



ZgZInfo



Paula Ezpeleta - 800692

Javier Valles - 798523

Juliana Zordan - 776390

Oscar Palacín - 797021

Trabajo final

| | |
|--|-----------|
| URL | 3 |
| URL donde se ha publicado la aplicación | 3 |
| La URL para acceder a la aplicación es la siguiente: | 3 |
| URL de la documentación Swagger/API REST. | 3 |
| Resumen del proyecto | 3 |
| Propuestas similares | 3 |
| Relación de módulos y utilidad | 4 |
| Módulos Backend | 4 |
| Módulos frontend | 4 |
| Implementación | 5 |
| Frontend | 5 |
| Backend | 5 |
| Análíticas | 6 |
| Despliegue del sistema | 9 |
| Frontend | 9 |
| Backend | 9 |
| Validación | 10 |
| Problemas encontrados | 11 |
| Análisis de problemas potenciales | 11 |
| Distribución del tiempo | 12 |
| Trabajo individual | 14 |
| Conclusiones | 15 |
| Valoración | 15 |
| Anexos | 16 |

URL

URL donde se ha publicado la aplicación

La URL para acceder a la aplicación es la siguiente:

<http://zgzinbucket1.s3-website.eu-west-3.amazonaws.com/>

- Para iniciar sesión con un usuario normal se deben introducir los siguientes datos:
Email: javi23@gmail.com
- Contraseña: javi23@gmail.com

Para iniciar sesión como administrador se deben introducir los siguientes datos:

- Email: admin@gmail.com
- Contraseña: 123456

URL de la documentación Swagger/API REST.

http://13.53.45.121:8080/api/doc/#/Incidencias/get_api_incidenciasLista

Resumen del proyecto

Esta aplicación se trata de una aplicación para poder ver las incidencias en tiempo real ocurridas en la ciudad de Zaragoza. Las incidencias se pueden ver sobre un mapa o en forma de listado.

Para obtener estas incidencias se utiliza la API proporcionada por el Ayuntamiento de zaragoza: <https://www.zaragoza.es/sede/portal/datos-abiertos/servicio/catalogo/67>

Se permite utilizar la aplicación también a usuarios que no estén registrados, pero no cuentan con todas las funcionalidades de un usuario registrado (suscribirse a un tipo de incidencia, foro o comentar en este).

Propuestas similares

Realmente, a la hora de presentar la propuesta de la aplicación y de recibir el visto bueno por parte del profesor, no existía ninguna aplicación que funcionase como esta, pero dos días más tarde, concretamente el 28 del mes de febrero, el ayuntamiento de Zaragoza lanzó la aplicación “Conecta Zaragoza”.

Esta aplicación cubría exactamente las mismas funcionalidades que la implementada para la asignatura, excepto la parte de constar con foros para cada incidencia y de poder comentar en ellos. En cuanto a esto, nuestra aplicación es más completa.

Una desventaja de la aplicación del ayuntamiento con respecto a la nuestra es que al no ser una aplicación web, es necesario descargarla (está disponible tanto para Android como para los).

Para conocer más acerca de la aplicación del ayuntamiento, se puede visitar este [link](#).

Relación de módulos y utilidad

Módulos Backend

Se ha utilizado un patrón de diseño MVC (modelo vista controlador). De esta forma hemos separado la lógica de negocio de la gestión de los datos.

En los controladores tenemos los módulos que implementan las operaciones que son invocadas desde las vistas. Tenemos 4 controladores para gestionar todos los recursos:

- Admin: Define las operaciones y las gráficas correspondientes al usuario administrador.
- Foro: Define las operaciones y gráficas relacionadas con los foros.
- Incidencia: Define las operaciones y gráficas relacionadas con las incidencias.
- Usuario: Define las operaciones y gráficas relacionadas con los usuarios.

Modelos: Se han definido los esquemas de todos los modelos de datos necesarios para la aplicación: Comentarios, Foro, Incidencias y Usuarios. Cada uno define las estructuras que deben guardar los documentos en la base de datos.

index: Se definen todas las rutas y operaciones disponibles en la API.

db: Establece y configura la conexión con la base de datos mongoDB.

