



Trabajo de Fin de Grado

Plataforma de emergencias

Emergency platform

Sonia Díaz Santos

La Laguna, 29 de mayo de 2020

Dña. **Pino Caballero Gil**, con N.I.F. 45.534.310-Z Catedrática de Universidad adscrita al Departamento de Ingeniería Informática y de Sistemas, como tutora

D. **Carlos Rosa Remedios**, con N.I.F. 43.786.084-H responsable de la Unidad de Tecnologías de la Información y la Comunicación de Gestión de Servicios para la Salud y Seguridad en Canarias, como cotutor

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

Que la presente memoria titulada:

"Plataforma de emergencias"

ha sido realizada bajo su dirección por D. **Sonia Díaz Santos**, con N.I.F. 42.237.482-Y.

Y para que así conste, en cumplimiento de la legislación vigente y a los efectos oportunos firman la presente en La Laguna a 29 de mayo de 2020

Agradecimientos

En primer lugar agradecer tanto a mi tutora, Pino Caballero Gil, como a mi cotutor, Carlos Rosa Remedios, por su ayuda, orientación y apoyo durante la realización de este proyecto.

También agradecer a mi familia, amigos y compañeros que me han ayudado y apoyado a lo largo de estos meses de trabajo.

Licencia



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

Resumen

Este trabajo se ha basado en crear una plataforma de emergencias para la empresa 112 [1] que consiste en crear una página web que sea un SMC (Sistema de Mando y Control) en la que se almacenen los INRs (Incidentes No Rutinarios) con toda la información, funcionalidad y recursos necesarios para su entendimiento. La finalidad es que la aplicación se asemeje de la manera más exacta con la aplicación utilizada en la empresa. Los usuarios finales de la aplicación van a ser los propios trabajadores de la empresa.

La idea de este proyecto es permitir que el usuario pueda hacer uso la aplicación web de manera adecuada, sencilla y segura. Por este motivo se ha hecho uso de la accesibilidad, usabilidad y seguridad. Cabe destacar que al ser una empresa tan importante para los ciudadanos asegurar la seguridad es primordial para evitar posibles ataques.

Finalmente para desarrollar este proyecto se han empleado diversas tecnologías, como el stack MEVN (Mongodb, Express, Vuejs, Nodejs) [2] para el desarrollo del proyecto, Heroku [3] para desplegar el proyecto de forma continua en la nube, y GitHub [4] como sistema de control de versiones Git [5]. El proyecto se encuentra separado en dos partes, el backend [6] que se encarga de la parte lógica de la página web y el frontend [7] que es la parte enfocada a la interacción con el usuario.

Palabras clave: SMC, INR, Accesibilidad, Usabilidad, Seguridad, MEVN, Heroku, GitHub, Backend, Frontend

Abstract

This work has been based on creating an emergency platform for the 112 company that consists of creating a web page that is an SMC (Command and Control System) in which the INRs (Non-Routine Incidents) are stored with all the information, functionality and resources necessary for its understanding. The purpose is to make the application more closely resemble to the application used in the company. The end users of the application will be the workers of the company.

The idea of this project is to allow the user to make use of the web application in an adequate, simple and safe way. For this reason, accessibility, usability and security have been used. It should be noted that being such an important company for citizens ensuring security is essential to avoid possible attacks.

Finally, to develop this project various technologies have been used such as the MEVN stack (Mongodb, Express, Vuejs, Nodejs) for the development of the project, Heroku to deploy the project continuously in the cloud and GitHub as a Git version control system. The project is divided into two parts, the backend that is in charge of the logical part of the web page and the frontend that is the part focused on user interaction.

Keywords: SMC, INR, Accessibility, Usability, Security, MEVN, Heroku, GitHub, Backend, Frontend

Índice general

1. Introducción	1
1.1. Motivación y justificación del proyecto	1
1.2. Antecedentes y estado actual	1
1.3. Objetivos	3
1.4. Fases del desarrollo	4
1.5. Estructura de la memoria	5
2. Herramientas y tecnologías	6
2.1. Bases de la aplicación y arquitectura software: Stack de desarrollo con MEVN	6
2.2. Herramientas de desarrollo e integración continua	9
3. Descripción de la aplicación y guía de uso	11
3.1. Diseño inicial: Bocetos	11
3.2. Usuarios y roles	15
3.3. Sección de inicio de sesión y registro	15
3.4. Sección de inicio	17
3.5. Sección de documentos	22
3.6. Sección de actividad	24
3.7. Sección de búsqueda	26
3.8. Sección de SGE-112	26
3.9. Sección de mapa	27
3.10 Sección de administrador	28
4. Desarrollo	29
4.1. Backend	29
4.2. Frontend	37
5. Evaluación	40
5.1. Usabilidad y accesibilidad	40
5.2. Seguridad de la página web	43
6. Presupuesto	46
7. Conclusiones y líneas futuras	47
8. Summary and conclusions	48
Bibliografía	49

Índice de Figuras

1.1. Hexagon Oncall Dispath	3
2.1. Docker utilizado en el proyecto	7
2.2. Ejemplo robo3t de este proyecto	8
2.3. Ejemplo de uso de Postman	8
2.4. Modificación package.json para despliegue en Heroku	9
2.5. Despliegue de la aplicación en Heroku	10
2.6. Estructura repositorio GitHub	10
3.1. Boceto de la sección de inicio	11
3.2. Boceto de la sección de documentos	12
3.3. Boceto de la sección de actividad	13
3.4. Boceto de la sección de búsqueda	13
3.5. Boceto de la sección de SGE-112	14
3.6. Boceto de la sección de capas	14
3.7. Ventana de inicio de sesión	15
3.8. Ventana de registro	16
3.9. Usuario pendiente de validación tras registrarse	16
3.10Alerta de contraseñas no coincidentes	16
3.11Sección de inicio como administrador	17
3.12Sección de inicio como usuario normal	18
3.13Formulario de crear nuevo INR	18
3.14Resultado de crear el INR	19
3.15Formulario de modificar un INR existente	19
3.16Resultado de modificar el INR existente	19
3.17Formulario para eliminar el INR	20
3.18Filtros por fechas	20
3.19Filtros por gravedad	21
3.20Filtros por simulacros	21
3.21Usuarios activos	21
3.22Sección de documentos como administrador	22
3.23Sección de documentos como usuario normal	23
3.24Ejemplo de sección documentos de otro INR	23
3.25Formulario de añadir documento	23
3.26Sección de actividad para el administrador	24
3.27Sección de actividad para un usuario normal	25
3.28Información para una guía de estilo común entre los trabajadores	25
3.29Formulario para rectificar una aportación	25
3.30Sección búsqueda para cualquier usuario	26

3.31 Sección SGE-112 para las llamadas	27
3.32 Sección mapa	27
3.33 Sección administrador	28
4.1. Código del servidor del backend	29
4.2. Código del modelo de la base de datos del usuario	30
4.3. Código del modelo de la base de datos del INR	31
4.4. Código del modelo de la base de datos de la carpeta	31
4.5. Código del modelo de la base de datos de la aportación	32
4.6. Código de las vistas del backend	33
4.7. Código de la función de inicio de sesión del usuario	34
4.8. Código de la función de eliminar un INR	35
4.9. Código de la función de modificar los datos de una aportación	36
4.10 Código de añadir un fichero en una carpeta	37
4.11 Vistas del frontend	38
4.12 Componentes del frontend	38
4.13 Código de las rutas del frontend	39
4.14 Ejemplo de los estados Vuex	39
5.1. Resultado del porcentaje de usabilidad con el método Sirius	40
5.2. Accesibilidad medida con WAVE inicialmente	42
5.3. Accesibilidad medida con WAVE tras la corrección de errores	42
5.4. Desencriptado bcrypt	45

Índice de Tablas

6.1. Modelo de presupuesto para los tres meses de duración del proyecto	46
---	----

Capítulo 1

Introducción

1.1. Motivación y justificación del proyecto

Es de vital importancia contar con un centro coordinador desde el que se gestionen los recursos de emergencias en las diferentes localidades. Esto se debe a que ante cualquier alerta en un amplio abanico de ámbitos como son el sanitario, la seguridad, el salvamento, la extinción y el rescate es muy importante recibir una respuesta inmediata. La seguridad de todos los ciudadanos, de la fauna y flora, y de cualquier terreno o aspecto que sea importante para la humanidad es clave para poder lidiar con los problemas y accidentes, tanto diarios como los incidentes no rutinarios (INR). Los INR son incidentes cuya magnitud genera la activación de medios extraordinarios, afecte o no a múltiples víctimas.

En este proyecto se ha desarrollado una plataforma web para tratar, documentar y visualizar todos los datos pertinentes acerca de cualquier INR que ocurra. A esta plataforma van a tener acceso una serie de usuarios autorizados, existiendo diferentes roles y permisos donde dependiendo de la función que desempeñe un usuario concreto va a poder realizar diferentes acciones o visualizar una determinada información.

La plataforma web que se va a desarrollar va a ser usable y accesible por los usuarios mencionados con anterioridad, además de contar con una parte de seguridad tanto en el sitio web como en los datos que se puedan tratar dentro del mismo.

1.2. Antecedentes y estado actual

La empresa internacional Hexagon [8] tiene una división especializada en el tratamiento de eventos e incidencias denominado Hexagon Oncall

Dispath para la gestión operativa de incidentes utilizando soluciones geo-espaciales para mostrar información en tiempo real como se muestra en la Figura 1.1. Es utilizada por diferentes cuerpos y fuerzas de seguridad de múltiples países. Se compone de diversos aspectos como la recepción de alarmas y tipificación de incidentes, la gestión de recursos, unidades que reciben, informan y comparten información mediante la utilización de tablets o radios y agentes de campo que reciben, informan y comparten la información. Este sistema cuenta con una arquitectura web que permite el despliegue SaaS, un cliente ligero con cloud híbrido, una accesibilidad desde cualquier dispositivo o plataforma y un mapa en el que se puede visualizar y ocultar información cuando se realiza zoom para no sobrecargar la interfaz ni al usuario. También tiene capacidad de agrupación (clustering) de incidentes, representación de diferentes tipos de accidente con iconos de incidentes con distintos estilos y menús contextuales radiales para unidades e incidentes.

La plataforma web que se utiliza actualmente como Sistema de Mando y Control (SMC) tiene como objetivo integrar exclusivamente los INR para manejar la documentación asociada a ellos y su clasificación mediante hitos o subincidencias. Por lo tanto, el SMC se utiliza principalmente como registro de los INR que se están produciendo convirtiéndolo en un gestor documental para añadir, guardar y ver hitos pertenecientes a los INR. La función principal es la realización de informes y estudios posteriores del INR con la línea temporal de todos los documentos y anexos de este. Asimismo la plataforma actual cuenta con una sección dedicada a la visualización de un mapa, con diferentes capas que pueden ser mostradas u ocultadas, que proceden de tomas de datos variadas. Estas capas son muy básicas por el momento por ello actualmente se centra más en la gestión documental y no en la visualización de datos.

Durante el desarrollo de la plataforma de emergencias se va a realizar no solo una mejor distribución de los contenidos sino la adaptación a las necesidades de los usuarios específicos que la vayan a utilizar, mejorando así la experiencia de usuario elaborando una plataforma más accesible y usable con un diseño minimalista e intuitivo. Además de reestructurar la página se van a añadir funcionalidades que puedan ser de utilidad para los usuarios y se va a securizar la página web y los datos que se utilicen dentro de ella, ya que tratándose de una plataforma de emergencias la protección de los datos de los incidentes y de los usuarios es un aspecto a tratar muy

importante. Con todo ello se pretende mejorar el diseño y adaptarlo a los usuarios específicos que lo van a utilizar añadiendo la mayor cantidad de funcionalidad posible.

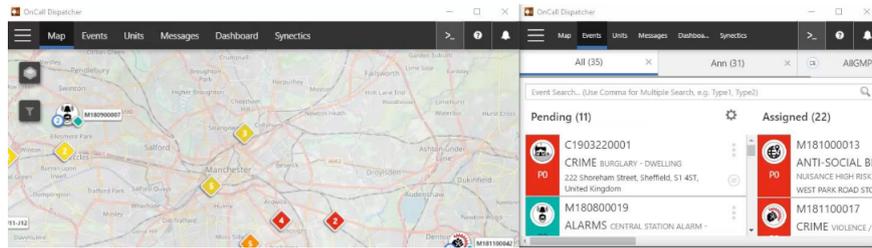


Figura 1.1: Hexagon Oncall Dispath

1.3. Objetivos

El objetivo principal del proyecto es crear una plataforma de emergencias para los INR. Dicha plataforma ha de ser usable, accesible y segura para el uso de los trabajadores del 112, pudiendo distinguir diferentes roles para dichos trabajadores.

Para poder llevar a cabo el proyecto se han establecido una serie de objetivos:

- **OBJ-1:** Realizar un diseño inicial de la plataforma con los bocetos.
- **OBJ-2:** Crear la interfaz gráfica basándose en los bocetos e implementar algunas funcionalidades:
 - Crear un inicio de sesión y registro para la identificación de éstos y la diferenciación de los diferentes roles.
 - Crear la página de inicio donde se tendrá acceso a los INR contando con una serie de filtros para mejorar la búsqueda de un INR concreto.
 - Crear la página de documentos para adjuntar documentos nuevos a un INR concreto contando con una estructura de sistemas de archivos y, con un buscador de documentos. Asimismo se contará con un apartado de carpetas “Favoritos” para un acceso más rápido a carpetas utilizadas con más frecuencia. Se tiene una sección de datos relevantes sobre el INR concreto y una sección de mapa en el que se ve la localización donde tiene lugar el INR con la opción de ampliar para ver mejor.
 - Crear la página de actividad para registrar la subida de documentos y las aportaciones realizadas por todos los usuarios en ese INR.

Así se pueden crear nuevas aportaciones o rectificar una previa. Además, cuenta con una pestaña de más información para consultar el formato conjunto a seguir al realizar nuevas aportaciones. Cuenta con las secciones de datos del INR y del mapa.

- Crear la página de SGE-112 para que se tenga acceso a las llamadas realizadas a la sala 112 relativas a ese INR. Cuenta con las secciones de datos del INR y del mapa.
 - Crear la página de capas para establecer las modificaciones oportunas sobre el mapa tales como las vistas de mapa, satélite o relieve, como la de visualización de las capas existentes si se quiere activar o desactivar una capa en concreto, la visualización históricas de las capas en una fecha determinada o importar o exportar capas. Todas estas opciones se encuentran también cuando se amplía el mapa a pantalla completa para un mejor manejo y visualización del mapa.
- **OBJ-3:** Mejorar la usabilidad y accesibilidad de la página web.
 - **OBJ-4:** Mejorar la seguridad de la página web, tanto de los datos que utiliza como de la propia página.

1.4. Fases del desarrollo

El desarrollo del presente proyecto se puede agrupar en cinco fases diferentes. Una primera fase se dedica a entender el funcionamiento de emergencias para establecer un diseño inicial de la plataforma de emergencias. Asimismo se realizan los bocetos de las páginas más relevantes de la plataforma priorizando el minimalismo, la accesibilidad y la claridad de la misma.

Una segunda fase en la que se desarrollan los bocetos definitivos y se realiza la etapa de aprendizaje orientada a la documentación acerca de las herramientas y tecnologías que se utilizarán en el proyecto. Al tratarse de tecnologías que no han sido utilizadas previamente se realiza la documentación tanto con búsquedas como con videotutoriales sobre Vue.js y el stack MEVN (MongoDB, Express, VueJS y NodeJS).

Una tercera fase que trata sobre el pilar fundamental del proyecto. Se crea la interfaz gráfica para que el usuario pueda ver e interactuar con la información de la página de forma digital, garantizando una navegación

simple y directa a través de la página. Además se desarrollan funcionalidades nuevas para mejorar la experiencia de usuario y la usabilidad de la página.

Una cuarta fase de verificación de la usabilidad y accesibilidad de la plataforma para los usuarios que la van a utilizar. Así como una mejora de la seguridad tanto de la página web como de los datos que se utilicen dentro de esta.

Por último, en la quinta fase se corrigen errores y se realizan los retoques finales del proyecto.

1.5. Estructura de la memoria

Este documento se encuentra estructurado en ocho capítulos. El primero es la introducción al trabajo en el que se explica la motivación y la justificación del proyecto, se describen los antecedentes y el estado actual del mismo, se establecen los objetivos y las fases del desarrollo que se han llevado a cabo para finalizar dicho proyecto.

El segundo capítulo introduce las herramientas y tecnologías que se han utilizado para llevar a cabo el trabajo propuesto. Estas tecnologías son desde las más básicas para que la aplicación web funcione que están compuestas por el stack MEVN (Mongodb, Express, Vuejs y Nodejs) como las secundarias para poder llevar a cabo funciones específicas. También se mencionan algunas herramientas de desarrollo como Heroku y GitHub para la integración continua del proyecto.

El tercer y cuarto capítulo pretenden dar una descripción detallada tanto del resultado final del proyecto como de las partes más relevantes del código desarrollado como puede ser la estructura de la base de datos, el diseño de las vistas y el diseño de los componentes.

El quinto capítulo es muy importante para poder concluir que se han cumplido los requisitos más importantes, entre los que se encuentran que la página web sea usable, accesible, minimalista y segura.

Los últimos capítulos comentan los problemas, las conclusiones obtenidas y la visión del futuro de este proyecto.

Capítulo 2

Herramientas y tecnologías

En el capítulo anterior se han introducido los antecedentes y el estado actual del tema, la justificación del proyecto, los objetivos del proyecto y las fases del desarrollo para llevarlo a cabo.

Este capítulo se va a centrar en describir las herramientas y tecnologías que se han utilizado para llevar a cabo este trabajo.

Se ha utilizado Visual Studio Code [9] como entorno de desarrollo de la aplicación. Este es un editor de código fuente potente que se ejecuta en el escritorio de código abierto y gratuito que ha sido desarrollado por Microsoft. Incluye soporte para la depuración, control integrado de Git, resaltado de sintaxis, finalización inteligente de código, fragmentos y refactorización de código. A lo largo de este proyecto se ha utilizado Visual Studio Code vinculado a la cuenta de GitHub utilizando Git como software de control de versiones.

