



# INGENIERIA EN SISTEMAS DE TELECOMUNICACIÓN

Curso Académico 2017/2018

Trabajo Fin de Grado

## PLANGO

Autor : Sara López Zambrano

Tutor : Dr. Pedro de las Heras



# Trabajo Fin de Grado

planGO

**Autor :** Sara López Zambrano

**Tutor :** Dr. Pedro de las Heras

La defensa del presente Proyecto Fin de Carrera se realizó el día                      de  
de 20XX, siendo calificada por el siguiente tribunal:

**Presidente:**

**Secretario:**

**Vocal:**

y habiendo obtenido la siguiente calificación:

**Calificación:**

Fuenlabrada, a                      de                      de 20XX



*Dedicado a  
mi familia*



# Agradecimientos

Me gustaria hacer especial mencion a mis padres en este momento. Les agradezco enormemente todo lo que han hecho por mi a lo largo de toda mi vida. Gracias a ellos he conseguido llegar a este momento tan importante y he logrado alcanzar ese objetivo que me habia marcado. Muchas gracias por apoyarme en todas mis decisiones, por ayudarme en los malos momentos y por disfrutar, igual o mas que yo, de los buenos momentos. Tambien me gustaria agradecer a mis abuelos





# Resumen

Aquí viene un resumen del proyecto. Ha de constar de tres o cuatro párrafos, donde se presente de manera clara y concisa de qué va el proyecto. Han de quedar respondidas las siguientes preguntas:

- ¿De qué va este proyecto? ¿Cuál es su objetivo principal?
- ¿Cómo se ha realizado? ¿Qué tecnologías están involucradas?
- ¿En qué contexto se ha realizado el proyecto? ¿Es un proyecto dentro de un marco general?

Lo mejor es escribir el resumen al final.



# Índice general

<b>1. Introducción</b>	<b>1</b>
1.1. Historia de la web . . . . .	1
1.2. Tecnologías . . . . .	1
1.2.1. HTML5 . . . . .	1
1.2.2. JavaScript . . . . .	3
1.2.3. TypeScript . . . . .	3
1.2.4. CSS3 . . . . .	4
1.2.5. Sass . . . . .	4
1.2.6. Node.js . . . . .	4
1.2.7. Ionic 2 . . . . .	5
1.2.8. Express . . . . .	7
1.2.9. MongoDB . . . . .	7
1.2.10. Express . . . . .	8
1.3. Estructura de la memoria . . . . .	9
<b>2. Objetivos</b>	<b>11</b>
2.1. Objetivo general . . . . .	11
2.2. Objetivos específicos . . . . .	11
2.3. Planificación temporal . . . . .	12
<b>3. Estado del arte</b>	<b>13</b>
3.1. Sección 1 . . . . .	13
<b>4. Diseño e implementación</b>	<b>15</b>
4.1. Frontend . . . . .	15

4.1.1. Mapa navegación . . . . .	15
4.1.2. Funcional . . . . .	16
4.2. Arquitectura general . . . . .	21
<b>5. Resultados</b>	<b>23</b>
<b>6. Conclusiones</b>	<b>25</b>
6.1. Consecución de objetivos . . . . .	25
6.2. Aplicación de lo aprendido . . . . .	25
6.3. Lecciones aprendidas . . . . .	25
6.4. Trabajos futuros . . . . .	26
<b>A. Manual de usuario</b>	<b>27</b>
<b>Bibliografía</b>	<b>29</b>

# Índice de figuras

4.1. Pantalla login . . . . .	16
4.2. Header . . . . .	18
4.3. Mis amigos . . . . .	19
4.4. Buscador amigos . . . . .	20



# Capítulo 1

## Introducción

En poco tiempo se ha producido un gran avance en el mundo de las tecnologías y en concreto en los dispositivos móviles, tanto ha sido así que estamos al alcance de la mayoría de las cosas solo disponiendo de un dispositivo móvil con conexión a Internet. Si navegamos en el tiempo, desde dispositivos que solo nos ofrecían la capacidad de mantener una comunicación instantánea entre dos personas, hemos llegado a incluso poder realizar pagos. Pero es tan importante la evolución a nivel de dispositivo como a nivel de aplicaciones. Se han desarrollado aplicaciones muy potentes que nos facilitan la vida. Un claro ejemplo es el poder gestionar tus cuentas del banco e incluso realizar operaciones desde cualquier lugar. Existen aplicaciones para todos los gustos, necesidades e intereses.