Módulos frontend

Los módulos creados en frontend a la hora de desarrollar la aplicación han sido los siguientes:

- Módulo pages: Contiene los ficheros necesarios para mostrar correctamente las páginas por pantalla y la implementación de la funcionalidad con el BackEnd. Dentro de éste podemos encontrar el módulo CSS, el cual sirve para definir el estilo que se le desea dar a la página al renderizarla.
- Módulo componentes: Aquí se encuentra todo lo que el módulo pages utilizará. Cada archivo dentro de este módulo puede que sea utilizado por uno o más archivos del módulo pages, como por ejemplo la barra de navegación (NavBar). El módulo componentes también incluye los gráficos utilizados. En resumen, el módulo pages

utilizará este módulo para mostrar información adicional por pantalla como pueden ser los gráficos estadísticos o la barra de navegación.

- Módulo urls: Este módulo se compone de un único fichero en donde se encuentran las urls del Backend correspondientes a las respectivas APIs.
- Ficheros adicionales:
 - Fichero index: Es el primer archivo que se ejecuta al iniciar la aplicación con node. En él aparece el siguiente fichero a ejecutarse es el fichero rutas, el cual se explica a continuación.
 - Fichero rutas: Indica qué páginas deben mostrarse al iniciar la aplicación, además de También sirve para redirigir a las páginas correspondientes si por ejemplo el usuario que ha iniciado sesión es administrador o no o si hay un usuario logueado o no.

Otros módulos que se han importado desde Internet que no hemos desarrollado nosotros pero han tenido especial relevancia a la hora de desarrollar la aplicación han sido los siguientes:

- Módulo axios: Es imprescindible para poder realizar las peticiones al BackEnd.
- Módulo proj4: Ha servido para poder transformar las coordenadas de un sistema a otro.
- Módulo Chart: Para poder representar los gráficos estadísticos.
- Módulo Cypress: Para poder realizar los test de e2e y de componentes.

Implementación

Frontend

Para la implementación del frontend no ha sido necesario ningún componente adicional, únicamente ha sido necesario transformar y filtrar los datos que se recibían de backend para que el usuario viera por pantalla lo que únicamente necesitaba ver. Los datos de las incidencias se almacenaban en el frontend en una variable “Map”, que consistía en un array de ids al que cada id se le asociaban los datos correspondientes (título, coordenadas, tipo de incidencias, etc).

Backend

Para el Backend, la implementación a grandes rasgos, ha sido consumir los datos de la api de Incidencias del gobierno de Aragón en formato Json (lenguaje nativo de Mongo) y transformar estos datos para poblar un conjunto de schemas.

El único schema para el que no se ha consultado la api, es el de Usuarios.

Posteriormente, para la realización de peticiones, los datos se transforman dependiendo de las necesidades de la aplicación y de lo que quería mostrar el Frontend y se enviaban en la respuesta.

Análíticas

Para el usuario normal se muestran 3 gráficas de análisis.

La primera de ellas, se encuentra en la página del listado de incidencias, y aporta información del número de incidencias de cada tipo existentes en tiempo real.

Se puede observar el gráfico en la siguiente imagen:

Número de incidencias por tipo



La siguiente gráfica con la que cuenta el usuario es una gráfica en la pantalla de “Mis suscripciones” (por tanto solo podrá acceder a ella el usuario registrado). Esta gráfica muestra el número de usuarios suscrito a los distintos tipos de incidencias:

Usuarios suscritos:



Por último, una gráfica que muestra el número total de foros guardados de cada tipo por los usuarios. Igualmente, sólo tendrán acceso a esta gráfica los usuarios que estén registrados.

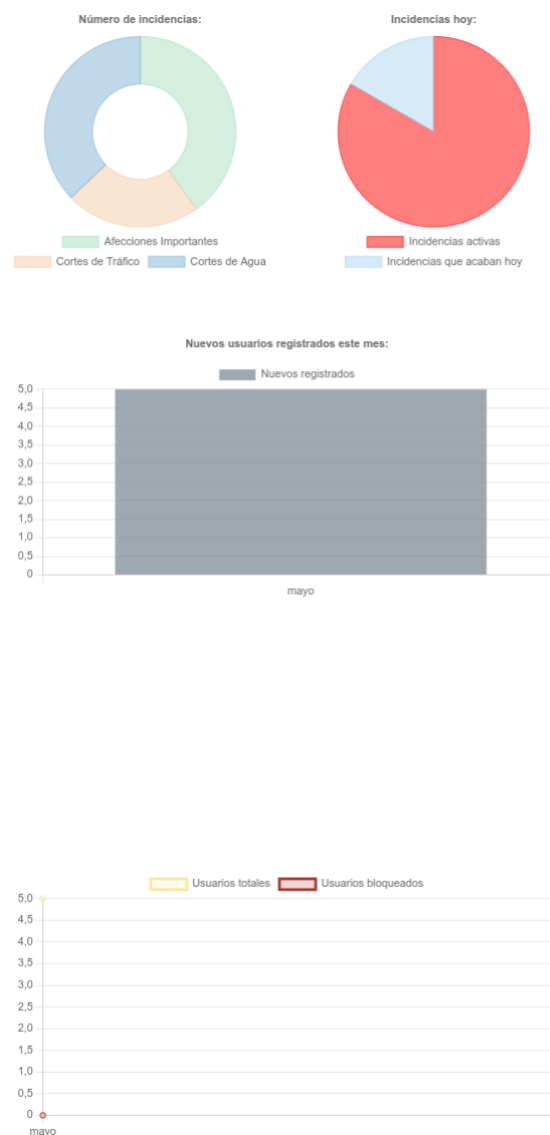
Foros con más "guardados":



