



UNIVERSIDAD DE MÁLAGA



GRADO EN INGENIERÍA DEL SOFTWARE

Aplicación web para el autoaprendizaje de  
tecnologías

Web application for self-learning technologies

Realizado por  
Javier Ortuño Roig

Tutorizado por  
Eduardo Guzmán de los Riscos

Departamento  
Lenguajes y Ciencias de la Computación  
UNIVERSIDAD DE MÁLAGA

MÁLAGA, JULIO 2022



UNIVERSIDAD  
DE MÁLAGA



ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INFORMÁTICA  
GRADUADO EN INGENIERÍA DEL SOFTWARE

**Aplicación web para el autoaprendizaje de  
tecnologías**

**Web application for self-learning  
technologies**

Realizado por  
**Javier Ortuño Roig**

Tutorizado por  
**Eduardo Guzmán de los  
Riscos**

Departamento  
**Lenguajes y Ciencias de la Computación**

UNIVERSIDAD DE MÁLAGA  
MÁLAGA, JULIO DE 2022

Fecha defensa: julio de 2022

# Abstract

In recent years there has been a growth in the amount of information on software development and along with this, an increase in resources such as books, videos and e-learning platforms to learn this knowledge. However, all this information is distributed all over the internet, making it difficult to find the right learning resource. This project tries to centralize, categorize and rate all this information thanks to the users who use it, thus simplifying the process of searching for resources when learning new knowledge about a technology or block of content. To this end, a web application has been developed with MERN technologies, where the storage is done in a NoSQL database, MongoDB, the front-end is built with the React library and the back-end with the Node.js Express framework.

**Keywords:** MERN, MongoDB, Express.js, React, Node.js

# Resumen

En los últimos años ha habido un crecimiento en la cantidad de información sobre desarrollo software y junto a este, un incremento de recursos como libros, vídeos y plataformas de e-learning para aprender dichos conocimientos. No obstante, toda esta información se encuentra distribuida por todo internet, lo que dificulta encontrar el recurso de aprendizaje adecuado. Este proyecto intenta centralizar, categorizar y puntuar toda esta información gracias a los usuarios que la usen, consiguiendo así simplificar el proceso de búsqueda de recursos a la hora de aprender nuevos conocimientos sobre alguna tecnología o bloque de contenido. Con este fin se ha desarrollado una aplicación web con tecnologías MERN, donde el almacenamiento se realiza en una base de datos NoSQL, MongoDB, el front-end está construido con la librería React y el back-end con el framework Express de Node.js.

**Palabras clave:** MERN, MongoDB, Express.js, React, Node.js



# Índice

<b>1. Introducción</b>	<b>9</b>
1.1. Motivación . . . . .	9
1.2. Objetivos . . . . .	10
1.3. Antecedentes . . . . .	11
1.4. Organización del documento . . . . .	13
<b>2. Tecnologías y herramientas utilizadas</b>	<b>15</b>
2.1. HTML . . . . .	15
2.2. CSS . . . . .	16
2.3. Javascript . . . . .	16
2.4. Node . . . . .	16
2.5. Gherkin . . . . .	17
2.6. Frontend . . . . .	17
2.6.1. React . . . . .	17
2.6.2. React Router . . . . .	18
2.6.3. Redux . . . . .	18
2.6.4. SASS . . . . .	21
2.7. Backend . . . . .	22
2.7.1. Express . . . . .	23
2.7.2. MongoDB . . . . .	24
2.7.3. Mongoose . . . . .	24
2.7.4. Bcryptjs . . . . .	25
2.8. Pruebas . . . . .	26
2.8.1. Jest . . . . .	26
2.8.2. Cypress . . . . .	26
2.9. Otros . . . . .	26
<b>3. Especificación y Análisis</b>	<b>29</b>

3.1. Actores del sistema . . . . .	29
3.2. Requisitos funcionales y no funcionales . . . . .	31
3.2.1. Requisitos funcionales . . . . .	32
3.2.2. Requisitos no funcionales . . . . .	35
3.3. Casos de uso . . . . .	35
3.3.1. Requisito 1 . . . . .	35
3.3.2. Requisito 2 . . . . .	38
3.3.3. Requisito 3 . . . . .	43
3.3.4. Requisito 4 . . . . .	46
3.3.5. Requisito 5 . . . . .	47
3.3.6. Requisito 6 . . . . .	50
3.3.7. Requisito 7 . . . . .	53
3.3.8. Requisito 8 . . . . .	54
3.3.9. Requisito 9 . . . . .	55
3.3.10. Requisito 10 . . . . .	57
3.3.11. Requisito 11 . . . . .	74
3.3.12. Requisito 12 . . . . .	75
3.3.13. Requisito 13 . . . . .	77
3.3.14. Requisito 14 . . . . .	79
3.3.15. Requisito 15 . . . . .	80
3.3.16. Requisito 16 . . . . .	82
<b>4. Diseño del sistema</b>	<b>91</b>
4.1. Diseño arquitectónico de la aplicación . . . . .	91
4.2. Diseño de la Base de Datos . . . . .	93
4.3. Mock-ups . . . . .	97
<b>5. Implementación</b>	<b>111</b>
5.1. Preparación para el proyecto . . . . .	111
5.2. Estructura del proyecto . . . . .	112
5.2.1. Seguridad . . . . .	116
5.3. Metodología de trabajo . . . . .	120