### 1.1. Historia de la web

### 1.2. Tecnologías

#### 1.2.1. HTML5

HTML, que significa Lenguaje de Marcado para Hipertextos, es el elemento de construcción más básico de una página web y se usa para crear y representar visualmente una página web. Determina el contenido de la página web, pero no su funcionalidad. Hiper Texto se refiere a enlaces que conectan una página Web con otra, ya sea dentro de una página web o entre diferentes sitios web. Un lenguaje de marcado hace referencia a aquellos lenguajes que emplean etiquetas.

Estas etiquetas ya están predefinidas dentro del lenguaje respectivo y contienen la información que ayudan a leer el texto. Su principal diferencia con los lenguajes de programación es que éstos últimos poseen funciones aritméticas o variables, mientras que los lenguajes de marcado no.

HTML5 es la quinta revisión del estándar que fue creado en 1990 y su versión definitiva se publicó en octubre de 2014. Con HTML5, los navegadores como Firefox, Chrome, Explorer, Safari y más pueden saber cómo mostrar una determinada página web, saber dónde están los elementos, dónde poner las imágenes, dónde ubicar el texto.

En HTML5, se han tomado en cuenta mejoras en la creación de la estructura del código web y en el manejo óptimo de las etiquetas web. De esta manera, se convierte en un estándar mucho más versátil, que permitirá realizar una interacción mucho más poderosa y simple, mejorando la experiencia de uso por parte del usuario y facilitando la depuración del código web.

Las ventajas principales de esta versión son:

- Nueva estructura de etiquetas: permite separar el encabezado, la barra de navegación, secciones de la página, textos, diálogos y el pie de página.
- Introducción de etiquetas video y audio: por medio de las etiquetas `<video>` y `<audio>` de HTML5, ahora puedes añadir videos o audio sin necesidad de usar Adobe Flash o cualquier otro plugin de tercero. Toda la acción sucede desde el propio navegador, lo que puede ayudar a disminuir el tamaño del archivo final de tu página.
- Geolocalización: permite al sitio detectar la ubicación de cada usuario que ingresa al sitio web.
- Aplicaciones web: desarrollar aplicaciones HTML5 tiene la ventaja de que el resultado final es completamente accesible, es decir, se puede acceder a esta aplicación desde un ordenador, tablet o móvil.
- Capacidad de realizar ejecuciones offline: esto permite realizar aplicaciones de escritorio.
- Canvas: nueva etiqueta de dibujo sobre la página web.



### 1.2.2. JavaScript

JavaScript es un lenguaje de programación que se utiliza principalmente para crear páginas web dinámicas. Una página web dinámica es aquella que incorpora efectos como texto que aparece y desaparece, animaciones, acciones que se activan al pulsar botones y ventanas con mensajes de aviso al usuario.

Surgió por la necesidad de ampliar las posibilidades del HTML. En efecto, al poco tiempo de que las páginas web apareciesen, se hizo patente que se necesitaba algo más que las limitadas prestaciones del lenguaje básico, ya que el HTML solamente provee de elementos que actúan exclusivamente sobre el texto y su estilo, pero no permite, como ejemplo sencillo, ni siquiera abrir una nueva ventana o emitir un mensaje de aviso. La temprana aparición de este lenguaje, es posiblemente la causa de que se haya convertido en un estándar soportado por todos los navegadores actuales.

