



**Universidad  
Europea**

**UNIVERSIDAD EUROPEA DE MADRID**

**ESCUELA DE ARQUITECTURA, INGENIERÍA Y DISEÑO**

**GRADO EN INGENIERÍA INFORMÁTICA**

**PROYECTO FIN DE GRADO**

**TICKETCLASS**

**ADRIAN SANTOS MENA**

**Dirigido por**

**BORJA MONSALVE PIQUERAS**

**CURSO 2019-2020**

**TÍTULO:** Ticketclass

**AUTOR:** ADRIÁN SANTOS MENA

**TITULACIÓN:** GRADO EN INGENIERÍA INFORMÁTICA

**DIRECTOR/ES DEL PROYECTO:** BORJA MONSALVE PIQUERAS

**FECHA:** JUNIO de 2020

## RESUMEN

En la Escuela de Arquitectura, Ingeniería y Diseño de la Universidad Europea de Madrid, el departamento de Ciencia, Computación y Tecnología es el encargado de gestionar y solucionar todas las incidencias que se producen en los laboratorios del Edificio C del campus de Villaviciosa de Odón y en varios del campus de Alcobendas.

Las incidencias son problemas que pueden surgir en cualquier momento y lugar dentro de una asociación o empresa, y siempre es importante mantenerlas gestionadas para conseguir que la experiencia del cliente sea la mejor posible.

El problema es que actualmente, el departamento de Ciencia, Computación y Tecnología no posee una herramienta decente para la gestión interna de todas las incidencias que van surgiendo dentro de los laboratorios, por lo que la gestión de estas es una tarea difícil y tediosa.

La idea de este proyecto es que, mediante la ayuda de las tecnologías de Laravel y React, se desarrolle una aplicación web portátil y responsive que ayude a gestionar todas estas incidencias y de esta forma conseguir un tiempo de resolución mayor y un aumento de la satisfacción de las personas que hagan uso de esos laboratorios.

## ABSTRACT

At the School of Architecture, Engineering and Design of the Universidad Europea of Madrid, the Science, Computing and Design Department is responsible for managing and solving all incidents produced in the build C laboratories of the Villaviciosa de Odon campus and in several of the Alcobendas campus.

The incidents are problems which can arise anywhere and anytime inside an association or company, and it is always something important to keep managed to ensure that the customer experience is the best possible.

The problem is that, actually, the Science, Computing and Design Department does not have a decent tool to manage internally all the incidents that arise within the laboratories, so the management of those is a difficult and tedious task.

The idea of this project is to make a portable and responsive web application using Laravel and React frameworks, to help the management of all of these incidents and, in this way, increase the achieve resolution and the satisfaction of those who use these laboratories.

## **AGRADECIMIENTOS**

Me gustaría agradecerle la ayuda a mi tutor de TFG, Borja Monsalve, por toda la ayuda, predisposición e ideas que me ha proporcionado durante el desarrollo del TFG.

A mis padres por haber estado apoyándome en todo momento.

Y finalmente a todos mis amigos por haber conseguido que estos últimos 4 años hayan sido los mejores de mi vida.

### **Dedicatoria**

Es verdad que una vida se divide por etapas, y puedo decir que la etapa como universitario y estudiante de ingeniería ha sido la mas feliz y triste a mismo tiempo. Ha sido un camino de 4 años no precisamente largo, porque aquí me encuentro, redactando la memoria final de mi Proyecto de Fin de Grado y a dos pasos y medio de terminar la carrera y matricularme.

En este apartado me gustaría dedicarle el trabajo de estos años a mis padres y a mi hermano pequeño, que han estado apoyándome incondicionalmente en todo momento desde que empecé la carrera, pero especialmente dedicárselo a mi abuelo, que hasta el día que nos dejó, no me di cuenta de todas las cosas que me aportó, sobre todo como trabajador y luchador nato.

*“El que da, no debe volver a acordarse, pero el que recibe nunca debe olvidar”*

## TABLA RESUMEN

	DATOS
<b>Nombre y apellidos:</b>	Adrián Santos Mena
<b>Título del proyecto:</b>	Ticketclass
<b>Directores del proyecto:</b>	Borja Monsalve Piqueras
<b>El proyecto se ha realizado en colaboración de una empresa o a petición de una empresa:</b>	SI El proyecto es una petición del departamento de Ciencias, Computación y Tecnología de la Escuela de Arquitectura, Ingeniería y Diseño de la UEM
<b>El proyecto ha implementado un producto:</b> (esta entrada se puede marcar junto a la siguiente)	SI
<b>El proyecto ha consistido en el desarrollo de una investigación o innovación:</b> (esta entrada se puede marcar junto a la anterior)	NO
<b>Objetivo general del proyecto:</b>	Implementar una mejor solución a la gestión de incidencias del área TIC de la Universidad Europea de Madrid.

# Índice

RESUMEN .....	3
ABSTRACT .....	4
TABLA RESUMEN .....	7
Índice .....	8
Índice de Figuras .....	11
Capítulo 1. RESUMEN DEL PROYECTO .....	15
1.1 Contexto y justificación .....	15
1.2 Planteamiento del problema .....	15
1.3 Objetivos del proyecto .....	15
1.4 Resultados obtenidos .....	15
1.5 Estructura de la memoria .....	15
1.5.1 Objetivos .....	16
1.5.2 Desarrollo del proyecto .....	16
1.5.3 Discusión .....	16
1.5.4 Conclusiones .....	16
1.5.5 Futuras líneas de trabajo .....	16
1.5.6 Referencias .....	16
Capítulo 2. ANTECEDENTES / ESTADO DEL ARTE .....	17
2.1 Estado del arte .....	18
2.1.1 FreshDesk .....	18
2.1.2 ZenDesk .....	19
2.1.3 SysAid .....	20
2.1.4 ZohoDesk .....	21
2.2 Contexto y justificación .....	22
2.3 Planteamiento del problema .....	23
Capítulo 3. OBJETIVOS .....	24
3.1 Objetivos generales .....	24
3.2 Objetivos específicos .....	24
3.3 Beneficios del proyecto .....	25



Capítulo 4.	DESARROLLO DEL PROYECTO .....	26
4.1	Planificación del proyecto .....	26
4.2	Descripción de la solución, metodologías y herramientas empleadas .....	27
4.2.1	Descripción de la solución .....	27
4.2.2	Metodología usada: SCRUM. ....	59
4.2.3	Herramientas empleadas. ....	63
4.3	Recursos requeridos.....	71
4.4	Presupuesto .....	72
4.5	Viabilidad.....	73
4.6	Resultados del proyecto.....	73
4.7	Plan de pruebas.....	73
4.7.1	Pruebas.....	74
Capítulo 5.	DISCUSIÓN.....	77
5.1	Metodología de trabajo empleada .....	77
5.2	¿Ha sido buena opción implementar React con Laravel? .....	77
5.3	Visualización de incidencias .....	77
5.3.1	Incidencias del supervisor .....	77
5.3.2	Incidencias de técnico .....	78
5.4	Uso de librería CSS externa (Bootstrap) o no.....	78
Capítulo 6.	CONCLUSIONES .....	79
6.1	Conclusiones del trabajo .....	79
6.2	Conclusiones personales .....	79
Capítulo 7.	FUTURAS LÍNEAS DE TRABAJO .....	81
Capítulo 8.	REFERENCIAS.....	82
Capítulo 9.	ANEXOS .....	84
9.1	Manual de instalación .....	84
9.1.1	Instalación de dependencias.....	85
9.2	Manual de usuario .....	86
9.2.1	Visión general de la aplicación Ticketing. Home. ....	86
9.2.2	Sidebar.....	87

---

9.2.3	Perfil .....	88
9.2.4	Página de incidencias. ....	91
9.3	Estructura completa del proyecto.....	98

## Índice de Figuras

Ilustración 1. Estructura básica de un sistema de gestión de incidencias .....	17
Ilustración 2. Página principal de FreshDesk.....	18
Ilustración 3. Página principal de ZenDesk.....	19
Ilustración 4. Página principal de SysAid.....	20
Ilustración 5. Página principal de ZohoDesk .....	21
Ilustración 6. Arquitectura Ticketclass .....	28
Ilustración 7. Diagrama de casos de uso Técnico-Supervisor .....	29
Ilustración 8. Diagrama de casos de uso Supervisor .....	30
Ilustración 9. Diagrama de casos de uso Administrador-Visitante .....	31
Ilustración 10. Fichero de configuración .env .....	47
Ilustración 11. Diagrama entidad-relación.....	50
Ilustración 12. Ejemplo 'Migración' .....	55
Ilustración 13. Ejemplo 'Seeder' .....	56
Ilustración 14. Creación petición HTTP .....	57
Ilustración 15. Realización de petición HTTP (React) .....	58
Ilustración 16. Ejemplo de media query .....	58
Ilustración 17. Paleta de colores ( <a href="https://colorhunt.co/palette/187385">https://colorhunt.co/palette/187385</a> ) .....	59
Ilustración 18. Histórico de los sprints .....	61
Ilustración 19. Tablero Scrum en Trello .....	61
Ilustración 20. Tableros creados en este proyecto .....	62
Ilustración 21. Tablero Trello de un Sprint.....	63
Ilustración 22. Ejemplo 1: Gráfica de barras .....	69
Ilustración 23. Ejemplo 2: tipos de gráficas de líneas .....	69
Ilustración 24. Ejemplo 'Button' de Bootstrap .....	78
Ilustración 25. Ejemplo 'Button' Ticketclass.....	78
Ilustración 26. Navbar .....	87
Ilustración 27. Página PERFIL .....	88
Ilustración 28. Visualización de gráficas usuario.....	89
Ilustración 29. Comparativas gráficas .....	89

---

Ilustración 30. Editar perfil usuario .....	90
Ilustración 31. Modal editar perfil .....	90
Ilustración 32. Actualización datos - editar perfil .....	91
Ilustración 33. Página incidencias .....	91
Ilustración 34. Filtro de incidencias.....	92
Ilustración 35. Visualización datos de incidencia .....	93
Ilustración 36. Editar incidencia. ....	94
Ilustración 37. Comentarios incidencia. ....	94
Ilustración 38. Confirmación eliminar incidencia.....	95
Ilustración 39. Crear incidencia.....	95
Ilustración 40. Grupos de técnicos.....	96
Ilustración 41. Añadir y eliminar técnico.....	96
Ilustración 42. Crear usuario. ....	97
Ilustración 43. Importar usuarios desde Excel. ....	97

## Índice de Tablas

Tabla 1. Requisitos deseables de una aplicación web .....	25
Tabla 2. Tabla ejemplo para los casos de uso .....	32
Tabla 3. CU-00 Iniciar sesión .....	32
Tabla 4. CU-01 Cerrar sesión .....	33
Tabla 5. CU-02 Visualizar listado incidencias .....	34
Tabla 6. CU-03 Ordenar incidencias .....	34
Tabla 7. CU-04 Filtrar incidencias .....	35
Tabla 8. CU-05 Visualizar datos incidencia .....	36
Tabla 9. CU-06 Cambiar estado de incidencia .....	36
Tabla 10. CU-07 Añadir comentario a incidencia .....	37
Tabla 11. CU-08 Crear incidencia .....	38
Tabla 12. CU-09 Editar incidencia .....	38
Tabla 13. CU-10 Eliminar incidencia .....	39
Tabla 14. CU-11 Asignar técnico o grupo a incidencia .....	40
Tabla 15. CU-12 Visualizar lista de equipos .....	40
Tabla 16. CU-13 Crear grupo de técnicos .....	41
Tabla 17. CU-14 Visualizar datos de un grupo .....	41
Tabla 18. CU-15 Añadir técnico a grupo existente .....	42
Tabla 19. CU-16 Eliminar técnico de grupo .....	42
Tabla 20. CU-17 Eliminar grupo de técnicos .....	43
Tabla 21. CU-18 Crear nueva categoría de equipo .....	44
Tabla 22. CU-19 Visualizar datos perfil .....	44
Tabla 23. CU-20 Editar perfil .....	45
Tabla 24. CU-21 Visualizar gráficas sobre datos de las incidencias .....	45
Tabla 25. CU-22 Crear usuario .....	46
Tabla 26. Ejemplos componentes Bootstrap .....	68
Tabla 27. Presupuesto del proyecto .....	72
Tabla 28. Plantilla tabla de pruebas .....	74
Tabla 29. CP-00 Datos de inicio de sesión erróneos .....	74

---

Tabla 30. CP-01 Campos rellenados incorrectamente al crear incidencia.....	75
Tabla 31. CP-02 Filtrar incidencias .....	75
Tabla 32. CP-03 Actualizar datos de incidencias filtradas .....	76
Tabla 33. CP-04 Editar perfil.....	76

## **Capítulo 1. RESUMEN DEL PROYECTO**

La finalidad principal de este apartado es proporcionar al lector qué tipo de proyecto se va a exponer, explicando su contexto, el problema que se pretende solucionar, los objetivos generales y el resultado final obtenido.

### **1.1 Contexto y justificación**

El proyecto se enfoca para satisfacer las necesidades del departamento de Ciencia, Computación y Tecnología, el cual es responsable de la gestión de las incidencias producidas en muchos de los laboratorios pertenecientes a los campus de Alcobendas y de Villaviciosa de Odón de la Universidad Europea de Madrid.

### **1.2 Planteamiento del problema**

El problema se encuentra en que, actualmente, no poseen de ningún sistema sencillo de usar mediante el cual puedan gestionar, de manera rápida y eficaz, todas las incidencias que se producen en los laboratorios de los que están a cargo.

Además, esto puede producir, entre muchas otras cosas, que la experiencia académica de los alumnos empeore y reciban correctamente las clases lectivas.

### **1.3 Objetivos del proyecto**

El objetivo principal de este proyecto es desarrollar una aplicación web mediante la cual se puedan gestionar todas esas incidencias que se vayan produciendo en las zonas de trabajo de las que esté a cargo el departamento de Ciencia, Computación y Tecnología.

De esta manera, mejorarán los tiempos de resolución de las incidencias y conseguirán que se encuentren a disposición todas las herramientas necesarias para impartir de manera satisfactoria las clases lectivas.

### **1.4 Resultados obtenidos**

El resultado es una aplicación web sencilla y muy intuitiva que permite la gestión de incidencias a cualquier usuario de la aplicación.

### **1.5 Estructura de la memoria**

El resto del documento se estructura de la siguiente manera:

#### **CAPITULO 2. ANTECEDENTES / ESTADO DEL ARTE.**

En este apartado se pone en contexto al lector. Se pretende explicar cuáles son los motivos por los que se ha decidido desarrollar este proyecto y qué otras herramientas parecidas existen en el mercado actualmente.

### **1.5.1 Objetivos**

Aquí se especifican cuáles son los objetivos principales y secundarios del proyecto.

### **1.5.2 Desarrollo del proyecto**

En este apartado se expone al lector todos los pasos, herramientas, metodologías, ..., que se han usado para llevar a cabo este proyecto y cual ha sido el resultado final del proyecto.

### **1.5.3 Discusión**

Este capítulo está dedicado a la discusión de varios temas que han ido surgiendo durante el desarrollo del proyecto y los cuales han producido incertidumbres en el desarrollador por no saber cómo implementarlos o desarrollarlos.

### **1.5.4 Conclusiones**

En este apartado se muestran las conclusiones del trabajo realizado, basándose en las ideas iniciales y las implementadas. También se describen los conocimientos adquiridos durante el desarrollo del proyecto.

### **1.5.5 Futuras líneas de trabajo**

En este apartado se toman en cuenta desarrollos que han quedado pendientes de hacer o ideas nuevas que se tienen pensadas para implementar en un futuro.

### **1.5.6 Referencias**

Este capítulo está única y exclusivamente destinado a la recopilación de las referencias usadas.



## Capítulo 2. ANTECEDENTES / ESTADO DEL ARTE

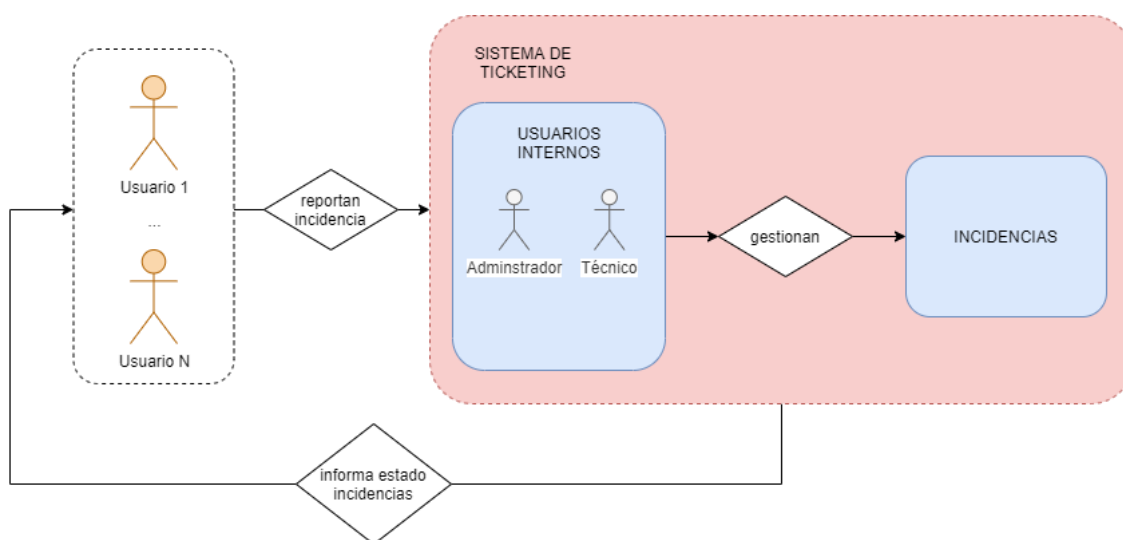
Para la comprensión del proyecto es necesario saber la importancia que tiene la resolución de incidencias en cualquier departamento de una asociación.

Las incidencias son problemas que pueden surgir en cualquier departamento, asociación u empresa sin aviso alguno, y que pueden acarrear una serie de problemas si su solución no se lleva a cabo en un determinado límite de tiempo. Estos problemas pueden ser desde el retraso de una sesión o reunión hasta la dada de baja de un cliente, las cuales son perjudiciales para la empresa en cuestión y no ayudarán en absoluto al crecimiento de esta.

Para ello se suelen crear lo que son llamadas herramientas de ticketing o gestores de incidencias, a través de las cuales, tanto los usuarios externos a la empresa o internos a ella reportan problemas o preguntas que les puedan surgir durante el periodo de instancia dentro de esa web/empresa/asociación...

Es necesario tener en cuenta los siguientes aspectos para crear una herramienta simple y funcional:

- Creación de incidencias/tickets para reportar un problema en específico.
- Visualización de los datos del ticket para su posible mejora de especificación o modificación de datos erróneos.
- El tiempo de respuesta debe de ser el menor posible. Como se suele decir: el tiempo es oro.
- Avisos o comentarios sobre el ticket para comunicar su estado y poder notificar en caso de resolución del ticket.



*Ilustración 1. Estructura básica de un sistema de gestión de incidencias*

En la Imagen *Ilustración 1*, se explica de forma más visual cómo debe de ser el funcionamiento básico de un sistema de gestión de incidencias, incluyendo así los principales roles y las acciones más importantes.

## 2.1 Estado del arte

Dentro del sector de gestión de incidencias en el mercado, se han mirado otras herramientas cuyas características tengan semejanza con la aplicación que se pretende desarrollar.

Actualmente hay bastantes herramientas públicas, por lo que se han podido sacar ideas y factores de diferenciación frente a ellas para el desarrollo de este proyecto.

### 2.1.1 FreshDesk

<b>Nombre</b>	FreshDesk
<b>URL</b>	<a href="https://freshdesk.com/">https://freshdesk.com/</a>
<b>Descripción</b>	Es una herramienta de la compañía <i>Freshwork</i> que se dedica a desarrollar software especializado para <b>mejorar el ambiente de trabajo de una empresa</b> .
<b>De pago</b>	Ofrece una versión gratuita, pero el resto de las <b>versiones son de pago</b>
<b>Dinero al año</b>	15 € / 29 € / 49 € / 109 € dependiendo la elección seleccionada
<b>Ventajas</b>	Tiene una interfaz amigable y muy intuitiva.
<b>Diferenciación frente a nuestra app</b>	Solo está hecha para empresas. A no ser que tengas una empresa, no vas a poder utilizarla de la manera más apropiada.

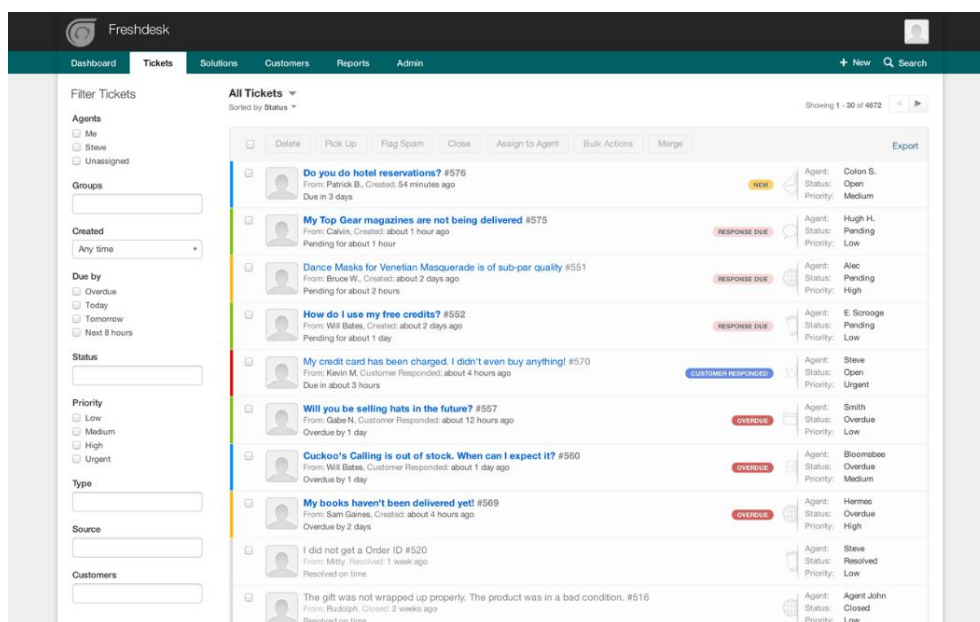
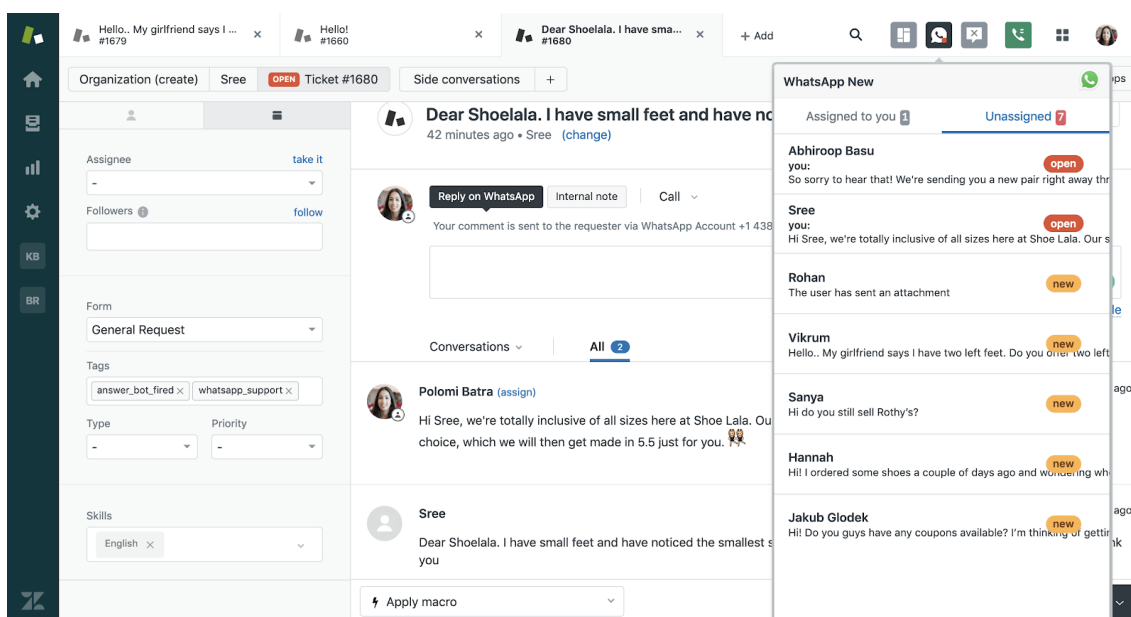


Ilustración 2. Página principal de FreshDesk

## 2.1.2 ZenDesk

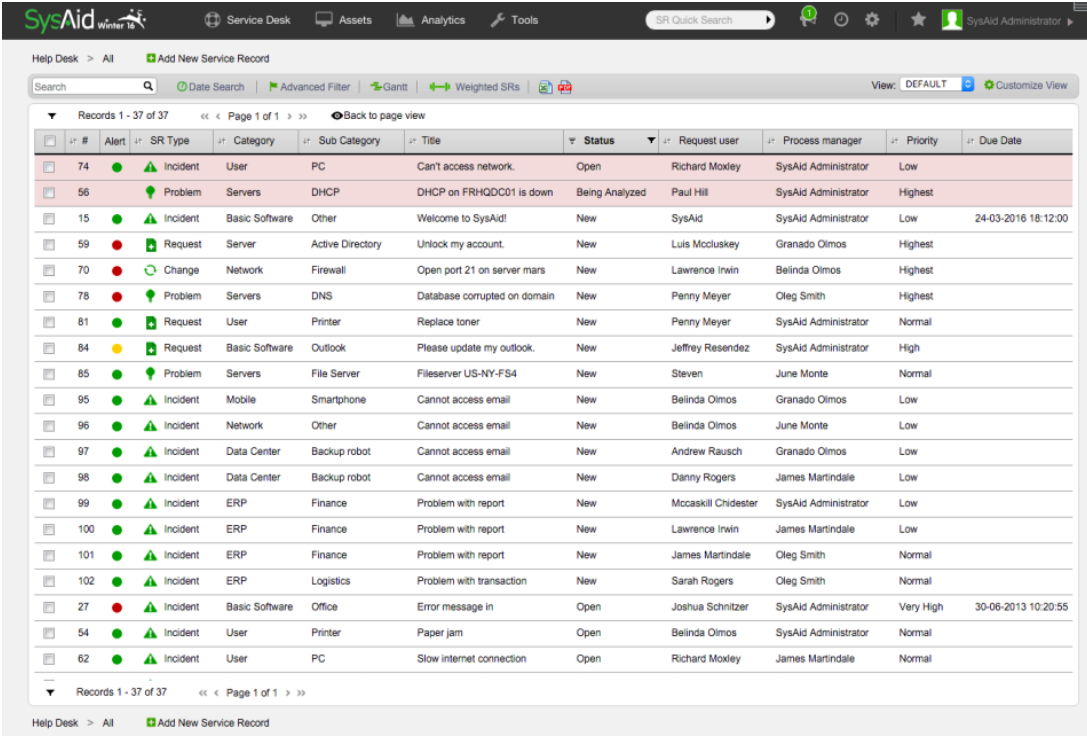
<b>Nombre</b>	ZenDesk
<b>URL</b>	<a href="https://www.zendesk.es">https://www.zendesk.es</a>
<b>Descripción</b>	Es una herramienta hecha para la gestión de <b>peticiones de un cliente</b> , permitiendo el envío de tickets por múltiples canales de conversación.
<b>De pago</b>	Pruebas gratuitas, pero contratación del servicio de pago.
<b>Dinero al mes</b>	5 opciones: <ul style="list-style-type: none"> <li>➤ 5 €/mes por agente.</li> <li>➤ 19 €/mes por agente.</li> <li>➤ 49 €/mes por agente.</li> <li>➤ 99 €/mes por agente.</li> <li>➤ 199 €/mes y agentes ilimitados.</li> </ul>
<b>Ventajas</b>	<ul style="list-style-type: none"> <li>- Chat.</li> <li>- Interfaz fácil de entender.</li> <li>- Comentarios a incidencias.</li> <li>- Llamadas online.</li> </ul>
<b>Diferenciación frente a nuestra app</b>	La principal diferencia es que va destinada originariamente a los clientes de la empresa y no para los trabajadores, aunque de todas formas, pone que hay una versión para los empleados de la empresa.