5.4. Diagrama de despliegue . . . . .	121
<b>6. Pruebas</b>	<b>123</b>
6.1. Backend - Pruebas de integración . . . . .	124
6.2. Frontend - Pruebas de aceptación . . . . .	126
<b>7. Conclusiones y Líneas Futuras</b>	<b>129</b>
7.1. Conclusiones . . . . .	129
7.2. Líneas Futuras . . . . .	130
<b>Referencias</b>	<b>133</b>
<b>Apéndice A. Manual de Usuario</b>	<b>137</b>
A.1. Página principal . . . . .	137
A.2. Registrarse . . . . .	137
A.3. Iniciar sesión . . . . .	138
A.4. Roadmaps . . . . .	139
A.5. Navbar para usuarios que han iniciado sesión . . . . .	140
A.6. Bloques de contenido . . . . .	142
A.7. Crear y ver recursos . . . . .	143
A.8. Denunciar . . . . .	146
A.8.1. Perfiles . . . . .	147
A.9. Editar perfil . . . . .	149
A.10. Crear, actualizar y eliminar bloques de contenido . . . . .	152
A.10.1. Crear un bloque contenido . . . . .	153
A.10.2. Actualizar bloque de contenido . . . . .	154
A.10.3. Eliminar bloque de contenido . . . . .	156
<b>Apéndice B. Manual de Instalación</b>	<b>157</b>





# 1

# Introducción

## 1.1. Motivación

Internet ha evolucionado mucho desde su creación, que es, sin embargo, muy reciente desde la perspectiva de la historia, pero por reciente que sea, es innegable el gran impacto que ha tenido en la sociedad.

Cada año salen nuevos productos, servicios o mejoras de alguno ya existente, pero todo esto es para el usuario medio, ¿y qué pasa con los desarrolladores? Para los desarrolladores esta situación es aún más descabellada. Cada año salen lenguajes, framework, librería o servicios nuevos y, de la misma manera, al igual que muchos de estos elementos aparecen, otros muchos se quedan en el olvido. Son tantos los cambios de las tecnologías en el mercado que obligan a las personas cercanas al mundo del software a estar en constante aprendizaje para aprender a usar las librerías y lenguajes más actuales. Como se puede ver en un análisis creado por la plataforma *classcentral* en el 2021, es innegable el crecimiento de cursos online para *classcentral* en el 2021: es innegable el crecimiento de cursos online para aprender estos nuevos conocimientos [1].