Por otra parte, se ha implementado una pantalla entera en la parte del administrador para que este observe todas las gráficas al mismo tiempo.

La pantalla se puede observar en forma de imagen debajo de este texto e incluye las siguientes gráficas:

1. En primer lugar, una gráfica que muestra el número de incidencias activas de cada tipo
2. En segundo lugar, una gráfica que muestra el número de incidencias totales activas vs. el número de esas incidencias que finalizan hoy
3. En tercer lugar, una gráfica con el número de nuevos usuarios registrados en el mes actual
4. En cuarto y último lugar, una gráfica que muestra por meses (actualmente sólo existe el mes de mayo) el número total de usuarios vs. el número de usuarios bloqueados



Despliegue del sistema

Frontend

Para realizar el despliegue del proyecto sería necesario en primer lugar descargar los repositorios de frontend y de backend. En segundo lugar, habría que desplegar el frontend.

Para desplegar el frontend, se ha utilizado AWS S3. Para desplegarlo en él, se ha creado un bucket sobre el que se han subido los archivos del interior del directorio “build” del repositorio de frontend (este directorio build se ha obtenido mediante el comando “npm run build” y contiene todos los ficheros necesarios para desplegar el frontend. En caso de querer desplegar el backend en una máquina diferente a la que se encuentra, sería necesario cambiar la dirección de la máquina donde está desplegada el backend, en el archivo “URLs.js” del repositorio de frontend y volver a crear el directorio build. También eliminar el directorio build que existía y utilizar el nuevo).

Una vez subidos los archivos, se ha configurado el bucket para ser público y que todos los usuarios puedan acceder a él desde cualquier parte, concediéndoles permisos de “GET object”.

Backend

Para el despliegue del Backend, se ha utilizado AWS EC2.

Esta herramienta ha sido muy fácil de utilizar ya que consistía en desplegar en Amazon una instancia de una máquina, para este caso se ha escogido una máquina Ubuntu, pues ha sido en este entorno donde se ha desarrollado el Backend.

A esta instancia desde el propio Amazon, se le han añadido reglas de seguridad de entrada para poder acceder a esta tanto desde un puerto local como desde el 8080, y se han generado claves del tipo clave, valor para la autenticación ssh.

| Versión de IP | Tipo | Protocolo | Intervalo de puertos | Origen |
|---------------|-------------------|-----------|----------------------|-----------|
| IPv4 | TCP personalizado | TCP | 8080 | 0.0.0.0/0 |
| IPv4 | RDP | TCP | 3389 | 0.0.0.0/0 |
| IPv4 | HTTP | TCP | 80 | 0.0.0.0/0 |
| IPv4 | HTTPS | TCP | 443 | 0.0.0.0/0 |
| IPv4 | SSH | TCP | 22 | 0.0.0.0/0 |
| IPv4 | TCP personalizado | TCP | 3000 | 0.0.0.0/0 |

Una vez la instancia estaba lanzada, se hacía un ssh a la máquina desde Ubuntu y se accedía a esta, se clonaba el repositorio, se cambiaban los puertos al 8080 y las direcciones del Swagger.

Para que el despliegue no parase cuando se cerraba la máquina se ha instalado el módulo PM2.

Validación

Para llevar a cabo la validación del sistema, en frontend se han implementado pruebas de componentes y pruebas de e2e.

Para la prueba de componentes, se ha comprobado que en la pantalla de inicio de la aplicación existan los botones necesarios y que además se puedan clicar.

Por otra parte, la validación con tests e2e ha sido más laboriosa. Se han realizado 3 tests con pruebas e2e.

El primero de ellos es para comprobar el correcto funcionamiento de la aplicación ingresando en ella como administrador.