2.1. Bases de la aplicación y arquitectura software: Stack de desarrollo con MEVN

Esta plataforma de emergencias se desarrolló bajo el stack de desarrollo MEVN (Mongodb, Express, Vuejs y Nodejs). Un stack de desarrollo es un conjunto de marcos y herramientas utilizadas para desarrollar un producto de software. Estos marcos y herramientas han sido elegidos específicamente para proporcionar la funcionalidad de su software de la manera más optimizada.

El stack MEVN tiene como objetivo crear aplicaciones web con Javascript. Es de código abierto y ha surgido como una nueva forma evolutiva de crear aplicaciones web potentes y dinámicas. Sus componentes de software se pueden utilizar para diseñar de manera efectiva el desarrollo de interfaz y backend y mejorar la funcionalidad de su sitio web o aplicación. El servidor del backend utiliza Node.js y Express para las API REST mientras que en el lado del frontend es un cliente Vue con Vue Router y axios.

Mongodb [10]

Es el encargado de almacenar los datos de la aplicación. Es una base de datos NoSql orientada a documentos que utiliza JSON como formato de almacenaje de datos, haciendo el desarrollo web más fácil.

Express [11]

Es el framework de Nodejs y Javascript encargado de crear la lógica del lado del servidor, las REST APIs, los archivos estáticos, conectarse a la base de datos y demás lógica de programación. Es utilizado para construir el backend de un sitio utilizando las funciones y estructuras de NodeJS.

Vue.js [12]

Es el framework de Javascript del lado del cliente encargado de crear toda la lógica de la interfaz. Utiliza los datos del servidor Nodejs y se utiliza especialmente en el desarrollo web frontend. Tiene enlace de datos bidireccional que permite un desarrollo frontend perfecto junto con la capacidad MVC y las aplicaciones interactivas del lado del servidor.

Node.js [13]

Es el encargado de poder ejecutar a Javascript fuera del navegador permitiendo escribir nuestro servidor a través de Javascript utilizando sus frameworks como Express.

Otras herramientas

- **Axios [14]** : es una librería JavaScript que puede ejecutarse en el navegador y que permite hacer sencillas las operaciones como cliente HTTP, por lo que se puede configurar y realizar solicitudes a un servidor y recibir respuestas fáciles de procesar.
- **Modelo Vista Controlador (MVC) [15]**: es un estilo de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos.
- **Docker [16]**: es un proyecto de código abierto que automatiza el despliegue de aplicaciones dentro de contenedores de software, proporcionando una capa adicional de abstracción y automatización de virtualización de aplicaciones en múltiples sistemas operativos. Se muestra el docker que se está utilizando y cómo se pone en funcionamiento en la Figura 2.1.

```
noLewen@noLewen-UX550VD:~/TF6$ sudo docker start c8e3
c8e3
noLewen@noLewen-UX550VD:~/TF6$ sudo docker ps -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
c8e3a3500dc2	mongo	"docker-entrypoint.s..."	6 weeks ago	Up 4 seconds	0.0.0.0:27017-27019->27017-27019/tcp	smcmongo

Figura 2.1: Docker utilizado en el proyecto

- **Vue CLI [17]**: es una herramienta para facilitar el rápido desarrollo de las aplicaciones Vue.
- **Robo3t [18]**: es un programa cuya función es administrar las bases de datos MongoDB visualmente. En la Figura 2.2 se muestra un ejemplo de su utilización en este proyecto.

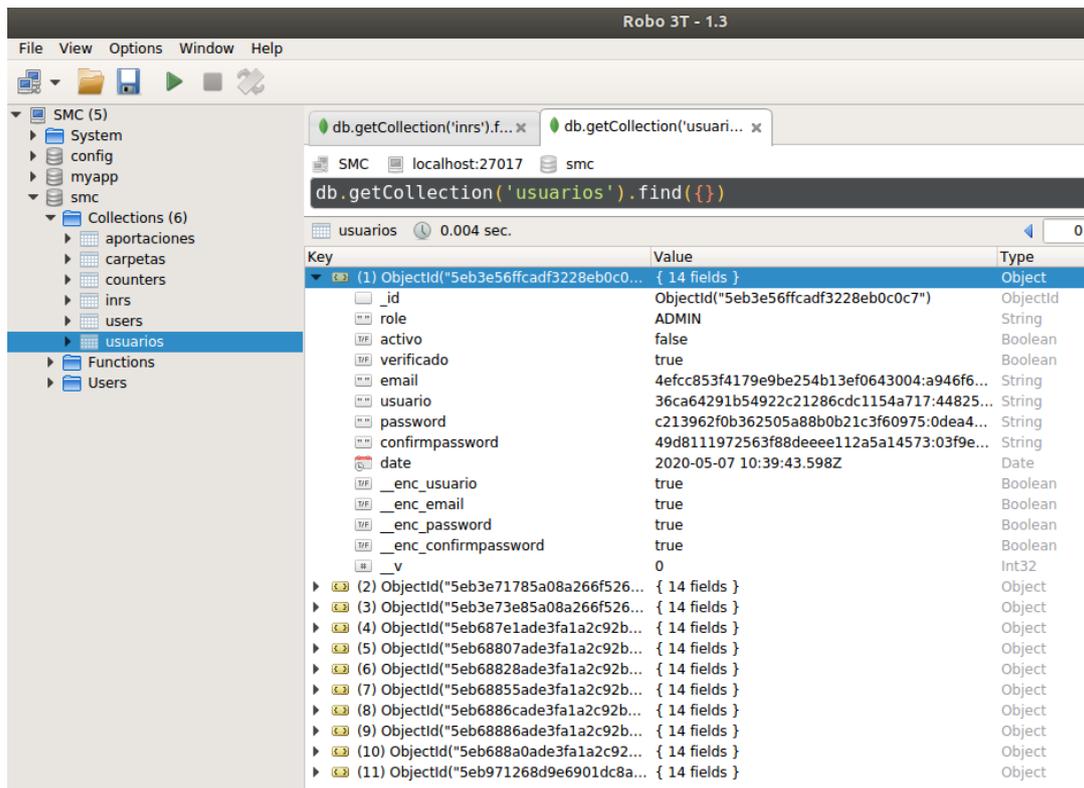


Figura 2.2: Ejemplo robo3t de este proyecto

- **Postman [19]:** es una herramienta que se utiliza, sobre todo, para el testing de API REST. Permite el envío de peticiones HTTP REST sin necesidad de desarrollar un cliente. En la Figura 2.3 se observa un ejemplo de una petición GET al backend a la función showinrs.

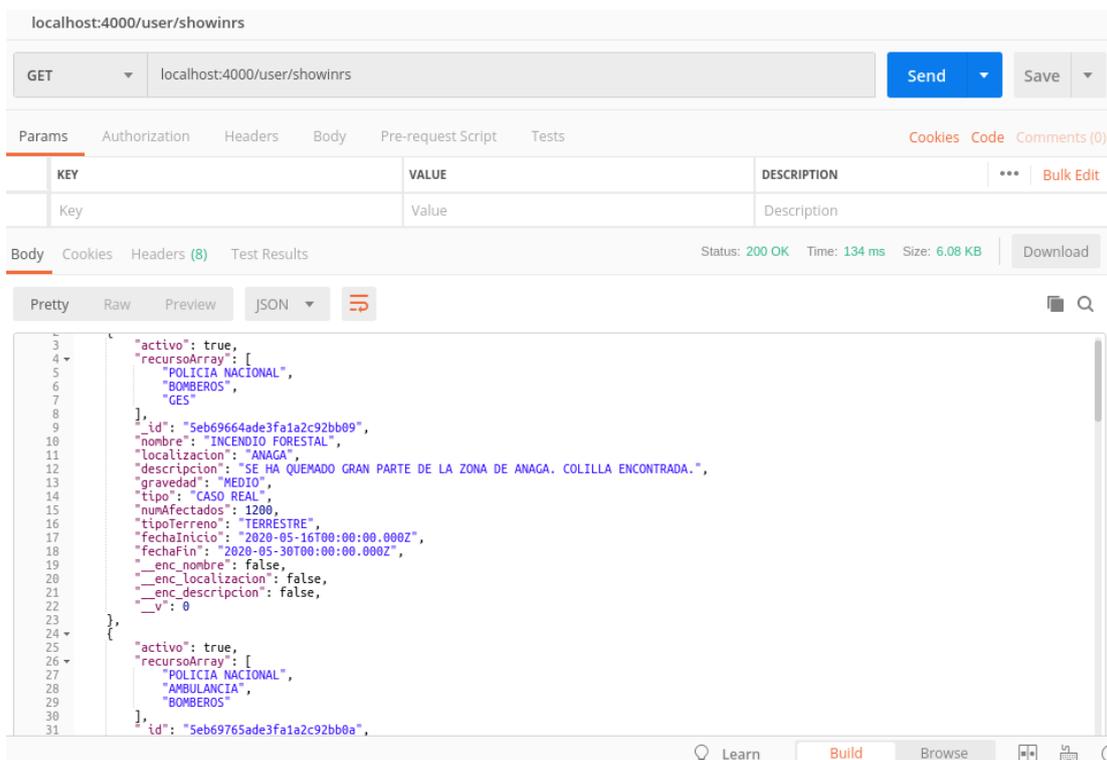


Figura 2.3: Ejemplo de uso de Postman

- **Middleware [20]:** Morgan [21] sirve para emitir las peticiones HTTP request que se solicitan a nuestra aplicación, CORS [22] se utiliza para realizar solicitudes de un servidor externo e impedir el bloqueo por CORS, y por último, JSON y urlencoded son middlewares de Express.
- **Bcrypt [23]:** es una función de hashing de contraseñas basado en el cifrado de Blowfish. Lleva incorporado un valor llamado salt, que es un fragmento aleatorio que se usará para generar el hash asociado a la password, y se guardará junto con ella en la base de datos. Así se evita que dos contraseñas iguales generen el mismo hash y los problemas que ello conlleva.
- **JSON Web Token (JWT) [24]:** es un conjunto de medios de seguridad para peticiones HTTP y así representar demandas para ser transferidos entre dos partes (cliente y servidor). Las partes de un JWT se codifican como un objeto JSON que está firmado digitalmente utilizando JSON Web Signature(JWS).
- **Bootstrap [25]:** es un framework CSS y Javascript diseñado para la creación de interfaces limpias y con un diseño responsive. Además, ofrece un amplio abanico de herramientas y funciones, de manera que los usuarios pueden crear prácticamente cualquier tipo de sitio web haciendo uso de los mismos.
- **Bootstrap Vue (BV) [26]:** sirve para construir proyectos responsivos y ARIA accesibles en la web usando Vue.js y la biblioteca de front-end CSS Bootstrap.
- **Marvel [27]:** es una herramienta utilizada para crear prototipos interactivos de plataformas digitales. Se ha usado para desarrollar los bocetos iniciales de la plataforma de emergencias.

2.2. Herramientas de desarrollo e integración continua

Despliegue: Heroku

Heroku es una aplicación web que permite el despliegue de un proyecto en la nube. El despliegue de la aplicación se ha decidido hacer mediante la herramienta Heroku. Para ello se ha de crear una cuenta y modificar el package.json como se muestra en la Figura 2.4.