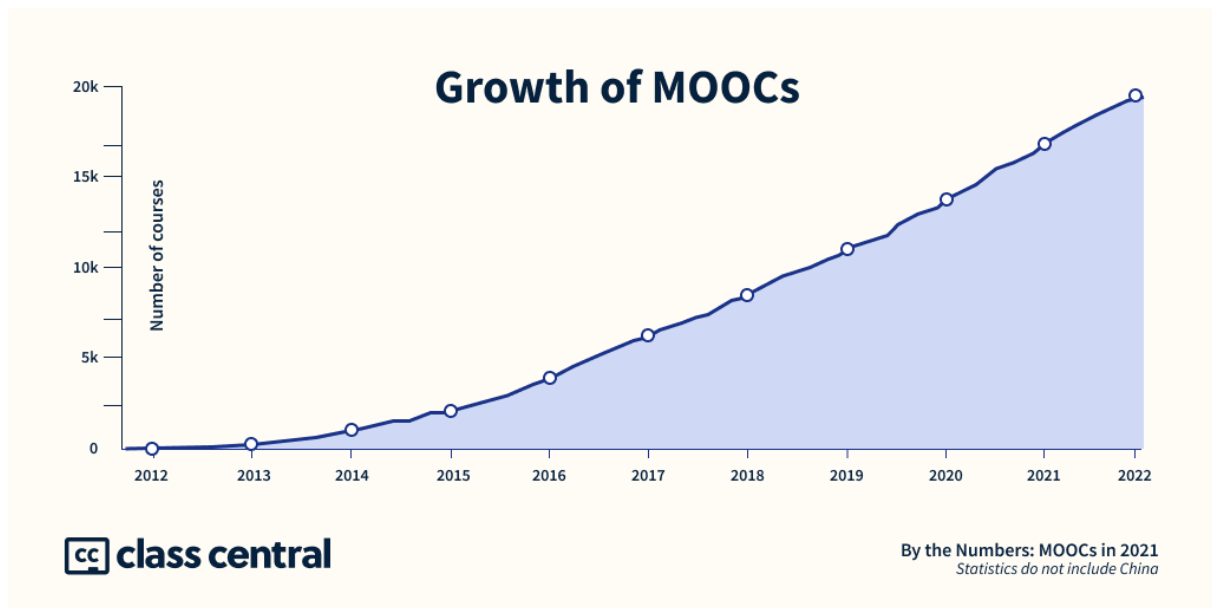


Figura 1: Crecimiento del número de cursos en plataformas MOOCs del 2012 al 2022 [1]

Pero siempre surgen las mismas preguntas: ¿Por dónde empiezo a estudiar una tecnología nueva? ¿Bastará con leer la documentación oficial? ¿Debería visualizar un vídeo de YouTube? ¿Comprar un libro? ¿Comprar un curso en alguna web dedicada a enseñar?

Estas preguntas son recurrentes en desarrolladores que intentan afrontar una nueva tecnología. Ante este problema la única solución es ir al buscador, plataforma de vídeo o de e-learning de confianza e intentar buscar cuáles podrían ser los mejores recursos disponibles en la web para aprender la tecnología deseada. ¿No hay ninguna solución mejor que la búsqueda exhaustiva durante horas?

## 1.2. Objetivos

El objetivo de este proyecto es poner a disposición de la gente un sitio en el que cualquier desarrollador podrá compartir y opinar sobre sus recursos favoritos. Gracias a esto se podrá acortar el tiempo de búsqueda de nuevos recursos para aprender gracias a la ayuda proporcionada por otros desarrolladores que ya hayan pasado por dicha búsqueda y quieran compartir, con otros desarrolladores, los recursos que más les hayan servido.

Esta aplicación estará dirigida a personas autodidactas, poniendo a su disposición recursos de muchas tecnologías entre las cuales elegir la más adecuada para formarse a través de

vídeos y libros compartidos por otros usuarios. Un usuario también podrá compartir sus conocimientos con otros apasionados del autoaprendizaje a través de sus recomendaciones de libros, cursos o tutoriales escritos por él mismo. Cualquiera persona puede crear tutoriales sobre la tecnología que desee y ver qué opinan los demás usuarios a través de likes y dislikes. A su vez podrá compartir sus recursos favoritos con los demás usuarios y ellos podrán valorar qué les ha parecido.

### 1.3. Antecedentes

Uno de los motivos que ha impulsado la realización de este proyecto ha sido la escasez de aplicaciones web que proporcionen esta solución. En internet se pueden encontrar páginas que reúnen herramientas útiles para desarrolladores web; otras recopilan todas las investigaciones que salen en el mundo de la inteligencia artificial, otras reúnen posibles librerías y frameworks existentes, pero después de una búsqueda por internet, solo hay una única página que coincide con la idea de este proyecto, la idea de, a través de los usuarios, recopilar y filtrar los mejores recursos de todo internet para **aprender** una tecnología nueva.

El sitio web mencionado anteriormente es [hackr.io](https://hackr.io). Esta es una aplicación web dirigida principalmente a:

- Recopilar tutoriales sobre lenguajes de programación.
- Recopilar tutoriales sobre librerías.
- Recopilar tutoriales sobre frameworks.
- Encontrar trabajo.

#### **Puntos fuertes**

- ✓ Un sitio web con muchos años y por tanto con un buen desarrollo y gran variedad de funcionalidades.
- ✓ Soporta el idioma de inglés y español.
- ✓ Se les puede poner etiquetas a los recursos, esto permite una mejor agrupación y filtración.

- ✓ Permite filtrar por etiquetas y dificultad.