Este test comprueba primero que al ingresar con el correo y contraseña de un usuario que es el administrador, acceda al inicio del administrador en lugar de al home, al que accedería en caso de ser una cuenta de un usuario normal.

Una vez en el inicio del administrador, comprueba que existen los 4 botones que deben haber y que al hacer click sobre ellos, redirecciona a las páginas necesarias y además realiza las peticiones correspondientes al backend.

Por ejemplo, en la pantalla de los gráficos, se comprueba que se hacen las peticiones correspondientes de algunos de los gráficos que se muestran en la pantalla.

En la pantalla correspondiente a administrar los usuarios, se comprueba que se realiza correctamente la petición para obtener la lista de los usuarios.

En la pantalla de administrar los foros, se comprueba que se realiza exitosamente la petición de la lista de los foros.

Por último, se comprueba que al cerrar la sesión, se redirige a la primera pantalla de la aplicación.

Se ha implementado un test para comprobar el correcto desarrollo del funcionamiento de la aplicación para un usuario que no está registrado.

Se comprueba que se llega a la pantalla de home y que se carga el mapa con las incidencias correspondientes.

También se comprueba que el listado de incidencias se carga exitosamente.

En siguiente lugar, se comprueba que no permite acceder a la pagina correspondiente a “Mis suscripciones” ni a “mis foros” al hacer click sobre estos elementos en la navbar, y que en su lugar redirige al usuario no registrado a una pagina de error que indica que para poder disfrutar de esas funcionalidades debes registrarte. También se comprueba que al pulsar la opción de “Volver” se vuelve a la página del mapa.

En último lugar, se ha implementado un test para comprobar el correcto desarrollo de un usuario normal cuando este inicia sesión ya que está registrado.

Se comprueba lo mismo en la página correspondiente al mapa y al listado del test anterior, pero además se comprueba que sí permite en este caso acceder a “Mis suscripciones” y a “Mis foros” y que se llevan a cabo las peticiones correspondientes de manera correcta.

El código correspondiente a estos tests, se puede encontrar en las primeras 4 secciones del anexo.

Problemas encontrados

Uno de los principales problemas ha sido la necesidad de adaptación al no haber utilizado ciertas tecnologías con anterioridad.

Por ejemplo, a la hora de hacer los tests, ha sido costoso entender por qué muchos de los errores. También aprender a configurar el entorno para poder llevarlos a cabo.

Por otra parte, en frontend ha surgido algún problema con la versión de las librerías utilizadas en los módulos. A mitad de proyecto surgió un problema con la librería de los charts utilizados y no se renderizaba ninguna página que contuviese ningún chart. Además daba error. Esto no fue muy costoso de solucionar en cuanto a aptitudes de los miembros pero sí en cuanto a tiempo ya que no se llegaba a comprender qué ocurría si no se había modificado nada en el proyecto.

Problema con el sistema de coordenadas: Se tuvo un problema a la hora de mostrar las coordenadas en el mapa. Nos dimos cuenta cuando del backend nos llegaba un array de coordenadas que se diferenciaba mucho al array de coordenadas que teníamos al probar la visualización de una coordenada en el frontend estático. Además de darnos cuenta de esto, también tuvimos que buscar cómo hacer para cambiar el sistema de coordenadas, y al final lo conseguimos gracias al módulo proj4, el cuál se ha explicado previamente.

No se mantenía la sesión del usuario: Cuando un usuario se registra o inicia sesión correctamente, en el frontend se guarda el email del usuario, para por ejemplo si se quiere acceder a sus foros, mandarle al backend la petición con este email, para que se obtengan los foros correspondientes. Sin embargo, el email del usuario se pierde si se refresca la página a través del navegador y también se perdía al hacer click sobre un botón que tenía un redirección a otra página. Esto se ha conseguido solucionar eliminando las redirecciones y añadiendo que al hacer click sobre el botón, se ejecute la función navigate del módulo react-router-dom, importado de Internet.

