; al hacerlo, el profesor de esa asignatura, debe dar parte de ello y aunque poner una abstención a un alumno es tarea fácil, cuando son varios el trabajo se torna pesado. Las ventajas que tiene la aplicación es que el profesorado podrá ponerle faltas de asistencia a varios alumnos a la vez en las asignaturas a las que ha faltado, otra de las virtudes de las que dispone es que el alumnado no tendrá la ocasión de poder manipular la hoja de faltas.
- **Mensajería interna.** Cuando un alumno se retrasa, falta al centro o es expulsado del mismo, el centro podrá realizar el envío de un mensaje generado a través de la aplicación indicando al estudiante, si este es mayor de edad, de las faltas pertinentes que ha tenido ese día, en cambio si es menor de edad el envío se realizará a los familiares.

4. Desarrollo del contenido

En primer lugar hay que tener en cuenta que la aplicación va a ser administrada desde el propio centro educativo, por lo que su administración debe ser sencilla e intuitiva, y hallarse integrada dentro de la misma aplicación. En cuanto a las funciones que realizará la aplicación, serán de administración de la propia aplicación, de introducción de datos o de lectura de datos, por lo que la mayor parte de las funcionalidades tendrán que acceder a la base de datos para resolverse.

4.1 Perfiles de usuarios

4.1.1 Usuario administrador

- Modificación de los datos (detallados antes) y su contraseña.
- Alta, baja y modificación de los datos del alumnado y de los datos del profesorado.
- Visualización de todos los estudiantes y profesores registrados, y a su vez los datos de cada uno de ellos.
- Ver los mensajes de todo el profesorado.

4.1.2 Usuario no administrador

- Modificación de sus datos (detallados antes) y su contraseña.
- Visualización del alumnado que esté matriculado en las asignaturas en las que el usuario imparte enseñanza.
- Baja del alumno en las asignaturas que esté matriculado.
- Visualización de los mensajes generados en las asignaturas en las que realiza la enseñanza.

4.2 Documentación de las tareas

Tarea	Frecuencia (1-4)	Usuarios	Entradas	Salidas	Problemas posibles
Login	3	Todos	Usuario y contraseña	Devuelve una autenticación de usuario	La entrada no es coincidente con la BBDD
Horarios y sesiones	4	Todos	Alumnos, faltas, fechas y horas	Inserta faltas/Borra faltas	Fechas mal introducidas
Mensajes	2	Todos	---	Muestra mensajes generados	---
Mensajes enviados	2	Todos	---	Muestra mensajes archivados	---
Ver detalles	2	Todos	Cuerpo del mensaje	Muestra el mensaje a enviar	---
Perfil	1	Todos	Usuario y contraseña	---	Usuario ya existente en BBDD
Subir csv	1	Admin.	.csv y curso	---	Curso mal seleccionado
Eliminar matricula	3	Todos	Curso, asignatura y	---	---

			alumno		
Eliminar alumno	2	Admin.	Curso y alumno	---	---
Insertar alumno	1	Admin.	Nia, nombre, apellidos, curso, correos, teléfonos	---	Campos obligatorios vacíos
Asociar profesor	1	Admin.	Curso, asignatura y profesor	---	---
Eliminar profesor	1	Admin.	Profesor	---	---
Insertar profesor	1	Admin.	Nombre, contraseña, tipo, tutor (Si/No), curso	---	El usuario ya ha sido insertado con anterioridad

4.3 Usabilidad

La aplicación ha de ser intuitiva para que pueda ser utilizada por un tipo de usuario sin conocimientos informáticos profesionales, para conseguir esto se ha optado por utilizar una interfaz gráfica simple, con títulos breves para que el usuario sepa intuitivamente para que sirve cada ventana.

5. Conclusión y valoración personal

En este capítulo se incluyen las conclusiones del proyecto y las dificultades que se han encontrado durante el desarrollo del proyecto.

5.1 Conclusión

Una vez finalizada la elaboración del TFG, es el momento de reflexionar sobre el trabajo realizado, se puede decir que se ha cumplido con los requisitos especificados y se ha alcanzado los objetivos planteados, ya que, se ha elaborado una aplicación híbrida para smartphones dotados con el sistema operativo Android e iOS como para ordenadores personales.

A lo largo del trimestre con el desarrollo del proyecto se ha adquirido y consolidado una serie de conocimientos que resultan importantes para la vida profesional de un programador web, entre ellas he aprendido a desarrollar paso a paso este software, analizando las peticiones del consumidor y se han realizado pruebas funcionales y de diseño, con el fin de asegurar una buena aceptación por parte del cliente y de los usuarios. De esta forma he conseguido realizar las tareas propuestas y cumplir con los objetivos planteados.

En definitiva, la aplicación resultada del proyecto es una herramienta para facilitar la evaluación del alumnado de forma precisa. Todos los objetivos establecidos al principio del desarrollo se han alcanzado con éxito teniendo la misma un futuro muy prometedor, por lo tanto se puede concluir que cumple con las condiciones necesarias para facilitar su uso al usuario.

5.2 Valoración personal

La elaboración del Trabajo de Fin de Grado (TFG) ha sido muy enriquecedor en todo su proceso de creación a la vez que ha supuesto también un reto personal al haberlo realizado por cuenta propia.

Al mismo tiempo, me he sumergido en un universo nuevo que ha sido el mundo de la investigación, a través del cual he podido conocer más sobre los procesos de búsqueda de información, análisis y toma de decisiones incrementando las capacidades de búsqueda, haciendo lecturas específicas sobre el tema del proyecto y realizando la escritura del mismo de una manera impersonal, etc.

Para finalizar, me ha servido para afianzar conocimientos, favorecer una actitud de compromiso personal responsable y consciente sobre las decisiones tomadas, poder profundizar más en la toma de decisiones que supone mi vocación, a ser consecuente con unas tareas definidas y participar de manera eficaz en la toma de decisiones.

6. Webgrafía

- Manuales de aprendizaje de Ionic
<https://ionicframework.com/docs/es>
- Manuales de aprendizaje de AngularJS
<https://docs.angularjs.org/guide>
- Manuales de aprendizaje de PHP
<https://www.php.net/manual/es/index.php>
- Foros para consulta y resolución de dudas
<https://stackoverflow.com/>
- Código abierto y librerías para gestión de copias de seguridad
<https://github.com/>