Los documentos HTML permiten incrustar fragmentos de código JavaScript, bien dentro del propio archivo HTML o bien realizando una carga de ese código indicando el archivo donde se encuentra el código JavaScript. Dentro de un documento HTML puede haber ninguno, uno o varios scripts de JavaScript.

Además, también es utilizado del lado del servidor, ya que tiene la ventaja de poseer un excelente modelo de eventos, ideal para la programación asíncrona.

### 1.2.3. TypeScript

TypeScript es un lenguaje de programación de alto nivel que implementa muchos de los mecanismos más habituales de la programación orientada a objetos, pudiendo extraer grandes beneficios que serán especialmente deseables en aplicaciones grandes, capaces de escalar correctamente durante todo su tiempo de mantenimiento. Puede ser usado para desarrollar aplicaciones JavaScript que se ejecutarán en el lado del cliente o del servidor (Node.js).

TypeScript convierte su código en Javascript común. Es llamado también Superset de Javascript, lo que significa que si el navegador está basado en Javascript, este nunca llegará a saber que el código original fue realizado con TypeScript y ejecutará el Javascript como lenguaje original.

### 1.2.4. CSS3

Es el lenguaje utilizado para describir la presentación de documentos HTML o XML. Al crear una página web, se utiliza en primer lugar el lenguaje HTML/XHTML para marcar los contenidos, es decir, para designar la función de cada elemento dentro de la página y una vez creados los contenidos, se utiliza el lenguaje CSS para definir el aspecto de cada elemento.

El objetivo inicial de CSS, separar el contenido de la forma, se cumplió ya con las primeras especificaciones del lenguaje. Sin embargo, el objetivo de ofrecer un control total a los diseñadores sobre los elementos de la página ha sido más difícil de cubrir. Las especificaciones anteriores del lenguaje tenían muchas utilidades para aplicar estilos a las webs, pero los desarrolladores aun continúan usando trucos diversos para conseguir efectos tan comunes o tan deseados como los bordes redondeados o el sombreado de elementos en la página.

### 1.2.5. Sass

Sass es un metalenguaje de Hojas de Estilo en Cascada (CSS). Es un lenguaje de script que es traducido a CSS.

Extiende CSS proveyendo de varios mecanismos que están presentes en los lenguajes de programación tradicionales, particularmente lenguajes orientados a objetos, pero éste no está disponible para CSS3 como tal. Cuando SassScript se interpreta, éste crea bloques de reglas CSS para varios selectores que están definidos en el fichero SASS. El intérprete de SASS traduce SassScript en CSS.

Sass permite la definición de variables de tipo number, string, colores y booleanos. Otra ventaja es que soporta mixins. Un mixin es una sección de código que contiene código Sass. Cada vez que se llama un mixin en el proceso de conversión el contenido del mismo es insertado en el lugar de la llamada. Los mixins permiten una solución limpia a las repeticiones de código, así como una forma fácil de alterar el mismo.

### 1.2.6. Node.js

Node.js es un entorno de ejecución multiplataforma de código abierto para desarrollar aplicaciones web. Esta librería se ejecuta sobre JavaScript y está basado en el motor V8 de Javascript de Google. Este motor está diseñado para correr en un navegador y ejecutar el código de

Javascript de una forma extremadamente rápida.

Se trata de un intérprete Javascript del lado del servidor, lo que permite utilizar el mismo lenguaje de programación tanto para cliente como para servidor.

Node sirve para facilitar la creación de aplicaciones web escalables de manera sencilla y con gran estabilidad, pudiendo ser utilizado para desarrollar cualquier tipo de aplicación. Además, es importante volver a destacar su altísima velocidad y su flexibilidad, dos de sus cualidades más importantes.

Trabaja con un único hilo de ejecución que es el encargado de organizar todo el flujo de trabajo que se deba realizar. Gestiona sus tareas de manera asíncrona y para trabajar de manera óptima delega todo el trabajo en un pool de threads. La librería que construye esto es Libuv, una vez que el trabajo ha sido completado emite un evento recibido por Node.js.