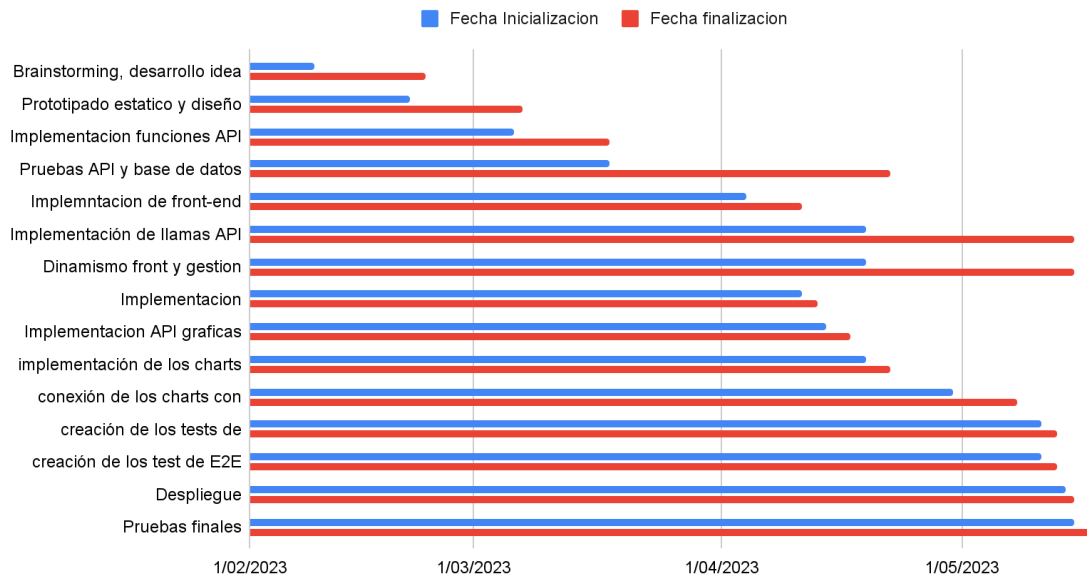
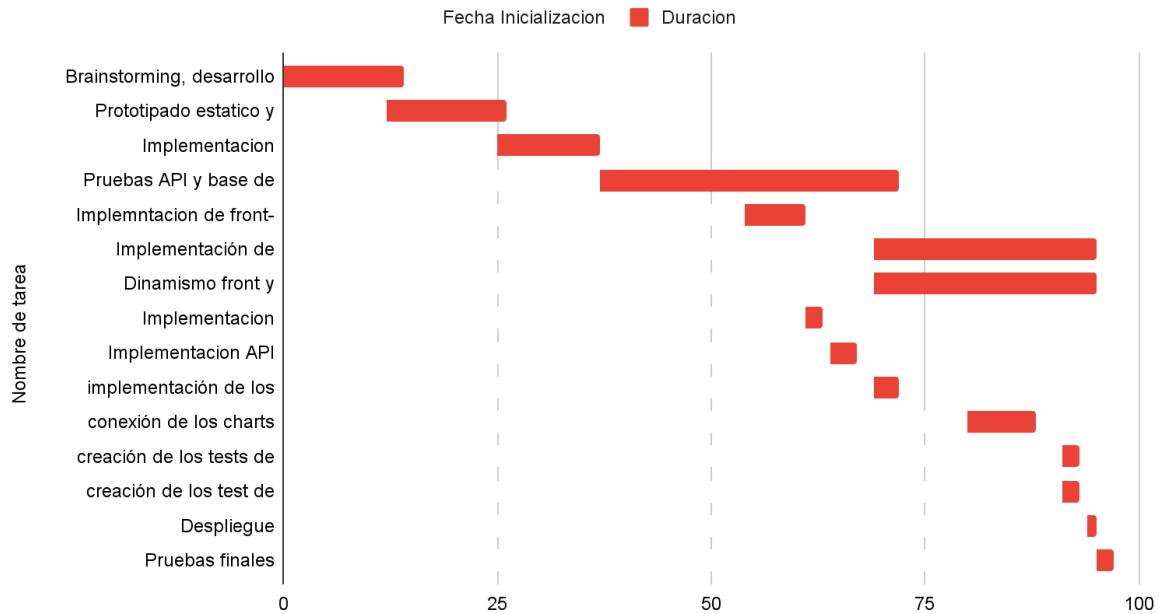
Otro problema encontrado también en el frontend fue a la hora de buscar. Cuando un usuario iba introduciendo datos en la barra de búsqueda de la calle, iban apareciendo las sugerencias. Cuando se seleccionaba una sugerencia, se autocompletaba el campo y se enviaba la petición al backend. El problema fue que el nombre no concordaba, ya que en el back se almacenaba con varios espacios entre las palabras y las calles que se mostraban en el autocompletado poseían únicamente un espacio entre palabras.

En el backend, no hubo demasiados problemas técnicos ya que Oscar y Juliana sabían mucho del tema. El mayor problema que hubo y el causante de que la aplicación no tenga todas las funcionalidades es que el backend se cambió unas cuatro veces entero pues a front no le servían los datos que se pasaban en las respuestas. Esto sucedió las dos últimas semanas antes de acabar el proyecto y el tiempo se nos vino encima.