```
"scripts": {
  "start": "babel-node app.js --exec",
}...
"dependencies": {
  "@babel/cli": "^7.6.0",
  "@babel/core": "^7.6.0",
  "@babel/node": "^7.6.1",
  "@babel/preset-env": "^7.6.0"
}
```

Figura 2.4: Modificación package.json para despliegue en Heroku

Una vez se ha modificado el fichero se suben los cambios a Heroku y la aplicación se despliega como se muestra en la Figura 2.5.

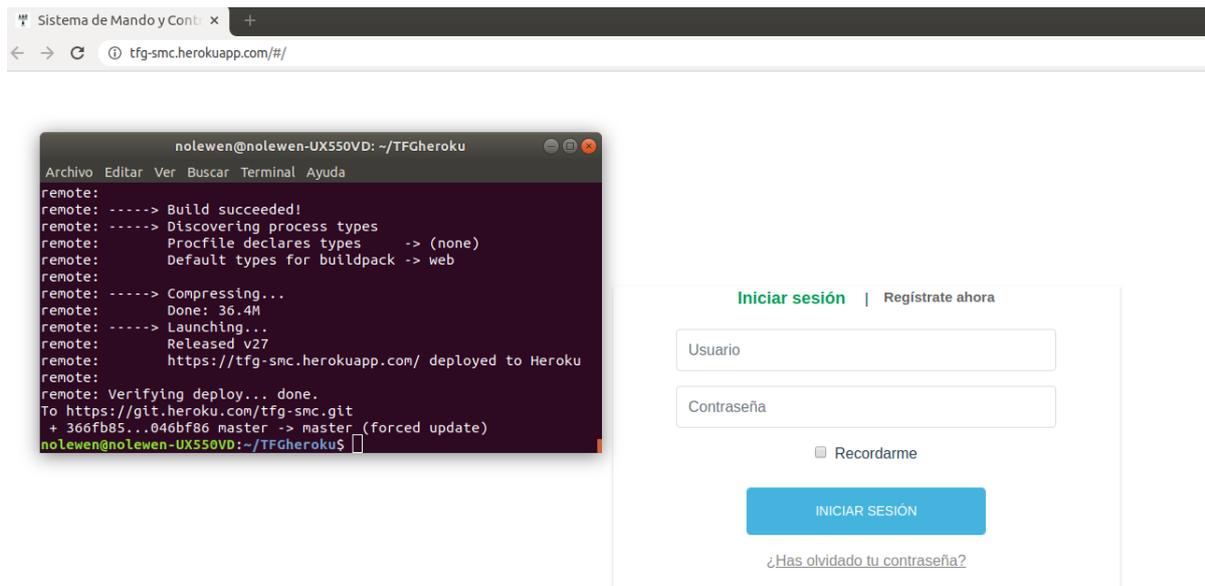


Figura 2.5: Despliegue de la aplicación en Heroku

Control de versiones GitHub

GitHub es una plataforma de desarrollo colaborativo para alojar proyectos utilizando el sistema de control de versiones Git. Para llevar el control de versiones de este proyecto se ha elegido crear un repositorio público en GitHub [28]. En la Figura 2.6 se muestra el repositorio del proyecto.

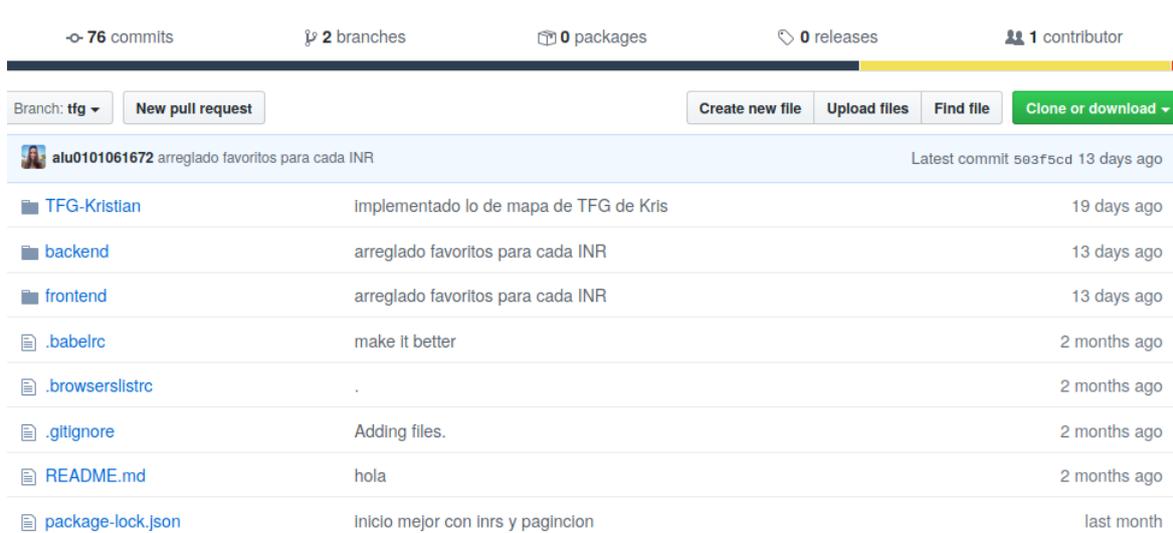


Figura 2.6: Estructura repositorio GitHub

Capítulo 3

Descripción de la aplicación y guía de uso

En este capítulo se expondrá el diseño de la aplicación desde la fase inicial de los bocetos hasta la versión final incluyendo las funcionalidades de cada sección.

3.1. Diseño inicial: Bocetos

Inicialmente en el proyecto se crearon una serie de bocetos que se perfeccionaron hasta que estos se acercaran lo máximo posible a la versión real final de la aplicación. En un principio no se incluyó ni el inicio de sesión ni el registro, por ese motivo no se encuentran en los bocetos. La sección de inicio se asemeja bastante a la idea inicial de la Figura 3.1.

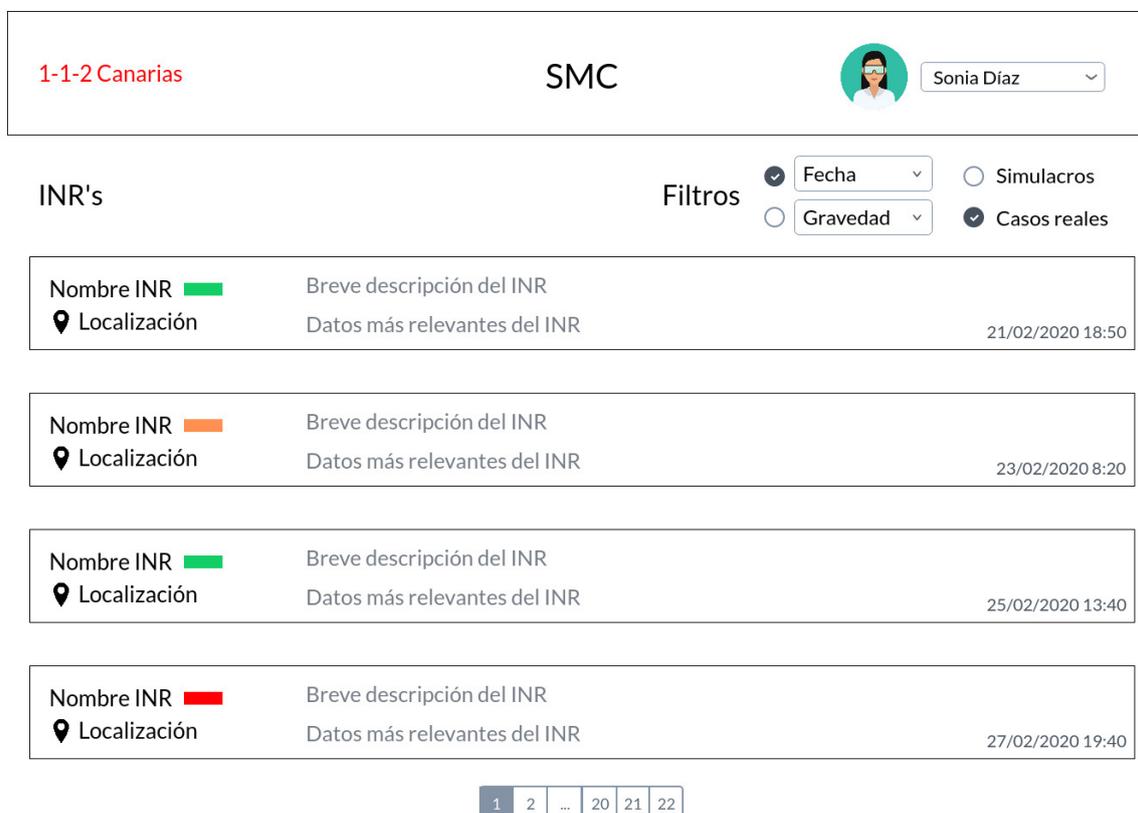


Figura 3.1: Boceto de la sección de inicio

La sección de documentos en los bocetos se diseñó de una manera parecida a la final como se muestra en la Figura 3.2. Sin embargo al final no se implementó la barra de búsqueda ya que existía una sección específica para realizar esta funcionalidad y se añadieron funcionalidades como crear y eliminar carpetas según el rol del usuario. Los datos del INR se muestran de forma muy similar pero la idea de mostrar el mapa a la izquierda se consideró inviable dado que una visualización tan pequeña del mapa no iba a permitir a los usuarios interactuar de una manera sencilla por ello se quitó el mapa y se añadió una sección específica para ver el mapa.

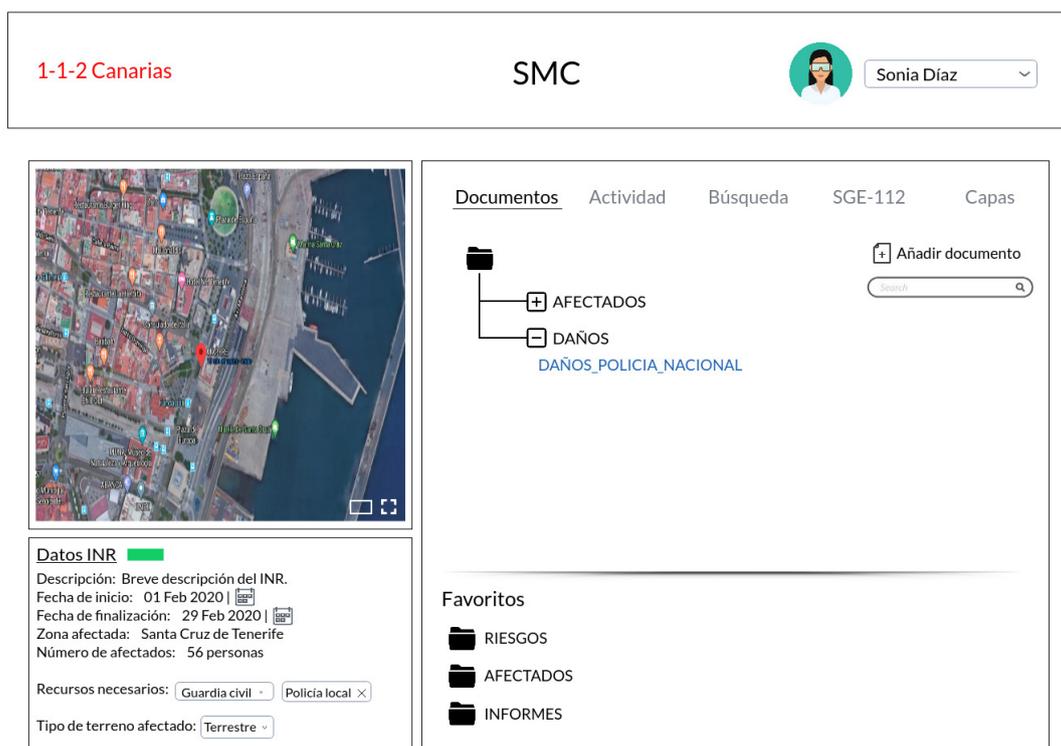


Figura 3.2: Boceto de la sección de documentos

La sección de actividad del boceto se asemeja bastante a la versión final añadiendo la funcionalidad de eliminar aportación según el rol del usuario. El boceto de esta sección se puede observar en la Figura 3.3.

La sección de búsqueda inicialmente se diseñó como se muestra en la Figura 3.4. A medida que se desarrollaba esta sección se notó que carecía de sentido buscar los datos del INR, ya que estos se encontraban en la parte izquierda de la sección, y además se buscaría una aportación única en la que se obtendría un resultado con todos los datos pertenecientes a dicha aportación y no varios resultados de aportaciones con títulos similares.

La sección de SGE-112 es para almacenar las llamadas relativas a dicho INR. En un principio se planteó como una tabla con datos por defecto como se muestra en la Figura 3.5.

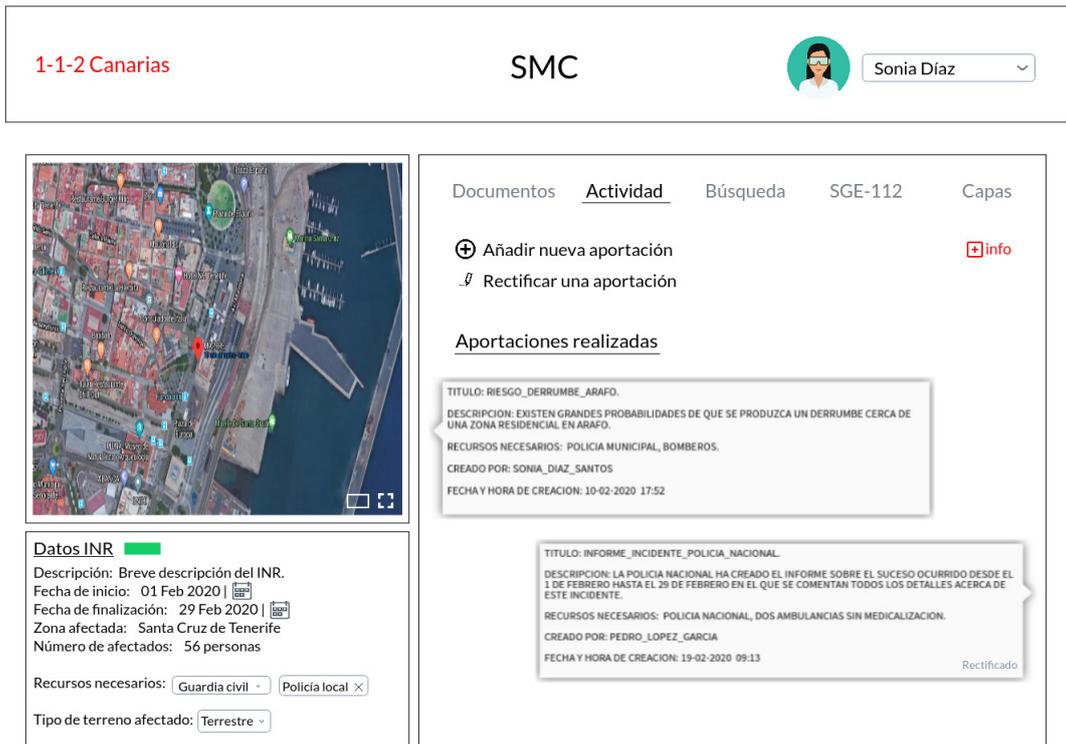


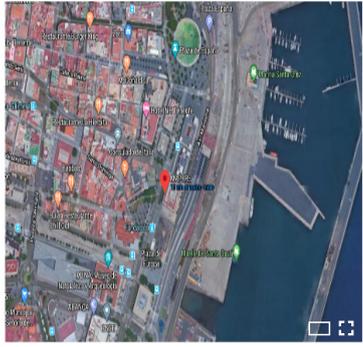
Figura 3.3: Boceto de la sección de actividad



Figura 3.4: Boceto de la sección de búsqueda

1-1-2 Canarias
SMC


Sonia Díaz



Datos INR

Descripción: Breve descripción del INR.

Fecha de inicio: 01 Feb 2020

Fecha de finalización: 29 Feb 2020

Zona afectada: Santa Cruz de Tenerife

Número de afectados: 56 personas

Recursos necesarios: Guardia civil Policia local

Tipo de terreno afectado: Terrestre

Documentos
Actividad
Búsqueda
SGE-112
Capas

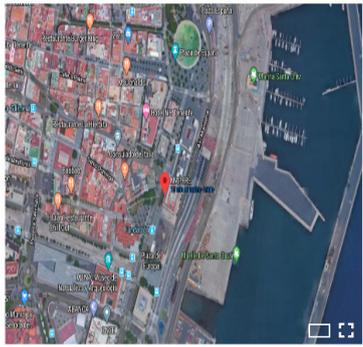
Clasificación	Cola	Nombre	Observaciones
SANIDAD	0	SUSANA	PERSONA CON MAREO CONSCIENTE
SEGURIDAD:GUARDIA CIVIL	1	RODRIGO	PELEA DE DOS CHICOS EN LA CALLE
SEGURIDAD:ACCIDENTE TRAFICO	0	MARTA	TF-1 TUNELES GUIMAR MOTO-COCHE
SEGURIDAD:POLICIA LOCAL	2	JOSUE	PERRO ABANDONADO EN LA ESPERANZA
SEGURIDAD:VIOLENCIA NO FÍSICA	0	PEDRO	MALTRATO VERBAL EN VIA PUBLICA
SEGURIDAD:MULTIDISCIPLINAR	0	SOFIA	BEBE ABANDONADO AL TRINIDAD
SEGURIDAD:POLICIA LOCAL	3	EDUARDO	DISTURBIO EN VIA PUBLICA

Figura 3.5: Boceto de la sección de SGE-112

La sección de Capas se pensó para la configuración y ajustes de la parte del mapa como se muestra en la Figura 3.6. Más tarde se eliminó esta sección y se creó una nueva para el mapa que ya cuenta con su propia configuración.

1-1-2 Canarias
SMC


Sonia Díaz



Datos INR

Descripción: Breve descripción del INR.

Fecha de inicio: 01 Feb 2020

Fecha de finalización: 29 Feb 2020

Zona afectada: Santa Cruz de Tenerife

Número de afectados: 56 personas

Recursos necesarios: Guardia civil Policia local

Tipo de terreno afectado: Terrestre

Documentos
Actividad
Búsqueda
SGE-112
Capas

Mapa

Satélite

Relieve

Capas existentes

Tráfico

Transporte público

Recursos disponibles

Hospitales

Importar capa

Exportar capa

Visualización histórica de las capas

18 Feb 2020

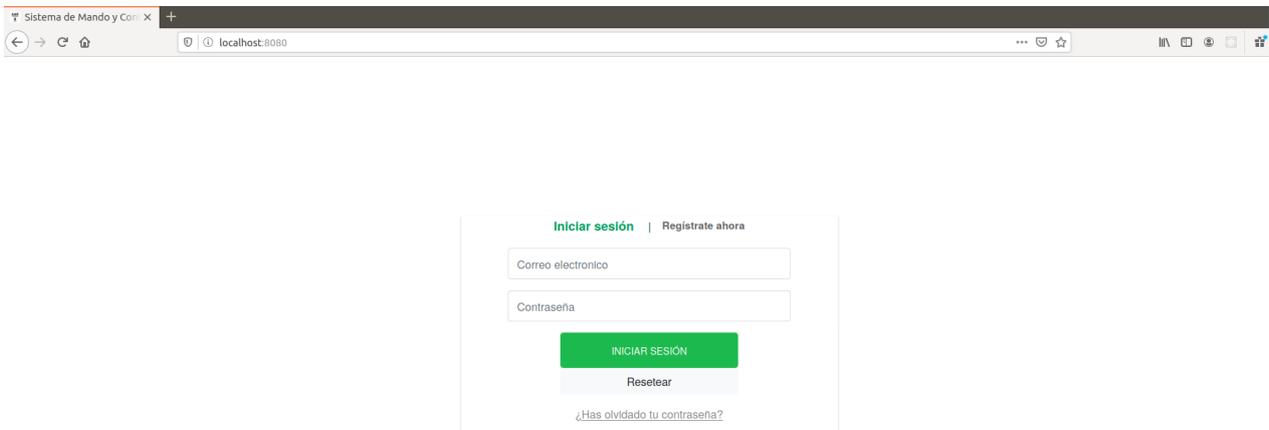
Figura 3.6: Boceto de la sección de capas

3.2. Usuarios y roles

Esta aplicación se encuentra orientada para que los usuarios finales que la utilicen sean los trabajadores de la empresa 112. Por este motivo se ha decidido diferenciar dos roles principales para los usuarios, el rol 'USER' y el rol 'ADMIN'. Esta distinción se realiza no solo por motivos de seguridad sino de funcionalidad. De esta manera un usuario que tenga el rol de administrador 'ADMIN' puede eliminar o añadir usuarios, cambiar el rol a un usuario, validar la cuenta de un usuario, añadir, modificar o eliminar ciertos datos. Un usuario con el rol por defecto de 'USER' se puede registrar pero debe esperar a que un administrador le valide la cuenta para poder iniciar sesión, a parte de no contar con las opciones mencionadas para el administrador. Se han creado once usuarios, un administrador con nombre de usuario admin y diez usuarios con rol 'USER' cuyos nombres de usuario son 'usu' seguido de un número. A continuación se va a comentar la estructura final del proyecto y se podrá observar la diferencia de funcionalidad entre los roles.

3.3. Sección de inicio de sesión y registro

A continuación, en la Figura 3.7 se puede observar el formulario de inicio de sesión para los usuarios. Este se realiza con el correo electrónico y una contraseña. Se pueden resetear los valores de los campos del formulario si el navegador guarda por defecto los datos no tener que quitarlos manualmente.



The image shows a web browser window with the address bar displaying 'localhost:8080'. The page content includes a login form with the following elements:

- Navigation links: 'Iniciar sesión' and 'Regístrate ahora'.
- Input field: 'Correo electrónico'.
- Input field: 'Contraseña'.
- Buttons: 'INICIAR SESIÓN' (green) and 'Resetear' (grey).
- Link: '¿Has olvidado tu contraseña?'.

Figura 3.7: Ventana de inicio de sesión

El formulario de registro de un usuario se muestra en la Figura 3.8 y consta de los campos de nombre de usuario, correo electrónico, la contraseña y la confirmación de la contraseña. Si las contraseñas no coinciden se produce una alerta como se observa en la Figura 3.10. Este formulario también cuenta con la opción de resetear los campos. Una vez un usuario se ha registrado se redirige a la pestaña de inicio de sesión. Cualquier

usuario que se registre tiene que esperar a que un administrador le valide dicho registro para poder iniciar sesión como se muestra en la Figura 3.9.

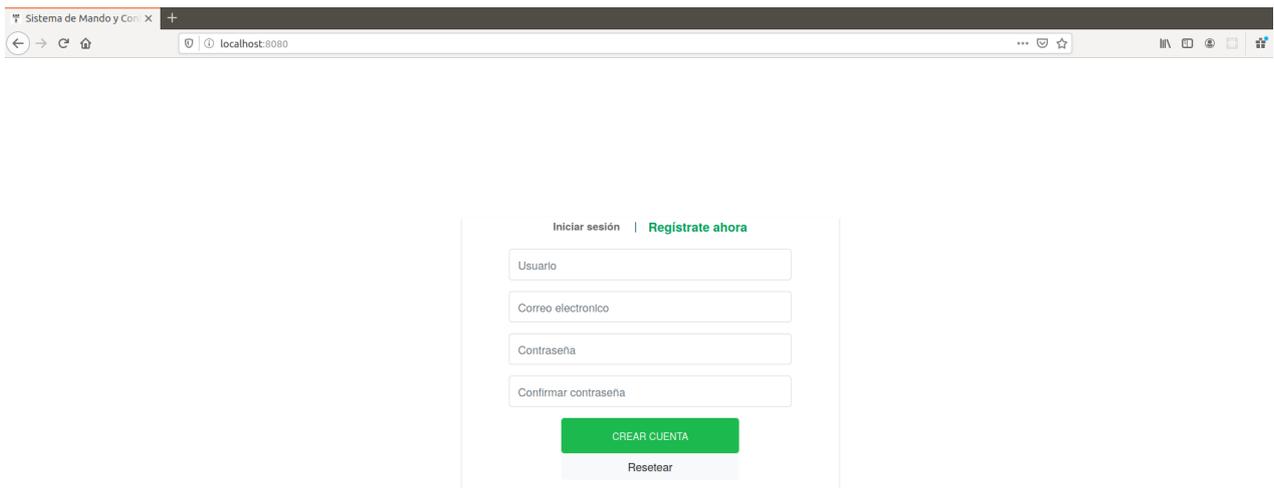


Figura 3.8: Ventana de registro

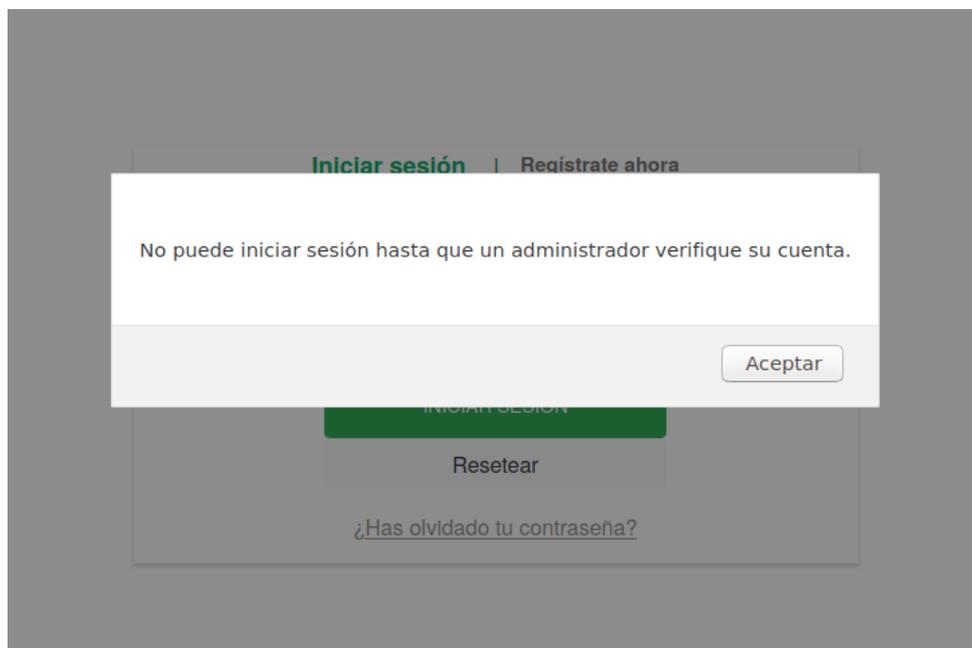


Figura 3.9: Usuario pendiente de validación tras registrarse

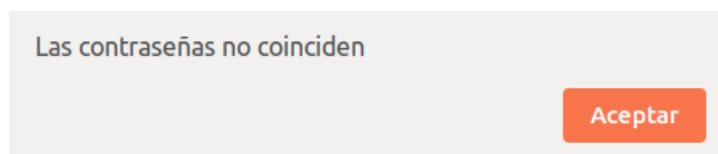


Figura 3.10: Alerta de contraseñas no coincidentes

3.4. Sección de inicio

Una vez un usuario ha iniciado sesión satisfactoriamente se redirige a la página de inicio del SMC(Sistema de Mando y Control) del 112. En esta sección se visualizan todos los INRs(Incidentes no rutinarios) creados por el administrador como se observa en la Figura 3.11. Si se trata de un usuario con el rol 'USER' esta página se verá como se muestra en la Figura 3.12. En estas figuras se puede observar cómo el administrador puede crear, modificar y eliminar los INRs mientras que un usuario normal no puede realizar estas tareas. Cada INR es un botón que al presionarlo se accede a la siguiente sección cuyo propósito es dar la información exclusiva a ese INR concreto que se ha seleccionado.

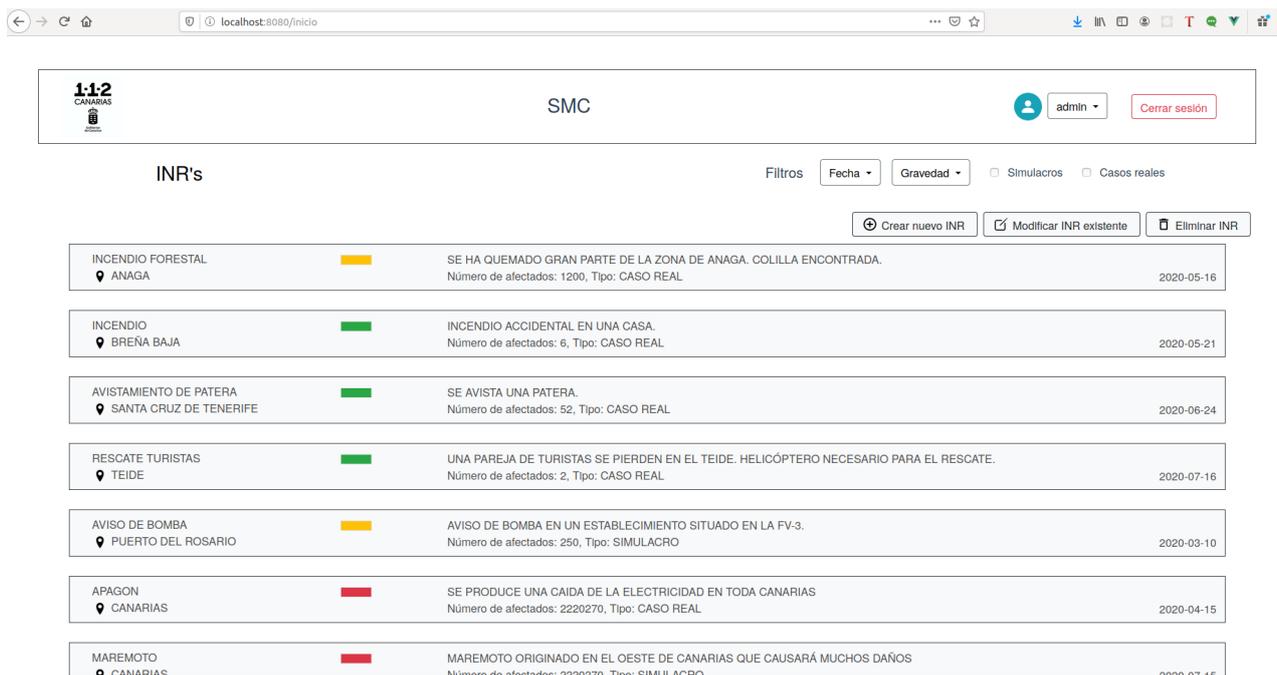


Figura 3.11: Sección de inicio como administrador

Cuando el administrador crea un nuevo INR se muestra el formulario de la Figura 3.13. Una vez creado se observa en la Figura 3.14 cómo se ha creado correctamente el nuevo INR. Si se modifica un INR existente se rellena el formulario de la Figura 3.15 que cuenta en cada campo posible con una lista de los elementos ya creados como se observa con el nombre de los INR existentes para escoger uno de ellos. Se muestra como la modificación se ha realizado en la Figura 3.16. En la Figura 3.17 se muestra el formulario para eliminar un INR.