### **Puntos débiles**

- × Permite iniciar sesión con varios métodos, entre ellos con Google, pero esta forma, a día de hoy, no funciona.
- × Altos tiempos de carga al intentar visualizar algunas páginas de la aplicación.
- × Anuncios en varias partes de la página, además aparece un pop-up cada vez que se entra en alguna página de su dominio, indicando que está instalado un adblock, aunque no haya ningún adblock activado.
- × Para quitar los diversos anuncios de la aplicación hay que “donar”.
- × No permite añadir una descripción a un nuevo recurso que se quiera subir a la aplicación web.
- × No es posible escribir un tutorial propio para compartir con la comunidad. Entendemos como tutorial propio como aquel escrito por una persona para un recurso, por ejemplo, “el usuario Juan escribe en la página un tutorial sobre cómo imprimir texto con colores en pantalla”.
- × Aunque es posible puntuar positivamente un recurso, no es posible darle una mala puntuación.
- × No se puede denunciar un recurso en caso de haber contenido inapropiado.

Dado el tiempo limitado para la realización del proyecto se hace imposible intentar implementar funcionalidades a la par de la competencia. En cambio se ha intentado realizar las características principales para tener una aplicación web a la que, aunque podría añadirse más funcionalidades, se podría decir que no falta ninguna para la idea con la que se diseñó. Dichas ideas son:

- Compartir los recursos favoritos de un desarrollador para aprender una tecnología.
- La posibilidad de filtrar recursos.

- La posibilidad de puntuar recursos, tanto positiva como negativamente.
- Denunciar recursos inapropiados por su contenido.
- Contactar con el administrador para sugerencias, problemas, etc.
- Tener un perfil público y personalizable.

Entonces, ¿en qué se diferencia esta aplicación de la competencia? El principal cambio con otras páginas como `hackr.io`, es poder poner una descripción a los recursos compartidos y poder crear tutoriales propios desde la misma aplicación.

Cuando se comparte un tutorial, libro, serie, restaurante, etc. a otras personas, siempre va acompañado de una explicación, por ejemplo, si le quisieras recomendar un editor de código a un amigo, ¿qué sería lo normal? ¿Decir “Usa Visual Studio Code” o “Usa Visual Studio code, porque es el editor de código más empleado por la comunidad, tiene cientos de miles de extensiones, tiene versión online, ...”?

Siempre que recomendamos a un conocido damos una explicación del porqué de dicha recomendación, por lo tanto consideramos que en una web de este estilo era una característica que no puede faltar.

En cuanto a escribir tutoriales, hay personas que después de aprender mucho descubren sus propios trucos, snippets de código, funciones útiles creadas por ellos mismos y es posible que quieran compartir sus ideas desde la propia aplicación web, por lo tanto, también veo de vital importancia poder escribir tutoriales propios en la página web, y por supuesto, en markdown uno de los lenguajes de marcado más usado por los desarrolladores a la hora de comunicar información en lenguaje natural a otros programadores.

## 1.4. Organización del documento

Esta memoria se encuentra dividida en ocho partes diferenciadas que se lista a continuación junto a una breve descripción de la misma.

### ❖ Capítulo 1: Introducción

Breve introducción sobre la motivación, objetivos y antecedentes del proyecto.

- ❖ Capítulo 2: Tecnologías y herramientas  
Breve explicación a los lenguajes, librerías y herramientas usadas para desarrollar el proyecto.
- ❖ Capítulo 3: Especificación y análisis  
Especificación de los actores, requisitos y casos de uso del proyecto.
- ❖ Capítulo 4: Diseño y arquitectura del sistema  
Breve explicación sobre el diseño de la base de datos, y el diseño arquitectónico y mock-ups del proyecto.
- ❖ Capítulo 5: Implementación  
Breve explicación sobre algunos detalles de la configuración e implementación del proyecto.
- ❖ Capítulo 6: Pruebas  
Breve explicación sobre cómo se han llevado a cabo las pruebas.
- ❖ Capítulo 7: Conclusiones y líneas futuras  
Conclusiones del proyecto y posibles ideas de cambios para el futuro.
- ❖ Capítulo 8: Documentación del proyecto  
Documentación adicional sobre el proyecto, como puede ser el manual de instalación y el manual de usuario.

# 2

## Tecnologías y herramientas utilizadas

Para este proyecto se ha usado principalmente el lenguaje Javascript, HTML, CSS y Gherkin. Partiendo de estos pocos lenguajes se han usado una variedad de librerías para ampliar su potencial al máximo. A continuación, se encuentra una breve explicación de las librerías y frameworks más importantes en el desarrollo de esta aplicación dividida en 3 categorías dependiendo de su temática.