*Ilustración 3. Página principal de ZenDesk*

### 2.1.3 SysAid

<b>Nombre</b>	SysAid
<b>URL</b>	<a href="https://www.sysaid.com/es">https://www.sysaid.com/es</a>
<b>Descripción</b>	Esta herramienta es otro software de ticketing que ofrece la opción de guardar los tickets en la nube o en local, además de poder visualizar todos los dispositivos conectados a la red de ticketing privada.
<b>De pago</b>	Si.
<b>Dinero al año</b>	Es necesario introducir un correo electrónico para poder visualizar los costes de contratación mensuales.
<b>Ventajas</b>	<ul style="list-style-type: none"> <li>➤ Guardar datos en local o en la nube.</li> <li>➤ Atención a incidencias en 42 idiomas diferentes.</li> </ul>
<b>Diferenciación frente a nuestra app</b>	Es muy antigua y está desactualizada con la tecnología actual.



The screenshot displays the SysAid web interface. At the top, there's a navigation bar with icons for Service Desk, Assets, Analytics, and Tools. Below this is a search bar and a 'View: DEFAULT' dropdown. The main content area shows a table of service records with columns for Alert, SR Type, Category, Sub Category, Title, Status, Request user, Process manager, Priority, and Due Date. The table lists various incidents and requests, such as 'Can't access network', 'DHCP on FRHQDC01 is down', and 'Welcome to SysAid!'. The interface also includes pagination controls at the bottom of the table.

Alert	SR Type	Category	Sub Category	Title	Status	Request user	Process manager	Priority	Due Date
74	Incident	User	PC	Can't access network.	Open	Richard Moxley	SysAid Administrator	Low	
56	Problem	Servers	DHCP	DHCP on FRHQDC01 is down	Being Analyzed	Paul Hill	SysAid Administrator	Highest	
15	Incident	Basic Software	Other	Welcome to SysAid!	New	SysAid	SysAid Administrator	Low	24-03-2016 18:12:00
59	Request	Server	Active Directory	Unlock my account.	New	Luis Mccluskey	Granado Olmos	Highest	
70	Change	Network	Firewall	Open port 21 on server mars	New	Lawrence Irwin	Belinda Olmos	Highest	
78	Problem	Servers	DNS	Database corrupted on domain	New	Penny Meyer	Oleg Smith	Highest	
81	Request	User	Printer	Replace toner	New	Penny Meyer	SysAid Administrator	Normal	
84	Request	Basic Software	Outlook	Please update my outlook.	New	Jeffrey Resendez	SysAid Administrator	High	
85	Problem	Servers	File Server	Fileserver US-NY-FS4	New	Steven	June Monte	Normal	
95	Incident	Mobile	Smartphone	Cannot access email	New	Belinda Olmos	Granado Olmos	Low	
96	Incident	Network	Other	Cannot access email	New	Belinda Olmos	June Monte	Low	
97	Incident	Data Center	Backup robot	Cannot access email	New	Andrew Rausch	Granado Olmos	Low	
98	Incident	Data Center	Backup robot	Cannot access email	New	Danny Rogers	James Martindale	Low	
99	Incident	ERP	Finance	Problem with report	New	Mccaskill Chidester	SysAid Administrator	Low	
100	Incident	ERP	Finance	Problem with report	New	Lawrence Irwin	James Martindale	Low	
101	Incident	ERP	Finance	Problem with report	New	James Martindale	Oleg Smith	Normal	
102	Incident	ERP	Logistics	Problem with transaction	New	Sarah Rogers	Oleg Smith	Normal	
27	Incident	Basic Software	Office	Error message in	Open	Joshua Schnitzer	SysAid Administrator	Very High	30-06-2013 10:20:55
54	Incident	User	Printer	Paper jam	Open	Belinda Olmos	SysAid Administrator	Normal	
62	Incident	User	PC	Slow internet connection	Open	Richard Moxley	James Martindale	Normal	

Ilustración 4. Página principal de SysAid

### 2.1.4 ZohoDesk

<b>Nombre</b>	ZohoDesk
<b>URL</b>	<a href="https://www.zoho.com/es-xl/desk/">https://www.zoho.com/es-xl/desk/</a>
<b>Descripción</b>	Esta herramienta ofrece la realización de llamadas a través de ella. Además, ofrece a sus usuarios una manera de implementar un formulario a su página web para poder generar desde ahí tickets que lleguen directamente a Zoho Desk (API).
<b>De pago</b>	Si. Como todas las demás trae una versión de prueba para poder testear todas las opciones.
<b>Dinero al año</b>	3 pagos: <ul style="list-style-type: none"> <li>➤ 12 €/mes por agente.</li> <li>➤ 20 €/mes por agente.</li> <li>➤ 35 €/mes por agente.</li> </ul>
<b>Ventajas</b>	<ul style="list-style-type: none"> <li>- Se adapta al problema del cliente, sea cual sea.</li> <li>- Tiene desarrollada un api para poder usar sus servicios en cualquier página web que la implemente.</li> </ul>
<b>Diferenciación frente a nuestra app</b>	No está especializada en la solución de un problema específico, cosa que puede afectar a la resolución de estos.

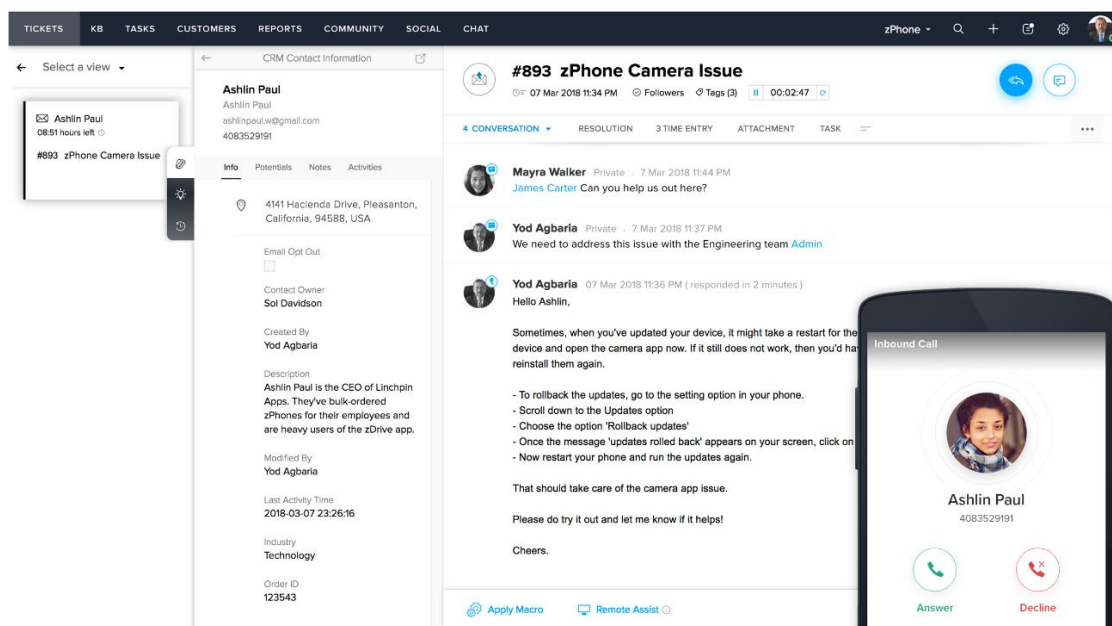


Ilustración 5. Página principal de ZohoDesk

Muchas de las ideas iniciales de proyecto se han sacado de todos los ejemplos anteriores.

Algunas de ellas son:

- **Agrupación de incidencias.** Se daría el caso en agrupar varias incidencias dependiendo del tipo de incidencia o categoría.
- **Notificaciones.** La idea inicial fue implementar un sistema de mensajería que notificara a los usuarios de los cambios que se iban produciendo en las incidencias a las que eran accesibles.
- **Asignación de incidencias.** Para ello es necesario tener un gestor de usuarios dentro de la herramienta o aplicación para así poder asignar incidencias específicas a los usuarios.

## 2.2 Contexto y justificación

En la Escuela de Arquitectura, Ingeniería y Diseño de la Universidad Europea de Madrid, el departamento de Ciencia, Computación y Tecnología es responsable, junto al departamento de TIC, de la gestión de todos los laboratorios con ordenadores que hay en el edificio C del Campus de Villaviciosa de Odón.

Esta gestión es realizada por un grupo de trabajo formado por un coordinador y dos monitores de trabajo, que a su vez son profesores de la universidad, y cuya función es, entre otras tareas, la correcta instalación del software necesario en dichas aulas, la gestión de las incidencias que se producen en ellas y su mantenimiento diario.

Teniendo en cuenta que al día pueden pasar por cada laboratorio más de 100 estudiantes, es muy importante que cada uno de ellos pueda encontrarse instalado en los equipos informáticos el software que necesite en cada asignatura para poderla seguir correctamente y con el mínimo de problemas posible. Sin embargo, con ese uso diario, es común encontrar desperfectos o problemas de configuración en el software instalado, y es conveniente que sean solucionados con la mayor velocidad posible para que no se interrumpa la docencia que se esté impartiendo.

Hasta la fecha, cuando un profesor encontraba una incidencia, podía, o darla de alta a través del sistema de ticketing institucional, o bien comunicarlo al coordinador de los laboratorios para su centralización. Sin embargo, los monitores de los laboratorios y el coordinador sólo contaban con permisos de usuario en esta plataforma, con lo que debían coordinar las incidencias que habían puesto en conocimiento del TIC de forma paralela. Esta tarea se venía tratando de centralizar a través de documentos compartidos o herramientas de terceros (como por ejemplo Trello), que no terminaban de cuajar y redundaban en descoordinación y aumento en los tiempos de respuesta.

Hablando con varios profesores y observando que la mayoría de ellos me exponían los problemas anteriormente mencionados, llegué a la conclusión de que, dentro del área de Tecnologías de la Información y de la Comunicación del campus de Villaviciosa, no existe un buen gestor de incidencias con el que profesores y trabajadores de mantenimiento puedan avisar de posibles fallos o perturbaciones en el material, tanto físico como virtual, de dentro de

las aulas del Edificio C, además de la imposibilidad de acceso a la plataforma por la mayoría de ellos.

### **2.3 Planteamiento del problema**

Para dar servicio a la necesidad expuesta en el anterior apartado, se plantea proyecto, en el que se va a idear y desarrollar una nueva aplicación de gestión de incidencias a través de la cual sea mucho más fácil e intuitivo la gestión de incidencias por parte de los que están a cargo de administraras, solucionarlas y verificarlas.

Esta aplicación web va a ser desarrollada con el fin de que los usuarios que la utilicen puedan gestionar todas las incidencias generadas diariamente mediante el uso de nuevas funcionalidades añadidas a las ya existentes del anterior gestor de incidencias.

De esta forma se conseguirá una herramienta útil para todos departamentos de la Escuela de Arquitectura, Ingeniería y Diseño, con la que la gestión de incidencias será un problema menos.

## Capítulo 3. OBJETIVOS

### 3.1 Objetivos generales

El objetivo principal de este proyecto es desarrollar un sistema de gestión de tickets/incidencias para la Escuela de Arquitectura, Ingeniería y Diseño de la Universidad Europea de Madrid mediante el cual sea más sencilla y rápida la administración de todas las incidencias producidas en los laboratorios de ordenadores en sus diferentes campus.

### 3.2 Objetivos específicos

Los objetivos específicos más remarcables del proyecto son los siguientes:

- **Crear incidencias.**
- **Gestión de los datos de una incidencia.**
- **Visualización de todos los datos** de incidencias.
- **Gestión de perfil** de usuario.
- **Administración de grupos** de técnicos.
- **Enviar comentarios** sobre una incidencia.
- **Accesibilidad** tanto desde ordenador como desde móvil.
- **Diseño UX<sup>1</sup> sencillo** y fácil de usar.
- Implementar **sistema de alertas** para los usuarios de sus incidencias.
- Seguimiento de cambios de una incidencia.

También se han tomado en cuenta algunos requisitos deseables para llevar a cabo el desarrollo de la aplicación web (Sara Escudero, s.f.):

<b>Diseño adaptado a todos los dispositivos</b>	Este es uno de los objetivos más importantes, ya que se ha pensado que el uso de la herramienta debería ser posible en todo momento.
<b>Diseño intuitivo y estético</b>	Se ha creado la página pensando siempre en el usuario que la va a usar, de esta manera se ha intentado crear una interfaz lo más sencilla e intuitiva posible.
<b>Descripciones y notificaciones</b>	Es importante mantener al usuario de lo que está ocurriendo en todo momento para así mejorar su experiencia usando la página web.

---

<sup>1</sup> **UX: user-experience.** Se refiere a cómo se siente un usuario al interactuar con un programa específico.



<b>Botones de acción correctamente definidos</b>	Todos los botones de acción de la aplicación deben de estar correctamente definidos para hacer entender al usuario su funcionamiento.
<b>Aplicación en constante actualización</b>	Conseguir que los datos de la aplicación se actualicen a tiempo real según el usuario vaya utilizándola.
<b>Efecto sorpresa</b>	Incluir alguna funcionalidad que el usuario no se espere y que le dé puntos a la aplicación.

*Tabla 1. Requisitos deseables de una aplicación web*

### 3.3 Beneficios del proyecto

El proyecto permitirá a los monitores y supervisores de la escuela, encargados de los laboratorios, tener un control total sobre sus incidencias, consiguiendo así una mayor respuesta por parte del departamento de soporte.

Otro de los beneficios de este proyecto es proveer al Departamento de Ciencia, Computación y Tecnología de una herramienta de gestión de incidencias. Esto quiere decir que no necesitarán depender de un tercero para acceder a la parte administrativa de la herramienta, y así poder tener accesos como:

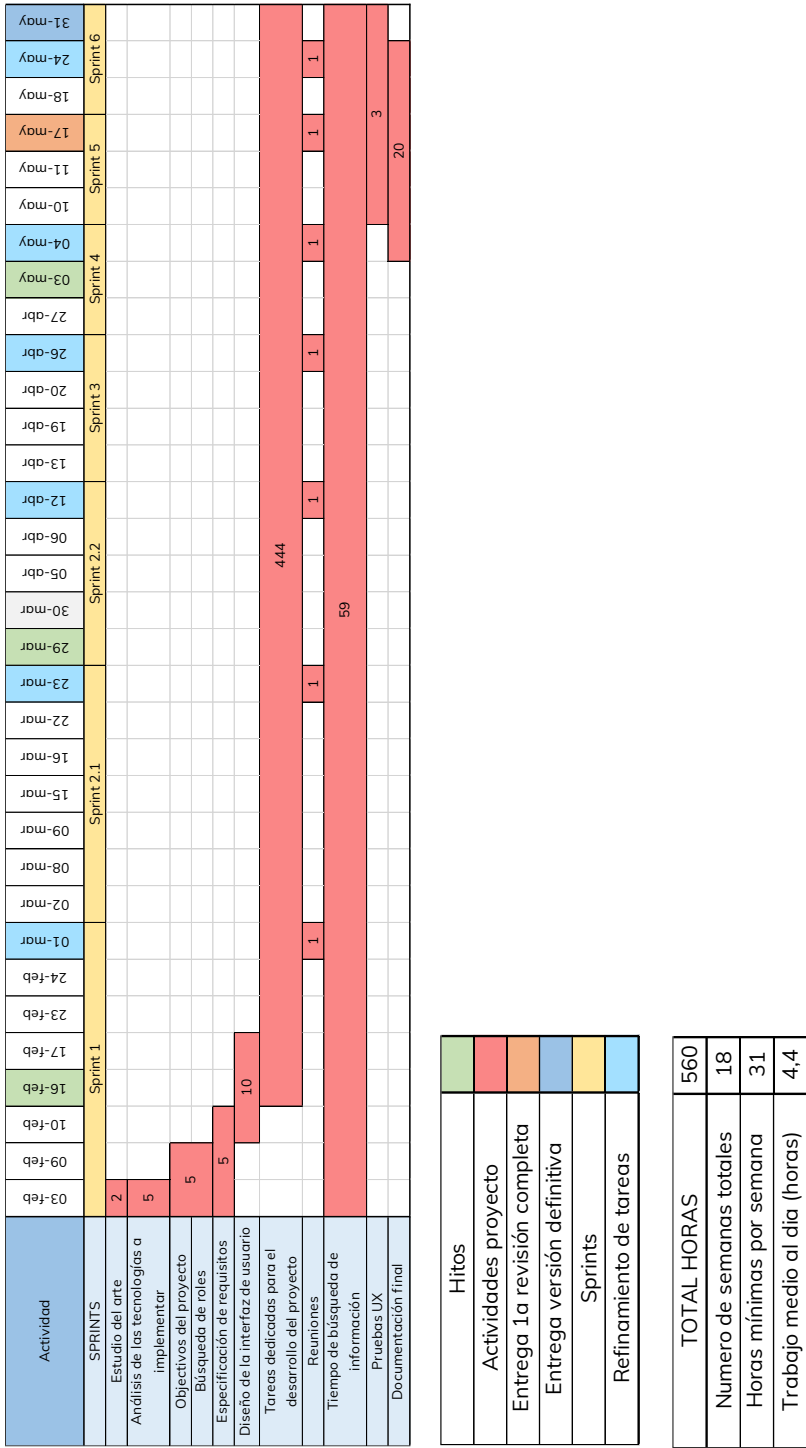
- Alta/Baja de usuarios.
- Peticiones directas para la mejora de las funcionalidades o interfaz de la herramienta.
- Generación de incidencias.
- ...

Gracias a un buen desarrollo del proyecto y un correcto uso de la herramienta se conseguirá que las instalaciones de la organización y los materiales que se usan para las clases lectivas se mantengan en el mejor estado posible. De esta forma, los cursos lectivos que hagan uso de todas estas herramientas se impartirán con una mayor agilidad y claridad consiguiendo así una mejor experiencia académica para todos los estudiantes.

Esto afectará indirectamente al NPS (Net Promoter Score) de la universidad ya que los estudiantes dejarán en las encuestas finales mejores valoraciones y feedback. Esto último se considera bastante importante dentro de cualquier universidad, ya que su imagen depende en gran parte las opiniones de sus estudiantes, que son los que van a salir de ella echándola flores o pedradas.

Capítulo 4. DESARROLLO DEL PROYECTO

4.1 Planificación del proyecto



Actividades mencionadas:

- **Sprints.** Es un intervalo de tiempo fijado por el proyecto durante el cual se realiza un avance del desarrollo del producto.
- **Estudio del arte.** Fase del anteproyecto en la que se estudia la competencia en el mercado y las posibles mejoras que se pueden implementar respecto a esa competencia.
- **Análisis de las tecnologías a implementar.** Es un proceso necesario ya que en él se deciden las tecnologías que se van a usar para desarrollar el proyecto.
- **Objetivos del proyecto.** Fase del proyecto en la que se especificaron cuáles eran los objetivos principales del proyecto y cuáles los específicos.
- **Búsqueda de roles.** Proceso mediante el cual se obtuvieron los principales roles de la aplicación.
- **Especificación de requisitos.** Especificación de las posibles acciones del usuario dentro de la aplicación web (inicio de sesión, creación de incidencia, ...).
- **Diseño de la interfaz de usuario.** Diseño superficial de la posible interfaz de usuario.
- **Estudio del arte.** Búsqueda de competencia en el mercado y obtención de nuevas posibles ideas.
- **Análisis de las tecnologías a implementar.** Búsqueda de las mejores opciones para cada apartado de desarrollo de la página web.
- **Tareas dedicadas para el desarrollo del proyecto.** Tiempo dedicado al desarrollo de la aplicación.
- **Reuniones.** Reuniones realizadas al final de cada sprint para la obtención de nuevos requisitos.
- **Tiempo de búsqueda de información.** Tiempo dedicado a la búsqueda de información necesaria para el desarrollo del proyecto.
- **Pruebas UX.** Pruebas realizadas a diferentes usuarios externos al proyecto.
- **Documentación final.** Documento explicativo del desarrollo del proyecto.

## 4.2 Descripción de la solución, metodologías y herramientas empleadas

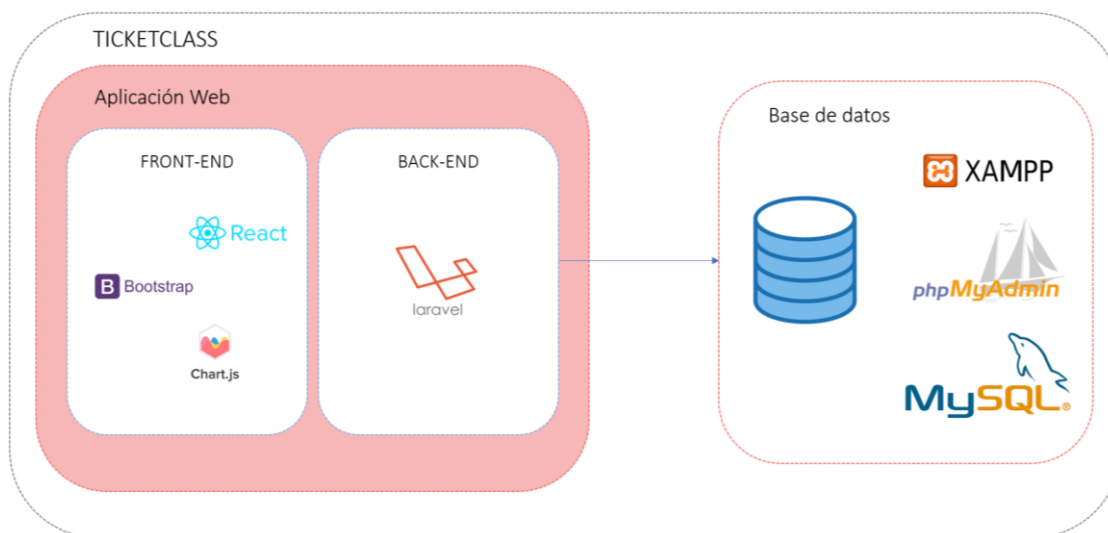
### 4.2.1 Descripción de la solución

El desarrollo de este proyecto da lugar a una **herramienta de gestión de incidencias** desarrollada para facilitar a sus usuarios la gestión de las incidencias que se vayan generando en el entorno en el que se encuentren.

La aplicación consta de dos tipos de usuarios, **supervisor** y **técnico**. Cada uno de ellos tiene una función dentro de la aplicación que se explicará más adelante.

El proyecto está desarrollado bajo dos tecnologías diferentes: Laravel y React. Cada una de ellas está dedicada a una parte diferente de la aplicación.

- **React**: librería JavaScript dedicada sobre todo para el diseño y la parte *front-end*<sup>2</sup> de la aplicación.
- **Laravel**: framework desarrollado en PHP para el manejo de la parte back-end de una aplicación web. Por defecto, viene preinstalado para utilizar en ella Vue.js como librería front-end, pero en el caso de este proyecto, *Vue.js* se ha cambiado por *React* ya que tiene muchas ventajas que se explicarán más adelante.