Distribución del tiempo

| Nombre de tarea | Fecha Inicialización | Fecha finalización |
|---|----------------------|--------------------|
| Brainstorming, desarrollo idea de producto | 9/02/2023 | 23/02/2023 |
| Prototipado estático y diseño inicial API | 21/02/2023 | 7/03/2023 |
| Implementación funciones API y diseño modelo de datos | 6/03/2023 | 18/03/2023 |
| Pruebas API y base de datos | 18/03/2023 | 22/04/2023 |
| Implementación de front-end estático | 4/04/2023 | 11/04/2023 |
| Implementación de llamas API del frontend al backend | 19/04/2023 | 15/05/2023 |
| Dinamismo front y gestión datos recibidos back | 19/04/2023 | 15/05/2023 |
| Implementación funcionalidades API admin | 11/04/2023 | 13/04/2023 |
| Implementacion API gráficas | 14/04/2023 | 17/04/2023 |
| implementación de los charts | 19/04/2023 | 22/04/2023 |
| conexión de los charts con backend | 30/04/2023 | 8/05/2023 |
| creación de los tests de componentes | 11/05/2023 | 13/05/2023 |
| creación de los test de E2E | 11/05/2023 | 13/05/2023 |
| Despliegue | 14/05/2023 | 15/05/2023 |

Fecha Inicializacion y Duracion



Trabajo individual

| Nombre de tarea | Alumnos |
|--|--|
| Brainstorming, desarrollo idea de producto | Paula Ezpeleta Javier Valles Juliana Zordan Oscar Palacin |
| Prototipado estático y diseño inicial API | Paula Ezpeleta Juliana Zordan Oscar Palacin |
| Implementación funciones API y diseño modelo de datos | Juliana Zordan Oscar Palacin |
| Pruebas API y base de datos | Juliana Zordan Oscar Palacin |
| Implementación de front-end estatico y rutas | Paula Ezpeleta |
| Implementación de llamadas API del frontend al backend | Javier Vallés |
| Dinamismo front y gestión datos recibidos back | Javier Vallés |
| Implementación funcionalidades API admin | Oscar Palacin |
| Implementación API gráficas | Oscar Palacin |
| Documentación API | Juliana Zordan |
| implementación de los charts | Paula Ezpeleta |
| conexión de los charts con backend | Paula Ezpeleta |
| creación de los tests de componentes | Paula Ezpeleta |
| creación de los test de E2E | Paula Ezpeleta |
| Despliegue | Paula Ezpeleta Juliana Zordan |
| Pruebas finales | Paula Ezpeleta Juliana Zordan |

Conclusiones

Valoración

Paula Ezpeleta: mi valoración personal del proyecto es que está bastante bien hecho y el equipo se ha esforzado para conseguirlo. He aprendido a utilizar nuevas tecnologías que hasta ahora no sabía que eran necesarias para la creación de una aplicación, como por ejemplo cypress. Y he hecho cosas como el despliegue, que hasta ahora nunca había llevado a cabo ya que en la asignatura de Proyecto Software no fue tarea mía. También considero que al final se nos ha echado el tiempo encima y que con unos días más podríamos haber tenido un mucho mejor producto final, pero por lo general estoy bastante contenta. Es una asignatura que he disfrutado haciendo.

Javier Vallés: Este proyecto ha sido muy didáctico, sin embargo creo que hemos enfocado bastante mal el proyecto. Creo que la parte de frontend únicamente se tendría que encargar de mostrar por pantalla lo que se quiere mostrar (que ya es bastante) y los datos recibidos de backend, además de la conexión con el backend. Sin embargo, hemos tenido que transformar muchos de los datos que se recibían de backend ya que no se pensó demasiado en las APIs que se harían falta en el frontend y se empezó directamente a implementar. En general ha habido bastante poca comunicación entre frontend y backend ya que cada uno iba a lo suyo. A mi me hubiera gustado trabajar en backend ya que nunca he trabajado ahí, pero en frontend he desarrollado principalmente lo que más me gustaba que son las funcionalidades y la conexión con el backend. Me hubiera gustado saber si nos hubiéramos distribuido al revés (cambiar frontend por backend) si mis compañeros que han hecho backend hubieran pensado lo mismo que yo y viceversa.