### 1.2.7. Ionic 2

Se trata de un framework destinado al desarrollo de aplicaciones híbridas, aunque también puede ser utilizado para implementar aplicaciones web. Una aplicación híbrida es aquella desarrollada por las tecnologías web: HTML, CSS Y JavaScript. Este tipo de aplicaciones tienen una serie de ventajas como ser compatibles para una gran cantidad de sistemas operativos con un tiempo de desarrollo menor, pero a cambio de esta gran ventaja el rendimiento es menor que en una aplicación nativa.

Su característica fundamental es que usa por debajo Angular, esto le da ventajas como tener una buena estructura de proyecto y contar con una buena gama de componentes y directivas.

#### Componentes

Los componentes se utilizan unos a otros para la obtención de objetivos globales de la aplicación. Están pensados para, de manera modular y encapsulada, resolver pequeños problemas. Ionic ofrece componentes fáciles de utilizar, pero para comportamientos más específicos de nuestro modelo de negocio, será necesario crear nuestros propios componentes.

Los componentes de Ionic 2 se adaptan al dispositivo estéticamente. Manteniendo el mismo código, en un dispositivo iOS tiene diferente vista que en un dispositivo Android, ya que se adapta al sistema operativo en el que se compila. Esto permite que una aplicación híbrida de

visión de nativa y además da al usuario una experiencia cercana a la que está acostumbrada en su teléfono. Sin embargo, es decisión del desarrollador mantener esta visión en su aplicación o personalizar la estética a su gusto.

## **Apache Cordova**

Para el acceso a componentes nativos desde la aplicación de Ionic, como la cámara, acelerómetro, teclado, usa plugins que proporciona Apache Cordova. También nos permite compilar el desarrollo realizado con Ionic con tecnologías web en aplicaciones para móviles instalables mediante tiendas de aplicaciones.

## **Estructura**

Un proyecto con Ionic contiene una lista de carpetas y archivos. Cada parte tiene su función:

- **SRC:** carpeta que contiene los archivos fuente con el código desarrollado de la aplicación.
- **WWW:** contiene los archivos que se producen al realizar la transpilación del TypeScript y compilado de los archivos Sass, es decir, la transformación de todos los archivos de 'src', de tal manera que el navegador sea capaz de entender.
- **PLUGINS:** contiene todos los plugins nativos que se utilizan para la aplicación.
- **PLATFORM:** archivos de cada plataforma a la que da soporte la aplicación. Suele ser Android e iOS.
- **RESOURCES:** contiene los iconos de la aplicación y el splash screen.
- **HOOKS:** scripts que se crean para ser ejecutados automáticamente después de algo específico.
- **NODE MODULES:** dependencias de npm que vienen definidas en el package.json e instaladas en local dentro de tu proyecto á

### 1.2.8. Express

Express es una infraestructura de aplicaciones web Node.js mínima y flexible que proporciona un conjunto sólido de características para las aplicaciones web y móviles. Contiene muchos métodos de programa de utilidad HTTP y middleware.

Node.js es una plataforma construida sobre el motor de JavaScript de Google Chrome (V8) que permite la ejecución de JavaScript en el lado del servidor. Permite montar un servidor HTTP utilizando el modulo http que viene incluido en el core de Node.

Para comenzar un proyecto en Express es necesario configurar en que puerto e IP va a estar escuchando el servidor para atender a las peticiones. Además, es necesario añadir las urls a las que tiene que atender y que método realizar para cada caso.

### 1.2.9. MongoDB

Se trata de una base de datos no relacional, es decir, NoSQL. Esto significa que los datos no son almacenados en tablas y no garantiza consistencia. Este tipo de bases de datos tienen ciertas desventajas, como: poca eficiencia en aplicaciones que necesiten usar los datos intensivamente, o si contiene gran número de indexaciones. A cambio de esta desventaja tiene la capacidad de manejar mucha cantidad de datos, no generan cuellos de botella y se ejecutan en clusters de máquinas baratas.