112 CANARIAS

SMC

usu1 Cerrar sesión

INR's Filtros Fecha Gravedad Simulacros Casos reales

INCENDIO FORESTAL ANAGA	SE HA QUEMADO GRAN PARTE DE LA ZONA DE ANAGA. COLILLA ENCONTRADA. Número de afectados: 1200, Tipo: CASO REAL	2020-05-16
INCENDIO BREÑA BAJA	INCENDIO ACCIDENTAL EN UNA CASA. Número de afectados: 6, Tipo: CASO REAL	2020-05-21
AVISTAMIENTO DE PATERA SANTA CRUZ DE TENERIFE	SE AVISTA UNA PATERA. Número de afectados: 52, Tipo: CASO REAL	2020-06-24
RESCATE TURISTAS TEIDE	UNA PAREJA DE TURISTAS SE PIERDEN EN EL TEIDE. HELICÓPTERO NECESARIO PARA EL RESCATE. Número de afectados: 2, Tipo: CASO REAL	2020-07-16
AVISO DE BOMBA PUERTO DEL ROSARIO	AVISO DE BOMBA EN UN ESTABLECIMIENTO SITUADO EN LA FV-3. Número de afectados: 250, Tipo: SIMULACRO	2020-03-10
APAGON CANARIAS	SE PRODUCE UNA CAIDA DE LA ELECTRICIDAD EN TODA CANARIAS Número de afectados: 2220270, Tipo: CASO REAL	2020-04-15
MAREMOTO CANARIAS	MAREMOTO ORIGINADO EN EL OESTE DE CANARIAS QUE CAUSARÁ MUCHOS DAÑOS Número de afectados: 2220270, Tipo: SIMULACRO	2020-07-15

Figura 3.12: Sección de inicio como usuario normal

Crear INR

CORONAVIRUS
Formato: NOMBRE INR

MUNDIAL
Formato: SANTA CRUZ

VIRUS CONTAGIOSO PELIGROSO PARA LA

GRAVE

CASO REAL

7625000000

POLICIA LOCAL
POLICIA NACIONAL
GUARDIA CIVIL
AMBULANCIA
Pulse Ctrl y seleccione para añadir múltiples opciones -->

TERRESTRE

30 de enero de 2020

30 de junio de 2020

Resetear

Cancelar

Guardar

Figura 3.13: Formulario de crear nuevo INR

1:12 CANARIAS SMC admin Cerrar sesión

INR's Filtros Fecha Gravedad Simulacros Casos reales

Crear nuevo INR Modificar INR existente Eliminar INR

INUNDACION CENTRO COMERCIAL SAN SEBASTIAN	EN LA GOMERA SE HA INUNDADO UN CENTRO COMERCIAL. Número de afectados: 501, Tipo: CASO REAL	2020-02-04
CORONAVIRUS MUNDIAL	VIRUS CONTAGIOSO PELIGROSO PARA LA SALUD Número de afectados: 7625000000, Tipo: CASO REAL	2020-01-30

First Previous 1 2 Next Last

Figura 3.14: Resultado de crear el INR

Modificar INR

NOMBRE DEL INR A MODIFICAR

- MAREMOTO
- TERREMOTO
- ERUPCION VOLCANICA
- DERRAME DE MATERIALES PELIGROSOS
- INUNDACION CENTRO COMERCIAL
- CORONAVIRUS
- LOCALIZACIÓN

Formato: SANTA CRUZ

DESCRIPCIÓN

GRAVEDAD

TIPO

Opciones: SIMULACRO, CASO REAL

NÚMERO DE AFECTADOS

- POLICIA LOCAL
- POLICIA NACIONAL
- GUARDIA CIVIL
- AMBULANCIA

Pulse Ctrl y seleccione para añadir múltiples opciones

TIPO DE TERRENO

No date selected

No date selected

Resetear

Figura 3.15: Formulario de modificar un INR existente

1:12 CANARIAS SMC admin Cerrar sesión

INR's Filtros Fecha Gravedad Simulacros Casos reales

Crear nuevo INR Modificar INR existente Eliminar INR

INUNDACION CENTRO COMERCIAL SAN SEBASTIAN	EN LA GOMERA SE HA INUNDADO UN CENTRO COMERCIAL. Número de afectados: 501, Tipo: CASO REAL	2020-02-04
COVID-19 MUNDIAL	VIRUS CONTAGIOSO PELIGROSO PARA LA SALUD Número de afectados: 7625000000, Tipo: CASO REAL	2020-01-30

First Previous 1 2 Next Last

Figura 3.16: Resultado de modificar el INR existente

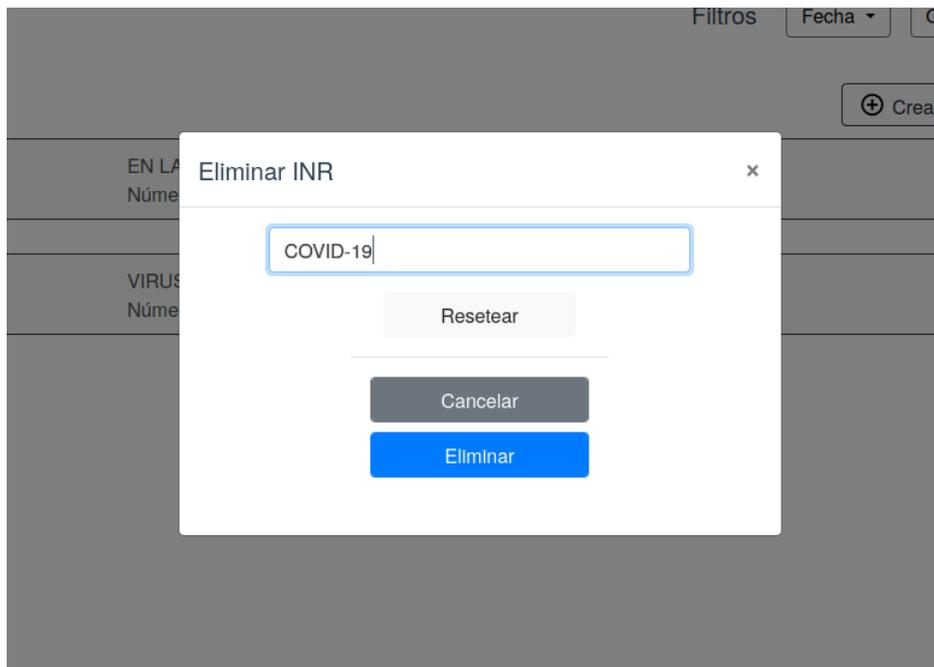


Figura 3.17: Formulario para eliminar el INR

En esta sección también se encuentran disponibles una serie de filtros para una búsqueda del INR más rápida y eficiente o incluso para una categorización de los INR. Se puede filtrar por fecha los INR que empiecen más temprano o más tarde como se muestra en la Figura 3.18, en la que los INR se encuentran ordenados por los que tienen una fecha de inicio más reciente. Por gravedad como se muestra en la Figura 3.19, en la que se encuentran ordenados de mayor gravedad a menor. Por tipo de INR, si son simulacros o casos reales como se muestra en la Figura 3.20, en la que se encuentran ordenados por simulacros.

SMC

admin
 Cerrar sesión

INR's

Filtros Fecha ▾ Gravedad ▾

Simulacros Casos reales

Comenzan antes
Comenzan más tarde
Sin filtro
Modificar INR existente
Eliminar INR

CORONAVIRUS MUNDIAL	<div style="width: 15px; height: 10px; background-color: red; margin: 0 auto;"></div>	VIRUS CONTAGIOSO PELIGROSO PARA LA SALUD Número de afectados: 7625000000, Tipo: CASO REAL	2020-01-30
INUNDACION CENTRO COMERCIAL SAN SEBASTIAN	<div style="width: 15px; height: 10px; background-color: orange; margin: 0 auto;"></div>	EN LA GOMERA SE HA INUNDADO UN CENTRO COMERCIAL. Número de afectados: 501, Tipo: CASO REAL	2020-02-04
AVISO DE BOMBA PUERTO DEL ROSARIO	<div style="width: 15px; height: 10px; background-color: orange; margin: 0 auto;"></div>	AVISO DE BOMBA EN UN ESTABLECIMIENTO SITUADO EN LA FV-3. Número de afectados: 250, Tipo: SIMULACRO	2020-03-10
APAGON CANARIAS	<div style="width: 15px; height: 10px; background-color: red; margin: 0 auto;"></div>	SE PRODUCE UNA CAIDA DE LA ELECTRICIDAD EN TODA CANARIAS Número de afectados: 2220270, Tipo: CASO REAL	2020-04-15
INCENDIO FORESTAL ANAGA	<div style="width: 15px; height: 10px; background-color: orange; margin: 0 auto;"></div>	SE HA QUEMADO GRAN PARTE DE LA ZONA DE ANAGA. COLILLA ENCONTRADA. Número de afectados: 1200, Tipo: CASO REAL	2020-05-16
INCENDIO BREÑA BAJA	<div style="width: 15px; height: 10px; background-color: green; margin: 0 auto;"></div>	INCENDIO ACCIDENTAL EN UNA CASA. Número de afectados: 6, Tipo: CASO REAL	2020-05-21

Figura 3.18: Filtros por fechas

1:12 CANARIAS SMC admin Cerrar sesión

INR's Filtros Fecha Gravedad Simulacros Casos reales

Más grave
Menos grave
Sin filtro

CORONAVIRUS MUNDIAL	■	VIRUS CONTAGIOSO PELIGROSO PARA LA SALUD Número de afectados: 7625000000, Tipo: CASO REAL	2020-01-30
APAGON CANARIAS	■	SE PRODUCE UNA CAIDA DE LA ELECTRICIDAD EN TODA CANARIAS Número de afectados: 2220270, Tipo: CASO REAL	2020-04-15
MAREMOTO CANARIAS	■	MAREMOTO ORIGINADO EN EL OESTE DE CANARIAS QUE CAUSARÁ MUCHOS DAÑOS Número de afectados: 2220270, Tipo: SIMULACRO	2020-07-15
ERUPCION VOLCANICA TEIDE	■	EL TEIDE ENTRA EN ERUPCIÓN. Número de afectados: 950000, Tipo: SIMULACRO	2021-05-11
INUNDACION CENTRO COMERCIAL SAN SEBASTIAN	■	EN LA GOMERA SE HA INUNDADO UN CENTRO COMERCIAL. Número de afectados: 501, Tipo: CASO REAL	2020-02-04
AVISO DE BOMBA PUERTO DEL ROSARIO	■	AVISO DE BOMBA EN UN ESTABLECIMIENTO SITUADO EN LA FV-3. Número de afectados: 250, Tipo: SIMULACRO	2020-03-10

Figura 3.19: Filtros por gravedad

1:12 CANARIAS SMC admin Cerrar sesión

INR's Filtros Fecha Gravedad Simulacros Casos reales

Crear nuevo INR Modificar INR existente Eliminar INR

AVISO DE BOMBA PUERTO DEL ROSARIO	■	AVISO DE BOMBA EN UN ESTABLECIMIENTO SITUADO EN LA FV-3. Número de afectados: 250, Tipo: SIMULACRO	2020-03-10
MAREMOTO CANARIAS	■	MAREMOTO ORIGINADO EN EL OESTE DE CANARIAS QUE CAUSARÁ MUCHOS DAÑOS Número de afectados: 2220270, Tipo: SIMULACRO	2020-07-15
ERUPCION VOLCANICA TEIDE	■	EL TEIDE ENTRA EN ERUPCIÓN. Número de afectados: 950000, Tipo: SIMULACRO	2021-05-11

First Previous 1 Next Last

Figura 3.20: Filtros por simulacros

Por último, en la cabecera de la página se encuentra el cierre de sesión que redirige a la ventana de inicio de sesión y, el nombre del usuario que ha iniciado sesión. Si se despliega el 'dropdown' dicho usuario puede ver al resto de usuarios activos como se muestra en la Figura 3.21.

admin Cerrar sesión

Usuarios activos

- usu1
- usu9

Figura 3.21: Usuarios activos

3.5. Sección de documentos

Se observa que al entrar en un INR concreto aparece una barra de navegación que contiene cada una de las secciones que se van a ir describiendo a continuación. La sección de documentos se encuentra implementada para poder guardar toda la información relacionada con la documentación del INR concreto que se haya seleccionado. En la siguiente Figura 3.22 se muestra el INR que se ha seleccionado. Como se puede observar en la parte izquierda se muestran todos los datos del INR que se ha seleccionado. En la parte derecha se ven las opciones de añadir carpeta, eliminar carpeta y añadir fichero con el rol de administrador. Se pueden ver las carpetas creadas para cada INR, con los ficheros que contienen si se presiona el botón de la carpeta. Las carpetas cuentan con un botón de un corazón para añadir dicha carpeta a la sección de Favoritos que se encuentra más abajo. El botón del corazón se encuentra lleno si se encuentra en la sección de favoritos y vacío si no. La sección de favoritos tiene la función de acceso directo para encontrar de manera más eficiente, rápida y efectiva la carpeta o documento que se desee si es una carpeta que suele visitar con mayor frecuencia. Para un usuario normal esta sección se muestra como en la Figura 3.23 a falta de las opciones de añadir y eliminar carpetas.

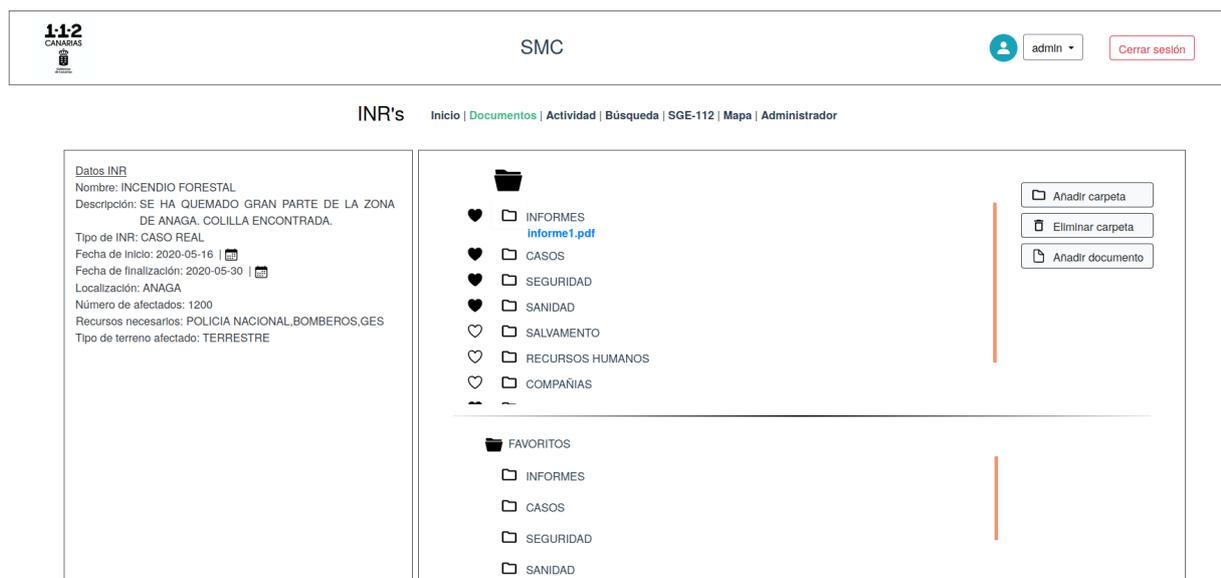


Figura 3.22: Sección de documentos como administrador

En la Figura 3.24 se puede observar como accediendo a otro INR diferente las carpetas y los favoritos son diferentes ya que cada INR tiene sus propios datos.

Cabe destacar el formulario de añadir un documento como se puede observar en la Figura 3.25 en el que se añaden los documentos a las carpetas existentes. En el campo del nombre de la carpeta se despliegan las carpetas que se encuentran creadas.

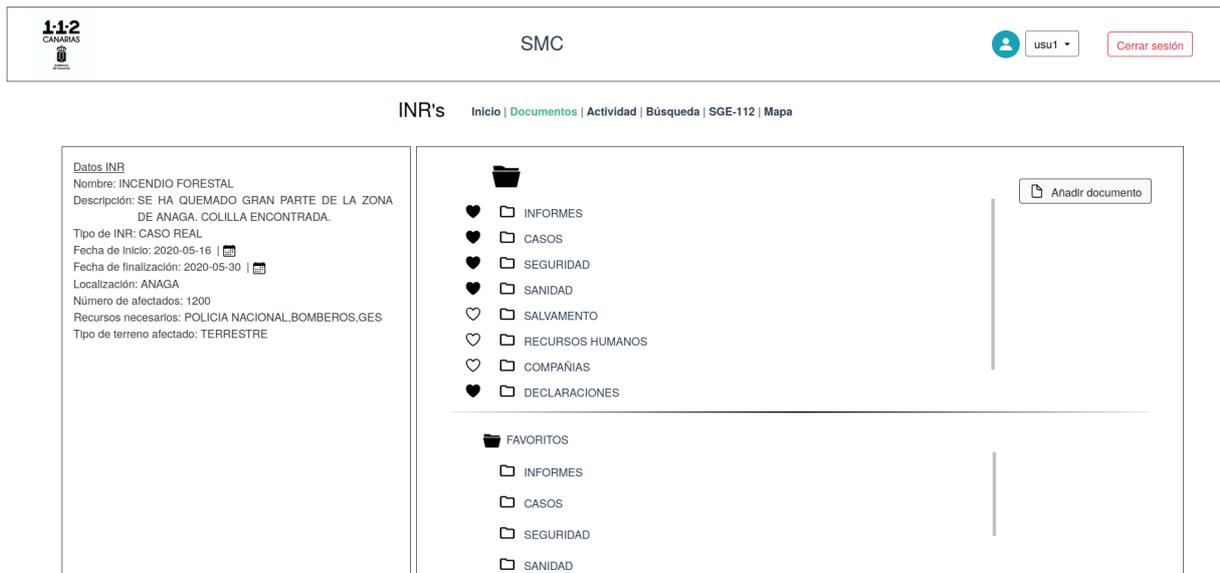