*Ilustración 6. Arquitectura Ticketclass*

#### 4.2.1.1 Casos de uso

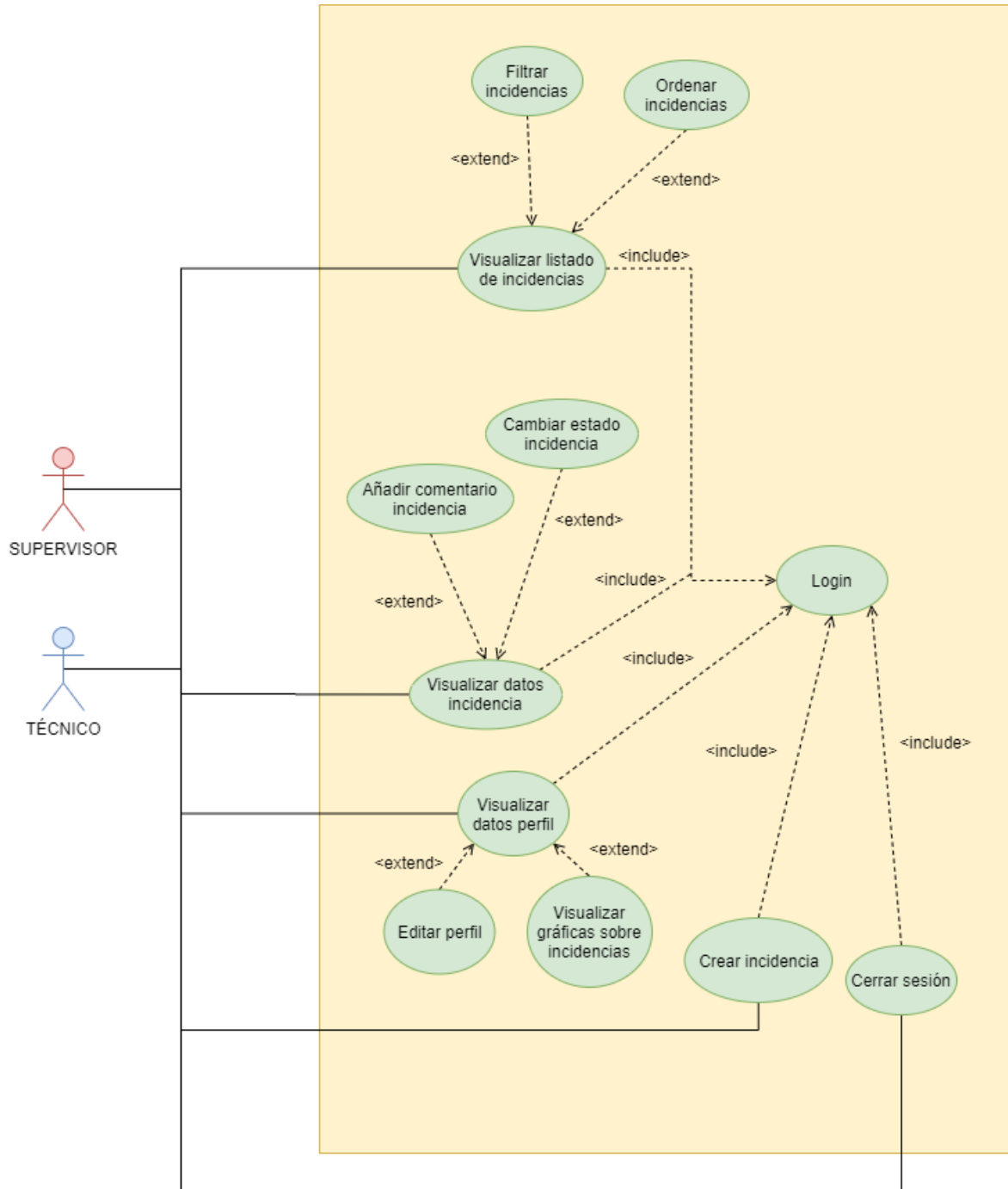
Dentro de este apartado se van a mostrar los diferentes **casos de uso** que tiene la aplicación.

Para entenderlo mejor, un caso de uso es la **descripción de una acción o actividad dentro de la aplicación**, describiendo también los pasos a seguir para llevarla a cabo.

---

<sup>2</sup> **Front-end**. Parte visual de una aplicación web y a través de la cual el usuario la maneja.

**DIAGRAMAS DE CASOS DE USO**



*Ilustración 7. Diagrama de casos de uso Técnico-Supervisor*

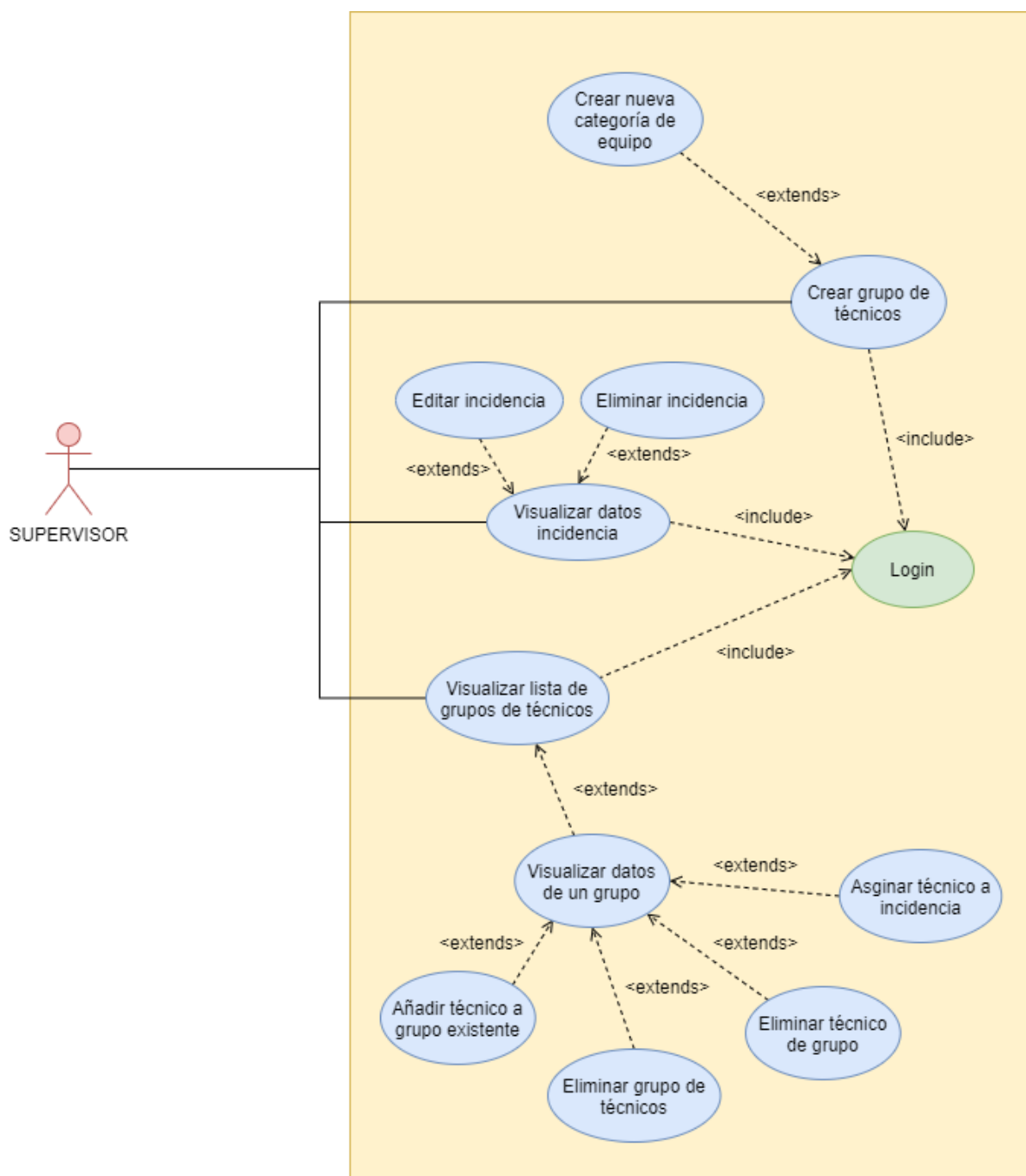
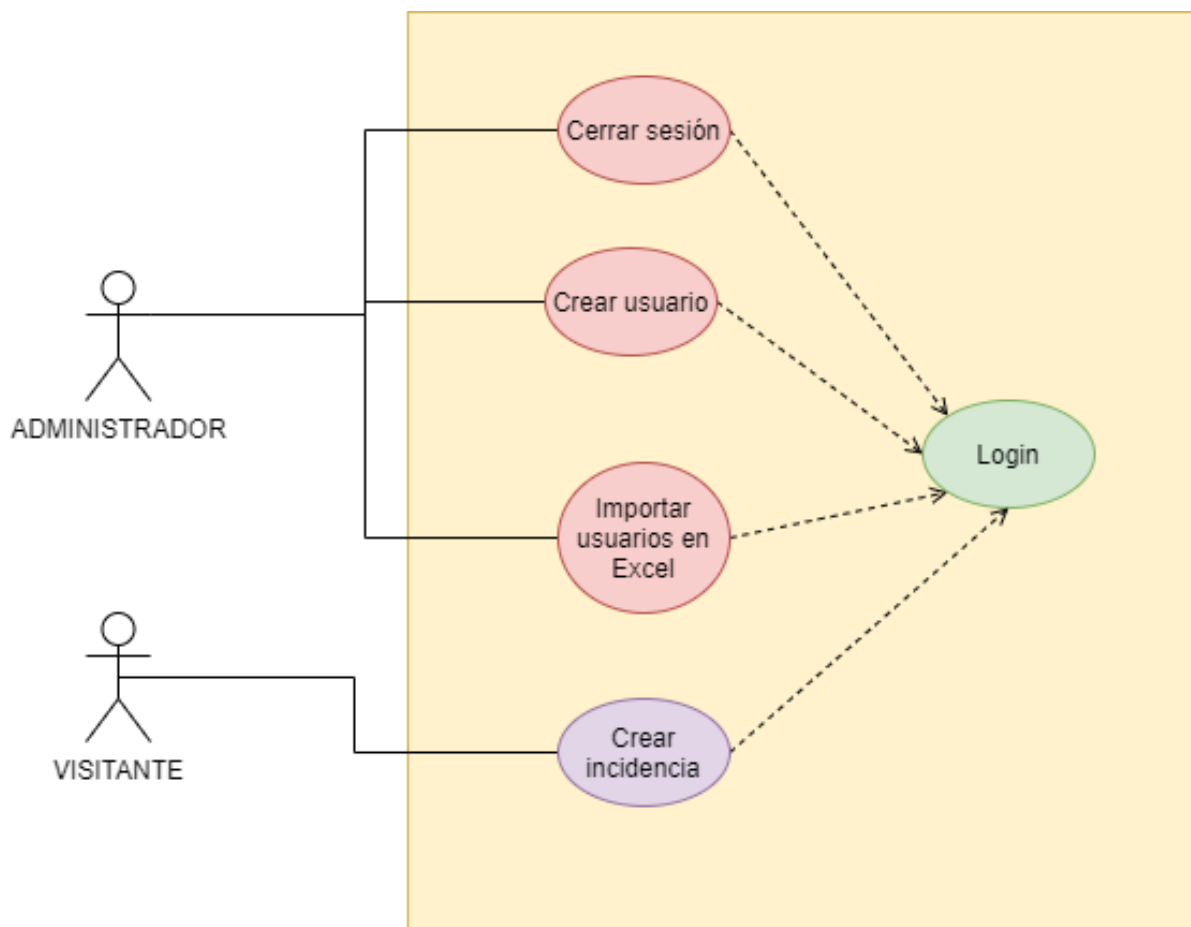


Ilustración 8. Diagrama de casos de uso Supervisor



*Ilustración 9. Diagrama de casos de uso Administrador-Visitante*

La tabla que se va a seguir es la siguiente (Guía para la redacción de casos de uso | Marco de Desarrollo de la Junta de Andalucía, s.f.):

CU-XX	Nombre del caso de uso
<b>Dependencias</b>	Requisitos de los que depende el caso de uso
<b>Precondición</b>	Precondición necesaria para llevar a cabo el caso de uso
<b>Descripción</b>	Explicación general del caso de uso
<b>Secuencia normal</b>	Secuencia y explicación de caso uno de los pasos a seguir para llevar a cabo el caso de uso

<b>Postcondición</b>	Postcondición necesaria después de realizar el caso de uso
<b>Excepciones</b>	Casos que se pueden dar durante la realización del caso de uso
<b>Comentarios</b>	Sección en la que se ponen comentarios necesarios para un mejor entendimiento del caso de uso

*Tabla 2. Tabla ejemplo para los casos de uso*

### **CASOS DE USO**

<b>CU-00</b>	<b>Iniciar sesión (Login)</b>	
<b>Precondición</b>	Estar registrado en la base de datos de la aplicación.	
<b>Descripción</b>	El sistema permite al usuario iniciar sesión con sus credenciales para poder acceder a la aplicación.	
<b>Secuencia normal</b>	<b>1</b>	Navegar a la página principal de la aplicación.
	<b>2</b>	Introducir su ID y su contraseña en los campos correspondientes.
	<b>3</b>	Si los campos son correctos, el usuario habrá iniciado sesión correctamente.
<b>Postcondición</b>	El usuario debe de tener la capacidad de navegar por la aplicación sin ningún problema.	
<b>Excepciones</b>	<b>2</b>	Si los campos introducidos son incorrectos, saldrá un mensaje de error y el usuario no podrá acceder a la aplicación.
<b>Comentarios</b>	--	

*Tabla 3. CU-00 Iniciar sesión*



<b>CU-01</b>	<b>Cerrar sesión</b>	
<b>Precondición</b>	Estar loggeado en la aplicación	
<b>Descripción</b>	El sistema permite al usuario cerrar sesión dentro de la aplicación.	
<b>Secuencia normal</b>	<b>1</b>	Estando dentro de la aplicación, pulsar el botón para abrir el <i>sidebar</i> .
	<b>2</b>	Para cerrar sesión, pulsar el botón rojo que se encuentra en la parte inferior del <i>sidebar</i> y en el que pone 'Cerrar sesión'.
	<b>3</b>	Si todo ha ido correctamente, el usuario ha cerrado sesión correctamente.
<b>Postcondición</b>	No poder acceder a ninguna de las URLs internas de la aplicación.	
<b>Excepciones</b>	--	
<b>Comentarios</b>	--	

Tabla 4. CU-01 Cerrar sesión

<b>CU-02</b>	<b>Visualizar listado incidencias</b>	
<b>Precondición</b>	Haber iniciado sesión en la aplicación.	
<b>Descripción</b>	El sistema permite al usuario visualizar un listado de las incidencias de las que está a cargo.	
<b>Secuencia normal</b>	<b>1</b>	CU-00
	<b>2</b>	Acceder al apartado de incidencias del <i>sidebar</i> .
<b>Postcondición</b>	Deberá aparecer en pantalla la tabla con las correspondientes incidencias.	

<b>Excepciones</b>	En el caso de que no haya incidencias, la tabla aparecerá vacía.
<b>Comentarios</b>	Dentro de la propia página de incidencias existen una serie de <i>tabs</i> , en los que pulsando la opción de ' <b>Mis incidencias</b> ', el sistema llevará al usuario a la misma página que pulsado la opción indicada del <i>sidebar</i> .

Tabla 5. CU-02 Visualizar listado incidencias

<b>CU-03</b>	<b>Ordenar incidencias</b>	
<b>Precondición</b>	Haber iniciado sesión en la aplicación y haber pulsado la opción de Incidencias del <i>sidebar</i> , o la opción ' <b>Mis incidencias</b> ' de los <i>tabs</i> .	
<b>Descripción</b>	El sistema permite al usuario ordenar las incidencias según la propiedad que seleccione.	
<b>Secuencia normal</b>	<b>1</b>	CU-00
	<b>2</b>	CU-03
	<b>3</b>	Seleccionar en el dropdown que se encuentra al lado de ' <b>Ordenar por</b> ' cualquiera de las opciones disponibles.
	<b>4</b>	La tabla cambiará visualizando las incidencias en el orden seleccionado.
<b>Postcondición</b>	Las incidencias de la tabla deben de aparecer ordenadas según la opción seleccionada.	
<b>Excepciones</b>	--	
<b>Comentarios</b>	--	

Tabla 6. CU-03 Ordenar incidencias

<b>CU-04</b>	<b>Filtrar incidencias</b>	
<b>Precondición</b>	Haber realizado el caso de uso CU-03	
<b>Descripción</b>	El sistema permite al usuario filtrar las incidencias dependiendo de la opción seleccionada en el 'Ordenar por'.	
<b>Secuencia normal</b>	<b>1</b>	CU-03.
	<b>2</b>	Seleccionar cualquiera de las opciones de filtro que aparecen a la derecha del dropdown de 'Ordenar por'
<b>Postcondición</b>	Solo aparecerán en la tabla las incidencias coincidentes con los filtros seleccionados.	
<b>Excepciones</b>	<b>2</b>	Si no se selecciona ningún filtro, aparecerán todas las incidencias ordenadas por la opción seleccionada.
<b>Comentarios</b>	--	

Tabla 7. CU-04 Filtrar incidencias

<b>CU-05</b>	<b>Visualizar datos incidencia</b>	
<b>Precondición</b>	Haber iniciado sesión en la aplicación.	
<b>Descripción</b>	El sistema permite al usuario visualizar los datos de una incidencia específica.	
<b>Secuencia normal</b>	<b>1</b>	Acceder a la vista de todas las incidencias.
	<b>2</b>	Pulsar sobre el nombre de una de ellas.
	<b>3</b>	Al pulsar, se redirigirá a la página con todos los datos de la incidencia.

<b>Postcondición</b>	Los datos de la incidencia se deben haber cargado correctamente en la nueva página.
<b>Excepciones</b>	--
<b>Comentarios</b>	--

Tabla 8. CU-05 Visualizar datos incidencia

<b>CU-06</b>	<b>Cambiar estado de incidencia</b>	
<b>Precondición</b>	Haber iniciado sesión en la aplicación.	
<b>Descripción</b>	El sistema permite al usuario cambiar el estado de cualquier incidencia.	
<b>Secuencia normal</b>	<b>1</b>	CU-05
	<b>2</b>	Cambiar el estado de la incidencia a través de un dropdown.
	<b>3</b>	El estado de la incidencia se cambia y se actualizan automáticamente los datos de la página.
<b>Postcondición</b>	<p>Los datos de la página se actualizan automáticamente una vez se ha cambiado de estado a la incidencia</p> <p>El sistema envía un correo a las personas responsables de dicha incidencia informándoles de que el estado de la incidencia ha cambiado.</p>	
<b>Excepciones</b>	--	
<b>Comentarios</b>	--	

Tabla 9. CU-06 Cambiar estado de incidencia

<b>CU-07</b>	<b>Añadir comentario a incidencia</b>	
<b>Precondición</b>	Acceder a la página de la información de la incidencia.	
<b>Descripción</b>	El sistema permite al usuario enviar un comentario sobre una incidencia específica.	
<b>Secuencia normal</b>	<b>1</b>	CU-05.
	<b>2</b>	Acceder a la opción de comentarios.
	<b>3</b>	Introducir el comentario en el campo de texto.
	<b>4</b>	Pulsar el botón situado a la derecha del campo de texto para enviar el comentario
<b>Postcondición</b>	Los usuarios encargados de la incidencia y lo que hayan comentado recibirán un mensaje con la información del nuevo comentario añadido.	
<b>Excepciones</b>	--	
<b>Comentarios</b>	--	

Tabla 10. CU-07 Añadir comentario a incidencia

<b>CU-08</b>	<b>Crear incidencia</b>	
<b>Precondición</b>	Haber iniciado sesión en la aplicación.	
<b>Descripción</b>	El sistema permite al usuario generar una incidencia.	
<b>Secuencia normal</b>	<b>1</b>	Acceder al apartado de incidencias.
	<b>2</b>	Seleccionar la opción 'Crear nueva incidencia'.
	<b>3</b>	Rellenar los campos correctamente.
	<b>4</b>	Seleccionar el botón crear incidencia.

	<b>5</b>	Si todo ha ido bien, aparece una ventana modal para confirmar.
<b>Postcondición</b>	La incidencia se guarda correctamente en la base de datos de la aplicación.	
<b>Excepciones</b>	<b>3</b>	Si los campos no se rellenan correctamente, aparecerán mensajes de error y no dejará crear la incidencia.
<b>Comentarios</b>	--	

Tabla 11. CU-08 Crear incidencia

<b>CU-09</b>	<b>Editar incidencia</b>	
<b>Precondición</b>	El usuario que edita la incidencia tiene que ser el creador de dicha incidencia o el supervisor.	
<b>Descripción</b>	El sistema permite al usuario editar una incidencia ya existente.	
<b>Secuencia normal</b>	<b>1</b>	CU-05.
	<b>2</b>	Seleccionar 'Editar incidencia'.
	<b>3</b>	Cambiar los campos que se quieran editar.
	<b>4</b>	Guardar cambios pulsando el botón de 'Editar incidencia'.
<b>Postcondición</b>	Los cambios en la incidencia se guardan en la base de datos.	
<b>Excepciones</b>	<b>3</b>	Si no se introducen los campos correctamente, se mostrarán mensajes de error hasta que los datos sean los correctos.
<b>Comentarios</b>	--	

Tabla 12. CU-09 Editar incidencia

<b>CU-10</b>	<b>Eliminar incidencia</b>	
<b>Precondición</b>	El usuario que elimina la incidencia tiene que ser el creador de dicha incidencia o el supervisor.	
<b>Descripción</b>	El sistema permite al usuario eliminar una incidencia.	
<b>Secuencia normal</b>	<b>1</b>	CU-05.
	<b>2</b>	Seleccionar la opción de 'Eliminar incidencia'.
	<b>3</b>	Se abrirá una modal para confirmar la eliminación de la incidencia.
<b>Postcondición</b>	La incidencia se elimina correctamente de la base de datos de la aplicación.	
<b>Excepciones</b>	--	
<b>Comentarios</b>	--	

Tabla 13. CU-10 Eliminar incidencia

<b>CU-11</b>	<b>Asignar técnico o grupo a incidencia</b>	
<b>Precondición</b>	El usuario debe tener el rol de supervisor dentro de la aplicación para poder asignar una incidencia.	
<b>Descripción</b>	El sistema permite al supervisor asignar la incidencia a un técnico.	
<b>Secuencia normal</b>	<p>Puede darse en dos casos, es el mismo proceso:</p> <ol style="list-style-type: none"> <li>1. CU-08.</li> <li>2. CU-09.</li> </ol>	

<b>Postcondición</b>	El usuario asignado recibe un correo notificándole que se le ha asignado una nueva incidencia.
<b>Excepciones</b>	--
<b>Comentarios</b>	--

Tabla 14. CU-11 Asignar técnico o grupo a incidencia

<b>CU-12</b>	<b>Visualizar lista de equipos (grupos de técnicos)</b>	
<b>Precondición</b>	El usuario debe tener el rol de supervisor.	
<b>Descripción</b>	El sistema permite al usuario visualizar los grupos de técnicos de los que está a cargo.	
<b>Secuencia normal</b>	<b>1</b>	Acceder al apartado incidencias del <i>sidebar</i> .
	<b>2</b>	Entrar a la sección 'Mis grupos de técnicos'.
<b>Postcondición</b>	Se muestra en pantalla el listado de los grupos de técnicos de los que está a cargo el supervisor.	
<b>Excepciones</b>	--	
<b>Comentarios</b>	--	

Tabla 15. CU-12 Visualizar lista de equipos

<b>CU-13</b>	<b>Crear grupo de técnicos</b>	
<b>Precondición</b>	El usuario debe tener el rol de supervisor.	
<b>Descripción</b>	El sistema permite al supervisor crear un equipo de técnicos.	
<b>Secuencia normal</b>	<b>1</b>	CU-12
	<b>2</b>	Acceder a la parte inferior de la pantalla.



	<b>3</b>	Rellenar el formulario correctamente.
	<b>4</b>	Pulsar el botón 'Crear grupo'.
<b>Postcondición</b>	El grupo se tiene que guardar correctamente en la base de datos y mostrarse en la lista de grupos automáticamente	
<b>Excepciones</b>	<b>3</b>	Si datos de los campos no son correctos, el sistema no dejará al usuario crear el grupo.
<b>Comentarios</b>	--	

Tabla 16. CU-13 Crear grupo de técnicos

<b>CU-14</b>	<b>Visualizar datos de un grupo</b>	
<b>Precondición</b>	El usuario debe tener el rol de supervisor.	
<b>Descripción</b>	El sistema permite al supervisor visualizar los datos de un grupo de técnicos, incluyendo el listado de técnicos asociado.	
<b>Secuencia normal</b>	<b>1</b>	CU-12
	<b>2</b>	Pulsar sobre el grupo del que se quieren ver los datos.
<b>Postcondición</b>	Actualización de la parte de la derecha de la página con los datos del grupo seleccionado.	
<b>Excepciones</b>	--	
<b>Comentarios</b>	--	

Tabla 17. CU-14 Visualizar datos de un grupo

<b>CU-15</b>	<b>Añadir técnico a grupo existente</b>	
<b>Precondición</b>	El usuario debe tener el rol de supervisor.	
<b>Descripción</b>	El sistema permite al supervisor añadir un técnico al grupo de técnicos correspondiente.	

<b>Secuencia normal</b>	<b>1</b>	CU-14
	<b>2</b>	Introducir un técnico en el input que se encuentra en la sección de la lista de técnicos.
	<b>3</b>	Una vez encontrado el técnico, presionar el botón 'Añadir técnico'.
<b>Postcondición</b>	El técnico añadido se debe de mostrar en la lista de técnicos de la pantalla y se debe guardar correctamente en la base de datos.	
<b>Excepciones</b>	<b>3</b>	Si el técnico seleccionado pertenece ya al grupo, se mostrar una modal indicándolo.
<b>Comentarios</b>	--	

Tabla 18. CU-15 Añadir técnico a grupo existente

<b>CU-16</b>	<b>Eliminar técnico de grupo</b>	
<b>Precondición</b>	El usuario debe tener el rol de supervisor.	
<b>Descripción</b>	El sistema permite al supervisor eliminar un técnico del grupo seleccionado.	
<b>Secuencia normal</b>	<b>1</b>	CU-14.
	<b>2</b>	Pulsar el botón rojo que se encuentra a la derecha del nombre del usuario.
	<b>3</b>	Se abre una ventana modal para confirmar la eliminación del técnico.
<b>Postcondición</b>	El usuario debe desaparecer de la lista de técnicos y de la base de datos.	
<b>Excepciones</b>	--	
<b>Comentarios</b>	--	

Tabla 19. CU-16 Eliminar técnico de grupo

<b>CU-17</b>	<b>Eliminar grupo de técnicos</b>	
<b>Precondición</b>	El usuario debe tener el rol de supervisor.	
<b>Descripción</b>	El sistema permite al supervisor eliminar un grupo de usuarios al completo.	
<b>Secuencia normal</b>	<b>1</b>	CU-14.
	<b>2</b>	Seleccionar los tres puntos de la parte superior derecha del cuadro de información del grupo.
	<b>3</b>	Seleccionar eliminar incidencia.
	<b>4</b>	Se abre una ventana modal para confirmar la acción.
<b>Postcondición</b>	Se deben eliminar toda la información sobre el grupo de la base de datos, las relaciones de los técnicos con el grupo y se debe actualizar el listado de grupos del supervisor.	
<b>Excepciones</b>	<b>3</b>	La funcionalidad aún no está implementada.
<b>Comentarios</b>	--	

Tabla 20. CU-17 Eliminar grupo de técnicos

<b>CU-18</b>	<b>Crear nueva categoría de equipo</b>	
<b>Precondición</b>	El usuario debe tener el rol de supervisor.	
<b>Descripción</b>	El sistema permite al supervisor crear una nueva categoría de grupo.	
<b>Secuencia normal</b>	<b>1</b>	CU-13
	<b>2</b>	En vez de seleccionar una categoría, introducir una nueva en el input de la derecha al dropdown de 'Categoría'.
	<b>3</b>	Crear grupo con la nueva categoría.