Este tipo de bases de datos surgieron por el Big Data, donde la información que se genera es muy grande, de manera rápida y constante, y que, además, en ocasiones la forma es no estructurada y cambiante. Las bases de datos relacionales tenían carencias para afrontar esto.

#### Colecciones

MongoDB es una base de datos orientada a documentos, es decir, los documentos son almacenados en BSON, que es una representación binaria de JSON. No siguen un esquema fijo, los documentos de una misma colección pueden tener esquemas diferentes.

En MongoDB los documentos se agrupan en colecciones. Aunque lo normal es que los documentos de una colección compartan estructura, puede ser flexible porque la estructura no se impone a ningún documento y dinámica porque la estructura puede cambiar. En el caso de que se decida variar la estructura de datos, no sería necesario crear ni modificar las colecciones, bas-

taría con almacenar los nuevos documentos, con una estructura distinta, en la misma colección o en otra.

No proporciona integridad referencial. Esto significa que, si en una colección hacemos referencia a un documento de otra colección, la base de datos no tiene la responsabilidad de comprobar que el documento referenciado existe.

## **Velocidad**

MongoDB tiene una baja velocidad en generar o modificar información. Esto provoca que el acceso a la base de datos este más tiempo bloqueado. Una operación de escritura bloquea el acceso a toda la base de datos en la que se efectúa la operación. Para solucionar esto hay que evitar actualizaciones que provoquen movimientos, es decir, que el crecimiento de un documento sea tan grande que suponga un cambio de cajón. Cada documento se encuentra almacenado en un cajón el cual cuenta con un espacio extra para posibles crecimientos futuros.

A cambio de esta baja velocidad de escritura, proporciona una gran rapidez de lectura.

### **1.2.10. Express**

Express es una infraestructura de aplicaciones web Node.js mínima y flexible que proporciona un conjunto sólido de características para las aplicaciones web y móviles. Contiene muchos métodos de programa de utilidad HTTP y middleware.

Node.js es una plataforma construida sobre el motor de JavaScript de Google Chrome (V8) que permite la ejecución de JavaScript en el lado del servidor. Permite montar un servidor HTTP utilizando el modulo http que viene incluido en el core de Node.

Para comenzar un proyecto en Express es necesario configurar en que puerto e IP va a estar escuchando el servidor para atender a las peticiones. Además, es necesario añadir las urls a las que tiene que atender y que método realizar para cada caso.

## **Estructura**

FOTO

## 1.3. Estructura de la memoria

En esta sección se deberá introducir la estructura de la memoria. Así:

- En el primer capítulo se hace una intro al proyecto.
- En el capítulo 2 (ojo, otra referencia automática) se muestran los objetivos del proyecto.
- A continuación se presenta el estado del arte.
- ...





# Capítulo 2

## Objetivos

### 2.1. Objetivo general

Mi trabajo fin de grado consiste en crear una aplicación que ayude a tomar decisiones a la hora de organizar eventos o reuniones, además de realizar encuestas de todo tipo.

El objetivo principal de este proyecto es el aprendizaje de tecnologías puntas en estos momentos y llevar a cabo su desarrollo.

### 2.2. Objetivos específicos

- Buscar información sobre las tecnologías empleadas y hacer tutoriales como método de aprendizaje.
- Realizar diseño de las pantallas de la aplicación.
- Desarrollar las pantallas con su correspondiente funcionalidad.
- Desarrollar la parte backend conectando con la base de datos.
- Conectar la parte front con la parte back.
- Testear la aplicación.

## 2.3. Planificación temporal

Este proyecto ha sido desarrollado en 4 meses a tiempo parcial, ya que ha sido compaginado con el trabajo.

En la primera fase del proyecto A mi  $\frac{1}{2}$  me gusta que aquí  $\frac{1}{2}$  pongi  $\frac{1}{2}$  is una descripción  $\frac{1}{2}$  de lo que os ha llevado realizar el trabajo. Hay gente que añ  $\frac{1}{2}$  ade un diagrama de GANTT. Lo importante es que quede claro cuñ  $\frac{1}{2}$  nto tiempo llevas (tiempo natural, p.ej., 6 meses) y a quí  $\frac{1}{2}$  nivel de esfuerzo (p.ej., principalmente los fines de semana).