Figura 3.23: Sección de documentos como usuario normal

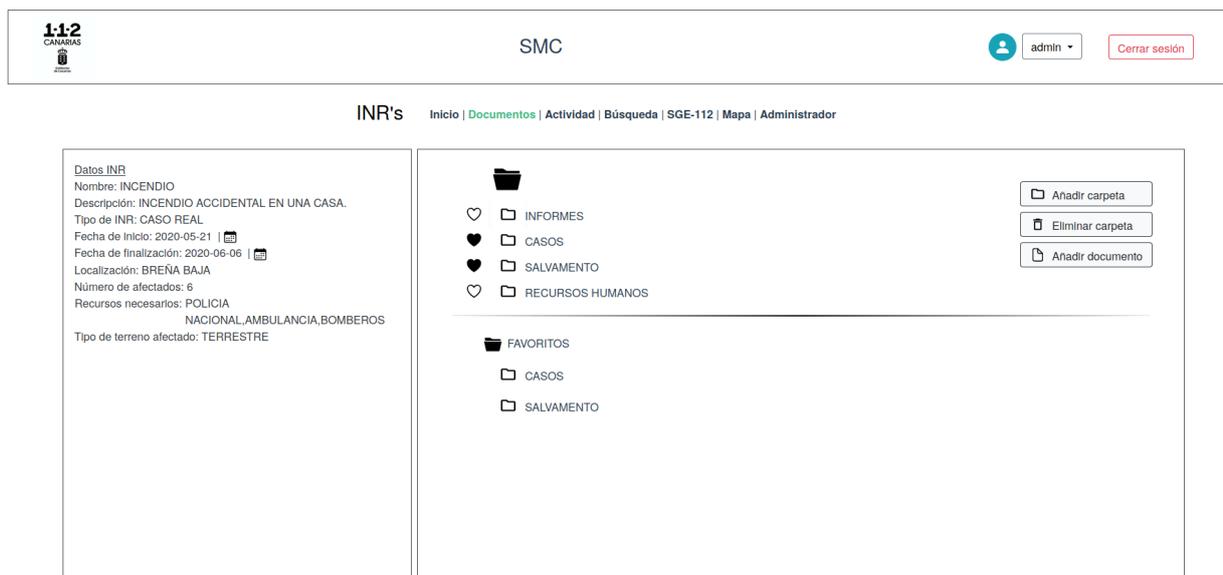


Figura 3.24: Ejemplo de sección documentos de otro INR

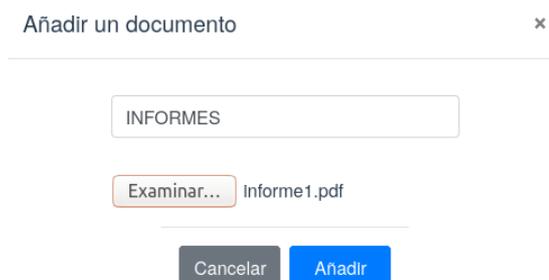


Figura 3.25: Formulario de añadir documento

3.6. Sección de actividad

Esta sección tiene como objetivo que los trabajadores del 112 realicen aportaciones acerca del INR para tener un registro de los datos como si fuera un chat informativo para los trabajadores, así se mantienen actualizados. En la Figura 3.26 se muestran las aportaciones realizadas para un INR concreto, para otro INR las aportaciones serán diferentes. Se pueden observar las funciones del administrador que puede añadir, rectificar y eliminar una aportación. En la Figura 3.27 se puede ver la interfaz de esta sección para un usuario normal de otro INR diferente. Se observa que un usuario normal no puede eliminar aportaciones. Por último, en la Figura 3.28 se muestra la ventana que se despliega si se presiona el botón más info situado a la derecha, el cual pretende ser una guía de estilo a seguir por los trabajadores para que todas las aportaciones sigan un mismo patrón independientemente del usuario que las escriba. El formulario más relevante de esta sección es el de rectificar una aportación que se muestra en la Figura 3.29.

112 CANARIAS

SMC

admin Cerrar sesión

INR's Inicio | Documentos | Actividad | Búsqueda | SGE-112 | Mapa | Administrador

Datos INR
Nombre: INCENDIO FORESTAL
Descripción: SE HA QUEMADO GRAN PARTE DE LA ZONA DE ANAGA, COLILLA ENCONTRADA.
Tipo de INR: CASO REAL
Fecha de Inicio: 2020-05-16 | 📅
Fecha de finalización: 2020-05-30 | 📅
Localización: ANAGA
Número de afectados: 1200
Recursos necesarios: POLICIA NACIONAL,BOMBEROS,GES
Tipo de terreno afectado: TERRESTRE

➕ Añadir nueva aportación

✎ Rectificar una aportación

🗑 Eliminar una aportación

Info

Aportaciones realizadas

TITULO: INFORME BOMBEROS
DESCRIPCION: LOS BOMBEROS ENCONTRARON UNA COLILLA PRUEBA QUE CONFIRMA QUE EL INCENDIO FUE INTENCIONADO. POSTERIORMENTE EL SOSPECHOSO CONFESÓ SU CRIMEN.
RECURSOS: BOMBEROSCREADO POR: ADMIN 2020-05-10

TITULO: INFORME POLICIA
DESCRIPCION: SE HA ENCONTRADO UN SOSPECHOSO CUYA HUELLA SE ENCONTRABA EN LA COLILLA
RECURSOS: POLICIA LOCAL,POLICIA NACIONAL,GUARDIA CIVILCREADO POR: ADMIN 2020-05-10

TITULO: UNIDAD DE SALVAMENTO
DESCRIPCION: SE ENCONTRÓ UN VARÓN CERCA DE LA ZONA INCENDIADA AL QUE HUBO QUE RESCATAR
RECURSOS: AMBULANCIA,BOMBEROS,GESCREADO POR: ADMIN 2020-05-10

Figura 3.26: Sección de actividad para el administrador

1.1.2
CAMBIOS

SMC

USU1 Cerrar sesión

INR's Inicio | Documentos | **Actividad** | Búsqueda | SGE-112 | Mapa

Datos INR

Nombre: INCENDIO
 Descripción: INCENDIO ACCIDENTAL EN UNA CASA.
 Tipo de INR: CASO REAL
 Fecha de inicio: 2020-05-21 | 📅
 Fecha de finalización: 2020-06-06 | 📅
 Localización: BREÑA BAJA
 Número de afectados: 6
 Recursos necesarios: POLICIA
 NACIONAL.AMBULANCIA.BOMBEROS
 Tipo de terreno afectado: TERRESTRE

➕ Añadir nueva aportación
Info

✍ Rectificar una aportación

Aportaciones realizadas

TITULO: INFORME BOMBEROS
 DESCRIPCION: SE DESCONOCE LA CAUSA DEL INCENDIO PERO SE SOSPECHA QUE HUBO UN ESCAPE DE GAS QUE PUEDO PROVOCARLO
 RECURSOS: BOMBEROSCREADO POR: ADMIN 2020-05-10

Figura 3.27: Sección de actividad para un usuario normal

Información del estilo de sintaxis a utilizar ✕

Normas a seguir

- Escribir en mayúsculas siempre
 - Escribir sin tildes
- Separación de palabras por un espacio

Salir

Figura 3.28: Información para una guía de estilo común entre los trabajadores

Rectificar una aportación ✕

Formato: NOMBRE APORTACION

POLICIA LOCAL
 POLICIA NACIONAL
 GUARDIA CIVIL
 AMBULANCIA

Pulse Ctrl y seleccione para añadir múltiples opciones

Resetear

Cancelar
Rectificar

Figura 3.29: Formulario para rectificar una aportación

3.7. Sección de búsqueda

La sección de búsqueda tiene como objetivo realizar la búsqueda de aportaciones por si se necesita consultar algún dato en particular sobre alguna aportación ya que en la sección aportaciones puede haber muchas aportaciones y puede llevar más tiempo consultar los datos. En la Figura 3.30 se observa como se busca una aportación en concreto para el INR seleccionado.

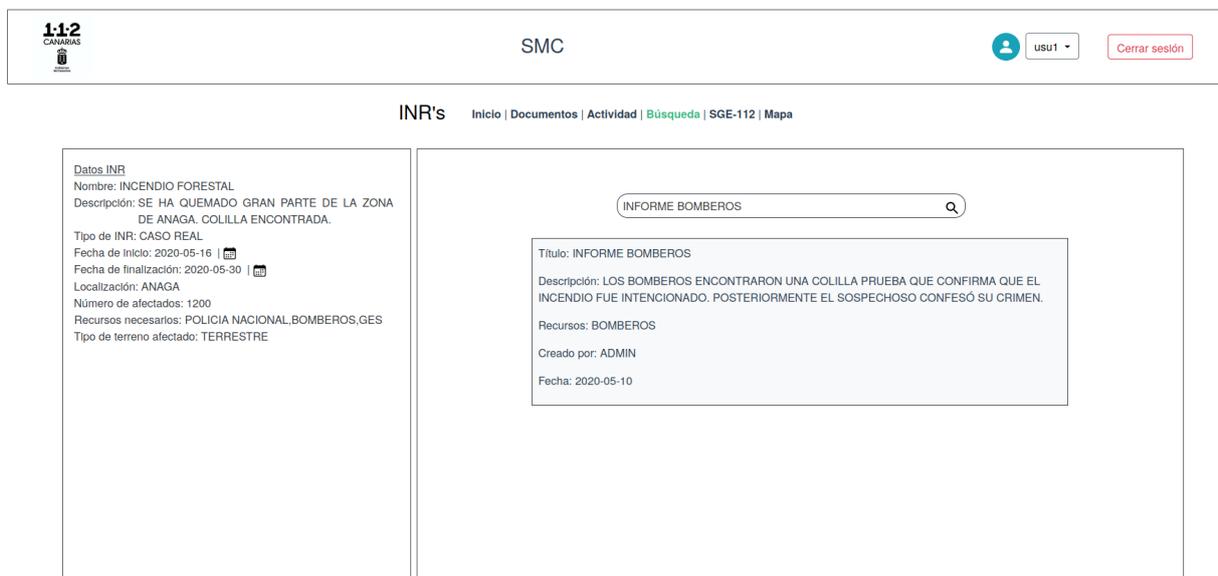


Figura 3.30: Sección búsqueda para cualquier usuario

3.8. Sección de SGE-112

Esta sección está pensada para ser vinculada con los datos reales de las llamadas que tengan que ver con este INR en concreto, teniendo la misma información que en la sala de llamadas de la empresa 112. La sección SGE-112 tiene actualmente una tabla estándar fija para cualquier INR que se consulte con el esquema de las llamadas reales y se encuentra implementada como se ve en la Figura 3.31.

<p>Datos INR</p> <p>Nombre: INCENDIO FORESTAL</p> <p>Descripción: SE HA QUEMADO GRAN PARTE DE LA ZONA DE ANAGA, COLILLA ENCONTRADA.</p> <p>Tipo de INR: CASO REAL</p> <p>Fecha de inicio: 2020-05-16</p> <p>Fecha de finalización: 2020-05-30</p> <p>Localización: ANAGA</p> <p>Número de afectados: 1200</p> <p>Recursos necesarios: POLICIA NACIONAL,BOMBEROS,GES</p> <p>Tipo de terreno afectado: TERRESTRE</p>	<table border="1"> <thead> <tr> <th>Clasificación</th> <th>Cola</th> <th>Nombre</th> <th>Observaciones</th> </tr> </thead> <tbody> <tr> <td>SANIDAD</td> <td>0</td> <td>SUSANA</td> <td>PERSONA CONSCIENTE CON MAREO</td> </tr> <tr> <td>SEGURIDAD: GUARDIA CIVIL</td> <td>1</td> <td>RODRIGO</td> <td>PELEA DE DOS PERSONAS EN LA CALLE</td> </tr> <tr> <td>SEGURIDAD: ACCIDENTE DE TRAFICO</td> <td>0</td> <td>MARTA</td> <td>TF-1 TUNELES GUIMAR MOTO-COCHE</td> </tr> <tr> <td>SEGURIDAD: POLICIA LOCAL</td> <td>2</td> <td>JOSUE</td> <td>PERRO ABANDONADO EN LA ESPERANZA</td> </tr> <tr> <td>SEGURIDAD: VIOLENCIA NO FISICA</td> <td>0</td> <td>PEDRO</td> <td>MALTRATO VERBAL EN VIA PUBLICA</td> </tr> <tr> <td>SEGURIDAD: MULTIDISCIPLINAR</td> <td>0</td> <td>SOFIA</td> <td>BEBE ABANDONADO AV. TRINIDAD</td> </tr> <tr> <td>SEGURIDAD: POLICIA LOCAL</td> <td>3</td> <td>EDUARDO</td> <td>DISTURBIO EN VIA PUBLICA</td> </tr> </tbody> </table>	Clasificación	Cola	Nombre	Observaciones	SANIDAD	0	SUSANA	PERSONA CONSCIENTE CON MAREO	SEGURIDAD: GUARDIA CIVIL	1	RODRIGO	PELEA DE DOS PERSONAS EN LA CALLE	SEGURIDAD: ACCIDENTE DE TRAFICO	0	MARTA	TF-1 TUNELES GUIMAR MOTO-COCHE	SEGURIDAD: POLICIA LOCAL	2	JOSUE	PERRO ABANDONADO EN LA ESPERANZA	SEGURIDAD: VIOLENCIA NO FISICA	0	PEDRO	MALTRATO VERBAL EN VIA PUBLICA	SEGURIDAD: MULTIDISCIPLINAR	0	SOFIA	BEBE ABANDONADO AV. TRINIDAD	SEGURIDAD: POLICIA LOCAL	3	EDUARDO	DISTURBIO EN VIA PUBLICA
Clasificación	Cola	Nombre	Observaciones																														
SANIDAD	0	SUSANA	PERSONA CONSCIENTE CON MAREO																														
SEGURIDAD: GUARDIA CIVIL	1	RODRIGO	PELEA DE DOS PERSONAS EN LA CALLE																														
SEGURIDAD: ACCIDENTE DE TRAFICO	0	MARTA	TF-1 TUNELES GUIMAR MOTO-COCHE																														
SEGURIDAD: POLICIA LOCAL	2	JOSUE	PERRO ABANDONADO EN LA ESPERANZA																														
SEGURIDAD: VIOLENCIA NO FISICA	0	PEDRO	MALTRATO VERBAL EN VIA PUBLICA																														
SEGURIDAD: MULTIDISCIPLINAR	0	SOFIA	BEBE ABANDONADO AV. TRINIDAD																														
SEGURIDAD: POLICIA LOCAL	3	EDUARDO	DISTURBIO EN VIA PUBLICA																														

Figura 3.31: Sección SGE-112 para las llamadas

3.9. Sección de mapa

Esta sección tiene como objetivo la visualización en un mapa del INR. La página web a la que se redirige es el TFG de un compañero cuyo trabajo es complementario a este proyecto por lo que se optó por unirlos. En esta sección hay un botón que al presionarlo se redirige al trabajo de fin de grado de mi compañero como se muestra en la Figura 3.32.

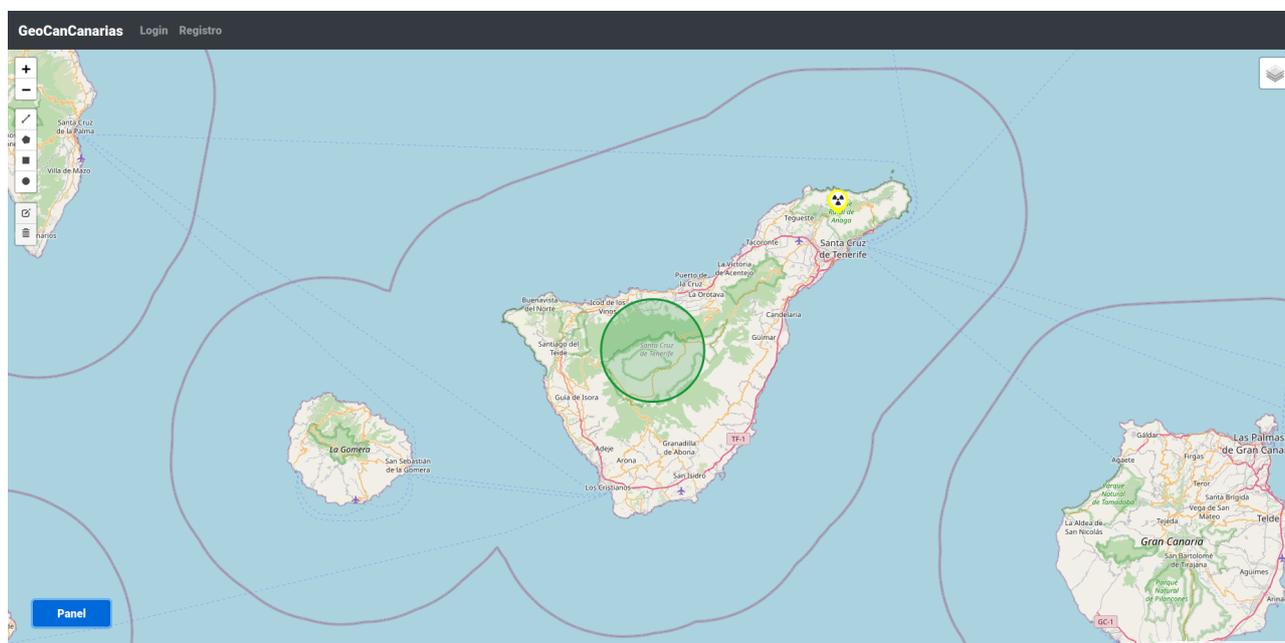


Figura 3.32: Sección mapa

3.10. Sección de administrador

Por último, se encuentra la sección para el usuario con el rol de administrador como se muestra en la Figura 3.33. Así un administrador puede eliminar un usuario, cambiar el rol a un usuario, añadir un nuevo usuario y verificar o validar una cuenta. Como se puede observar todos los campos cuentan con una lista de los usuarios existentes para eliminarlos o cambiarles el rol. También los usuarios que se han registrado y necesitan la validación del administrador para poder iniciar sesión. El formulario para añadir un nuevo usuario es muy parecido al del registro mostrado anteriormente.

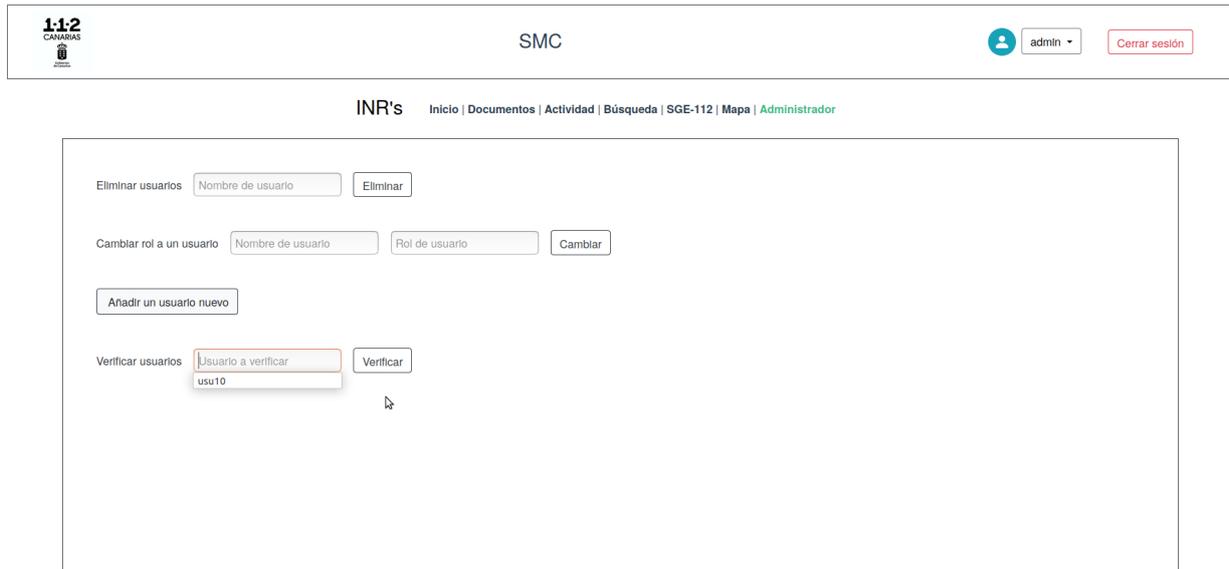


Figura 3.33: Sección administrador

Capítulo 4

Desarrollo

En el capítulo anterior se describió el diseño inicial y final con todas sus funcionalidades. En este capítulo se van a describir las partes más relevantes del backend y el frontend de la aplicación. Este proyecto se encuentra dividido en dos partes principales que son el backend y el frontend.

4.1. Backend

El backend es la parte del desarrollo web que se encarga de que toda la lógica de una página web funcione.

Servidor

En el servidor del backend se incluyen todas las herramientas necesarias mencionadas en el capítulo 2. Es muy importante para poder conectarse a la base de datos. El servidor se muestra en la Figura 4.1.