<b>Postcondición</b>	El grupo se debe crear correctamente con la nueva categoría.
<b>Excepciones</b>	--
<b>Comentarios</b>	--

Tabla 21. CU-18 Crear nueva categoría de equipo

<b>CU-19</b>	<b>Visualizar datos perfil</b>	
<b>Precondición</b>	Haber iniciado sesión en la aplicación.	
<b>Descripción</b>	El sistema permite al usuario visualizar sus datos dentro de la aplicación.	
<b>Secuencia normal</b>	<b>1</b>	Acceder al apartado de perfil en el <i>sidebar</i> .
<b>Postcondición</b>	El sistema debe de mostrar una tabla con los datos generales del usuario.	
<b>Excepciones</b>	--	
<b>Comentarios</b>	--	

Tabla 22. CU-19 Visualizar datos perfil

<b>CU-20</b>	<b>Editar perfil</b>	
<b>Precondición</b>	Haber iniciado sesión en la aplicación.	
<b>Descripción</b>	El sistema permite al usuario editar varios datos de su perfil.	
<b>Secuencia normal</b>	<b>1</b>	CU-19.
	<b>2</b>	Pulsar la opción de editar perfil, debajo de la foto del usuario.
	<b>3</b>	Rellenar los campos que se deseen editar.

	<b>4</b>	Pulsar el botón correspondiente a los campos modificados para guardar los cambios.
<b>Postcondición</b>	Los datos del usuario se deben de cambiar correctamente tanto en la página como en la base de datos.	
<b>Excepciones</b>	<b>3</b>	El sistema lanzará un mensaje de error si los campos no están correctamente rellenados. No se finalizará el proceso de modificación de perfil hasta que los campos estén correctamente completados.
<b>Comentarios</b>	--	

Tabla 23. CU-20 Editar perfil

<b>CU-21</b>	<b>Visualizar gráficas sobre datos de incidencias</b>	
<b>Precondición</b>	Haber iniciado sesión en la aplicación.	
<b>Descripción</b>	El sistema permite al usuario ver gráficas sobre las incidencias de las que está a cargo. Dependiendo del rol del usuario se podrán visualizar un tipo de gráficas u otras.	
<b>Secuencia normal</b>	<b>1</b>	CU-19.
	<b>2</b>	Pulsar la opción de 'Visualizar gráficas' que se encuentra debajo de la foto del usuario.
<b>Postcondición</b>	Los datos mostrados en las gráficas tienen que ser conformes a los reales.	
<b>Excepciones</b>	--	
<b>Comentarios</b>	--	

Tabla 24. CU-21 Visualizar gráficas sobre datos de las incidencias

<b>CU-22</b>	<b>Crear usuario</b>
--------------	----------------------

<b>Precondición</b>	El rol del usuario tiene que ser administrador.	
<b>Descripción</b>	El sistema permite al administrador crear un usuario en la aplicación.	
<b>Secuencia normal</b>	<b>1</b>	Iniciar sesión con la cuenta de usuario.
	<b>2</b>	Rellenar los campos correctamente.
	<b>3</b>	Una vez rellenados todos los campos, pulsar el botón de 'Crear usuario'.
<b>Postcondición</b>	El usuario se debe crear y almacenar en la base de datos correctamente.	
<b>Excepciones</b>	<b>2</b>	Si los campos no se rellenan correctamente, saldrá un mensaje de error y no se realizará la creación del usuario.
<b>Comentarios</b>	--	

Tabla 25. CU-22 Crear usuario

#### 4.2.1.2 Conexión a Base de Datos

Los datos usados por la aplicación se almacenan en una base de datos manejada por XAMPP.

**XAMPP** es un paquete de software libre para el **manejo de bases de datos en local y servidores Apache**. Esta es la explicación de su nombre, ya que lo forman las siglas de **X** (compatible con todos los Sistemas Operativos), **A** (Apache), **M** (MySQL o MariaDB) y **PP** (PHP y Perl).

Para conseguir la conexión con la base de datos, Laravel proporciona un archivo de configuración (.env) en el que se introducen los datos necesarios para establecer conexión con la base de datos en local.

```
APP_NAME=Laravel
APP_ENV=local
APP_KEY=base64:PiJg6xhIIWe1SueQMUYWnN3euEJYIdAy86eePZZ+qc8=
APP_DEBUG=true
APP_URL=http://localhost

LOG_CHANNEL=stack

DB_CONNECTION=mysql
DB_HOST=localhost
DB_PORT=3306
DB_DATABASE=TFG_Ticketing_System
DB_USERNAME=root
DB_PASSWORD=
```

*Ilustración 10. Fichero de configuración .env*

#### 4.2.1.3 Patrón de arquitectura MCV (Model-Controller-View)

**MCV (Modelo-Controllador-Vista)** es un **patrón de arquitectura** que se encarga de distinguir entre lo que son los datos y la lógica de negocio, dando lugar a tres tipos de componentes: **modelos, vistas y controladores**.

#### 4.2.1.4 Estructura del proyecto

En este apartado se va a exponer de forma minimizada la estructura de carpetas del proyecto.

En el apartado 0, 'Estructura completa del proyecto', se expone de una manera más detallada la estructura del proyecto junto a los archivos más importantes de cada nivel.

- App
  - Http
    - Controllers
      - Auth
      - DataGraphs
      - Incidencias
      - Pages
      - Users
      - Utilidades
    - Middleware
  - Mail
- Bootstrap
- Config
- Database
  - Factories
  - Migrations
  - Seeds
- Git
- Node\_modules

- Public
  - Bootstrap-4.4.1-dist
  - Css
  - Images
  - Js
  - Json
  - Sass
- Resources
  - Js
    - Components
      - AutocompleteInput
      - Button
      - DataCard
      - Dropdown
      - Input
      - Modal
      - Navbar
      - Selectbox
      - Tabs
      - UploadFile
    - Model
    - Pages
      - IncidenciasPage
        - CommentPage
        - IncidenciaViewPage
        - TabOptions
          - CreateIncidenciaPage
          - MostrarIncidenciasPage
          - TechnicalGroupsPageIncidenciasPage.sc  
ss
      - Login
      - MainPage
      - PerfilPage
        - GraphsPage
        - SettingsPage
    - Utilities
      - Graphics
      - Incidencias
    - Widgets
  - Lang
  - Sass
  - Sketches
  - Views



- Wireframes
- Routes
- Storage
- Tests

#### **4.2.1.5 Diagrama entidad-relación**

Un **diagrama entidad-relación** es un diagrama que expone al usuario la estructura de la base de datos y las relaciones que existen entre las entidades de ésta.

Son diagramas que se usan para:

- El diseño de una base de datos.
- Soluciones para problemas en una base de datos.
- Sistemas de información empresarial.
- Educación.
- Investigación.

Su principal función en el proyecto ha sido la **de tener un diseño previo a la creación de la base de datos**. Ha sido muy útil tener el diagrama E-R<sup>3</sup>, ya que es muy fácil realizar modificaciones sobre la base de datos una vez creada de manera visual y así evitar posibles errores.

---

<sup>3</sup> **E-R**: entidad-relación

A continuación, se muestra el diagrama E-R de la aplicación:

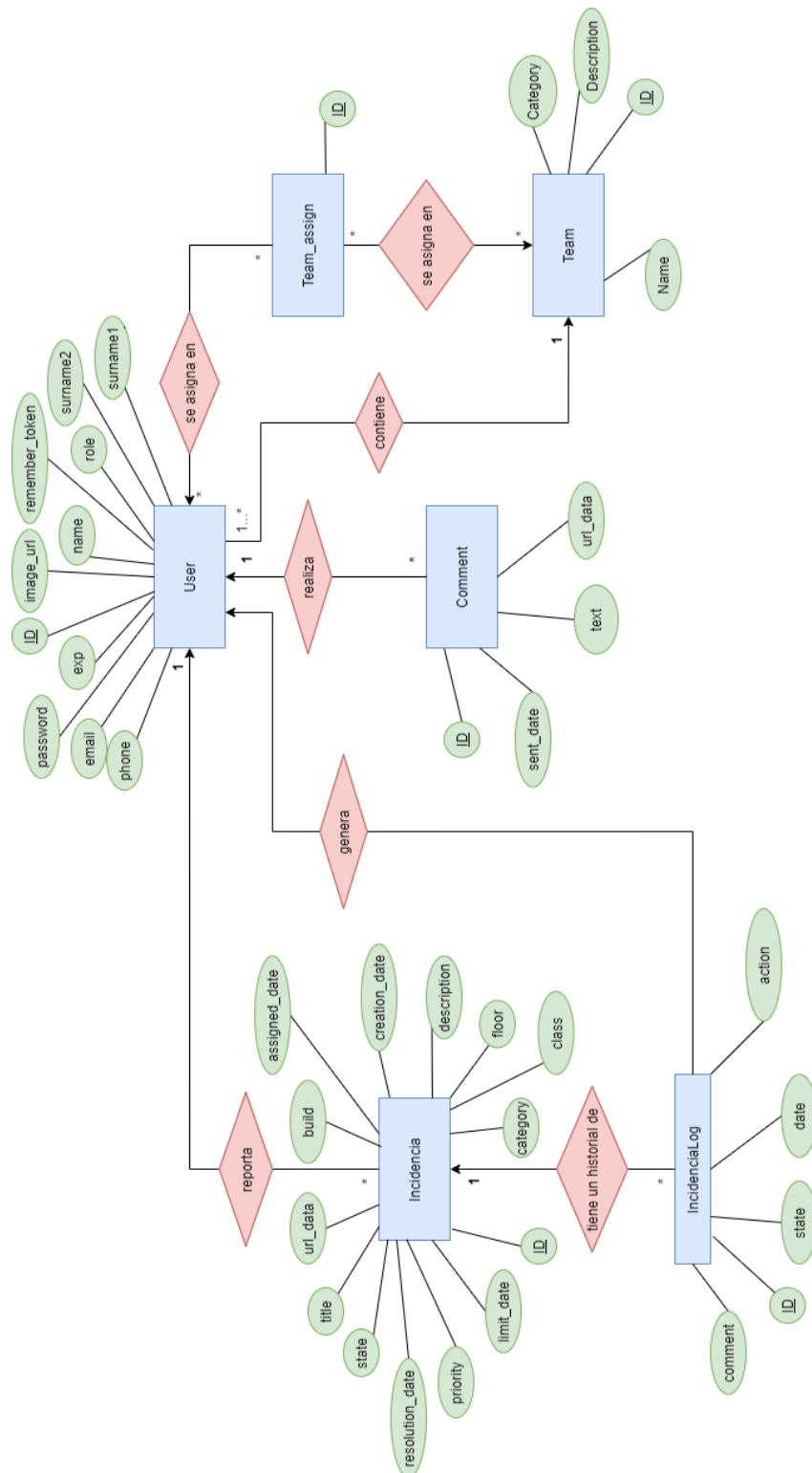


Ilustración 11. Diagrama entidad-relación

#### 4.2.1.6 Estructura de la base de datos

La estructura de las tablas que forman la base de datos usada por la aplicación son las siguientes:

**Tabla "COMMENTS"**

Nombre columna	Tipo de dato	Descripción
ID	bigInt(20)	Identificador del comentario.
incidencia_id	int(11)	Incidencia a la que está asociado el comentario.
user_id	int(11)	Usuario que genera el comentario.
text	text	Texto del comentario.
send_date	datetime	Fecha en la que se realiza el comentario.
url_data	long_text	Url al archivo adjunto (Sin implementar).

**Tabla "INCIDENCIAS"**

Nombre columna	Tipo de dato	Descripción
ID	bigint(20)	Identificador de la incidencia.
group_id	int(11)	--
id_reporter	int(11)	Identificador del usuario que ha reportado la incidencia.
id_assigned	int(11)	Identificador del usuario al que se le asigna la incidencia.
supervisor	int(11)	Identificador del supervisor encargado de revisar la incidencia.
id_team	int(11)	Incidencia

<b>title</b>	varchar(191)	Título principal de la incidencia.
<b>description</b>	longtext	Descripción mas detallada de lo que trata la incidencia.
<b>category</b>	varchar(191)	Categoría a la que se asocia la incidencia.
<b>build</b>	varchar(191)	Edificio del campus en el que se produce la incidencia.
<b>floor</b>	int(11)	Piso del edificio.
<b>class</b>	varchar(191)	Clase o aula en la que se produce la incidencia.
<b>url_data</b>	varchar(191)	Url de los datos adjuntos a la incidencia (Sin implementar).
<b>creation_date</b>	datetime	Fecha de creación de la incidencia.
<b>limit_date</b>	datetime	Límite de resolución de la incidencia.
<b>assigned_date</b>	datetime	Fecha de asignación de la incidencia.
<b>resolution_date</b>	datetime	Fecha de resolución de la incidencia.
<b>priority</b>	enum('critical', 'important', 'trivial')	Prioridad establecida a la incidencia.
<b>state</b>	enum('todo', 'doing', 'blocked', 'done')	Estado actual en el que se encuentra la incidencia.

**Tabla “INCIDENCIA\_LOGS”**

Nombre columna	Tipo de dato	Descripción
<b>id</b>	bigint(11)	Identificador del log
<b>incidencia_id</b>	int(11)	Identificador de la incidencia a la que va asociado el log.
<b>user_id</b>	int(11)	Identificador del usuario que genera el log.
<b>state</b>	enum ('todo', 'doing', 'blocked', 'done')	Estado en el que se encuentra la incidencia en el momento en el que se genera el log.
<b>comment</b>	text	Text que el usuario puede añadir al log.
<b>date</b>	datetime	Fecha en la que se genera el log.
<b>action</b>	enum ('Nuevo comentario', 'Cambio de estado', 'Editar incidencia', 'Crear incidencia')	Acción que se está realizando al generar el log.

**Tabla “TEAMS”**

Nombre columna	Tipo de dato	Descripción
<b>id</b>	bigint(20)	Identificador del equipo.
<b>name</b>	varchar(191)	Nombre del equipo
<b>description</b>	varchar(191)	Descripción del equipo
<b>category</b>	varchar(191)	Categoría a la que pertenece el equipo.
<b>id_supervisor</b>	int(11)	Id del supervisor que está a cargo del equipo.

**Tabla “TEAM\_ASSIGNS”**

Nombre columna	Tipo de dato	Descripción
id	bigint(20)	Identificador de la asignación.
id_user	int(11)	Identificador del técnico.
id_team	int(11)	Identificador del equipo.

**Tabla “USERS”**

Nombre columna	Tipo de dato	Descripción
id	bigint(20)	Identificador del usuario.
name	varchar(191)	Nombre del usuario.
surname1	varchar(191)	Primer apellido del usuario.
surname2	varchar(191)	Segundo apellido del usuario.
exp	varchar(191)	Código del usuario.
password	varchar(191)	Contraseña para iniciar sesión.
email	varchar(191)	Email asociado al usuario.
phone	varchar(191)	Nº de teléfono del usuario.
role	enum('admin', 'supervisor', 'technical', 'visitor')	Rol del usuario dentro de la aplicación
image_url	varchar(191)	Imagen de perfil.
remeber_token	varchar(100)	Token de inicio de sesión.

#### 4.2.1.7 Migraciones, seeders y factories

Es muy llamativo usar Laravel para la gestión de bases de datos debido a la cantidad de funcionalidades que tiene para su manejo.

Tres de estas funcionalidades son las **migraciones**, los **seeders** y las **factories**.

Las **migraciones** son ficheros programados en PHP que contienen la **estructura de una de las tablas** de la base de datos. El uso principal de las migraciones, tal y como dice su nombre, cuando se produce cuando hay que hacer una **migración de una base de datos antigua a una nueva**.

Como la estructura de las tablas está almacenada en estos ficheros, simplemente con configurar correctamente la conexión con la nueva base de datos y ejecutar **php artisan migrate:refresh**, serviría para migrar toda la estructura de la base de datos antigua a la nueva.

```
public function up()
{
    Schema::create('comments', function (Blueprint $table) {
        $table->bigIncrements('id');
        $table->integer('incidencia_id');
        $table->integer('user_id');
        $table->text('text');
        $table->dateTime('sent_date');
        $table->longText('url_data')->nullable();
    });
}
```

*Ilustración 12. Ejemplo 'Migración'*

En la Ilustración 12. Ejemplo 'Migración', se está creando la tabla *comments* en la base de datos con *id*, *incidencia\_id*, *user\_id*, *text*, *sent\_date* y *url\_data* como columnas.

Los **seeders** a su vez, sirven para **rellenar una tabla específica con datos**. Para ejecutar los seeders en Laravel es necesario ejecutar **php artisan migrate:refresh --seed** en la línea de comandos.

```
public function run()
{
    DB::table('incidencias')->insert([
        'group_id' => 1,
        'id_reporter' => 3,
        'id_assigned' => 5,
        'id_team' => null,
        'title' => 'Problema software instalado',
        'description' => 'El problema se encuentra en que el ordenador XX no tiene
instalado el Software xxxx para que el alumno que lo use pueda seguir la clase al mis
mo ritmo que todos los demás.',
        'category' => 'Wi-Fi',
        'build' => 'C',
        'floor' => 3,
        'class' => 'C307',
        'url_data' => '',
        'creation_date' => '2014-10-25 20:00:00',
        'limit_date' => '2009-12-30 14:34:29',
    ]);
}
```

```
        'assigned_date' => null,  
        'resolution_date' => null,  
        'priority' => 'critical',  
        'state' => 'todo'  
    });  
}
```

*Ilustración 13. Ejemplo 'Seeder'*

Las **factories** son un tipo de seeder que sirven para insertar **conjuntos enormes de datos random en la base de datos**. En el caso de este proyecto, se han usado para generar incidencias randomizadas y así poder trabajar con ellas.

Al igual que los seeders, se ejecutan con el comando **php artisan migrate:refresh --seed**.

#### **4.2.1.8 Autenticación de usuario**

Nada más entrar en la aplicación, se muestra en pantalla la página de **Login**. A través de ella el usuario puede **iniciar sesión** si posee las credenciales necesarias.

Para el inicio de sesión ha sido necesaria la implementación de una librería con la que poder crear un **JWT**<sup>4</sup> y de esta manera tener un control sobre el usuario que ha iniciado sesión.

Un **JWT** o **JSON Web Token**, es un estándar que permite la **transmisión de información de un usuario entre dos partes** de forma **segura** y otorgando una serie de privilegios, todo en formato **JSON**.

La librería usada es **JWT-auth**. Si el usuario que está intentando iniciar sesión se encuentra guardado en la base de datos, **JWT-auth** va a generar un token **JWT** que va a permitir al usuario navegar por toda la aplicación sin problema. Una vez el token desaparezca por inactividad o por necesidad de volver a generar uno nuevo, el usuario no podrá seguir navegando por la aplicación hasta que inicie sesión de nuevo.

El ciclo de vida de un token **JWT** sería de la siguiente manera:

- 1) Realización de petición **POST**<sup>5</sup> para realizar la búsqueda del usuario dentro de la base de datos.
- 2) Una vez encontrado, se genera el token **JWT** para asociarlo al usuario.
- 3) Dentro de la aplicación, todas las peticiones se realizarían con ese token dentro del encabezado → **Authorization: Bearer XXXXXXXX**, donde **Bearer** es el tipo de prefijo y **XXXXXXX** el token almacenado.

---

<sup>4</sup> JWT: JSON Web Token.

<sup>5</sup> **POST**. Es una petición en la que se envía un objeto u entidad a un recurso en concreto, causando cambios o efectos en el servidor.



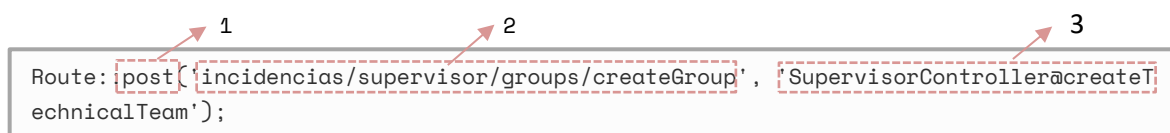
#### 4.2.1.9 Conexión Laravel-React: APIs

La aplicación, al estar desarrollada bajo dos diferentes herramientas (Laravel y React), la conexión entre ellas se ha tenido que realizar mediante **llamadas HTTP** usando **APIs**.

Una **API** es un **conjunto de definiciones y protocolos** que se utiliza para conseguir la **comunicación entre distintos productos**, o en este caso servicios, sin necesidad de saber como funcionan o como están implementados.

Para ello, dentro del proyecto de Laravel, existe una carpeta llamada `api` en la que se encuentran todas las posibles llamadas HTTP del proyecto.

Cada una de estas llamadas está se crea de la siguiente manera:



```
Route::post('incidencias/supervisor/groups/createGroup', 'SupervisorController@createTechnicalTeam');
```

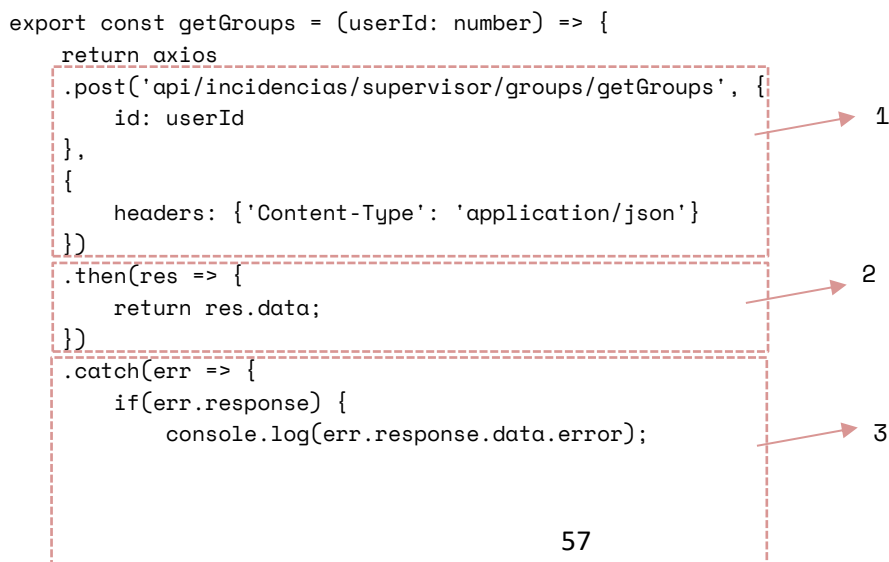
Ilustración 14. Creación petición HTTP

- 1) Tipo de **petición HTTP** que se va a producir. Pueden ser *GET* y *POST*.  
La diferencia entre ellos es básicamente la manera de enviar los datos a la página, ya que *GET* los envía a través de la URL y *POST* de una forma en la que no se pueden ver.
- 2) La **URL** de acceso a la petición HTTP.
- 3) Controlador de la petición HTTP. Aquí es donde se va a manejar toda la petición HTTP:
  - a. Recibe los datos de petición.
  - b. Obtiene datos de respuesta.
  - c. Maneja los datos.
  - d. Devuelve los datos requeridos.

La petición HTTP se realiza desde React mediante la librería **Axios**.

**Axios** es un **cliente HTTP basado en Promesas para JavaScript**, que se puede usar tanto para el front-end como el back-end utilizando *Node.js*. y permite obtener los datos de las peticiones HTTP realizadas con React. (apтивaweb.com, s.f.)

Una petición con esta librería se realiza de la siguiente manera:



```
export const getGroups = (userId: number) => {  
  return axios  
    .post('api/incidencias/supervisor/groups/getGroups', {  
      id: userId  
    },  
    {  
      headers: {'Content-Type': 'application/json'}  
    })  
    .then(res => {  
      return res.data;  
    })  
    .catch(err => {  
      if(err.response) {  
        console.log(err.response.data.error);  
      }  
    });  
}
```

```
console.log(err.response.status);  
} else if (err.request) {  
  console.log(err.request);
```

```
  } else  
    console.log(err);  
})
```

Ilustración 15. Realización de petición HTTP (React)

- 1) Petición *HTTP* junto a los *headers* correspondientes. En este caso, la petición va a devolver un fichero *JSON*, por lo que *header Content-type* tiene que ser de tipo *JSON*.
- 2) Se obtienen los datos devueltos por la petición *HTTP*.
- 3) En caso de que la petición dé error, se informará por consola del error que se ha producido.

#### 4.2.1.10 Diseño “responsive”

La aplicación se ha desarrollado con la idea de que **se pueda usar en cualquier dispositivo**.

Para ello, mediante la ayuda de **Sass**, se han implementado a los estilos del proyecto una característica CSS llamada ‘**media query**’.

Las **medias queries** son reglas o conjunto de reglas que se introducen dentro de la hoja de estilos con el fin de establecer una serie de propiedades a los estilos, dependiendo del tipo de dispositivo en el que se use la aplicación.

```
.incidenciaview2-container {  
  width: 100%;  
  height: auto;  
  min-height: 100%;  
  display: flex;  
  flex-direction: row;  
  
  @media screen and (max-width: 1350px) {  
    flex-direction: column;  
    align-items: center;  
  }  
}
```

Ilustración 16. Ejemplo de media query

En la Ilustración 16. Ejemplo de *media query*, se puede observar cómo dependiendo del ancho de la pantalla en la que se esté visualizando la aplicación, en este caso *1350 px*, los objetos de ese contenedor se van a situar en fila o en columna.

Estas reglas CSS han ayudado a conseguir que la aplicación sea **responsive**, que consiste en que se adapte correctamente en cualquier dispositivo.

Para probar la aplicación en diferentes dispositivos móviles se ha hecho uso de Google Chrome, ya que trae una herramienta de desarrollador que viene implementada para probar la aplicación en diferentes resoluciones.

#### **4.2.1.11 Paleta de colores**

Para la obtención de una paleta de colores similares o compatibles entre sí se ha utilizado la herramienta en línea <https://colorhunt.co/>.