# Capítulo 3

## Estado del arte

Existen otras aplicaciones que permiten la votación de encuestas como por ejemplo Doodle

### 3.1. Sección 1

Hemos hablado de cómo incluir figuras. Pero no hemos dicho nada de tablas. A mí me gustan las tablas. Mucho. Aquí un ejemplo de tabla, la Tabla ??.

1	2	3
4	5	6
7	8	9

Cuadro 3.1: Ejemplo de tabla



# Capítulo 4

## Diseño e implementación

### 4.1. Frontend

#### 4.1.1. Mapa navegación

El término navegación describe la acción de moverse entre las páginas y dentro de la página. Es el punto de partida de la experiencia del usuario. Es la forma en que los usuarios buscan el contenido y las características que les interesan.

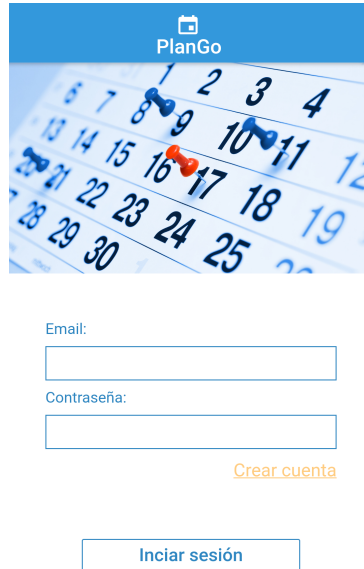
Existen dos tipos de estructura de navegación: jerárquica y plana. En una estructura jerárquica, las páginas se organizan en una estructura parecida a un árbol. Cada página secundaria tiene un único elemento primario, pero un elemento primario puede tener una o más páginas secundarias. Para llegar a una página secundaria, hay que moverse a través del elemento primario. En el caso de la estructura plana o lateral, las páginas existen en paralelo. Puedes ir de una página a otra en cualquier orden.

Esta aplicación sigue una estructura combinada entre jerárquica y plana. Se usan estructuras planas para las páginas de nivel superior que pueden verse en cualquier orden, y estructuras jerárquicas para las páginas que tienen relaciones más complejas.

FOTO

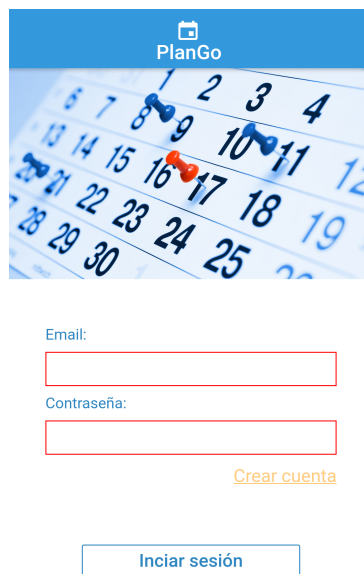
### 4.1.2. Funcional

#### Login



The image shows the login screen of the PlanGo app. At the top, there is a blue header with the PlanGo logo. Below the header, there is a background image of a calendar with several dates marked by pushpins. The login form consists of two input fields: 'Email:' and 'Contraseña:'. Below the 'Contraseña:' field, there is a link that says 'Crear cuenta'. At the bottom, there is a button labeled 'Iniciar sesión'.

(a) Login



This image is identical to the one above, showing the PlanGo app login screen. However, the input fields for 'Email:' and 'Contraseña:' are outlined with red borders, indicating an error state. The 'Iniciar sesión' button remains visible at the bottom.

(b) Error

Figura 4.1: Pantalla login

En la pantalla de login podemos encontrar dos inputs para poder introducir el correo y la contraseña. De esta forma se puede comprobar si el usuario tiene cuenta en la aplicación y

además se identifica al usuario que está usando la aplicación.