Oscar Palacin: Se han conseguido los objetivos propuestos en el proyecto y la asignatura. Considero que se escogió un tema muy interesante y se desarrolló un buen prototipo y algunas funcionalidades que pueden ser diferenciales con el resto de aplicaciones similares. Una de las lecciones importantes aprendidas es la importancia de establecer y tener muy machacados los esquemas de datos porque cualquier cambio en estos implica muchas veces rehacer gran parte del código de la API.

Juliana Zordan: Me ha gustado la realización de este proyecto, además me apasiona trabajar en la parte de Backend y he disfrutado mucho realizando el Swagger y el despliegue. Hemos tenido contratiempos pues la Api se ha cambiado demasiadas veces por no compartir objetos claros entre el Frontend y el Backend, por lo que considero que Oscar y yo en Back, hemos acabado agotados de realizar tantos cambios para el Front. Lo que se tuvo que haber hecho es una reunión y haber aclarado todas las peticiones a la api (suerte hemos tenido que la base de datos era Mongo, que si hubiese sido SQL aún no habríamos acabado de hacer tantos cambios). Esto además provocó discusiones en el grupo. Estoy contenta de todas formas, aunque tengo que comentar que al principio se han desarrollado test unitarios, se comenzaron a producir tantos cambios que decidimos quitarlos y probar en Postman varios casos, pues haciendo test unitarios el tiempo dedicado crecía exponencialmente. Me habría gustado cuidar muchos más aspectos de la aplicación y haber entregado una que fuese perfecta, pero por lo mencionado anteriormente, al menos las funcionalidades básicas están.

Anexos

1. Código del test de componente

Unset

```
/*
  Autor: Paula Ezpeleta
  Fecha de creación: Marzo/Abril/Mayo 2023
  Descripción: fichero de la aplicacion ZgzInfo
*/

import React from 'react';
import { BrowserRouter as Router } from 'react-router-dom'; //
Importa el componente Router
import Inicio from './Inicio';

describe('<Inicio />', () => {
  it('renders', () => {
    const setShowNavbar = () => {}; // Define una función simulada
    setShowNavbar
    cy.mount(
      <Router>
        <Inicio setShowNavbar={setShowNavbar} />
      </Router>
    );
    cy.get('.btn-info-1').should('have.length', 2); // existan 2
    botones de esa clase
    cy.get('.continuar-sinregistro').should('have.text',
'Continuar sin registrarse'); // exista ese "link"
    // se comprueba que se puedan clicar los 3
    cy.contains('Iniciar sesión').click();
    cy.contains('Registrarme').click();
    cy.contains('Continuar sin registrarse').click();
  });
});
```

2. Código del test e2e del administrador

Unset

```
/*
  Autor: Paula Ezpeleta
  Fecha de creación: Marzo/Abril/Mayo 2023
  Descripción: fichero de la aplicacion ZgzInfo
*/
describe('e2e desarrollo admin', () => {
  it('passes', () => {
    cy.visit('http://localhost:3001');
    cy.contains('Iniciar sesión').click();
    cy.get('#formGroupExampleInput').type('admin@gmail.com');
    cy.get('#formGroupExampleInput2').type('123456');
    cy.get('.btn-info-2').click();
    // me deberia llevar al inicio del admin pero esta mal
    cy.url().should('include', '/home');
    // aqui ya hemos iniciado sesion y hemos entrado en el
    inicio del admin
    //estamos en la pantalla de home
    // comprobar que puede acceder a los graficos
    cy.contains('Gráficos').click();
    cy.url().should('include', '/admingraficos');
    cy.request('GET',
'/api/admin/grafica/numUsuariosBloqueados')
      .its('body')
      .should('have.length.greaterThan', 0);
```



```
        cy.request('GET',
'/api/admin/grafica/numeroIncidenciasTipo')
            .its('body')
            .should('have.length.greaterThan', 0);
        cy.get('.flecha-admin').click();
        // se presiona la flecha para volver al inicio del admin
        cy.url().should('include', '/home');

        cy.contains('Administrar foros').click();
        cy.url().should('include', '/adminlistaforos');
        cy.request('GET', '/api/admin/listadoForos')
            .its('body')
            .should('have.length.greaterThan', 0);
        cy.get('.flecha-admin').click();
        cy.url().should('include', '/home');

        cy.contains('Administrar usuarios').click();
        cy.url().should('include', '/adminlistausers');
        cy.request('GET', '/api/admin/listadoUsuarios')
            .its('body')
            .should('have.length.greaterThan', 0);
        cy.get('.flecha-admin').click();
        // se presiona la flecha para volver al inicio del admin
        cy.url().should('include', '/home');

        cy.contains('Cerrar sesión').click();
        cy.url().should('include', '/');

    })
})
```