En el caso de este proyecto, la paleta usada ha sido la siguiente:



*Ilustración 17. Paleta de colores (<https://colorhunt.co/palette/187385>)*

#### **4.2.2 Metodología usada: SCRUM.**

La metodología que se ha usado para el desarrollo de este proyecto es la famosa metodología *Agile*. La metodología *agile*, según empresas que lo usan como el BBVA, no es una forma de trabajar, sino una filosofía que implica una forma diferente de trabajar y organizarse. Esta metodología se suele utilizar en proyectos en los que, o son de extensión grande y se necesita dividir en equipos más pequeños que desarrollen una parte específica del proyecto, o en proyectos en los que las prioridades del cliente cambien constantemente y a medida que se desarrolla el proyecto cambien más rápido, o ambas opciones.

Por esto mismo se ha pensado en utilizar una metodología *agile* en el proyecto, ya que los objetivos del anteproyecto eran muy dispersos y se supuso que las prioridades principales del proyecto iban a ir cambiando a medida que se iba a ir desarrollando.

#### 4.2.2.1 Metodología agile “SCRUM”

Dentro de las metodologías *agile* se pueden identificar varios tipos como Scrum, Kanvan, etc.

En este proyecto se ha elegido la metodología Scrum<sup>6</sup>, ya que se pensó desde un principio que era la mejor opción para el tipo de proyecto que se iba a desarrollar.

Scrum es una metodología *agile* pensada para empresas o proyectos en los que:

- Existe **incertidumbre** sobre cómo llegar al objetivo planteado al principio del proyecto. En este caso, el objetivo principal estaba claro desde un principio: desarrollar un gestor de incidencias sencillo y fácil de usar. El problema era cómo llevarlo a cabo y qué herramientas o frameworks usar, por lo que se fue aprobando e implementando a medida que iba transcurriendo el desarrollo.
- Los equipos tengan capacidad de **autoorganización**, es decir, no necesitan ninguna figura que se encargue de organizar al equipo. Para que haya autoorganización, los integrantes del equipo tienen que cumplir tres características:
  - Capacidad de autosuperación.
  - Autonomía. Esto quiere decir que no tiene que haber nadie detrás de nadie para que se haga el trabajo indicado.
  - Auto-enriquecimiento: al ser un equipo multidisciplinar, al final todos aprenderán un poco de todos.

Este proyecto, al ser desarrollado por una única persona, la premisa de autoorganización se va a cumplir al 100%, ya esa persona va a tener que asumir todos los roles, sea necesario o no.

- Haya un control moderado de la creatividad y espontaneidad de los integrantes.
- Haya transmisión de conocimiento, haciendo alusión a la tercera característica del segundo punto.

Los últimos dos puntos no se van a cumplir en el desarrollo de este proyecto, ya que no es posible por ser llevado por una única persona. (Gallego & Domingo Troncho).

#### 4.2.2.2 Tipos de roles

Dentro de un proyecto desarrollado bajo la metodología agile SCRUM, tiene que haber que siempre una serie de figuras esenciales:

- **Product owner:** Es el encargado de obtener información nueva del cliente acerca del producto que se esté desarrollando. Además es el encargado de ocuparse de que el producto se envíe correctamente y en las fechas indicadas, ya que toca la responsabilidad recaerá sobre él.
- **Scrum master:** Es la figura encargada de que se lleve a cabo correctamente los procesos de sprint planning y de creación de tareas.

---

<sup>6</sup> **SCRUM.** Tipo de metodología ágil.

- **Equipo de desarrollo:** son los encargados de realizar el trabajo de desarrollo.

Al ser un proyecto desarrollado por una única persona, la metodología *agile* SCRUM se ha tenido que adaptar y modificar en cierto modo para poderse llevar a cabo, de esta forma, la misma persona asumiría las responsabilidades de todos los roles.

#### 4.2.2.3 Implementación de SCRUM

Para la correcta implementación de esta metodología *agile* se siguieron las siguientes premisas:

- **Creación de sprints<sup>7</sup>.** Al principio del proyecto los sprints eran más largos, ya que la cantidad de tareas era menor pero sí más extensas, por lo que era necesario más tiempo para su correcto desarrollo.

A medida que avanzaba el proyecto, el tiempo de duración de un *sprint* se fue disminuyendo hasta que se dejó en una semana de duración.

Al final de cada *sprint* se realizaba una reunión con el cliente para agregar nuevas tareas o funcionalidades y validar las que ya se habían llevado a cabo.

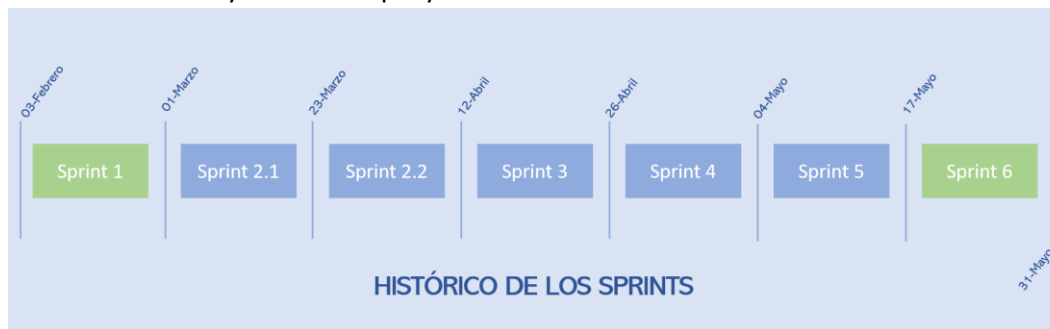


Ilustración 18. Histórico de los sprints

- **Uso de un tablero Scrum.** Para ello se usó una herramienta corporativa llamada Trello, a través de la cual se puede visualizar el estado de todas las tareas y tener un recuento de las tareas realizadas en cada *sprint*.

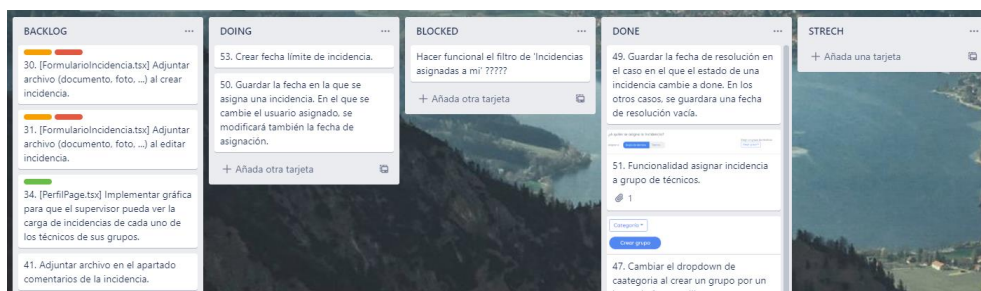


Ilustración 19. Tablero Scrum en Trello

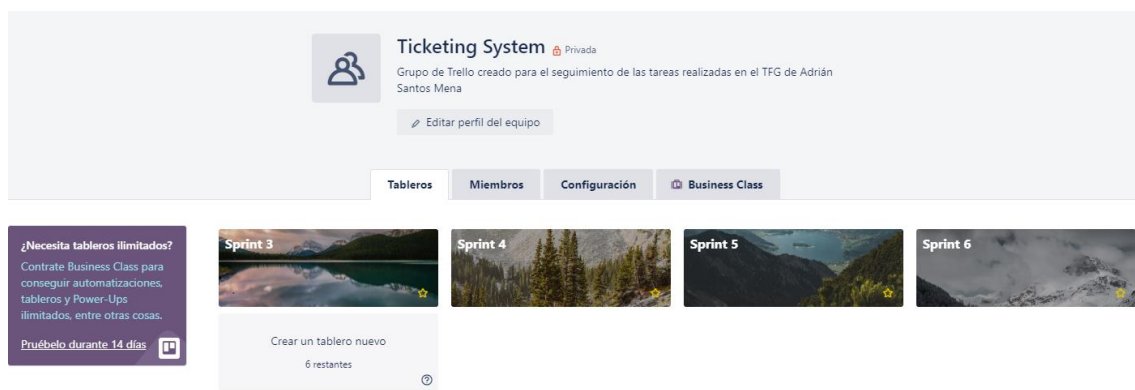
<sup>7</sup> **Sprint.** Es un intervalo de tiempo fijado por el proyecto durante el cual se realiza un incremento del desarrollo del producto.

- **Tareas etiquetadas.** Se le asigna un identificador a cada una de las tareas dentro de un sprint. De esta forma es más fácil identificarla dentro de la herramienta de control de versiones.

Si se sabe dónde se tiene que producir la tarea, se pone seguido al identificador, entre corchetes, el nombre del archivo de *typescript* que corresponde.

Trello se empezó a usar en Sprint 3, ya que el número de tareas iba creciendo a medida que iba avanzando el proyecto y llegó un punto en el que hizo falta el uso de un gestor de tareas para tener un historial de las finalizadas y las que quedaban por hacer.

Dentro del Trello se pueden generar tableros, por lo que se creó uno por cada sprint, quedando de la siguiente manera:



*Ilustración 20. Tableros creados en este proyecto*

Y dentro de cada uno de los tableros se crearon varias columnas siguiendo las pautas de un tablero SCRUM. Por ello, cada uno de los tableros tiene 5 columnas correspondiendo cada una a los siguientes posibles estados:

- **Backlog.** Es la columna en la que se encuentran todas las tareas que se han arrastrado del anterior sprint y las nuevas.
- **To do.** Son las tareas que se ha comprometido en realizar el desarrollador para el siguiente sprint.
- **Doing:** Son las tareas que se están desarrollando en ese momento.
- **Done.** Son las tareas cuyo desarrollo ya ha finalizado. En el caso en el que el proyecto tuviera un equipo de desarrollo formado por más personas, el *product-owner* sería el encargado de validar que las tareas se han realizado correctamente, entonces, una vez finalizada una de las tareas, ésta tendría que ir a la columna de *Ready to verify*, de esta manera el *product-owner* podría validarla.
- **Stretch.** Es la columna a la que se pasan las tareas que no entran en ese sprint, pero quedan a disposición de los desarrolladores en el caso en el que todas las tareas de la columna *to do* se terminen de desarrollar.

El tablero de la siguiente manera:

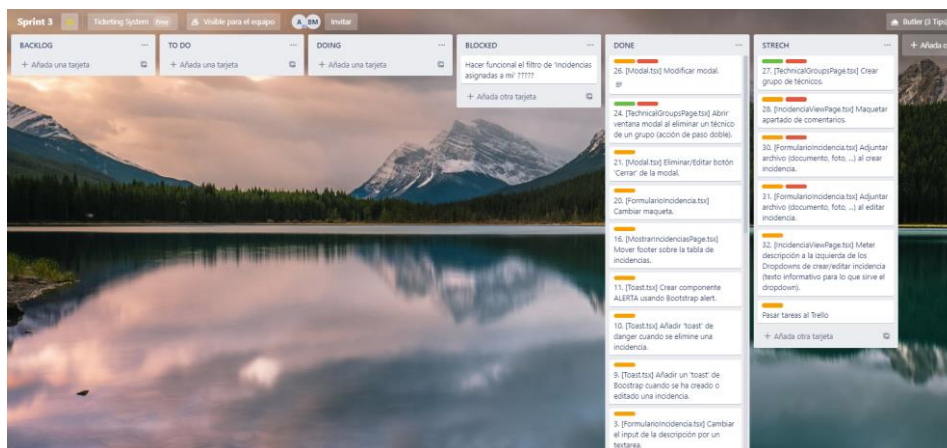


Ilustración 21. Tablero Trello de un Sprint

Los colores de las etiquetas de cada una de las tareas son identificadores para señalar al desarrollador sobre qué rol de la aplicación va a afectar el desarrollo de esa tarea o la importancia de ésta.

- **Rojo.** Señaliza que la tarea es importante y cuyo desarrollo es prioritario. Esto quiere decir que si no se realiza el desarrollo de esa tarea, otras tareas futuras no se podrán desarrollar.
- **Naranja.** Indica que la tarea es neutra, es decir, o es común a todos los roles de la aplicación o es completamente indiferente a ellos.
- **Verde.** Indica que el desarrollo va a ser sobre el rol de *Supervisor*.
- **Amarillo.** Indica que el desarrollo se va a realizar sobre el rol de *Técnico*.

#### 4.2.3 Herramientas empleadas.

Las herramientas que se han usado para el desarrollo han sido muy variadas y, en general, muy conocidas por la comunidad de desarrolladores web.

Todas estas herramientas pueden agruparse dependiendo de su tipo de finalidad, pudiendo estar la misma en dos grupos diferentes.

- **Herramientas de desarrollo.** Dentro de este grupo se encuentran las herramientas usadas para el desarrollo del proyecto. Se encuentran: *Visual Studio Code*, *Node.js*, *Composer*, *React*, *Laravel*, *Bootstrap*, *Chart.js*.
- **Herramientas de diseño.** Son herramientas destinadas para el diseño de la interfaz de la aplicación. Podemos encontrar las siguientes: *Bootstrap* y *Lunacy*.
- **Herramientas de control de versiones.** Estas herramientas sirven para tener un historial de todos los cambios producidos dentro del proyecto, dando opciones como restauración de una versión anterior y comprobación del historial de cambios. Dentro de este grupo podemos encontrar *Git*, *Github* y *Trello*.
- **Herramientas de comunicación con servidor.** Son herramientas para la conexión con el servidor necesaria para la visualización de la aplicación. *XAMPP* y *Postman* son herramientas pertenecientes a este grupo.

- **Herramientas para reuniones o llamadas.** Son herramientas para la comunicación entre usuarios. Son muy útiles para la realización de reuniones o *sprint plannings*<sup>8</sup>. Dentro de este grupo podemos encontrar aplicaciones como el *Team*, *Discord*, *Skype*, *Hangouts*, ...

#### 4.2.3.1 Visual Studio Code



Visual Studio Code es una herramienta de escritorio muy liviana para el desarrollo de código y que consta de muchas funcionalidades prácticas de las que el usuario puede hacer uso para amenizar su escritura de código y desarrollo de aplicaciones.

La edición de código no está limitada en esta herramienta, ya que existen lo que se llaman extensiones que permiten el uso de cualquier lenguaje disponible (Java, Python, C, C++, HTML, CSS, PHP, Ruby, PowerShell, JSON, ...).

Está disponible para todos los sistemas operativos (herramienta multiplataforma), y consta de *intellisense*<sup>9</sup>, lo que ayuda a predecir lo que el usuario pretende escribir a medida que programa.

Dentro de este proyecto ha sido esencial, ya que se ha usado como la herramienta de trabajo principal para el desarrollo de la aplicación web. En ella se han usado lenguajes de programación como PHP, Typescript, SCSS, HTML, ..., y ha sido de gran ayuda gracias a las características anteriormente mencionadas, consiguiendo amenizar el trabajo y por tanto la implementación de una mayor cantidad de funcionalidades.

#### 4.2.3.2 Git



Git es un software que se encarga del control de versiones de otro software o aplicación, pero para entenderlo mejor, habría que saber más detalladamente lo que es el control de versiones.

---

<sup>8</sup> **Sprint planning.** Reunión que se realiza en el comienzo de un sprint nuevo para establecer las nuevas tareas y la estimación correspondiente a cada una de ellas.

<sup>9</sup> **Intellisense.** Funcionalidades que implementan ejemplos como sugerencias de métodos, implementación de nueva sintaxis, estructuras completas de código, ...



El control de versiones dentro de una herramienta es la gestión de todos los cambios que se producen dentro de ella.

Sabiendo esto, ya se puede explicar con más detalle lo que es Git. Git se encarga básicamente de proporcionar a los usuarios un sistema de trabajo en ramas que ayude a mejorar la eficiencia y rapidez del trabajo en equipo.

En este proyecto ha sido una herramienta de gran importancia, ya que ha permitido mantener un historial de versiones, tanto para los sprint plannings como para los posibles errores que iban surgiendo, pudiendo de esta forma volver a una versión anterior y así evitarlos.

#### 4.2.3.3 *Github*



GitHub es un sistema de gestión de proyectos que implementa Git, y a su vez es una red social creada e implementada para la comunicación entre desarrolladores.

La ventaja de usar este sistema en el proyecto es que se ha podido gestionar y almacenar en un repositorio<sup>10</sup> al cual se podía acceder desde varios equipos. De esta forma se ha podido trabajar desde ordenadores diferentes sin importar si el proyecto se había pasado de un ordenador a otro o no con los cambios.

#### 4.2.3.4 *Trello*



Trello es una aplicación que se usa para la gestión de un proyecto, tanto de manera profesional como personal. Es una herramienta muy simple pero a la vez muy útil, ya que permite organizar las tareas de cualquier proyecto con un seguimiento exhaustivo de todas las acciones producidas sobre ellas.

Dentro del proyecto, Trello ha sido muy importante para llevar un seguimiento del estado de las tareas que se iban metiendo y realizando en cada uno de los Sprints.

---

<sup>10</sup> **Repositorio.** Es un directorio en el que se almacenan todos los archivos de un proyecto. Puede estar almacenado tanto en GitHub como en tu ordenador local.

#### 4.2.3.5 Node.js



Node.js es un **entorno de tiempo de ejecución de JavaScript**. Se encarga de transformar algo que solo se podía ejecutar en los navegadores, en algo que se puede ejecutar en ordenadores como si se tratase de una aplicación independiente.

También permite instalar paquetes externos en el proyecto y hacerlos funcionar correctamente (React.js, Char.js, Bootstrap, ...), por lo es un **manejador de paquetes**.

En este proyecto se usa *Typescript*, pero es algo que le da igual a Node.js, ya que *Typescript* es una modificación de *JavaScript*. Además, existen *loaders* que adaptan el código *Typescript* a código que *Node.js* puede entender.

#### 4.2.3.6 Composer



Al igual que *Node.js*, *Composer* es un **manejador de paquetes** para PHP con el que se pueden administrar, descargar e instalar dependencias y librerías.

En este proyecto ha sido necesario para la instalación de Laravel 7.

#### 4.2.3.7 React



React es una librería JavaScript enfocada en la creación de **interfaces de usuario**. Es una librería sobre la que gira un entorno lleno de componentes, herramientas, ..., que ayudarán a exprimirla al máximo. Su principal es la creación de SPAs<sup>11</sup>, aunque con ella también se pueden crear aplicaciones móviles.

---

<sup>11</sup> **SPAs**. *Single Page Applications*. Son aplicaciones que se montan sobre un solo archivo HTML.

React da muchas ventajas a la hora de usarla:

- **DOM<sup>12</sup> virtual.** Esta sería la ventaja principal de *React*, ya que se consigue tener un *DOM* dinámico. Esto quiere decir que se pueden recargar partes de la página sin necesidad de tener que recargarla entera, es decir, no cambia el *DOM* del navegador, sino el *DOM* virtual generado por *React*.
- Es una librería muy usada, por lo que la **comunidad** en internet es **muy extensa**. La ventaja se nota al encontrarse con alguna dificultad, ya que buscando un poco en la web se encontrarán muchas soluciones respecto a dicha dificultad.
- Se basa en **componentes reutilizables**. Con esto se consigue dividir la aplicación en componentes y a la hora de tener fallos, estos sean más fáciles de encontrar, ya que sabremos al instante en qué componente se está produciendo dichos fallos. Esta ventaja también permite que la aplicación sea escalable, esto quiere decir que una vez creados los componentes, será más fácil crear páginas nuevas, ya que se hará uso de los componentes creados.
- No tiene **lenguaje propio** (como puede ser con *Angular* y *Vue*). Esto quiere decir que funcionará siempre con el último estándar JavaScript (actualmente *ECMAScript 6*) y se podrá implementar sin ningún problema en cualquier navegador.

Se ha elegido *React* para el desarrollo de la parte *front-end* debido a que una de las exigencias propias del proyecto era trabajar con las tecnologías más punteras y potentes de la actualidad.

Otro motivo fue el hecho de poder reutilizar código. La *componentización* de la página web ha ahorrado mucho tiempo de desarrollo, pudiendo reutilizar componentes y generar páginas bonitas y sencillas en poco tiempo.

#### 4.2.3.8 *Laravel*



Laravel es un framework cuyo objetivo es la programación con una sintaxis refinada y expresiva para crear código simple y legible, evitando lo que se llama “código espagueti”.

Laravel es una librería que está compuesta por una gran cantidad de dependencias, sobre todo de *Symfony*<sup>13</sup>, por lo que su desarrollo depende en gran parte del desarrollo de estas dependencias.

---

<sup>12</sup> **DOM.** *Document Object Model* o *Modelo de Objetos del Documento*. Define la estructura lógica de una página web a través de la cual se modifican sus elementos.

<sup>13</sup> **Symfony.** Es un entorno de trabajo estandarizado utilizado para los desarrollos de aplicaciones web en PHP.

En este proyecto se ha usado Laravel para:

- Establecer conexión servidor-cliente.
  - Llamadas a la base de datos con el *Eloquent ORM* que incluye Laravel.
  - Petición de datos por API para el paso de datos con React.
- Gestión de la base de datos.
- Generación de modelos y controladores dentro del patrón de arquitectura software MCV.

#### 4.2.3.9 *Bootstrap*



Bootstrap es una **librería CSS** que permite crear interfaces de usuario mediante componentes ya creados en la propia librería. También consta de un sistema *grid* para conseguir que los contenedores de la página web sean adaptables en cualquier tipo de dispositivo y pantalla.

Es compatible con todos los navegadores, modificable con CSS y consta de muchos tipos de componentes y configuraciones.

*Tabla 26. Ejemplos componentes Bootstrap*

Ejemplos		
Botones	Inputs	Forms (login, upload file, ...)
Dropdowns	Collapses	Modales
Navbars	Progress bar	...

En este proyecto solo se ha hecho uso de Bootstrap para tener una base firme de algún componente en concreto y así poder modificarlo en base al diseño de la interfaz.

El resto de las modificaciones y modo *responsive*<sup>14</sup> de la aplicación se han hecho con SCSS puro.

---

<sup>14</sup> **Responsive.** Adaptable en cualquier tipo de pantalla o dispositivo.

#### 4.2.3.10 Chart.js

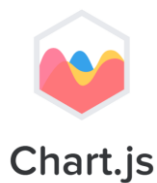
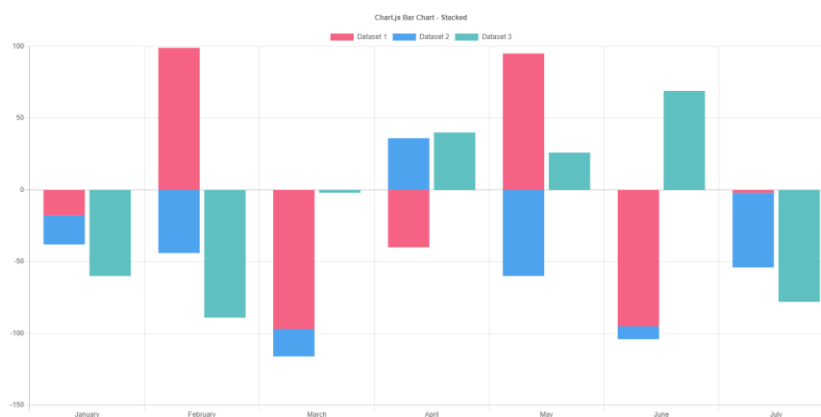


Chart.js es una librería JavaScript que permite la visualización de datos mediante el uso de gráficas de todo tipo.

Se pueden encontrar desde gráficas de barras hasta gráficas radar. Estos serían algunos ejemplos de las posibilidades que ofrece esta librería:



*Ilustración 22. Ejemplo 1: Gráfica de barras*



*Ilustración 23. Ejemplo 2: tipos de gráficas de líneas*

Durante el desarrollo del proyecto, se ha hecho uso de esta librería para la visualización de datos sobre el estado de las incidencias, el número de incidencias asignadas, historial de incidencias, ..., todo dependiendo del tipo de rol de la aplicación.

#### 4.2.3.11 XAMPP



XAMPP es un paquete de software libre que incluye gestión de bases de datos locales, un servidor *Apache* e intérpretes para los lenguajes de programación *PHP* y *Perl*. Esto explica su nombre, ya que está compuesto por las iniciales de **X** (funcional en cualquier SO), **A**(*Apache*), **M**(*MySQL* o *MariaDB*) y **PP**(*Php* y *Perl*).

XAMPP ha sido esencial dentro del proyecto, ya que se ha podido generar un servidor local y poder probar la aplicación en él. Con la base de datos igual, ha sido muy simple la conexión de las bases de datos de *XAMPP* con *Laravel* para poder hacer toda la extracción y guardado de datos necesario.

#### 4.2.3.12 Postman



Postman es una herramienta que permite **realizar peticiones HTTP**<sup>15</sup> a cualquier *API*. Es una herramienta muy útil, ya que además de poder obtener información de *APIs* de terceros, permite realizar pruebas sobre las peticiones *HTTP* que se están realizando dentro de un proyecto en local.

En este proyecto ha sido muy importante su uso, ya que el paso de datos entre *Laravel* y *React* se realiza mediante peticiones *HTTP*. Gracias a *Postman*, estas peticiones se han podido testear y modificar hasta que devolviesen los datos que se requerían en cada situación.

#### 4.2.3.13 Lunacy



*Lunacy* es un **editor gráfico** gratis para Windows que provee al usuario de muchas funcionalidades para el diseño gráfico.

---

<sup>15</sup> **Peticiones HTTP.** Son peticiones que se realizan para obtener unos datos específicos. HTTP corresponde al Protocolo de Transferencia de Hipertexto (Hypertext Transfer Protocol), que es el protocolo de transmisión de documentos HTML.

Permite la exportación del CSS usado en el diseño para que sea más fácil llevarlo al desarrollo y la exportación de los proyectos a archivos *PNG* o *SVG*, aunque lo más destacable de Lunacy es la biblioteca de recursos, ofreciendo desde iconos de todo tipo hasta imágenes completas.

En este proyecto se ha usado para el diseño de las primeras páginas de la aplicación, tomando importancia sobre todo al inicio.

#### 4.2.3.14 Microsoft Teams



*Teams* es una herramienta que ofrece Microsoft para **crear un entorno virtual de colaboración y trabajo** para las empresas, mejorando la comunicación y colaboración entre sus integrantes.

Es la herramienta que se ha usado para realizar todas las reuniones de los *sprint plannings* con el cliente, y así conseguir un avance rápido pero controlado del proyecto.