Se trata de un formulario reactivo, esto permite que los valores no tienen que ser recuperados de un servidor, se pueden validar de forma inmediata. Ambos son campos requeridos por lo que se comprueba que han sido introducidos, y en el caso de la contraseña que contenga mínimo 6 caracteres. Para el caso del email, se comprueba que sea un formato válido.

También encontramos un enlace para que aquellos usuarios que aún no han utilizado esta aplicación puedan crearse una cuenta. Este enlace lleva a la página de registro.

Finalmente encontramos un botón para iniciar sesión. Al pulsar en este botón se realiza la validación del formato de email y se manda una petición al servidor.

## **Registro**

Contiene un formulario para los datos personales del usuario. Es un formulario reactivo con sus correspondientes validaciones. Estas no se comprueban hasta que no se pulsa el botón 'Registrate'.

- Usuario: mediante una petición al servidor, se comprueba que no hay ningún usuario con este nombre.
- Email: se comprueba que sea un formato válido y, además, mediante una petición al servidor, que no esté ya registrado.
- Contraseña: se comprueba que no contenga menos de 6 caracteres.

He considerado que para este tipo de aplicación no son necesarios más datos sobre el usuario, lo más importante es identificar quien está votando y a quien va dirigida la encuesta.

## **Home**

En esta pantalla se encuentran accesos directos a distintas partes de la aplicación. El usuario puede acceder a las encuestas que ha creado, crear una nueva encuesta, consultar y añadir a sus amigos y acceder a las votaciones.

Al pulsar alguno de los accesos directos, se produce la navegación a la pantalla que corresponde.

## Header

El header cuenta con 4 botones:

1. Menú latera: nos permite acceder a cualquier parte de la aplicación.
2. Icono planGO: este icono te dirige a la pantalla de Home.
3. Icono ayuda: este icono te dirige a la pantalla de Ayuda.
4. Icono flecha: este icono te permite acceder a la pantalla anterior.



(a) Icono menú



(b) Icono atrás

Figura 4.2: Header

Como vemos en la 4.2a y 4.2b el menu lateral y el icono flecha no se muestran a la vez. Esto va a depender de la pantalla en la que nos encontremos. Cuando se produce navegación jerárquica, aparece el icono atrás, para el resto de los casos aparece el icono de menú.

## Mis encuestas

En esta pantalla el usuasio tiene acceso a todas las encuestas que han sido creadas por él. Como vemos en la figura, aparece un listado con todas las encuestas. Es posible hacer click en cualquiera de las encuestas para acceder al detalle de la misma.



## Mis amigos

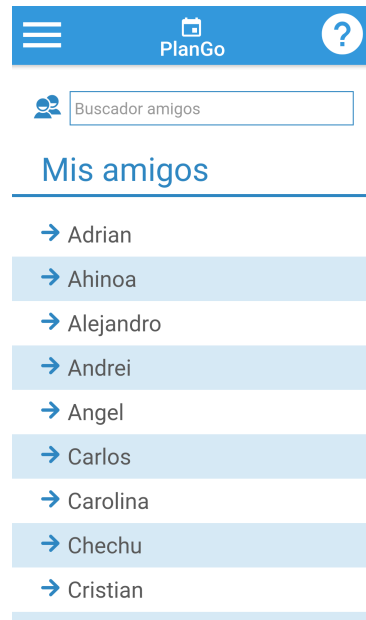


Figura 4.3: Mis amigos

Esta pantalla contiene un buscador en la parte superior y un listado, como se puede ver en la figura 4.3.

El buscador permite encontrar usuarios, que tengan cuenta en la aplicación, introduciendo su nombre. Este filtro tiene autocompletar, de tal manera que el listado de búsqueda se va actualizando a la vez que el usuario va introduciendo caracteres. Esto es muy útil para el usuario, pues no tiene que saber el nombre exacto del usuario que quiere buscar. Solo aparecen los usuarios que aún no tiene en su lista de amigos. En el caso de querer cerrar esta búsqueda, existe un botón con forma de flecha, para volver al estado anterior.