3. Código del test e2e del usuario registrado

Unset

```
/*
  Autor: Paula Ezpeleta
  Fecha de creación: Marzo/Abril/Mayo 2023
  Descripción: fichero de la aplicacion ZgzInfo
*/
describe('e2e desarrollo iniciando sesion', () => {
  it('passes', () => {
    cy.visit('http://localhost:3001');
    cy.contains('Iniciar sesión').click();
    cy.get('#formGroupExampleInput').type('javi23@gmail.com');
    cy.get('#formGroupExampleInput2').type('javi23@gmail.com');
    cy.get('.btn-info-2').click();
    cy.url().should('include', '/home');
    // aqui ya hemos iniciado sesion y hemos entrado en el home
    // estamos en la pantalla de home
    // espera a que se carguen las incidencias
    cy.wait(16000);
    cy.get('.leaflet-marker-icon').should('have.length.gt', 0);
    // espera a que se cargue el mapa
    cy.get('.leaflet-container').should('exist');
    // le doy a incidencias en la navbar

    cy.get('[data-cy="nav-listado-link"]').click()
    cy.url().should('include', '/listado')
    // ahora estoy en la pagina de listado de incidencias
    (numIncidencias > 0)
    cy.request('GET', '/api/incidencias')
      .its('body')
      .should('have.length.greaterThan', 0);
    // ya se ha comprobado que carga el listado de incidencias

    // probar la pagina de mis suscripciones
  })
})
```

```

cy.get('[data-cy="nav-susc-link"]').click()
cy.url().should('include', '/missuscripciones')
cy.request('GET', '/api/grafica/NumUsuariosIncidencia')
  .its('body')
  .should('have.length.greaterThan', 0);
// falta de añadir aqui lo que tene que mirar dentro de
missuscripciones

// probar la pagina de mis foros
cy.get('[data-cy="nav-foros-link"]').click()
cy.url().should('include', '/misforos')
cy.request('GET', '/api/getForosUsuario/javi23@gmail.com')
  .its('body')
  .should('have.length.greaterThan', 0);

cy.get('[data-cy="nav-cerrar-link"]').click()
cy.url().should('include', '/login')

})
})

```

4. Código del test e2e del usuario no registrado

Unset

```

/*
  Autor: Paula Ezpeleta
  Fecha de creación: Marzo/Abril/Mayo 2023
  Descripción: fichero de la aplicacion ZgzInfo
*/

```

```

describe('e2e desarrollo sin registrarse', () => {
  it('passes', () => {
    cy.visit('http://localhost:3001');
    cy.contains('Continuar sin registrarse').click();
    cy.url().should('include', '/home');
    // aqui ya hemos iniciado sesion y hemos entrado en el home
    // estamos en la pantalla de home
    // espera a que se carguen las incidencias
    cy.wait(16000); // para dar tiempo a la respuesta de la api
    cy.get('.leaflet-marker-icon').should('have.length.gt', 0);
    // espera a que se cargue el mapa
    cy.get('.leaflet-container').should('exist');
    // le doy a incidencias en la navbar
    cy.get('[data-cy="nav-listado-link"]').click();
    cy.url().should('include', '/listado');
    // ahora estoy en la pagina de listado de incidencias
    (numIncidencias > 0)
    cy.request('GET', '/api/incidencias')
      .its('body')
      .should('have.length.greaterThan', 0);

    // ya se ha comprobado que carga el listado de incidencias
    // probar que no permite entrar a la pagina de mis
    suscripciones
    cy.get('[data-cy="nav-susc-link"]').click();
    //Comprobamos que si no estas retgistrado sale la panatlla
    correspondiente
    cy.contains('¿Quieres registrarte?');
    // Se comprueba que esté el link de "volver"
    cy.contains('Volver').click();
    // Se comprueba que desde ese link se vuelva al home
    cy.url().should('include', '/home');
    // probar que no permite entrar a la pagina de mis foros
    cy.get('[data-cy="nav-foros-link"]').click();
  });
});

```

```
    //Comprobamos que si no estas retgistrado sale la panatalla  
correspondiente
```

```
    cy.contains('¿Quieres registrarte?');
```

```
    // Se comprueba que esté el link de "volver"
```

```
    cy.contains('Volver').click();
```

```
    // Se comprueba que desde ese link se vuelva al home
```

```
    cy.url().should('include', '/home');
```

```
    })
```

```
  })
```