### 4.3 Recursos requeridos

Los recursos necesarios para el desarrollo de este proyecto han sido:

- Ordenador de sobremesa:
  - Procesador: Intel® Core™ i5-4690 CPU @ 3.50GHz .
  - Memoria RAM: 12,0 Gb.
  - Sistema Operativo Windows 10 Pro - 64 bits.
  - Tarjeta gráfica: NVIDIA GeForce GTX 960 4G.
- Ordenador portátil:
  - Procesador: Intel® Core™ i5-7200U CPU @ 2.50GHz.
  - Memoria RAM: 8,0 Gb.
  - Sistema Operativo: Windows 10 Home - 64 bits.
  - Tarjeta gráfica: Intel® HD Graphics Family.
- Dos monitores.
  - DELL S2715H 27" (1920x1080).
  - DELL P2720DC 27" (2560x1440).
- Visual Studio Code (v.1.45).
- Github (v.2.26.2).
- React (v.16.13.1).
- Laravel (v.7).
- Composer (v.1.10.6).
- Bootstrap (v.4.4).
- XAMPP (v.7.3.4).
- Trello (v.).
- Teams (v.1.3.00.8663).
- Postman (v.7.23.0).

- Lunacy (v.5.3.1).
- Spotify (v.1.1.32).
- Wifi.
- Navegador (Google Chrome).

## 4.4 Presupuesto

*Tabla 27. Presupuesto del proyecto*

Tipo de coste	Valor	Comentarios
Horas de trabajo en el proyecto	21 000 €	Teniendo en cuenta que el sueldo medio de un programador web son 37,5€/hora, el total durante todo el proyecto han sido:  $560 * 37,5 = 21\ 000\ €$
Equipo técnico utilizado	3 144, 17 €	<ul style="list-style-type: none"> <li>➤ Ordenador de sobremesa: 960 €</li> <li>➤ Portátil: 1200 €</li> <li>➤ Pantallas: 306,00 € + 458,18 €</li> <li>➤ Teclado: 149,99 €</li> <li>➤ Ratón: 70 €</li> </ul>
Software utilizado	0 €	Todos los softwares que se han usado en este proyecto han sido de licencias gratuitas.
Estudios e informes	60 €	Todos los datos usados para la realización de este proyecto han sido obtenidos de internet, por lo que no han salido por el pago del wifi durante los 4 meses de desarrollo.  Wi-Fi: 15 €/mes durante 4 meses: 60 €
Materiales empleados	0 €	Todos los materiales necesarios ya han sido mencionados.
<b>TOTAL (sin IVA)</b>	<b>24 204, 17 €</b>	

(Enlace a la portada de la Web oficial de la Junta de Andalucía, s.f.)



## 4.5 Viabilidad

Los costes de una página web es muy diversos y diferentes. Se podría contrastar con los precios de los ordenadores: no existe un precio universal para todos los ordenadores ya que siempre dependerán de sus especificaciones y el uso que se les vaya a dar. Pues en las páginas web pasa lo mismo.

Teniendo en cuenta que el precio de una página puede ser desde los 500 € a más de 50 000 € y sabiendo el trabajo que lleva detrás el proyecto, tanto en tiempo como en esfuerzo, 24 204, 17 € es un precio bastante acertado y razonable.

Es una herramienta que siempre va a ir a más y tiene una escalabilidad brutal gracias a que su desarrollo se ha realizado bajo los Frameworks de *React* y *Laravel*.

Su disponibilidad la hace una herramienta de trabajo ideal, ya que al ser una página web es accesible desde cualquier dispositivo con acceso a Internet. De esta manera, el seguimiento del trabajo realizado puede ser constante independientemente del lugar en el que se encuentre el usuario.

En definitiva, teniendo en cuenta el trabajo y el esfuerzo dedicado, y las características de la aplicación, el presupuesto obtenido sale más que rentable.

## 4.6 Resultados del proyecto

El resultado del proyecto es una aplicación web desarrollada bajo las tecnologías de Laravel y React, y la cual permite las siguientes acciones:

- Generación de incidencias.
- Modificación de incidencias.
- Eliminación de incidencia específica.
- Visualización de datos sobre una incidencia.
- Realización de comentarios sobre incidencias.
- Visualización del perfil del usuario.
- Modificación de perfil.
- Visualización de distintos tipos de historiales de incidencias.

También es una aplicación web *responsive*, esto quiere decir que se puede usar desde cualquier dispositivo.

Los datos de la aplicación se visualizan a tiempo real, es decir, cualquier modificación sobre la aplicación se visualizará al instante sin necesidad de recargar.

## 4.7 Plan de pruebas

En este apartado se exponen todas las pruebas que se han ido realizando durante el desarrollo del proyecto para una mejor experiencia de usuario y un mejor funcionamiento de la aplicación.

La cantidad de pruebas realizadas no es extensa debido a que las interacciones dentro de la aplicación son bastante cerradas y guiadas, por lo que el usuario introduce datos en formato de texto libre en pocas situaciones.

La tabla que se va a usar como referencia es la siguiente, en la que se explica cuál es la prueba que se ha realizado, los pasos a seguir para realizar la prueba y los resultados esperados y obtenidos.

<Código del CP>	<Nombre caso de prueba>
<b>Descripción</b>	<Descripción del caso de prueba>
<b>Pasos</b>	<Enumerar los pasos a seguir para realizar la prueba>
<b>Resultado esperado</b>	<Resultado esperado de la prueba>
<b>Resultado obtenido</b>	<Resultado obtenido en la prueba>

Tabla 28. Plantilla tabla de pruebas

#### 4.7.1 Pruebas

CP-00	Datos de inicio de sesión erróneos
<b>Descripción</b>	El sistema notifica al usuario de que los datos introducidos en los campos del login no coinciden.
<b>Pasos</b>	<ol style="list-style-type: none"><li>1. Acceder a la página principal de la aplicación.</li><li>2. Introducir datos erróneos para iniciar sesión en la aplicación.</li></ol>
<b>Resultado esperado</b>	Los campos deben ponerse de color rojo y debe de aparecer bajo ellos el error producido.
<b>Resultado obtenido</b>	El resultado obtenido coincide con el esperado.

Tabla 29. CP-00 Datos de inicio de sesión erróneos

CP-01	Campos rellenados incorrectamente al crear incidencia
<b>Descripción</b>	El sistema notifica al usuario que los datos introducidos al crear una incidencia son erróneos.
<b>Pasos</b>	<ol style="list-style-type: none"><li>1. Acceder a la página de 'Crear nueva incidencia'.</li><li>2. Introducir datos erróneos o dejar los campos vacíos para crear la incidencia.</li></ol>
<b>Resultado esperado</b>	Los campos deben ponerse de color rojo y debe de aparecer bajo ellos el error producido y si se ha introducido bien el dato volver a su estado original.
<b>Resultado obtenido</b>	El resultado obtenido coincide con el esperado.

Tabla 30. CP-01 Campos rellenados incorrectamente al crear incidencia

CP-02	Filtrar incidencias
<b>Descripción</b>	El sistema muestra solo las incidencias filtradas. Al deseleccionar todas debe mostrar todas las incidencias ordenadas por la opción seleccionada.
<b>Pasos</b>	<ol style="list-style-type: none"><li>1. Acceder a 'Mis incidencias'.</li><li>2. Seleccionar una opción del 'ordenar por'.</li><li>3. Seleccionar un filtro.</li><li>4. Deseleccionar filtro.</li></ol>
<b>Resultado esperado</b>	Al seleccionar el filtro deben aparecer solo las incidencias coincidentes con el filtro seleccionado. Al deseleccionarlo, deben aparecer todas las incidencias de nuevo ordenadas por la opción seleccionada en 'Ordenar por'.
<b>Resultado obtenido</b>	El resultado obtenido coincide con el esperado.

Tabla 31. CP-02 Filtrar incidencias

CP-03	Actualizar datos de incidencias filtradas
<b>Descripción</b>	Al filtrar incidencias, debe cambiar el numero de incidencias filtradas en:  <div> Incidencias filtradas: 15    critical: 8    important: 6    trivial: 1 </div>
<b>Pasos</b>	<ol style="list-style-type: none"> <li>1. Acceder a 'Mis incidencias'.</li> <li>2. Seleccionar una opción del 'ordenar por'.</li> <li>3. Seleccionar un filtro.</li> <li>4. Deseleccionar filtro.</li> </ol>
<b>Resultado esperado</b>	Al seleccionar los filtros, el dato de <b>incidencias filtradas</b> deba actualizarse y mostrar la cantidad de incidencias mostradas. Si se deselectionan los filtros, ese dato debe cambiar tambien.
<b>Resultado obtenido</b>	El resultado obtenido coincide con el esperado.

Tabla 32. CP-03 Actualizar datos de incidencias filtradas

CP-04	Editar perfil
<b>Descripción</b>	El sistema muestra mensaje de error en caso de que los campos para modificar el perfil estén vacíos.
<b>Pasos</b>	<ol style="list-style-type: none"> <li>1. Acceder a 'Perfil'.</li> <li>2. Pulsar opción 'Editar perfil'.</li> <li>3. Guardar cambios campos vacíos.</li> </ol>
<b>Resultado esperado</b>	Al clickar el botón de 'Guardar cambios', debe mostrarse un mensaje de error y no guardarse los cambios realizados.
<b>Resultado obtenido</b>	El resultado obtenido coincide con el esperado.

Tabla 33. CP-04 Editar perfil

## Capítulo 5. DISCUSIÓN

Dentro del proyecto, ha habido varios puntos en los que su decisión ha sido brevemente discutida y dialogada con el cliente en cuestión. Todos y cada uno de ellos han tenido un factor positivo sobre el desarrollo del proyecto.

### 5.1 Metodología de trabajo empleada

La metodología de trabajo empleada ha sido la metodología agile SCRUM, aunque al principio no se tenía del todo claro.

Esto es debido a que estos tipos de metodologías están pensadas para el trabajo en equipo, por lo que en este proyecto, al ser un trabajo individual, se pensaba que el modelo de trabajo no iba a ser del todo óptimo.

Después de buscar y buscar en cómo adaptarlo, se consiguió implementar con varias adaptaciones. Esto quiere decir que, por ejemplo, en vez de estimar el tiempo de trabajo de cada una de las tareas, se dejaban en el *Strech* y se iban cogiendo a medida que se iban completando.

Esta forma de trabajo ayudó mucho a agilizar el desarrollo del proyecto, ya que se ha podido priorizar en las tareas y dejar las menos importantes para el final, llegando así a la última semana con la mayoría de las tareas prioritarias finalizadas.

### 5.2 ¿Ha sido buena opción implementar React con Laravel?

Buenísima. Esta decisión fue tomada debido al conocimiento mínimo sobre *Laravel* y las ganas de aprender React, y como se detectó que React puede implementarse para el desarrollo de la parte *front* dentro de *Laravel*, se tiró para adelante con la decisión.

La primera semana fue dura, ya que la mayoría del tiempo de desarrollo consistió en la configuración del proyecto, pero una vez configurado correctamente, el desarrollo se realizó sin ningún tipo de problema.

### 5.3 Visualización de incidencias

Dentro de la aplicación, como ya se ha explicado, existen dos roles principales: el supervisor y el técnico.

La idea de la discusión es saber qué incidencias pueden visualizar, modificar o eliminar cada uno de los roles.

#### 5.3.1 Incidencias del supervisor

Dentro del rol de supervisor, se llegó a la conclusión de que es importante que puedan acceder a todas las incidencias que estén asignadas a los equipos de los que cada uno de los técnicos está a cargo. De esta manera, podría reasignar o desechar incidencias que no pertenecen a un equipo de técnicos específico.

También debe tener la posibilidad de visualizar y editar todas las incidencias que pertenezcan a los técnicos que forman los grupos de técnicos que lleva. Esto ayudaría a visualizar la carga de los técnicos y poder reasignar tareas en base a su perspectiva.

Por último, tiene que ser capaz de modificar las incidencias que haya creado, independientemente de si están asignadas a un grupo o a un técnico. Esto se pensó porque se puede dar la casualidad de que exista alguna errata al crear una incidencia, pudiendo así acceder a dicha incidencia y modificarla las veces que hagan falta.

### 5.3.2 Incidencias de técnico

Dentro del rol de técnico, es importante que pueda ver todas las incidencias que le hayan sido asignadas o que hayan sido asignadas a algún equipo al que pertenezca. A la hora de poder modificar o eliminar alguna incidencia, solo dispondrá de esas dos opciones si ha sido él el que ha creado la incidencia, en caso contrario, solo podrá visualizar los datos y añadir comentarios.

Esto quiere decir que cualquier usuario, tanto técnico como supervisor, puede añadir tantos comentarios como desee a la incidencia que quiera.

De todas formas, esta opción está “limitada”, ya que no van a tener acceso directo a todas las incidencias, pudiendo solo acceder a todas a través de su ID en la URL.

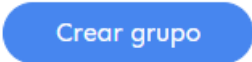
## 5.4 Uso de librería CSS externa (Bootstrap) o no

Otra de las decisiones que más costaron tomar fue la implementación de una librería CSS externa o no. Esta decisión se basaba en la capacidad que existía de poder hacerlo sin ninguna librería externa, aunque la preocupación era el tiempo, ya que maquetar una página web entera lleva bastante tiempo.

La decisión final fue la ayuda mediante Bootstrap y, ¿qué quiere decir “ayuda”? Pues bien, es una manera de tener una base hecha y modificarla a gusto del desarrollador, consiguiendo así exactamente el diseño deseado.



*Ilustración 24. Ejemplo 'Button' de Bootstrap*



*Ilustración 25. Ejemplo 'Button' Ticketclass*

## Capítulo 6. CONCLUSIONES

### 6.1 Conclusiones del trabajo

El trabajo realizado durante los 4 meses de desarrollo se puede considerar satisfactorio aunque no se hayan cumplido el 100% de los objetivos planteados en la memoria del anteproyecto.

La idea inicial y principal del proyecto se ha desarrollado completamente, ya que la aplicación es una aplicación web que permite la gestión de incidencias, además de que se han implementado nuevas ideas que no surgieron en el anteproyecto como:

- Un sistema de correos para notificar sobre los cambios producidos en las incidencias.
- Filtro y ordenación en la visualización de la lista de incidencias.
- Visualización de datos en gráficas.

### 6.2 Conclusiones personales

Ha sido un proyecto largo y tedioso de desarrollar, pero a la vez divertido y muy interesante.

La idea de meterme a programar con unas herramientas con las que mi experiencia era leve o nula ha sido la mejor idea de todo el proyecto, ya que mi capacidad de entendimiento sobre desarrollo web, bajo mi impresión, ha mejorado muchísimo.

Al principio del proyecto pensaba que me había metido en un berenjenal muy gordo, ya que hasta no terminar de desarrollar la idea del anteproyecto, no sabía siquiera si *React* y *Laravel* se podían implementar en un mismo proyecto de trabajo.

Por lo que pude ver, hay personas que implementaban cada *framework* en un proyecto diferente y después conseguían establecer comunicación entre ambos proyectos mediante peticiones *HTTP* a *APIs*. En mi caso no es así, ya que no son proyectos diferentes, sino que están implementados dentro de un mismo proyecto generado con *Laravel* sobre el que se trabaja con *React*.

La parte que me supuso más quebraderos de cabeza del proyecto fue la parte del *login*, ya que al quererlo hacer con *React* tuve que utilizar otra dependencia diferente a la que viene implementada en *Laravel*. La implementación de esta nueva dependencia me llevó más tiempo del que esperaba, ya que solo existe un video en el que basarme y la propia documentación.

Una vez terminado el *Login*, me puse a mirar los *hooks*, una nueva funcionalidad que se ha implementado en *React* a partir de la versión 16.8. Los *hooks* hacen que la funcionalidad de recarga y actualización de datos a tiempo real sea más sencilla de entender, por lo que, sin tener ni idea de cómo funcionaba al principio, no dudé en implementarlo al proyecto para darle esa mejora a la funcionalidad del proyecto.

Otra parte que me pareció bastante tediosa de desarrollar fueron las llamadas a la base de datos para que los filtros funcionasen correctamente. Como el filtro se realiza directamente sobre las incidencias guardadas en la base de datos, la visualización de estas es mucho más rápida, por lo que se consigue una mejor experiencia de usuario.

En definitiva, el proyecto me ha ayudado a:

- Entender mejor cómo funcionan las metodologías *agile*, tanto sus ventajas como desventajas.
- Mejorar como desarrollador web:
  - Aprendiendo *React*, una librería *JavaScript* que está actualmente en auge dentro del mundo del desarrollo web.
  - Aprendiendo y mejorando las bases obtenidas en tercero de carrera sobre el framework de *Laravel*.
- Aprender a organizarme dentro de un proyecto real y saber a priorizar entre tareas, mejorando así la eficacia de desarrollo.



## Capítulo 7. FUTURAS LÍNEAS DE TRABAJO

Son muchas las líneas de mejora para el futuro del proyecto. Varias de ellas no se han implementado por la falta de tiempo en el desarrollo del proyecto, y muchas otras ya se tenían en cuenta para futuras líneas de trabajo.

1. Una y bastante importante, es **implementar un filtro en el que los datos se puedan filtrar desde varias características diferentes**. Esto quiere decir, que si el usuario quiere solamente ver las incidencias coincidentes con el estado 'Pendiente' y la categoría 'Wi-Fi', debe tener la posibilidad de marcar estas dos opciones y que se le muestren todas las incidencias coincidentes.
2. **Implementar un 'Trello' al técnico dentro de la aplicación**. De esta manera, el técnico podrá saber de una manera más visual el estado de sus incidencias, sin necesidad de tener que filtrar o buscar en la tabla de incidencias.
3. **Disponibilidad de los laboratorios**. Es importante que el usuario que vaya a resolver una incidencia sepa la disponibilidad de las aulas para saber si puede ir sin molestar o interrumpir alguna clase.
4. **Mayor exportación y visualización de datos**. Generar una pantalla para el administrador en la que pueda ver el movimiento general de las incidencias mediante gráficas de todo tipo, y exportar los datos en formato Excel.
5. **Aumentar la configuración del perfil**. Meter opciones como; factor de doble autenticación, eliminación de cuenta, ...
6. **Mayor cantidad de opciones de gráficas**:
  - a. Comparativas entre grupos.
  - b. Comparativas entre empleados y sus medias.
  - c. Comparativas entre los grupos y sus medias.
  - d. Comparativas de un técnico con uno de los grupos a los que pertenece.
  - e. Gráficas temporales como las que ya existen actualmente, pero pudiendo seleccionar un periodo de tiempo.
7. Barajear la posibilidad de **acceder al LDAP de la Universidad Europea** para realizar los inicios de sesión. Esta línea de trabajo es costosa y con pocas probabilidades de realización, pero es factible.
8. **Exportación de incidencias** a EXCEL para las autorías.
9. **Adjunto de archivos** a las creaciones de incidencias, comentarios y cambios de estado.

## Capítulo 8. REFERENCIAS

- apptivaweb.com*. (s.f.). Obtenido de Qué es Axios.js. Librería de promesas en aplicaciones web: <https://apptivaweb.com/publicacion/desarrollo-web/axiosjs-una-libreria-de-promesas>
- BBVA NOTICIAS*. (28 de November de 2018). Obtenido de Metodología 'Agile'. La revolución de las formas de trabajo: <https://www.bbva.com/es/metodologia-agile-la-revolucion-las-formas-trabajo/>
- Blogs.itpro.es*. (s.f.). Obtenido de Visual Studio Code ¿Qué es? y ¿Qué no es?: <https://blogs.itpro.es/eduardocloud/2016/08/22/visual-studio-code-que-es-y-que-no-es/>
- Boada, N. (s.f.). *Cyberclick.es*. Obtenido de ¿Por qué es tan importante el User Experience o Experiencia del Usuario?: <https://www.cyberclick.es/numerical-blog/por-que-user-experience-o-experiencia-del-usuario>
- Code, V. (s.f.). *Code.visualstudio.com*. Obtenido de Documentation for Visual Studio Code: <https://code.visualstudio.com/docs>
- CódigoFacilito*. (s.f.). Obtenido de Qué es Git: <https://codigofacilito.com/articulos/que-es-git>
- Desarrolloweb.com*. (s.f.). Obtenido de Qué es React. Por qué usar React: <https://desarrolloweb.com/articulos/que-es-react-motivos-uso.html>
- Devs, Q. (s.f.). *Quality Devs*. Obtenido de ¿Qué es Symfony? Y por qué es el mejor framework para Apps: <https://www.qualitydevs.com/2019/08/05/que-es-symfony/>
- Dlsi.ua.es*. (s.f.). Obtenido de <https://www.dlsi.ua.es/~jaime/webe/articulos/s112.pdf>
- Documentación web de MDN*. (s.f.). Obtenido de HTTP: <https://developer.mozilla.org/es/docs/Web/HTTP>
- Enlace a la portada de la Web oficial de la Junta de Andalucía*. (s.f.). Obtenido de Plantilla Plan de Pruebas Funcionales: <http://www.juntadeandalucia.es/servicios/madeja/contenido/recurso/462>
- Es.wikipedia.org*. (s.f.). Obtenido de Net Promoter Score: [https://es.wikipedia.org/wiki/Net\\_Promoter\\_Score](https://es.wikipedia.org/wiki/Net_Promoter_Score)
- Es.wikipedia.org*. (s.f.). Obtenido de Modelo–vista–controlador: <https://es.wikipedia.org/wiki/Modelo%E2%80%93vista%E2%80%93controlador>
- Es.wikipedia.org*. (s.f.). Obtenido de XAMPP: <https://es.wikipedia.org/wiki/XAMPP>
- Gallego, M. T., & Domingo Troncho, A. C. (s.f.). *Gestión de proyectos informáticos. Metodología Scrum*. Obtenido de <http://openaccess.uoc.edu/webapps/o2/bitstream/10609/17885/1/mtrigasTFC0612memoria.pdf>

- García, J. (s.f.). *Blog de arsys.es*. Obtenido de ¿Qué es Laravel? - Blog de arsys.es:  
<https://www.arsys.es/blog/programacion/que-es-laravel/>
- García, J. (s.f.). *Blog de arsys.es*. Obtenido de ¿Qué es Laravel? - Blog de arsys.es:  
<https://www.arsys.es/blog/programacion/que-es-laravel/>
- Getbootstrap.com*. (s.f.). Obtenido de <https://getbootstrap.com/docs/4.4/components/>
- González, G. (s.f.). *Genbeta.com*. Obtenido de Lunacy es un excelente editor de gráficos para Windows fácil de usar, gratis y con muchos recursos de diseño integrados:  
<https://www.genbeta.com/imagen-digital/lunacy-excelente-editor-graficos-para-windows-facil-usar-gratis-muchos-recursos-diseno-integrados>
- Guía para la redacción de casos de uso | Marco de Desarrollo de la Junta de Andalucía*. (s.f.). Obtenido de <http://www.juntadeandalucia.es/servicios/madeja/contenido/recurso/416>
- Lucidchart.com*. (s.f.). Obtenido de Adapting Agile Scrum Methodology for One:  
<https://www.lucidchart.com/blog/what-is-agile-scrum-methodology>
- Lucidchart.com*. (s.f.). Obtenido de Adapting Agile Scrum Methodology for One:  
<https://www.lucidchart.com/blog/what-is-agile-scrum-methodology>
- Moisset, D. (s.f.). *Tutorialesprogramacionya.com*. Obtenido de IntelliSense:  
<https://www.tutorialesprogramacionya.com/herramientas/vscodeya/detalleconcepto.php?punto=11&codigo=11&inicio=0>
- Openaccess.uoc.edu*. (s.f.). Obtenido de <http://openaccess.uoc.edu/webapps/o2/bitstream/10609/17885/1/mtrigasTFC0612memoria.pdf>
- OpenWebinars.net*. (s.f.). Obtenido de Qué es NodeJS y para qué sirve:  
<https://openwebinars.net/blog/que-es-nodejs/>
- OpenWebinars.net*. (s.f.). Obtenido de Qué es Json Web Token y cómo funciona:  
<https://openwebinars.net/blog/que-es-json-web-token-y-como-funciona/>
- Qué es Sass, como usar Sass*. (s.f.). Obtenido de Desarrolloweb.com:  
<https://desarrolloweb.com/articulos/que-es-sass-usar-sass.html>
- Raiola Networks*. (s.f.). Obtenido de ¿Que es Bootstrap?: <https://raiolanetworks.es/blog/que-es-bootstrap/>
- Sara Escudero*. (s.f.). Obtenido de 10 Requisitos indispensables de una buena página Web:  
<https://www.saraescudergarcia.org/2016/05/26/10-requisitos-indispensables-de-una-buena-p%C3%A1gina-web/>
- Styde.net*. (s.f.). Obtenido de Qué es Composer y cómo usarlo: <https://styde.net/que-es-composer-y-como-usarlo/>

## Capítulo 9. ANEXOS

En este apartado se van a introducir extras que puedan ayudar a entender de mejor manera el proyecto.

### 9.1 Manual de instalación

Para poder instalar y correr la aplicación correctamente es necesario seguir una serie de pasos:

- 1) Clonar el repositorio git en una carpeta vacía.  
`git clone repo_url`
- 2) Ejecutar los comandos de *composer* y *npm* para instalar todas las dependencias necesarias.  
`composer install` y `npm install`
- 3) Crear un fichero *.env* en la raíz del proyecto mediante la siguiente plantilla:

```
*****
APP_NAME=Laravel
APP_ENV=local
APP_KEY=
APP_DEBUG=true
APP_URL=http://localhost

LOG_CHANNEL=stack

DB_CONNECTION=mysql
DB_HOST=localhost
DB_PORT=3306
DB_DATABASE=TFG_Ticketing_System
DB_USERNAME=root
DB_PASSWORD=

BROADCAST_DRIVER=log
CACHE_DRIVER=file
QUEUE_CONNECTION=sync
SESSION_DRIVER=file
SESSION_LIFETIME=120

REDIS_HOST=127.0.0.1
REDIS_PASSWORD=null
REDIS_PORT=6379

MAIL_DRIVER=smtp
MAIL_HOST=smtp.mailtrap.io
MAIL_PORT=2525
MAIL_USERNAME=null
MAIL_PASSWORD=null
MAIL_ENCRYPTION=null
MAIL_FROM_ADDRESS=null
MAIL_FROM_NAME="${APP_NAME}"

AWS_ACCESS_KEY_ID=
AWS_SECRET_ACCESS_KEY=
```

```
AWS_DEFAULT_REGION=us-east-1
AWS_BUCKET=

PUSHER_APP_ID=
PUSHER_APP_KEY=
PUSHER_APP_SECRET=
PUSHER_APP_CLUSTER=mt1

MIX_PUSHER_APP_KEY="${PUSHER_APP_KEY}"
MIX_PUSHER_APP_CLUSTER="${PUSHER_APP_CLUSTER}"

JWT_SECRET=
*****
```

- 4) Generar una **nueva key para identificar el proyecto** local sobre el que se está trabajando.  
`php artisan key:generate`
- 5) Hacer lo mismo con la **key de la librería JWT-auth**.  
`php artisan jwt:secret`
- 6) Crear una **base de datos en local** con el nombre puesto en el dato de DB\_DATABASE dentro del fichero `.env`.
- 7) Crear una **migración a la base de datos** de todos los seeders y tablas que se han creado en el proyecto.  
`php artisan migrate:refresh --seed`
- 8) Iniciar la aplicación ejecutando los siguientes comandos de manera simultánea en dos consolas diferentes y estando dentro de la carpeta raíz del proyecto:  
`php artisan serve`  
`npm run watch`

### 9.1.1 Instalación de dependencias.

Las dependencias, tanto en *Composer* como en *Node.js*, son **librerías externas necesarias para el correcto funcionamiento** del proyecto.