Para añadir un nuevo amigo, tras haber realizado su búsqueda, hay que pulsar en el botón 'añadir'.

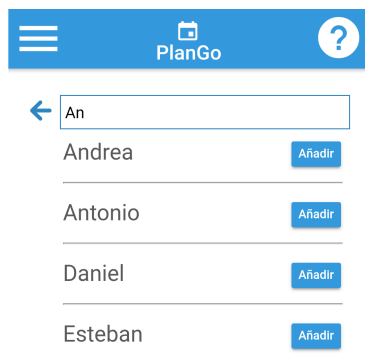


Figura 4.4: Buscador amigos

En el listado se ven todos los usuarios que contiene en su lista de amigos.

### Perfil

Esta pantalla contiene dos desplegables, cada uno de los cuales esta destinado a una función diferente. Es una pantalla bastante intuitiva como se ve en la fig.

En el desplegable 'Cambiar Username' el usuario podrá cambiar el nombre con el que esta identificado en la aplicación. En el caso del desplegable 'Cambiar contraseña' el usuario podrá cambiar su contraseña actual.

Para todos los cambios que se realicen en el perfil del usuario, por seguridad, es necesario que introduzca la contraseña actual. De esta manera solo el propio usuario conocedor de su contraseña podrá realizar cambios.

### Nueva encuesta

Esta es la parte principal de la aplicación, la creación de la encuesta.

Existen dos tipos de encuesta: por fecha o por texto. En una encuesta por fecha, se podrá votar sobre fechas y horas. En el caso de una encuesta por texto las votaciones se realizaran entre textos que pueden ser de cualquier tipo. Como se ve en la figura, esta eleccion se hace pulsando en el boton 'Nueva encuesta'.

Una vez selecionado el tipo de encuesta que se quiere crear, se accede a rellenar los datos necesarios para la encuesta como se ve en la figura. El formulario contiene varios campos, los cuales algunos son opcionaes y otros no. Los campos requeridos son el título y las opciones, si esto no se cumple, como se ve en la figura, se mostrara el error.

## 4.2. Arquitectura general

Si tu proyecto es un software, siempre es bueno poner la arquitectura (que es cómo se estructura tu programa a “vista de pájaro”).

Por ejemplo, puedes verlo en la figura ??.

Si utilizas una base de datos, no te olvides de incluir también un diagrama de entidad-relación.



# Capítulo 5

## Resultados

En este capítulo se incluyen los resultados de tu trabajo fin de grado.

Si es una herramienta de análisis lo que has realizado, aquí puedes poner ejemplos de haberla utilizado para que se vea su utilidad.



# Capítulo 6

## Conclusiones

### 6.1. Consecución de objetivos

Esta sección es la sección espejo de las dos primeras del capítulo de objetivos, donde se planteaba el objetivo general y se elaboraban los específicos.

Es aquí donde hay que debatir si se ha conseguido y si no. Cuando algo no se ha conseguido, se ha de justificar, en términos de qué problemas se han encontrado y qué medidas se han tomado para mitigar esos problemas.

### 6.2. Aplicación de lo aprendido

Aquí viene lo que has aprendido durante el Grado/Máster y que has aplicado en el TFG/TFM. Una buena idea es poner las asignaturas más relacionadas y comentar en un párrafo los conocimientos y habilidades puestos en práctica.

1. a
2. b

### 6.3. Lecciones aprendidas

Aquí viene lo que has aprendido en el Trabajo Fin de Grado/Máster.

1. a

2. b

## 6.4. Trabajos futuros

Una vez que el software se termina, así que aquí vienen ideas y funcionalidades que estarían bien tener implementadas en el futuro.

Es un apartado que sirve para dar ideas de cara a futuros TFGs/TFM.



**Apíndice A**

**Manual de usuario**



# **Bibliografía**