```
> JS server.js > ...
'use strict';
var express = require('express');
var serveStatic = require('serve-static');
const morgan = require('morgan');
const cors = require('cors');
const config = require('./config');
const user = require('./routes/index');
const app = express();

app.use(cors({origin: '*', credentials: true}));
app.use(express.json());
app.use(express.urlencoded({extended: true}));
app.use(morgan("tiny"));
app.use('/user', user)
app.use(serveStatic(__dirname + "/dist"));

const mongoose = require('mongoose');

mongoose
  .connect(config.db, config.options)
  .then(console.log("Conectado a la bbdd"))
  .catch(err => console.log(err));

app.listen(config.port, () => console.log(`Escuchando en el puerto ${config.port}`));
```

Figura 4.1: Código del servidor del backend

Base de datos: Models

En este apartado se van a explicar los modelos de la base de datos para un mayor entendimiento de los datos utilizados.

Primero se creó el modelo para los datos del usuario. Este modelo se muestra en la Figura 4.2. Los datos más relevantes e importantes son el nombre de usuario y el correo electrónico debido a que es lo que permite identificar de manera única a cada usuario. También se necesita el rol de cada usuario para la distinción de funcionalidades entre los usuarios. Estos pueden ser 'ADMIN' o 'USER', siendo este último el valor por defecto. La contraseña y la contraseña confirmada sirven para el inicio de sesión. El atributo activo se utiliza para saber si un usuario ha iniciado sesión y se encuentra activo o, por el contrario ha cerrado sesión y se encuentra inactivo. La fecha y el último inicio de sesión son datos extra para tener más datos de usuario. Por último el atributo verificado sirve para conocer si un usuario se ha registrado y se encuentra pendiente de validación de su cuenta por un administrador y necesita que este la valide para que el usuario puede iniciar sesión.

```
const mongoose = require('mongoose');
const Schema = mongoose.Schema;
const mongooseFieldEncryption = require("mongoose-field-encryption").fieldEncryption;
const key = "mf68snf6k1p3j5d8g9c2j4b6";

// Roles
const roles = {
  values: ['ADMIN', 'USER'],
  message: '{VALUE} no es un rol válido'
}

const usuarioSchema = new Schema({
  usuario: {type: String, required: [true, 'Nombre obligatorio'], unique:true },
  email: {type: String, required: [true, 'Email obligatorio'], unique:true },
  password: { type: String, required: [true, 'Pass es necesario'] },
  confirmpassword: { type: String, required: [true, 'Pass es necesario'] },
  role: { type: String, default: 'USER', enum: roles, select: true },
  activo: { type: Boolean, default: true },
  date:{type: Date, default: Date.now},
  lastLogin: Date,
  verificado: {type: Boolean, default: true},
});

usuarioSchema.plugin(mongooseFieldEncryption, {
  fields: ["usuario", "email", "password", "confirmpassword"], secret: key});
module.exports = mongoose.model('usuario', usuarioSchema);
```

Figura 4.2: Código del modelo de la base de datos del usuario

A continuación se creó el modelo de datos para los INRs debido a que son los datos más relevantes e importantes para este proyecto. Este se muestra en la Figura 4.3. El dato único que identifica a este modelo es el nombre del INR. Otros datos importantes a conocer son la localización donde ha tenido lugar dicho INR, una descripción breve de lo ocurrido, el número de personas afectadas y las fechas de inicio y finalización del mismo. Se ha categorizado el INR según gravedades, tipos de INR que define si se trata de un caso real o de una simulación para estar más preparados ante un posible caso real, recursos necesarios y el tipo de terreno en el que ha tenido lugar.

```

const mongoose = require('mongoose');
const Schema = mongoose.Schema;
const mongooseFieldEncryption = require("mongoose-field-encryption").fieldEncryption;
const key = "j5o7m9gawg1jtydfh23gfc456k";

const gravedades = {
  values: ['GRAVE', 'MEDIO', 'BAJO'],
  message: '{VALUE} no es una gravedad válida'
}

const tipos = {
  values: ['SIMULACRO', 'CASO REAL'],
  message: '{VALUE} no es un tipo válido'
}

const recursos = {
  values: ['POLICIA LOCAL', 'POLICIA NACIONAL', 'GUARDIA CIVIL', 'AMBULANCIA', 'BOMBEROS', 'GES'],
  message: '{VALUE} no es un recurso válido'
}

const terrenos = {
  values: ['TERRESTRE', 'ACUATICO', 'AEREO'],
  message: '{VALUE} no es un tipo de terreno válido'
}

const inrSchema = new Schema({
  nombre: {type: String, required: [true, 'Nombre obligatorio'], unique:true },
  localizacion: {type: String, required: [true, 'Localización obligatoria'] },
  descripcion: { type: String, required: [true, 'Descripción es necesario'] },
  gravedad: { type: String, enum: gravedades, select: true, required: [true, 'Gravedad es necesario'] },
  tipo: { type: String, enum: tipos, select: true, required: [true, 'Tipo de INR es necesario'] },
  activo: { type: Boolean, default: true },
  numAfectados: {type: Number, required: [true, 'Número de afectados necesario']},
  recursosNecesarios: {type: String, enum: recursos, select: true },
  tipoTerreno: {type: String, enum: terrenos, select: true,required: [true, 'Tipo de terreno obligatorio']},
  fechaInicio:{type: Date, default: Date.now, required: [true, 'Fecha de inicio del INR obligatorio']},
  fechaFin:{type: Date, required: [true, 'Fecha de finalización del INR obligatorio']},
  recursoArray: {type:Array, select: true, required: [true, 'Recursos necesarios obligatorio']},
});

inrSchema.plugin(mongooseFieldEncryption, {
  fields: ["nombre", "localizacion", "descripcion"], secret: key});
module.exports = mongoose.model('inr', inrSchema);

```

Figura 4.3: Código del modelo de la base de datos del INR

Para la sección de documentos se ha definido un modelo de datos para las carpetas como se muestra en la Figura 4.4. El nombre de la carpeta es el dato que tiene que ser único para identificar la carpeta. El atributo 'file' es un 'array' que contiene los archivos que se encuentran en una carpeta determinada. El 'array' favoritos contiene las carpetas que cada usuario ha puesto en favoritos para agilizar la búsqueda de las carpetas más frecuentada por ellos. Fav sirve para identificar si una carpeta se encuentra en favoritos para mostrar el corazón de favorito lleno o si por el contrario no pertenece y este se muestra vacío. El atributo inr sirve para identificar el INR para el que se ha creado dicha carpeta.

```

const mongoose = require('mongoose');
const Schema = mongoose.Schema;
var INR = mongoose.model('inr');
const mongooseFieldEncryption = require("mongoose-field-encryption").fieldEncryption;
const key = "iug45klj3srdf2fcg34v6dx7j8b2";

const carpetaSchema = new Schema({
  nombre: {type: String, required: [true, 'Nombre de la carpeta obligatorio'], unique:true },
  file: {type: Array, select: true},
  inr: {type: Schema.ObjectId, ref: INR},
  fav: {type: Boolean, default: false},
  favoritos: {type: Array, select: true},
});

carpetaSchema.plugin(mongooseFieldEncryption, {
  fields: ["nombre", "inr"], secret: key});
module.exports = mongoose.model('carpetas', carpetaSchema);

```

Figura 4.4: Código del modelo de la base de datos de la carpeta

Para la sección de aportaciones se ha definido un modelo de datos para las aportaciones como se muestra en la Figura 4.5. El título es el dato único que identifica a cada aportación. Cada aportación consta de una breve descripción, los recursos necesarios que ha solicitado, la fecha, el usuario que ha creado la aportación y si dicha aportación ha sido o no rectificada. El atributo inr sirve para identificar el INR para el que se ha creado dicha aportación.