En el caso de *Composer*, el listado de dependencias se genera y se guarda dentro del fichero `composer.json`, que se encuentra en la raíz del proyecto.

Para instalar todas las dependencias necesarias de *Composer* hay que ejecutar el siguiente comando en la consola: **composer install**.

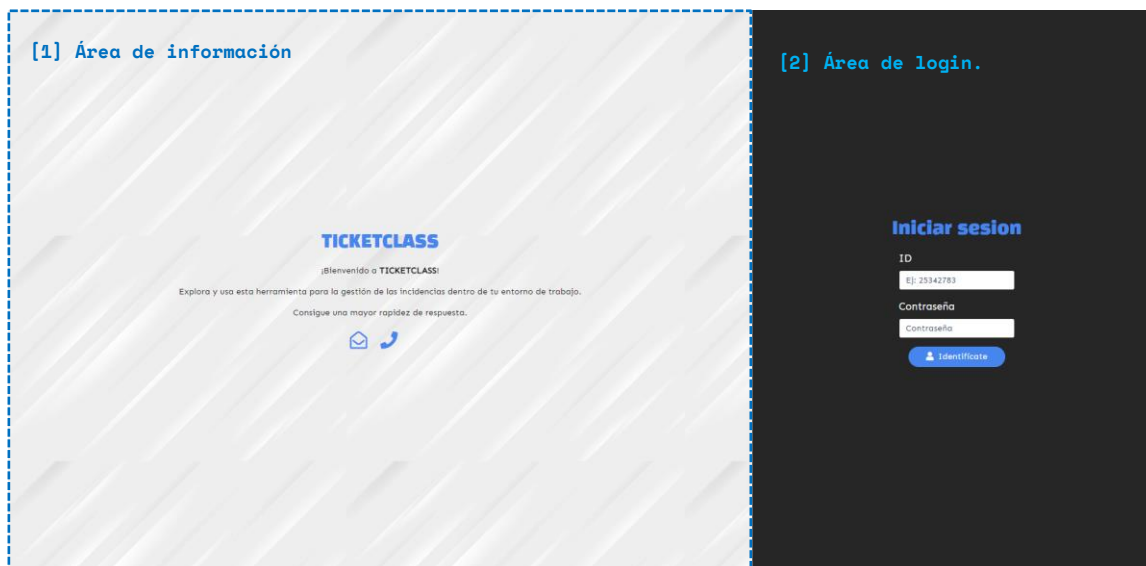
Al igual que *Composer*, las dependencias de *Node.js* se encuentran en el fichero de `package.json`, en la raíz del proyecto, y para instalar todas las dependencias que están listadas en él hay que ejecutar el siguiente comando: **npm install**.

Cada vez que se origine un cambio dentro de cualquiera de los ficheros de instalación de dependencias, será necesario ejecutar los comandos mencionados anteriormente para tener el proyecto actualizado.

## 9.2 Manual de usuario

### 9.2.1 Visión general de la aplicación Ticketing. Home.

La primera página o *home* de la aplicación es el *Login*.



Esta se divide en dos zonas muy simples y fácilmente reconocibles:

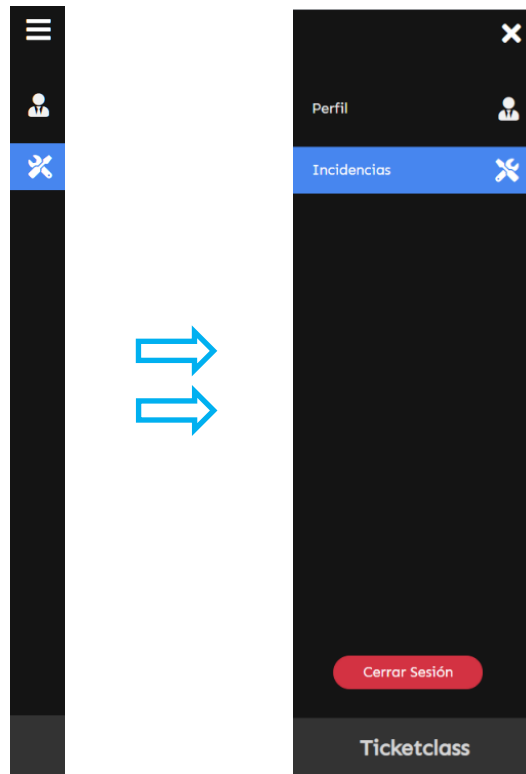
1. **[1] Área de información.** En esta área se encuentra una pequeña descripción de la aplicación web con la que el usuario se pueda hacer una idea simple y rápida de su funcionalidad principal.
2. **[2] Área del Login.** En esta área se realiza únicamente el inicio de sesión. Para ello es necesario tener una cuenta creada dentro del sistema de la aplicación.  
Para iniciar sesión es tan sencillo como introducir el ID o identificador del usuario y la contraseña correspondiente, y finalmente pulsar el botón de 'Identificate' para completar el inicio de sesión.

Una vez dentro de la aplicación, el usuario puede acceder a dos apartados o páginas diferentes mediante un **sidebar**:

1. **Perfil.** Página en la que el usuario puede visualizar sus datos y acceder a dos distintas opciones:
  - a. Editar perfil.
  - b. Visualización de gráficas sobre las incidencias.
2. **Incidencias.** Visualización de incidencias y opciones como:
  - a. Visualización de los datos de la incidencia.
  - b. Comentarios sobre incidencia.
  - c. ...

### 9.2.2 Sidebar

El **sidebar** de la aplicación es el elemento mediante el cual se realizará la navegación entre las distintas páginas de la aplicación: **perfil** y **página de incidencias**. También contiene un botón específico para poder cerrar sesión de manera segura.



*Ilustración 26. Navbar*

### 9.2.3 Perfil

En esta página se pueden visualizar todos los datos correspondientes al usuario que ha iniciado sesión, más opciones de modificación de perfil y visualización de datos sobre las incidencias asignadas.




Ilustración 27. Página PERFIL

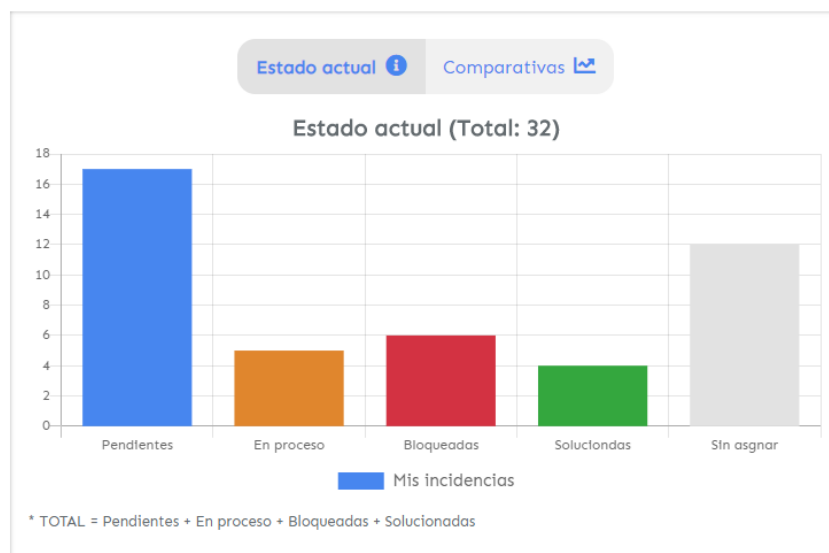
- **[1] Imagen de perfil y opciones.** En esta sección de la página se puede visualizar el nombre del usuario, la imagen de perfil y su rol en la aplicación. Además se encuentran los botones para acceder a las dos opciones disponibles dentro del perfil: visualización de datos sobre incidencias y editar perfil.
- **[2] Información sobre el usuario.** Esta sección recoge todos los datos visibles del usuario.
- **[3] Panel de visualización de las dos opciones.** En este panel se visualizará cada una de las opciones disponibles dependiendo cual se elija.



### **VISUALIZACIÓN DE DATOS SOBRE INCIDENCIAS**

Para poder acceder a ella, hay que pulsar el icono  o directamente entrar en la página de perfil mediante el *sidebar*, ya que viene como predeterminada.

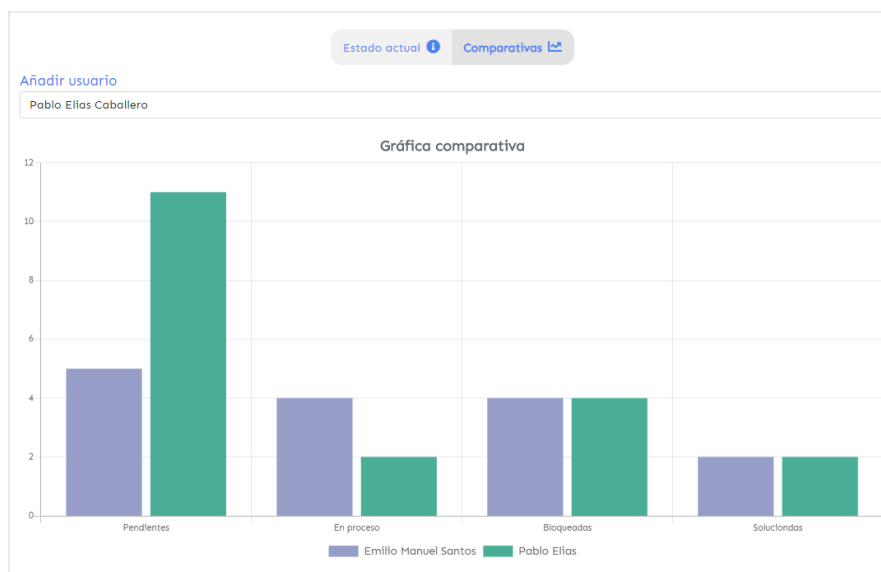
Una vez clicado se cambiará el panel de la derecha de la página mostrando las gráficas disponibles.



*Ilustración 28. Visualización de gráficas usuario.*


### **COMPARATIVAS ENTRE GRÁFICAS**

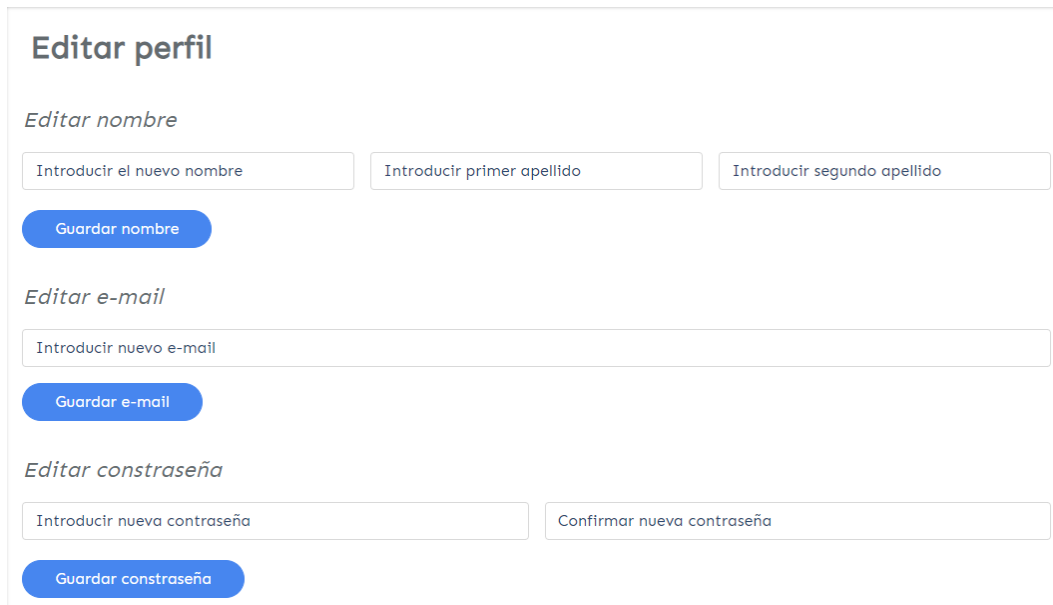
Este apartado solo está disponible para un supervisor. En él puede buscar tantos técnicos como desee y comparar las incidencias que tienen asignadas según el estado de estas.



*Ilustración 29. Comparativas gráficas*

### **EDITAR PERFIL USUARIO**

Para acceder a este panel es necesario pulsar el botón  .



**Editar perfil**

*Editar nombre*

Introducir el nuevo nombre    Introducir primer apellido    Introducir segundo apellido

**Guardar nombre**

*Editar e-mail*

Introducir nuevo e-mail

**Guardar e-mail**

*Editar contraseña*

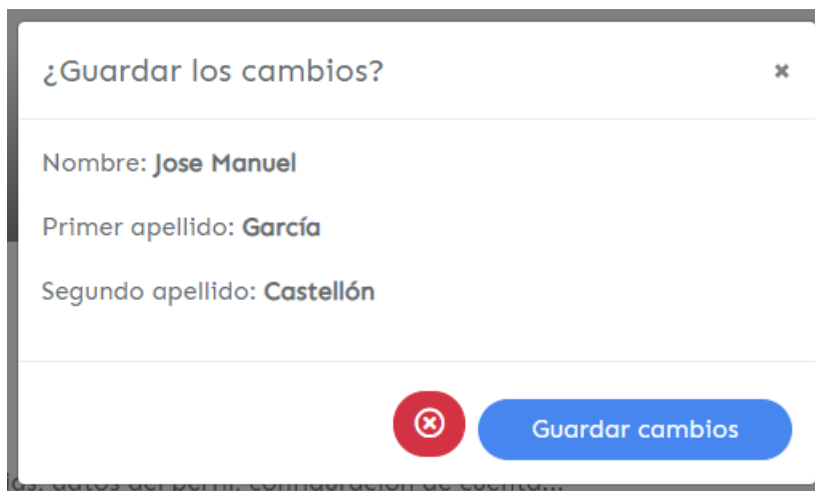
Introducir nueva contraseña    Confirmar nueva contraseña

**Guardar contraseña**

*Ilustración 30. Editar perfil usuario*

En esta sección se pueden realizar 3 acciones: editar nombre completo, editar correo electrónico y cambiar contraseña actual. Cualquiera de estas opciones viene con su comprobación de errores correspondiente.

Una vez estén los datos correctamente introducidos, saldrá una ventana modal con los datos que se van a cambiar.




¿Guardar los cambios?

Nombre: Jose Manuel

Primer apellido: García

Segundo apellido: Castellón

 **Guardar cambios**

*Ilustración 31. Modal editar perfil*

Al guardar los cambios, los datos se actualizarán automáticamente en el apartado sobre la información del usuario.

Información sobre el usuario

Nombre:

Jose Manuel

Primer apellido:

García

Segundo apellido:

Castellón

Expediente:

21619919

Dirección de correo:

Santos2menaaa@gmail.com

Rol:

Supervisor

Teléfono de contacto:

608650958

Ilustración 32. Actualización datos - editar perfil

## 9.2.4 Página de incidencias.

Esta es la página principal y más importante de la aplicación, por lo que nada más realizar el login es la página que se va a encontrar abierta.

Gestor de incidencias - Supervisor

En este apartado usted podrá gestionar todas las incidencias que hayan sido reportadas y asignadas. Tendrá las siguientes opciones:

- Visualizar todas las incidencias.
- Filtrar incidencias para una mejor búsqueda.
- Acceder a los datos de una incidencia específica.
- Editar o eliminar cualquier incidencia una vez se ha accedido a sus datos.
- Visualizar y añadir comentarios a la incidencia.

Mis incidencias

Crear nueva incidencia

Mis grupos de técnicos

Ordenar por:

Seleccionar...

Filtrar por:

Incidencias filtradas: 41

critical: 18

important: 4

trivial: 6

Id	Nombre	Descripción	Categoría	Prioridad	Estado	Fecha límite	¿Asignada?	TTC
#9	Incidencia individual 1	Incidencia individual	WI-FI	crítica	Pendiente	--	SI	21 días
#8	Prueba de fuego	Prueba de fuego	WI-FI	crítica	Pendiente	--	NO	2040 días
#22	zDLzbESXB	2jnGk5oUrWLzIKT85pVtXowYpQXyoo	Red	crítica	Bloqueado	2009-12-30 01:03:46	NO	3793 días
#2	Incidencia grupo 1.1	Incidencia grupo 1.1	WI-FI	crítica	Pendiente	2009-12-30 14:34:29	SI	2040 días
#3	Incidencia grupo 1.2	Incidencia grupo 1.2	WI-FI	crítica	Pendiente	2009-12-30 14:34:29	SI	2040 días
#4	Incidencia grupo 2.1	Incidencia grupo 2.1	WI-FI	crítica	Pendiente	2009-12-30 14:34:29	SI	2040 días
#5	Incidencia grupo 2.2	Incidencia grupo 2.2	WI-FI	crítica	Pendiente	2009-12-30 14:34:29	SI	2040 días
#1	Problema software instalado	El problema se encuentra en que el ordenador XX no tiene instalado el Software xxxx para q...	WI-FI	crítica	Pendiente	2009-12-30 14:34:29	SI	2040 días
#6	Incidencia grupo 3.1	Incidencia grupo 3.1	WI-FI	crítica	Pendiente	2009-12-30 14:34:29	SI	2040 días
#7	Incidencia grupo 3.2	Incidencia grupo 3.2	WI-FI	crítica	Pendiente	2009-12-30 14:34:29	SI	2040 días
#36	XFKjbjuohv	Ox5yXp4Mx5K62IfvV5G4E13a2gx527	Hardware	crítica	En proceso	2009-12-31 02:03:12	NO	3798 días

Ilustración 33. Página incidencias

Dependiendo del tipo de rol que cumpla el usuario que ha iniciado dentro de la aplicación, va a tener más o menos opciones disponibles dentro de esta página:

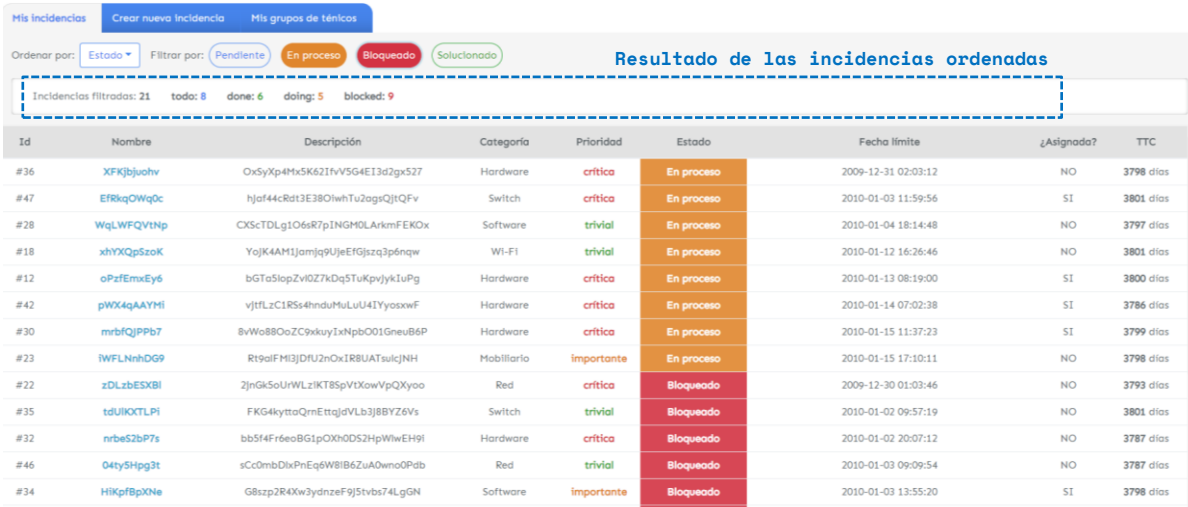
- Supervisor.** El usuario tiene las opciones de 'Mis incidencias', 'Crear nueva incidencia' y 'Mis grupos de técnicos'.

- **Técnico.** Solamente tendrá las dos primeras opciones del supervisor: ‘Mis incidencias’ y ‘Crear nueva incidencia’.

#### 9.2.4.1 Mis incidencias

En este apartado se visualiza una tabla junto a todas las incidencias que están en relación con el usuario que ha iniciado sesión. Dependiendo del rol:

- **Supervisor.** Visualiza las incidencias sin asignar, las creadas y asignadas por él, y las pertenecientes a los grupos de técnicos de los que está a cargo.
- **Técnico.** Visualiza las asignadas a él y a los grupos a los que pertenece.



The screenshot shows the 'Mis incidencias' (My Incidents) interface. At the top, there are tabs for 'Mis incidencias', 'Crear nueva incidencia', and 'Mis grupos de técnicos'. Below the tabs, there are filters for 'Ordenar por:' (Estado) and 'Filtrar por:' (Pendiente, En proceso, Bloqueado, Solucionado). A summary bar indicates 'Incidencias filtradas: 21', 'todo: 8', 'done: 6', 'doing: 5', and 'blocked: 9'. The main table displays the following data:

Id	Nombre	Descripción	Categoría	Prioridad	Estado	Fecha límite	¿Asignada?	TTC
#36	XFKjbuohv	OxSyXp4Hx5K62IvV5G4E13d2gx527	Hardware	crítica	En proceso	2009-12-31 02:03:12	NO	3798 días
#47	EFKqOWqOc	hJof44cRdt3E38OIwhTu2agsQjtQFv	Switch	crítica	En proceso	2010-01-03 11:59:56	SI	3801 días
#28	WqLWfQVtNp	CXScTDLg1O6sR7pINGM0LAkmFEKOx	Software	trivial	En proceso	2010-01-04 18:14:48	NO	3797 días
#18	xhYXQp5zoK	YojK4AM1Jamjg9UjeEIGjszq3p6nqw	Wi-Fi	trivial	En proceso	2010-01-12 16:26:46	NO	3801 días
#12	oPzfEmxEy6	bGTa5lopZvI0Z7kDq5TuKpvjyKtUpG	Hardware	crítica	En proceso	2010-01-13 08:19:00	SI	3800 días
#42	pWXqAAAYMl	vJtTLzC1RS4hnduMuLuU4IYyosxwF	Hardware	crítica	En proceso	2010-01-14 07:02:38	SI	3786 días
#30	mrbfQjPPb7	8vWo88OoZC9xkuyIxNpbO01GneuB6P	Hardware	crítica	En proceso	2010-01-15 11:37:23	SI	3799 días
#23	lWFLNnhDG9	Rt9oIFMI3JDFU2nOx1R8UATsulcJNH	Mobiliario	importante	En proceso	2010-01-15 17:10:11	NO	3798 días
#22	zDLzbESXBli	2JnGk5oUrWLzIKT8SpVixKovVpQXyoo	Red	crítica	Bloqueado	2009-12-30 01:03:46	NO	3793 días
#35	tdUliKXtLPi	FKG4kyttaQmEttqjvLb3j8BYZ6Vs	Switch	trivial	Bloqueado	2010-01-02 09:57:19	NO	3801 días
#32	nrbE52bP7s	bb5f4Fr6eoBG1pOXhODS2HpWlwEH9l	Hardware	crítica	Bloqueado	2010-01-02 20:07:12	NO	3787 días
#46	04tySHpg3t	sCc0mbDlxPnEq6W8IB6ZuA0wnoOPab	Red	trivial	Bloqueado	2010-01-03 09:09:54	NO	3787 días
#34	HtkpfbpXNe	G8szp2R4Xw3ydnzeF9j5tvbs74LgGN	Software	importante	Bloqueado	2010-01-03 13:55:20	SI	3798 días

Ilustración 34. Filtro de incidencias

El usuario tiene a su disposición un filtro mediante el cual puede ordenar según la propiedad de la incidencia que él desee. Depende de la propiedad seleccionada, podrá filtrar o no, tal y como se refleja en la **Ilustración 34. Filtro de incidencias**.

La tabla se encuentra dividida en 9 columnas: **Id**, **Nombre**, **Descripción**, **Categoría**, **Propiedad**, **Estado**, **Fecha límite**, **¿Asignada?** y **TTC** (Tiempo Transcurrido desde la Creación).

La única columna que cambia respecto a los diferentes roles es la de **Asignada**, ya que en el rol de supervisor se muestra únicamente si la incidencia se encuentra asignada o no, mientras que en el rol de técnico, muestra si la incidencia está asignada al usuario en cuestión o a alguno de los equipos a los que pertenece.

### **VISUALIZACIÓN DE LOS DATOS DE UNA INCIDENCIA ESPECÍFICA**

Para acceder a los datos de una incidencia solamente hay que pinchar sobre su nombre en la tabla y automáticamente se redirigirá al usuario a la página con todos los datos de dicha incidencia.



**[1] Nombre de incidencia, cambio de estado y opciones disponibles**

**[2] Datos sobre la incidencia**

Reportar	Adrian Santos Mena
Técnico asignado (ID)	--
Equipo asignado	Equipo de aulas
Supervisor	Adrian Santos Mena
Título	Incidencia grupo 2.1
Descripción	Incidencia grupo 2.1
Categoría	Wi-Fi
Edificio	C
Piso	3
Aula	C307
URL de acceso al archivo	--
Fecha límite de realización	2009-12-30 14:34:29
Fecha de creación	2014-10-25 20:00:00
Fecha de asignación	--
Fecha de resolución	--
Prioridad	critical
Estado actual	todo

[Volver a la página anterior](#)

**[3] Panel para las diferentes opciones**

Comentario

Enviar comentario...