```
const mongoose = require('mongoose');
const Schema = mongoose.Schema;
var INR = mongoose.model('inr');
const mongooseFieldEncryption = require("mongoose-field-encryption").fieldEncryption;
const key = "56hv3g32kjin2gvcdxf8gvh90jh6bb4hv";

const aportacionesSchema = new Schema({
  titulo: {type: String, required: [true, 'Título obligatorio'], unique:true },
  descripcion: { type: String, required: [true, 'Descripción es necesario'] },
  recursosAportacion: {type: Array, select: true, required: [true, 'Recursos necesarios obligatorio'] },
  date:{type: Date, default: Date.now},
  createdBy:{type: String},
  inr: {type: Schema.ObjectId, ref: INR},
  rectificado: {type: Boolean, default: false}
});

aportacionesSchema.plugin(mongooseFieldEncryption, {
  fields: ["titulo", "descripcion", "createdBy", "inr"], secret: key});
module.exports = mongoose.model('aportaciones', aportacionesSchema);
```

Figura 4.5: Código del modelo de la base de datos de la aportación

Router

En este apartado se van a explicar las rutas utilizadas en el backend que se muestran en la Figura 4.6 para gestionar las rutas de las peticiones recibidas del frontend.

Existen cuatro tipo de peticiones. Las peticiones GET son para recuperar datos, las POST se utilizan para enviar datos a un recurso en específico, las PUT sirven para reemplazar representaciones actuales del recurso de destino con la carga útil de la petición, y por último, las DELETE se utilizan para borrar un recurso en específico.

Se puede observar como existen cuatro bloques principales según cada modelo de datos que han sido explicados en el apartado anterior.

Para un usuario las funciones que se pueden realizar son: iniciar sesión, registrarse, eliminar un usuario, obtener todos los usuarios, un usuario concreto por su correo electrónico, los roles existentes, los usuarios que no se encuentran validados, cerrar sesión cambiando el estado del atributo activo de un usuario, cambiar el rol a un usuario y cambiar el estado del atributo verificado de un usuario concreto.

Para un INR se pueden mostrar todos los INRs existentes, los niveles de gravedad, los tipos de INR, los tipos de terreno y los recursos existentes, crear un INR, modificar los datos de un INR concreto y eliminar un INR.

Para una aportación se pueden mostrar todas las aportaciones existentes, una en concreto, crear una aportación, cambiar los datos de una aportación concreta o eliminar una aportación.

Para una carpeta se pueden obtener todas las carpetas, las que se encuentren en favoritos, añadir una carpeta, añadir un fichero a una carpeta, añadir a favoritos una carpeta, eliminar de favoritos una carpeta y eliminar una carpeta.

Por último cabe destacar el uso del middleware auth para comprobar que el token del usuario sea válido.

```
const express = require('express');
const api = express.Router();
const user = require('../controllers/user');
const auth = require("../middlewares/auth");
const inr = require('../controllers/inr');
const carpeta = require('../controllers/carpeta');
const aportacion = require('../controllers/aportaciones');

api.post('/signin', user.signIn);
api.post('/signup', user.signUpNoAdmin);
api.post('/signupadmin', user.signUpAdmin);
api.delete('/delete/:usuario', user.deleteUserByUsuario);
api.get('/showall', user.showAll);
api.get('/showuser/:email', user.showUser);
api.get('/getrole', user.getRole);
api.get('/shownoverificados', user.showNoVerificados);
api.put('/logout/:usuario', user.logOut);
api.put('/changerol/:usuario', user.changeRolToUser);
api.put('/changeverificado/:usuario', user.changeVerificado);

api.get('/showinrs', inr.showAll);
api.get('/getgravedad', inr.getGravedad);
api.get('/gettipo', inr.getTipo);
api.get('/getterreno', inr.getTipoTerreno);
api.get('/getrecursos', inr.getRecursos);
api.post('/inr', inr.dataINR);
api.put('/changeDataINR/:inr', inr.changeDataINR);
api.delete('/deleteINR/:inr', inr.deleteINR);

api.get('/showaportaciones', aportacion.showAll);
api.get('/showaportacion/:titulo', aportacion.showAportacion);
api.post('/aportacion', aportacion.dataAportacion);
api.put('/changeDataAportacion/:aportacion', aportacion.changeDataAportacion);
api.delete('/deleteAportacion/:aportacion', aportacion.deleteAportacion);

api.get('/showcarpetas', carpeta.showAll);
api.get('/showfavoritos', carpeta.showFavs);
api.post('/addCarpeta', carpeta.dataCarpeta);
api.put('/addFile', carpeta.fileInCarpeta);
api.put('/addFavorito', carpeta.addFavorito);
api.put('/removeFavorito', carpeta.removeFavorito);
api.delete('/deleteCarpeta/:nombre', carpeta.deleteCarpeta);

api.get('/isAuth', auth, (req, res) => {
  res.status(200).send({ auth: true });
});

module.exports = api;
```

Figura 4.6: Código de las vistas del backend

Controllers

En este apartado se van a explicar algunas de las funcionalidades de los modelos mencionados anteriormente.

Primero, se va a explicar la función de inicio de sesión del modelo de datos del usuario. En la Figura 4.7 se puede observar el código desarrollado para realizar dicha tarea. Hay que destacar que existen campos en la base de datos que se encuentran encriptados, como el email del usuario en este caso, y para poder comprobar que coinciden primero se han de obtener todos los usuarios de la base de datos y comprobar cual es el correo electrónico que coincide con el del usuario que inicia sesión. A continuación, se accede a la base de datos del usuario que trata de iniciar sesión utilizando bcrypt para la encriptación

de la contraseña del usuario. Si existe dicho usuario y no hay ningún error se envía una respuesta con los distintos datos del usuario: token, email, role, usuario y verificado, seguido de un mensaje para verificar el correcto inicio de sesión del usuario. También se puede observar la función para cambiar el último inicio de sesión del usuario.

```
async function signIn(req, res) {  
  
  var usu = {};  
  var usuarios = await User.find();  
  
  for (var item in usuarios){  
    console.log(usuarios[item].email)  
  
    if(usuarios[item].email === req.body.email){  
  
      usu = usuarios[item];  
  
    }  
  }  
  
  User.findById({ _id: usu._id })  
  .select('password')  
  .then((user) => {  
  
    if (!user)  
      return res.status(404).send({  
        message: 'Este usuario no está en la BBDD',  
      });  
  
    var salt = bcrypt.genSaltSync(10);  
    var hash = bcrypt.hashSync(usu.password, salt);  
    bcrypt.compare(usu.password, hash, function(  
      err,  
      result,  
    ) {  
      if (err)  
        return res.status(500).send({  
          message: 'Error al realizar la petición ${err}',  
        });  
      if (result) {  
        let changelastSignIn = changelastLogin(  
          usu.email,  
        );  
        changelastSignIn.then(response =>  
  
          | console.log(`Last Login: ${response.lastLogin}`)  
        );  
  
        res.status(200).send({  
          message: 'Te has logueado correctamente',  
          token: services.createToken(  
            user,  
            req.originalUrl,  
          ),  
          email: usu.email,  
          role: usu.role,  
          usuario: usu.usuario,  
          verificado: usu.verificado,  
        });  
      } else {  
        res.status(500).send({  
          message: 'Email o Contraseña incorrectos',  
        });  
      }  
    }  
  });  
});
```

Figura 4.7: Código de la función de inicio de sesión del usuario

Ahora se va a explicar la función de eliminar un INR del modelo de datos de los INR. En la Figura 4.8 se muestra el código desarrollado para realizar dicha tarea. Nuevamente el primer paso es obtener el INR concreto que se desea eliminar buscando en la base de datos el nombre del INR, campo que también se encuentra encriptado. Una vez se obtiene dicho INR se accede a este en la base de datos y se elimina. Si no hay ningún error se envía una respuesta con los datos del INR eliminado.

```
async function deleteINR(req,res){  
  
    var i = {};  
    var inrs = await INR.find();  
  
    for (var item in inrs){  
        if(inrs[item].nombre === req.params.inr){  
            i = inrs[item];  
        }  
    }  
  
    try{  
        const inrDB = await INR.findByIdAndRemove(i._id);  
  
        if(!inrDB){  
            return res.status(400).json({  
                mensaje: 'No se encontró el nombre del INR indicado'  
            })  
        }  
        res.json(inrDB);  
    } catch (error) {  
        return res.status(400).json({  
            mensaje: 'Ocurrió un error',  
            error  
        });  
    }  
}
```

Figura 4.8: Código de la función de eliminar un INR

Para el modelo de datos de las aportaciones se ha decidido explicar el código desarrollado de la función de modificar los datos de una aportación concreta que se muestra en la Figura 4.9. Para ello, se obtiene la aportación concreta de la que se quieren modificar los datos a través del campo título, que se encuentra encriptado en la base de datos. Posteriormente se accede a dicha aportación en la base de datos y se actualizan los datos antiguos con los nuevos enviados desde el frontend. Se observa como el campo rectificado cambia de valor a 'true' ya que se ha rectificado la aportación cambiando sus datos. Si no ocurre ningún error se envía una respuesta de la aportación con los nuevos datos.

```

async function changeDataAportacion(req,res){

    var ap = {};
    var aprts = await Aportaciones.find();

    for (var item in aprts){

        if(aprts[item].titulo === req.params.aportacion.toUpperCase()){

            ap = aprts[item];

        }

    }

    try{

        const aportacionDb = await Aportaciones.findByIdAndUpdate(
            {_id: ap._id},
            {
                titulo: req.body.aportacion.aportacion.titulo.toUpperCase(),
                descripcion: req.body.aportacion.aportacion.descripcion.toUpperCase(),
                recursosAportacion: req.body.aportacion.aportacion.recursosAportacion,
                rectificado: true,
            }
        );

        res.json(aportacionDb);

    } catch (error) {

        return res.status(400).json({
            mensaje: 'Ocurrio un error',
            error
        });

    }

}

```

Figura 4.9: Código de la función de modificar los datos de una aportación

Por último, para el modelo de datos de las carpetas se ha escogido explicar el código desarrollado de la función de añadir un archivo o fichero a una carpeta concreta que se puede observar en la Figura 4.10. Primero, se obtiene la carpeta concreta en la que se quiere añadir un fichero determinado a través del campo del nombre de la carpeta, que se encuentra encriptado en la base de datos. Más adelante, se añade al 'array' de ficheros de dicha carpeta el fichero que se quiere añadir a la carpeta. Por último, si no ocurre ningún error se accede a la base de datos de la carpeta concreta y se modifica el 'array' para que se añada el fichero nuevo y, se envía una respuesta de la carpeta con los datos modificados.

```

async function fileInCarpeta(req,res) {

    var car = {};
    var crpts = await Carpetas.find();

    for (var item in crpts){

        if(crpts[item].nombre == req.body.nombre){

            car = crpts[item];

        }

    }

    const file = req.body.file;
    var array = [];

    array.push(file);
    console.log(array)

    try{

        const carpetaDb = await Carpetas.findByIdAndUpdate(
            {_id: car._id},
            {
                file: array,
            }
        );
        console.log(carpetaDb)
        res.json(carpetaDb);

    } catch (error) {

        return res.status(400).json({
            mensaje: 'Ocurrio un error',
            error
        });

    }

}

```

Figura 4.10: Código de añadir un fichero en una carpeta

4.2. Frontend

El frontend es la parte de un sitio web que interactúa con los usuarios. Son todas las tecnologías de diseño y desarrollo web que corren en el navegador y que se encargan de la interactividad con los usuarios.

Diseño de vistas

En este proyecto se han creado ocho vistas que se corresponden con las secciones explicadas en apartados anteriores. Estas vistas son las que se encuentran en la Figura 4.11.



Figura 4.11: Vistas del frontend

Diseño de componentes

En este proyecto se han creado siete componentes que conforman la página web. Estos componentes son los que se encuentran en la Figura 4.12. Existe un componente para la sección de inicio de sesión y registro. Para la sección de inicio se encuentran los componentes de la página principal de inicio y los filtros. Para la sección de documentos, actividad, búsqueda y SGE-112 se encuentran los componentes de los datos del INR y la barra de navegación. El componente de la cabecera se encuentra durante toda la página en todas las secciones salvo en la sección de inicio de sesión y registro. Para la sección del administrador se encuentra el componente admin.



Figura 4.12: Componentes del frontend

Router

En este apartado se van a explicar las rutas utilizadas en el frontend. Algunas de ellas se muestran en la Figura 4.13. Se observa como se requiere autorización para todas las vistas salvo para el inicio de sesión.

```

Vue.use(VueRouter)

const routes = [
  {
    path: '/',
    name: 'InicioSesion',
    meta: { isAuth: false },
    component: InicioSesion
  },
  {
    path: '/inicio',
    name: 'Inicio',
    meta: { isAuth: true },
    component: Inicio
  },
  {
    path: '/documentos',
    name: 'Documentos',
    meta: { isAuth: true },
    component: function () {
      return import('../views/Documentos.vue')
    }
  }
],

```

Figura 4.13: Código de las rutas del frontend

Vuex

Vuex es una librería que se encarga de la gestión del estado de las aplicaciones Vue. Sirve como un almacén centralizado para todos los componentes de una aplicación Vue, con reglas que garantizan que el estado se puede cambiar de manera predecible. En la Figura 4.14 se representa un ejemplo de los estados que se han utilizado.

```

state: {
  session_out: true,
  token: '',
  email: '',
  role: '',
  usuario: '',
  inr: {
    nombre: "",
    localizacion: "",
    descripcion: "",
    gravedad: "",
    tipo: "",
    numAfectados: "",
    recursosNecesarios: [],
    tipoTerreno: [],
    fechaInicio: '',
    fechaFin: ''
  }
},

```

Figura 4.14: Ejemplo de los estados Vuex

Capítulo 5

Evaluación

5.1. Usabilidad y accesibilidad

Usabilidad

La usabilidad es una disciplina que se encarga de aquellos aspectos de una web orientados a mejorar y hacer más cómoda la interacción de los usuarios con la propia página. Para medir la usabilidad se ha empleado el método Sirius y se han evaluado las diez heurísticas de Nielsen.

Sirius [29] es un sistema de evaluación de la usabilidad web orientado al usuario y basado en la determinación de tareas críticas. Este sistema utiliza la evaluación heurística para detectar deficiencias en las interfaces de los sistemas interactivos. En la Figura 5.1 se muestra el resultado del porcentaje de usabilidad de la página de inicio de la aplicación aplicando el método Sirius.

CÁLCULO DEL PORCENTAJE DE USABILIDAD

Porcentaje de usabilidad*	96,70
---------------------------	-------

Datos para el cálculo:

Nº de criterios evaluados	67
El sumatorio de los valores de relevancia de los criterios evaluados es:	159
Sumatorio de la columna <i>i</i> de cada criterio	9,6698113208
Sumatorio de la columna <i>h</i> *10 de cada criterio	10,0000000000

Figura 5.1: Resultado del porcentaje de usabilidad con el método Sirius

Con la realización del método Sirius se ha podido observar el fallo existente en la búsqueda en la que si se realiza la búsqueda de una aportación no existente no se produce ningún aviso para el usuario, una sencilla tarea de implementar de la cual no se era consciente sin la realización de la evaluación Sirius.

El método de Jakob Nielsen [30] es una evaluación ligera que permite identificar problemas de usabilidad en un sitio web sin la participación de los usuarios. Existen diez heurísticas de Nielsen:

- **Visibilidad del estado del sistema:** se puede observar un ejemplo en la Figura 3.10 en la que el usuario no introduce de forma correcta la misma contraseña en los campos de contraseña y confirmar contraseña lo que genera el aviso mostrado en la imagen.

- **Utilizar el mismo lenguaje que el usuario** : el lenguaje empleado en toda la página es el usado por los usuarios que la van a utilizar.
- **Control y libertad para el usuario** : dado que los usuarios finales de la aplicación van a ser los trabajadores de la empresa 112, la libertad y el control que tendrán van a ser el adecuado ya que se trata de un grupo específico de usuarios para el que se ha diseñado esta aplicación.
- **Consistencia y estándares**: esta aplicación está diseñada para seguir una serie de estándares que los usuarios van a entender como se muestra en la Figura 3.28, en la que se despliega información acerca de las guías de uso de la plataforma. Además, el usuario es capaz de entender el significado del lenguaje y las acciones o situaciones del sistema.
- **Prevención de errores**: actualmente en la página web no se producen errores que puedan ser visualizados por los usuarios. Los errores ocurridos a lo largo del proyecto se han solucionado sin recurrir a la ocultación de los mismos.
- **Minimizar la carga de memoria del usuario**: se han utilizado elementos visuales para minimizar la carga de memorizar información a los usuarios. Uno de estos elementos son los colores utilizados para definir los tipos de gravedad para un INR. Asimismo la aplicación cuenta con una visualización de datos constante sin llegar a ser cargante para la vista del usuario, así éste no tiene que memorizar la información.
- **Flexibilidad y eficiencia de uso**: como se ha mencionado en el apartado de consistencia y estándares se ha añadido información sobre los estándares a seguir por los usuarios de esta aplicación. Así, esta aplicación se encuentra preparada para que lo puedan utilizar todos los trabajadores necesarios del 112, sin necesidad de la realización de grandes tutoriales o gastar muchas horas utilizando la aplicación para poder usarla correctamente, por este motivo es una página bastante usable.
- **Diálogos estéticos y diseño minimalista** : la aplicación tiene un diseño minimalista sin sobrecarga de información para que el usuario no tenga ninguna carga visual y pueda navegar por la página sin ningún tipo de impedimento. Este aspecto se puede comprobar en la Figura 3.12, en la que se muestra la sección de inicio y a lo largo de la navegación por la aplicación.
- **Ayudar a los usuarios a reconocer, diagnosticar y recuperarse de los errores**: se puede observar un ejemplo en la Figura 3.9, en la que se observa que si un usuario se registra no puede iniciar sesión sin que un administrador verifique su cuenta.
- **Ayuda y documentación** : en esta aplicación se ha decidido no añadir una sección de ayuda porque se ha visto innecesario debido a que los usuarios finales son los trabajadores del 112. Sin embargo, en los apartados anteriores se ha explicado el uso de ayudas para orientar al usuario a seguir un estándar en la página web.

Accesibilidad

La accesibilidad es un diseño web que va a permitir que personas con algún tipo de discapacidad puedan percibir, entender, navegar e interactuar con la web, aportando a su vez contenidos.

Para medir la accesibilidad de este proyecto se ha optado por utilizar la extensión de Firefox WAVE [31]. Cuando esta herramienta es activada la extensión WAVE inyecta iconos e indicadores en la página web para brindar comentarios sobre la accesibilidad y facilitar la evaluación manual. Inicialmente se ha evaluado la accesibilidad con el resultado que se muestra en la Figura 5.2.

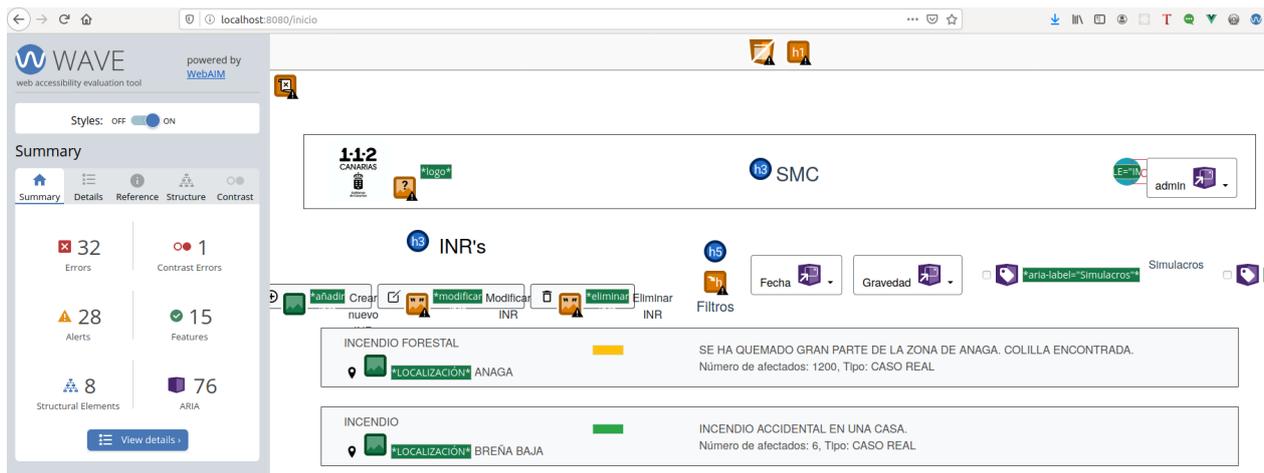


Figura 5.2: Accesibilidad medida con WAVE inicialmente

Después de realizar la primera evaluación de la accesibilidad se han corregido la mayoría de errores y advertencias. Algunos de los errores se han producido por la falta de la etiqueta 'label' antes de los 'input' dentro de los formularios. Algunas advertencias se debían al uso inadecuado del atributo 'tabindex' en los formularios. Otros errores o advertencias indicaban unas referencias vacías en los atributos 'aria-describedby' y 'aria-labelledby'. Se han solucionado bastantes de los problemas de accesibilidad detectados y se ha vuelto a evaluar obteniendo los resultados de la Figura 5.3.

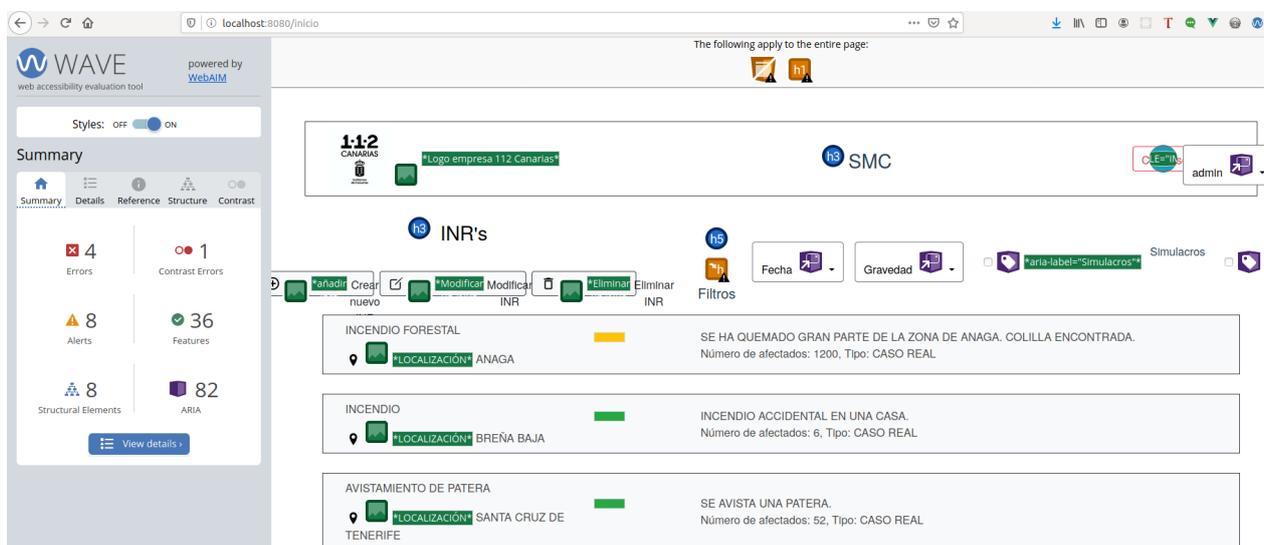


Figura 5.3: Accesibilidad medida con WAVE tras la corrección de errores

Los errores restantes se deben al uso de un elemento relativamente nuevo de Bootstrap Vue usado para escoger las fechas en los formularios 'b-form-datepicker'. Las advertencias y elementos estructurales se deben en su mayoría a los niveles de los encabezados utilizados. Por último, el error de contraste se debe al elemento que sirve para cambiar las páginas de los INR. Al tratarse de un componente incluido de npm no se ha podido modificar ni el color que provoca el error de contraste ni el lenguaje en el que se encuentra.

5.2. Seguridad de la página web

La seguridad de una página web es muy importante y necesaria si se quiere proteger a los clientes de posibles ataques informáticos o robos de datos personales e información de pago. El propósito de la seguridad web es proteger el sitio web del acceso, uso, modificación, destrucción o interrupción de personas no autorizadas.

Es preciso señalar que al tratarse de una empresa de especial importancia para todos los ciudadanos y, tratándose de datos especialmente importantes y privados, es de gran importancia proteger los datos tanto de los ciudadanos como de los incidentes, ya que un mal uso de estos datos podría conllevar una problemática muy grave. Además de la seguridad de los datos almacenados es muy importante la seguridad de la página web para que no se produzcan ataques y se pueda tener acceso a los datos.

Seguridad de los datos

Para poder contar con la seguridad de los datos almacenados en la base de datos de esta aplicación se ha utilizado el paquete de npm 'mongoose-field-encryption' [32]. Se trata de un complemento de cifrado simétrico simple para campos individuales basado en el módulo criptográfico de Node. Su objetivo es encriptar datos permitiendo la búsqueda de dichos datos en la base de datos. El cifrado y descifrado de los datos ocurren de forma transparente durante las acciones de búsqueda y guardar datos.

El cifrado se realiza utilizando AES-256-CBC. Para cifrar, los campos relevantes se cifran con el secreto proporcionado, denominado 'key', y el 'salt' aleatorio. El 'salt' generado y el valor cifrado resultante se concatenan juntos usando un carácter : y la cadena final se coloca en lugar del valor real para los valores de cadena. Se agrega un campo booleano adicional con el prefijo '__enc_' al documento que indica si el campo proporcionado está encriptado o no. Esto se puede observar en la Figura 4.2 de los modelos de la base de datos como por ejemplo el del usuario, en el que se observa la 'key' y los campos que se van a encriptar que en este caso son el nombre de usuario, el email, la contraseña y la confirmación de la contraseña. Además se muestra en la Figura 2.2 la base de datos almacenada en el programa Robo3t comprobando como los datos se encuentran almacenados encriptados y los campos encriptados tienen un campo booleano adicional con el prefijo '__enc_'.

Seguridad de la página web

La seguridad de la aplicación se encuentra dividida en tres elementos importantes:

- **Distinciones de roles de usuarios:** a lo largo de diferentes capítulos se ha comentado que existen dos tipos de usuarios en la aplicación, los usuarios con el rol de

administrador 'ADMIN' y los que tienen el rol por defecto como usuarios normales 'USER'. Se ha observado que durante la navegación de la página web, según el rol del usuario con el que se haya iniciado sesión, se tienen unas funcionalidades u otras. Esto aporta la seguridad necesaria para que haya unas personas responsables con un cargo superior que sean los administradores y otras que sean usuarios normales que no tengan tanto control sobre las funcionalidades de la aplicación. Como ejemplo en la Figura 3.11 se muestra la página de inicio de la aplicación en la que se ve como un administrador puede añadir, modificar o eliminar un INR mientras que en la Figura 3.12 se muestra como un usuario normal no puede realizar ninguna de estas tres tareas.

- **Registro con validación:** se ha mencionado en capítulos anteriores y se trata de que un usuario cuando se registra en la página web no pueda iniciar sesión si un administrador no le ha validado la cuenta previamente. Es un punto importante de seguridad para controlar los registros de sesión y que solo los administradores puedan realmente permitir que se creen cuentas nuevas de usuarios que sean activas. Si un usuario se registra e intenta iniciar sesión sin que el administrador le haya validado la cuenta se muestra el mensaje de la Figura 3.9.
- **Bcrypt:** es una función hashing de contraseñas que tiene dos valores, 'salt', que genera un dato aleatorio y 'hash', que utiliza el valor de 'salt' junto con la contraseña (valor a encriptar). El objetivo es que dos contraseñas idénticas no generen el mismo hash. El proceso de desencriptado se muestra en la Figura 5.4, en el que se envía la contraseña del usuario en el inicio de sesión para aplicarle el hash y, si esta coincide con la almacenada en la base de datos implica que el inicio de sesión es satisfactorio y válido, si no coincidieran las contraseñas significaría que se habría introducido una contraseña incorrecta y no se podría iniciar sesión.
- **Tokens:** se ha comentado en el capítulo del backend el uso de tokens. En esta aplicación se ha utilizado el token JSON Web Token que es una cadena de texto que consta de tres partes codificadas separadas por puntos que serían la cabecera para indicar el algoritmo y tipo de token, la carga útil donde aparecen los datos de usuario y privilegios, y la firma que permite verificar si el token es válido. Para la verificación de tokens se ha utilizado el middleware isAuth en el backend.

```

var salt = bcrypt.genSaltSync(10);
var hash = bcrypt.hashSync(usu.password, salt);
bcrypt.compare(usu.password, hash, function(
  err,
  result,
) {
  if (err)
    return res.status(500).send({
      message: `Error al realizar la petición ${err}`,
    });
  if (result) {
    let changeLastSignIn = changelastLogin(
      usu.email,
    );
    changeLastSignIn.then(response =>
      console.log(`Last Login: ${response.lastLogin}`)
    );

    res.status(200).send({
      message: 'Te has logueado correctamente',
      token: services.createToken(
        user,
        req.originalUrl,
      ),
      email: usu.email,
      role: usu.role,
      usuario: usu.usuario,
      verificado: usu.verificado,
    });
  } else {
    res.status(500).send({
      message: 'Email o Contraseña incorrectos',
    });
  }
});

```

Figura 5.4: Desencriptado bcrypt

Capítulo 6

Presupuesto

Este capítulo consta de una propuesta del presupuesto que estima el coste del trabajo según el tiempo empleado en el desarrollo de las tareas a lo largo del proyecto.

La aplicación al encontrarse desplegada en Heroku se ha decidido escoger el servicio Estándar 2x que incluye 1 GB de RAM y permite la escalabilidad horizontal simple. Este servicio sirve para obtener una mayor visibilidad, rendimiento y disponibilidad para potenciar la aplicación. A continuación se presentan los resultados del presupuesto:

Tareas	Horas	Presupuesto
Investigación de conceptos previos	20 horas	7€/hora
Análisis de requisitos	5 horas	7€/hora
Investigación de las tecnologías adecuadas	20 horas	7€/hora
Diseño de bocetos	15 horas	10€/hora
Diseño de la aplicación	20 horas	15€/hora
Desarrollo backend	60 horas	20€/hora
Desarrollo frontend	70 horas	20€/hora
Despliegue Heroku	2 horas	7€/hora
Hosting Heroku	3 meses	150 €
Evaluación usabilidad y accesibilidad	25 horas	10€/hora
Evaluación seguridad	30 horas	20€/hora
Total	267 horas	4.379€

Tabla 6.1: Modelo de presupuesto para los tres meses de duración del proyecto

Capítulo 7

Conclusiones y líneas futuras

Para concluir, resaltar el gran aprendizaje que ha conllevado la realización de este proyecto, con herramientas que no había utilizado anteriormente. Lograr alcanzar las metas establecidas y adquirir nuevos conocimientos acerca del desarrollo web ha sido un reto satisfactorio.

Los objetivos generales propuestos inicialmente se han cumplido, desarrollando la plataforma de emergencias tanto con la parte del frontend como con la parte del backend, en la que se han implementado incluso más funcionalidades de las esperadas en los objetivos iniciales.

Sin embargo, este es un trabajo abierto así que en un futuro se podrían realizar algunas mejoras en el proyecto. Estas mejoras podrían ser por ejemplo: habilitar en la página web el modo oscuro, en la sección SGE-112 crear un modelo de datos u obtener los datos reales relativos a cada INR de la sala de llamadas, en la sección de actividad incluir una parte de los registros de las llamadas del SGE-112, en la sección de documentos contar con un botón que al presionarlo despliegue una ventana en la que se tenga acceso a los teléfonos, radios o contactos de recursos que puedan ser necesarios o en el componente de navegación incluir una sección de mi perfil para cada usuario. Otras mejoras podrían incluir incrementos en la accesibilidad, usabilidad y seguridad de la aplicación.

Por lo tanto, se trata de una aplicación que puede llegar a ser muy potente y útil para la empresa 112, realizando algunas mejoras y manteniendo la página web. Por tanto, el proyecto cumple con el objetivo de ser una plataforma de emergencias del Sistema de Mando y Control en la que hay distinción de roles, es segura, usable y accesible, cuenta con un diseño claro, sencillo y minimalista, sirve como documentación para la gestión de los INRs, e incluye las distintas funcionalidades mencionadas en capítulos anteriores.

Capítulo 8

Summary and conclusions

To conclude, highlight the great learning that the project realization has entailed with tools that have not been used before. Therefore, achieving the established goals and acquiring new knowledge about web development has been a satisfactory challenge.

The general objectives initially proposed have been met by developing the emergency platform with both the frontend and backend parts, in which even more functionalities than expected in the initial objectives have been developed.

However, some improvements could be made to the project in the future. These improvements could be to enable dark mode on the web page, in the SGE-112 section create a data model or obtain the real data related to each INR of the call room, in the activity section include a part of the records of the calls of the SGE-112, in the documents section have a button that when pressed displays a window in which you have access to the telephones, radios or resource contacts that may be necessary or in the navigation component include a section of my profile for each user. Other improvements could include further improving the accessibility, usability and security of the application.

Therefore, it is an application that can be very powerful and useful for the 112 company, making some improvements and maintaining the website. Whereby the project complies with the objective of being a Command and Control System emergency platform in which there is a distinction of roles, it is safe, usable and accessible, it has a clear, simple and minimalist design, it serves as documentation for INRs management and performs the different functionalities mentioned in previous chapters.

Bibliografía

- [1] Empresa 112 Canarias -><http://www.112canarias.com/info/>
- [2] Stack MEVN -><https://www.faztweb.com/curso/mevn-stack>
- [3] Heroku -><https://www.heroku.com/what>
- [4] GitHub -><https://github.com/about>
- [5] Git -><https://git-scm.com/>
- [6] Backend -><https://www.qualitydevs.com/2018/11/19/que-es-un-programador-backend/>
- [7] Frontend -><https://www.strato.es/wordpress-hosting/wordpress-frontend/>
- [8] Hexagon -><https://hexagon.com/>
- [9] Visual Studio Code -><https://code.visualstudio.com/>
- [10] Calvo K., Durán J., Quirós E. y Malinowski E., *MongoDB: alternativas de implementar y consultar documentos*, Facultad de Ciencias de la Computación e Informática, Universidad de Costa Rica, Costa Rica, 8, 2017
- [11] Express-><https://expressjs.com/es/>
- [12] Vue.js -><https://vuejs.org/>
- [13] Íscar Martínez J., *NODE.JS Do's and Don'ts*, Escola Tècnica Superior d'Enginyeria Informàtica, Universitat Politècnica de València, Valencia, 66, 2015
- [14] Axios -><https://codingpotions.com/vue-axios>
- [15] López S., *Cómo mantener el patrón modelo vista controlador en una aplicación orientada a la WEB*, INVENTUM, Colombia, 7, 2009
- [16] Binker C., Tantignone H., Buranits G., Moreira R.D. y Zurdo E., *Interacción de elementos de networking con contenedores de software (docker)*, SEDICI, Universidad Nacional de la plata, Red de Universidades con Carreras en Informática, 10, 2018
- [17] Vue CLI -><https://cli.vuejs.org/>
- [18] Robo 3T-><https://robomongo.org/>
- [19] Postman-><https://www.postman.com/>
- [20] Middleware -><https://bluuweb.github.io/mevn/01-primeros-pasos>

- [21] Morgan -><https://www.npmjs.com/package/morgan>
- [22] CORS -><https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/cross-origin-resource-sharing/>
- [23] Bcrypt -><https://www.npmjs.com/package/bcrypt>
- [24] Olivera J., *JWT: Json Web Token*, Universidad Católica "Nuestra Señora de la Asunción", Teoría y Aplicación de la Informática 2, 18, 2014
- [25] Bootstrap -><https://axarnet.es/blog/bootstrap>
- [26] Bootstrap Vue -><https://bootstrap-vue.org/>
- [27] Marvel -><https://marvelapp.com/>
- [28] Repositorio GitHub -><https://github.com/alu0101061672/TFG>
- [29] Perurena Cancio L. y Moráguez Bergues M., *Usabilidad de los sitios Web, los métodos y las técnicas para la evaluación*, Revista Cubana de Información en Ciencias de la Salud, Universidad de La Habana, 19, 2013
- [30] Paz Espinoza F.A., *HEURÍSTICAS DE USABILIDAD PARA SITIOS WEB TRANSACCIONALES*, Pontificia Universidad Católica del Perú, Escuela de Posgrado, 60, 2013
- [31] Buj Gelonch M., *Accesibilidad web*, Universitat Oberta de Catalunya , Cataluña, 113-118, 2011
- [32] Mongoose field encryption -><https://www.npmjs.com/package/mongoose-field-encryption>