No hay comentarios disponibles

*Ilustración 35. Visualización datos de incidencia*

En la **Ilustración 35. Visualización datos de incidencia**, se visualiza como es dentro de la aplicación la página de información sobre una incidencia.

1. **[1] Nombre de incidencia, cambio de estado y opciones disponibles.** En esta parte de la página se realiza el cambio de estado de la incidencia. Cada vez que se realice un cambio de estado, se podrá guardar el porqué de ese cambio y, a su vez, llegará un email informativo tanto al supervisor como al técnico asignado de que ha habido un cambio en el estado de la incidencia.
2. **[2] Datos sobre la incidencia.** En este apartado se pueden visualizar todos los datos informativos sobre la incidencia.
3. **[3] Panel para las diferentes opciones.** Este apartado sirve para poder visualizar las opciones disponibles en la parte [1] de la imagen.  
Las opciones disponibles son: editar incidencia, comentarios y eliminar incidencia.

### **EDITAR INCIDENCIA**

Opción mediante la cual se puede editar la incidencia. Esta opción solamente tienen acceso los supervisores y el creador de la incidencia.

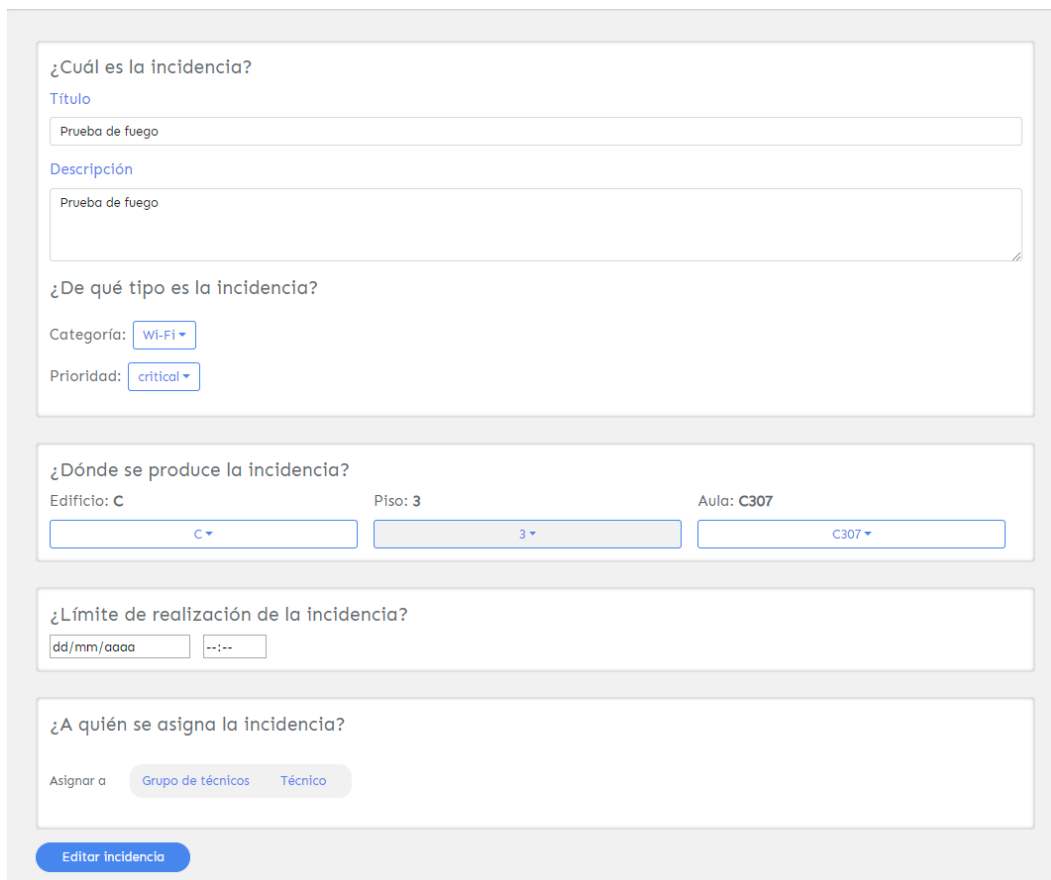


Ilustración 36. Editar incidencia.

## **COMENTARIOS**

En este apartado aparecen todos los **comentarios que se han ido realizando sobre la incidencia**. Para mandar el comentario simplemente hay que poner dentro del *textarea* el texto y darle al botón que se encuentra a la izquierda.

Una vez se haya enviado el comentario, llegará un correo informativo al supervisor de la incidencia, el técnico asignado y todos los usuarios que hayan comentado sobre la incidencia.

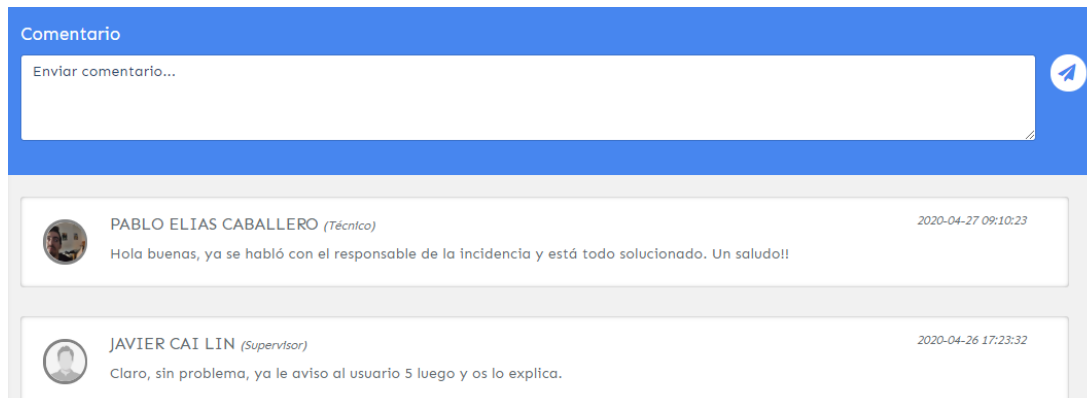
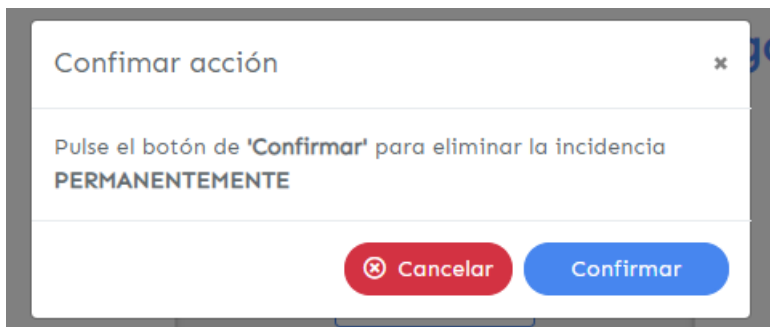


Ilustración 37. Comentarios incidencia.

### **ELIMINAR INCIDENCIA**

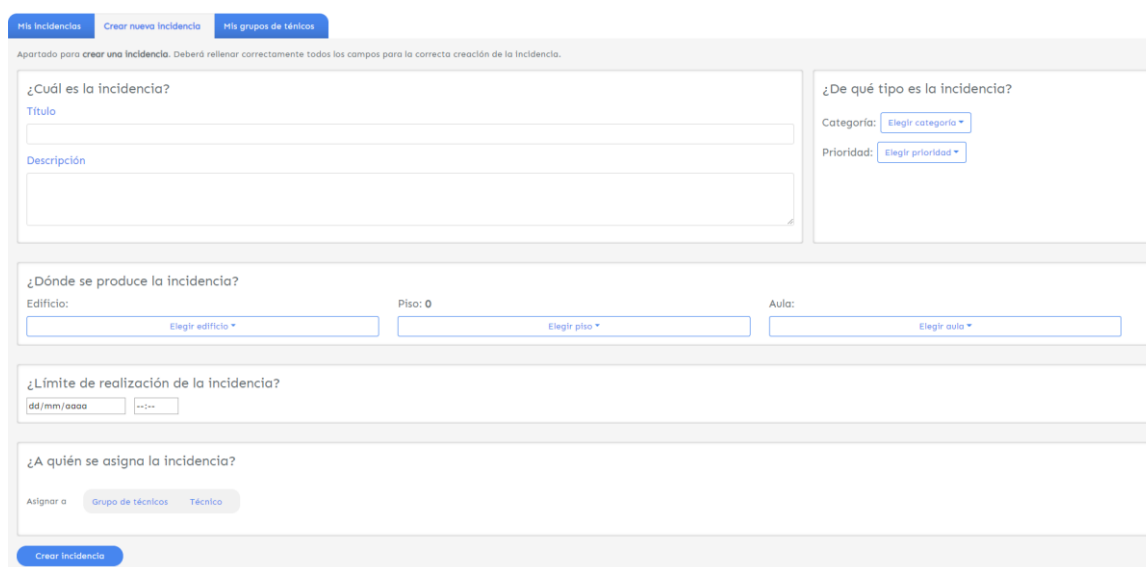
Al clicar sobre 'eliminar incidencia', aparecerá una modal para confirmar la acción. La acción de eliminar incidencia solo está disponible para los supervisores y los creadores de la incidencia.



*Ilustración 38. Confirmación eliminar incidencia.*

#### **9.2.4.2 Crear nueva incidencia**

En este apartado se genera una incidencia. Para ello es necesario rellenar todos los apartados correctamente, ya que estos tienen sus respectivas comprobaciones de errores para evitar generar incidencias incompletas o mal creadas.

El formulario está dividido en secciones. La primera sección, '¿Cuál es la incidencia?', contiene campos para 'Título' y 'Descripción'. La segunda, '¿De qué tipo es la incidencia?', tiene selectores para 'Categoría' y 'Prioridad'. La tercera, '¿Dónde se produce la incidencia?', incluye campos para 'Edificio', 'Piso' y 'Aula'. La cuarta, '¿Límite de realización de la incidencia?', tiene campos para la fecha y hora. La quinta, '¿A quién se asigna la incidencia?', tiene un selector 'Asignar a' con opciones 'Grupo de técnicos' y 'Técnico'. En la parte superior hay pestañas: 'Mis incidencias', 'Crear nueva incidencia' (activa) y 'Mis grupos de técnicos'. En la parte inferior hay un botón 'Crear incidencia'.

*Ilustración 39. Crear incidencia.*

Hay que señalar que si el rol del usuario es supervisor, tendrá la opción de asignar directamente la incidencia al crearla, en caso de ser técnico, la incidencia se creará pero sin asignación.

#### **9.2.4.3 Mis grupos de técnicos.**

Este apartado es exclusivo de los supervisores. En él podrá manejar los grupos de técnicos de los que esté a cargo, añadir técnicos, eliminarlos y crear nuevos grupos de técnicos.

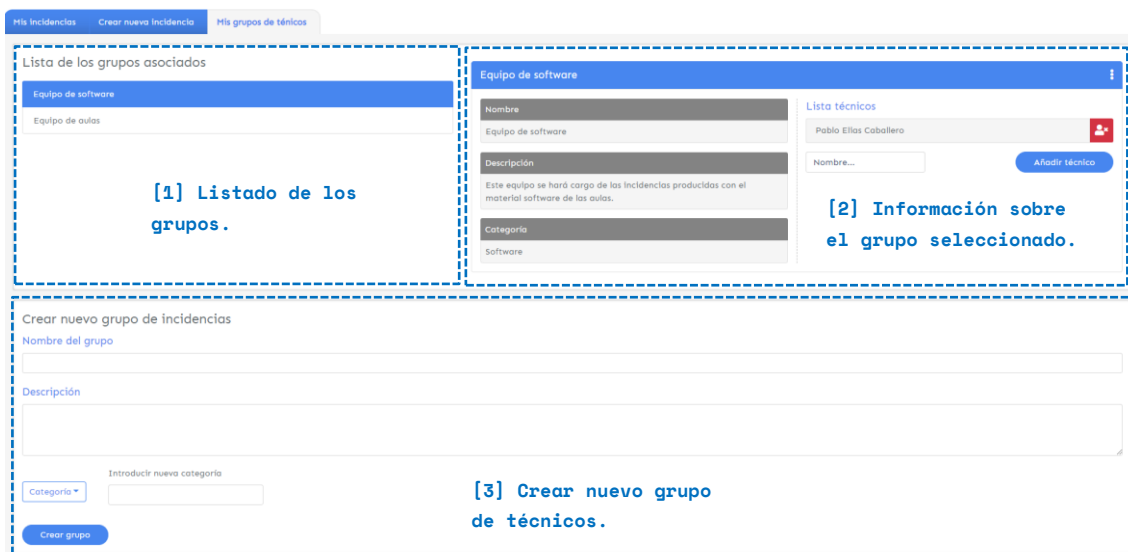


Ilustración 40. Grupos de técnicos.

En la Ilustración 40. Grupos de técnicos., se muestra la maquetación de la página de grupos de técnicos. En ella se pueden diferenciar 3 secciones:

1. **[1] Listado de grupos.** En esta sección se muestra un listado con todos los grupos de los que el supervisor está a cargo.
2. **[2] Información sobre el grupo seleccionado.** Una vez seleccionado un grupo de la lista, se visualizarán los datos en esta sección, junto a una lista de los técnicos que están asignados a él.

En esta sección se pueden también eliminar y añadir técnicos según el supervisor vea conveniente.

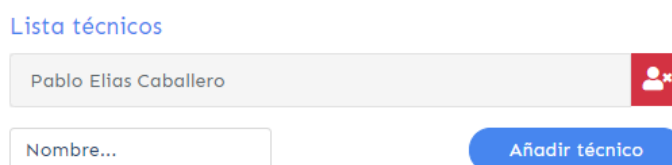


Ilustración 41. Añadir y eliminar técnico.

Para eliminar el técnico es tan simple como clickar sobre el botón rojo que hay a la derecha de su nombre.

Para añadirlo hay que primero buscarlo mediante el input y una vez seleccionado, añadirlo al grupo clickando sobre el botón azul de 'Añadir técnico'.

3. **[3] Crear nuevo grupo de técnicos.** En esta sección el supervisor puede crear un nuevo grupo de técnicos, pudiendo a su vez crear una nueva categoría para este o eligiendo una existente.

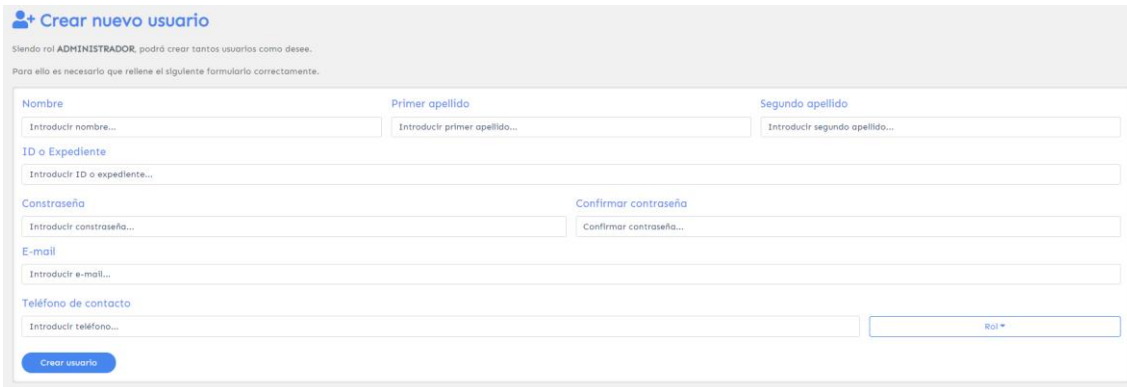


#### 9.2.4.4 Crear nuevo usuario

La aplicación también dispone de un rol medio ‘incógnito’ al que se le ha llamado *administrador*. Con él se pueden crear nuevos usuarios en la aplicación e importarlos.

##### **CREAR USUARIO**

Al igual que el resto de los formularios de la aplicación, es necesario introducir correctamente los datos en los campos para poder crear el usuario correctamente. En el caso en el que algún campo no esté bien introducido, aparecerá en él un mensaje de error indicando qué error es el que se produce.



Crear nuevo usuario

Siendo rol ADMINISTRADOR, podrá crear tantos usuarios como desee.  
Para ello es necesario que rellene el siguiente formulario correctamente.

Nombre: Introducir nombre... Primer apellido: Introducir primer apellido... Segundo apellido: Introducir segundo apellido...

ID o Expediente: Introducir ID o expediente...

Contraseña: Introducir contraseña... Confirmar contraseña: Confirmar contraseña...

E-mail: Introducir e-mail...

Teléfono de contacto: Introducir teléfono... Rol: \*

Crear usuario

Ilustración 42. Crear usuario.

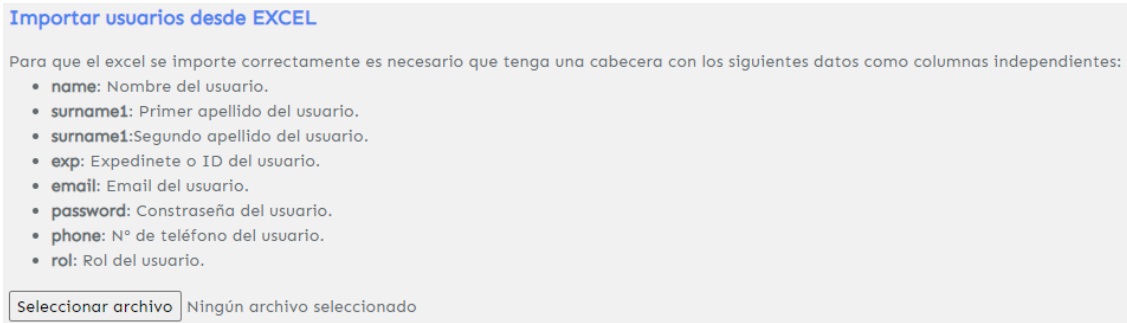
##### **IMPORTAR USUARIOS DESDE EXCEL**

La aplicación también da la posibilidad al administrador de crear muchos usuarios a la vez mediante un archivo EXCEL. Es necesario que este archivo Excel tenga una estructura específica para que se almacenen correctamente los nuevos usuarios.

La estructura tiene que ser la siguiente, aunque ya viene indicada dentro de la propia aplicación:

name	surname1	surname2	exp	password	email	phone	role
------	----------	----------	-----	----------	-------	-------	------

donde en cada columna tiene que ir el dato correspondiente al usuario.



Importar usuarios desde EXCEL

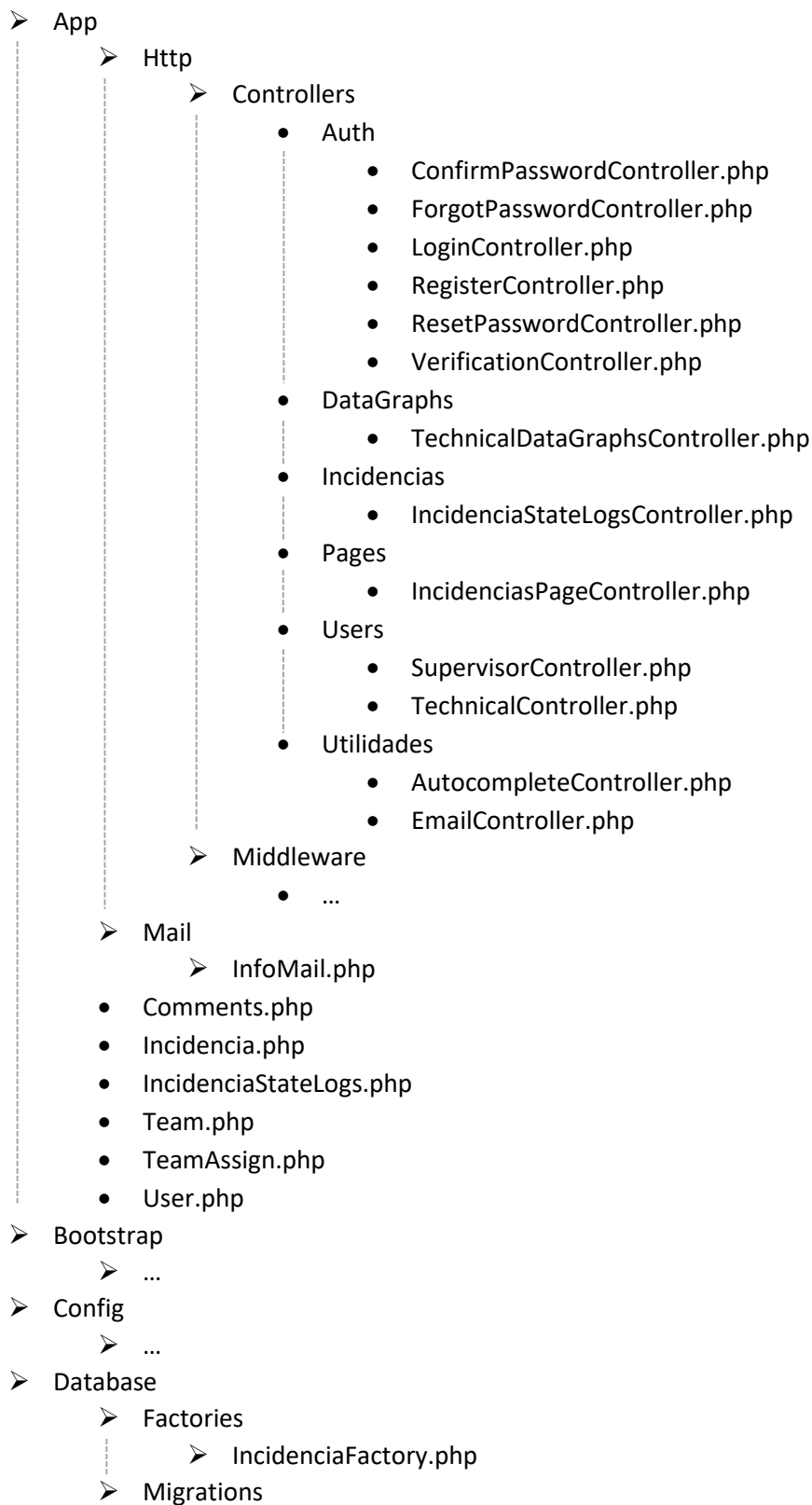
Para que el excel se importe correctamente es necesario que tenga una cabecera con los siguientes datos como columnas independientes:

- **name:** Nombre del usuario.
- **surname1:** Primer apellido del usuario.
- **surname2:** Segundo apellido del usuario.
- **exp:** Expediente o ID del usuario.
- **email:** Email del usuario.
- **password:** Contraseña del usuario.
- **phone:** N° de teléfono del usuario.
- **rol:** Rol del usuario.

Seleccionar archivo Ningún archivo seleccionado

Ilustración 43. Importar usuarios desde Excel.

### 9.3 Estructura completa del proyecto



- 2014\_10\_12\_000000\_create\_users\_table.php
- 2014\_10\_12\_100000\_create\_password\_resets\_table.php
- 2019\_08\_19\_000000\_create\_failed\_jobs\_table.php
- 2020\_02\_22\_162256\_create\_incidencias\_table.php
- 2020\_03\_22\_154915\_create\_teams\_table.php
- 2020\_03\_23\_003711\_create\_team\_assigns\_table.php
- 2020\_04\_27\_153729\_create\_comments\_table.php
- 2020\_04\_29\_090821\_create\_incidencia\_state\_logs\_table.php
- Seeds
  - CommentsSeeder.php
  - DatabaseSeeder.php
  - IncidenciasStateLogsSeeder.php
  - TeamAssignsSeeder.php
  - TeamSeeder.php
  - UsuariosTableSeeder.php
- Git
- Node\_modules
- Public
  - Bootstrap-4.4.1-dist
  - Css
  - Images
  - Js
  - Json
  - Sass
- Resources
  - Js
    - Components
      - AutocompleteInput
        - AutocompleteInput.scss
        - AutocompleteInput.tsx
      - Button
        - Button.scss
        - Button.tsx
      - DataCard
        - DataCard.scss
        - DataCard.tsx
      - Dropdown
        - Dropdown.scss
        - Dropdown.tsx
      - Input
        - Input.scss
        - Input.tsx

- Modal
  - Modal.scss
  - Modal.tsx
- Navbar
  - Navbar.scss
  - Navbar.tsx
- Selectbox
  - Selectbox.scss
  - Selectbox.tsx
- Tabs
  - Tabs.scss
  - Tabs.tsx
- Toast
  - Toast.scss
  - Tabs.tsx
- UploadFile
  - UploadFile.scss
  - UploadFile.tsx
- Model
  - Model.tsx
- Pages
  - IncidenciasPage
    - CommentPage
      - CommentPage.scss
      - CommentPage.tsx
    - IncidenciaViewPage
      - IncidenciaViewPage.scss
      - IncidenciaViewPage.tsx
    - TabOptions
      - CreateIncidenciaPage
        - CreateIncidenciaPage.scss
        - CreateIncidenciaPage.tsx
      - MostrarIncidenciasPage
        - MostrarIncidenciasPage.scss
        - MostrarIncidenciasPage.tsx
      - TechnicalGroupsPage
        - TechnicalGroupsPage.scss
        - TechnicalGroupsPage.tsx
    - IncidenciasPage.scss
    - IncidenciasPage.tsx
  - Login

- Login.scss
        - Login.tsx
      - MainPage
        - MainPage.scss
        - MainPage.tsx
      - PerfilPage
        - GraphsPage
          -
        - SettingsPage
        - PerfilPage.scss
        - PerfilPage.tsx
    - Utilities
      - Graphics
        - TechnicalDataGraphs.tsx
      - Incidencias
        - IncidenciaStateLogsUtilities.tsx
        - IncidenciasUtilities.tsx
        - SupervisorUtilities.tsx
        - TechnicalUtilities.tsx
      - API.tsx
      - Authentication.tsx
      - Autocomplete.tsx
    - Widgets
      - Navigation.tsx
      - Root.tsx
  - Lang
    - ...
  - Sass
    - App.scss
  - Sketches
    - Navbar.sketch
    - Perfil\_v1.sketch
  - Views
    - Emails
      - changeState.blade.php
    - Root.blade.php
  - Wireframes
- Routes
  - Api.php
  - Channels.php
  - Console.php
  - Web.php

- Storage
- Tests
- .env
- .gitattributes
- .gitignore
- .styleci.yml
- Artisan.php
- Composer.json
- Composer.lock
- Helper.txt
- LICENSE
- Package-lock.json
- Package.json
- Phpunit.xml
- README.md
- Server.php
- Tsconfig.json
- Webpack.mix.js