### 2.1. HTML

HTML (son las siglas en inglés de *HyperText Markup Language*) es el lenguaje más importante a la hora de crear una página o aplicación web, sin él no habría páginas web. En un archivo HTML se define el contenido que tendrá una página web, como texto, vídeo, audio, imágenes a través de etiquetas. Con estas etiquetas también se puede definir la estructura del documento con etiquetas como: header, nav, footer, etc. Este lenguaje fue creado por Tim Berners-Lee en 1991 y el W3C (*World Wide Web Consortium*) [2] creó en el año 1993 un estándar para este. A través de los años HTML ha ido evolucionando y actualmente se encuentra en la versión 5 (HTML5) publicada por el W3C en 2008 [3].

HTML se ha convertido en un estándar para los navegadores y páginas web. Este lenguaje solo define el contenido de la página pero no aporta estilo ni interacción con este. Por ello, comúnmente, HTML es usado junto a CSS y Javascript.



## 2.2. CSS

CSS (son las siglas en inglés de Cascading Style Sheets) es un lenguaje de diseño gráfico que aporta estilos a lenguajes de marcado, como por ejemplo HTML. Con CSS es posible crear diseños atractivos para el usuario en una página web. Con el paso de los años su uso se ha ido extendiendo dando lugar a que no sea solamente usado para el desarrollo web sino que puede ser usado para desarrollo de aplicaciones móviles o aplicaciones de escritorio, entre otros [4].

El organismo W3C propuso crear un lenguaje para dar estilos a HTML. Las mejores propuestas fueron CHSS creada por Håkon Wium Lie y SSP creada por Bert Bos. El W3C añadió a Lie y Bos a su plantilla para intentar definir un lenguaje uniendo lo mejor de sus propuestas, y finalmente, a finales de 1996, el W3C publicó “CSS nivel 1”. A lo largo de los años CSS ha ido evolucionando y actualmente se encuentra en la versión CSS3 [5].

## 2.3. Javascript

Javascript es un dialecto de ECMAScript y es un lenguaje de programación interpretado, con tipado débil y dinámico y es multiparadigma, con el que se puede hacer programación funcional, imperativa, orientada a objetos, entre otros [6].

Todos los navegadores modernos soportan Javascript, es decir, es posible usar Javascript en cualquier navegador moderno y dan acceso al DOM (siglas de Document Object Model) para poder modificar el HTML de la página web.

Javascript fue creado en el año 1995 por Brendan Eich bajo el nombre de LiveScript, aunque finalmente acabó llamándose Javascript. Con el paso de los años el lenguaje ha evolucionado mucho, en un principio solo era posible usar Javascript en el lado del cliente (en los navegadores), pero actualmente esto ya no es así. Actualmente es posible ejecutar Javascript en el lado del servidor gracias a los entornos en tiempo de ejecución, esto ha permitido que Javascript pueda usarse no solo para el desarrollo web sino también para crear aplicaciones de escritorio, aplicaciones móviles, aplicaciones de consola, etc [7].

## 2.4. Node

Node es un motor de ejecución para Javascript más famoso y usado hoy en día lanzado en 2009 por Ryan Dahl. Es de código abierto y permite usar Javascript en el lado del servidor,

lo que le da mucha potencia al lenguaje. Está basado en el motor V8 de Google, tiene una arquitectura dirigida a eventos con E/S de datos y es asíncrono [8, 9].

Todo el proyecto, ya sea cliente (Frontend), servidor (Backend) o las pruebas, ha sido escrito en tecnologías basadas en Node.

## 2.5. Gherkin

Gherkin es un DSL (Domain-Specific Languages) que se centra en la creación de casos de pruebas y en un desarrollo BDD (Behavior Driven Development). Es un conjunto de reglas gramaticales originalmente para ser usado por la herramienta Cucumber. Hay numerosas herramientas que han integrado Gherkin para definir casos de prueba en un lenguaje natural. Un ejemplo de esto es el complemento de Cypress usado en este proyecto para la realización de casos de prueba para pruebas de aceptación. Dicho complemento lee las pruebas especificadas en Gherkin y las asocia con las pruebas correspondientes en Cypress [10].

## 2.6. Frontend

Aquí vienen descritas las tecnologías más importantes respecto al desarrollo frontend de la aplicación. Cabe mencionar que hay varias librerías y componentes usadas en el proyecto para la creación del frontend, pero que no van a ser explicadas ya que han tenido poco impacto sobre el proyecto y no tienen la suficiente relevancia como para destacarse. Algunas de estas librerías son: SweetAlert 2, bootstrap, react-dropzone, react-markdown, nextui, ...

### 2.6.1. React

React es una librería de Javascript creada por Meta en 2013. Se centra en el desarrollo de interfaces web basados en componentes y es una de las librerías más famosas de Javascript. Después de la votación del último *State of Javascript*, queda claro que React es la librería más usada para el desarrollo frontend actualmente. [11].

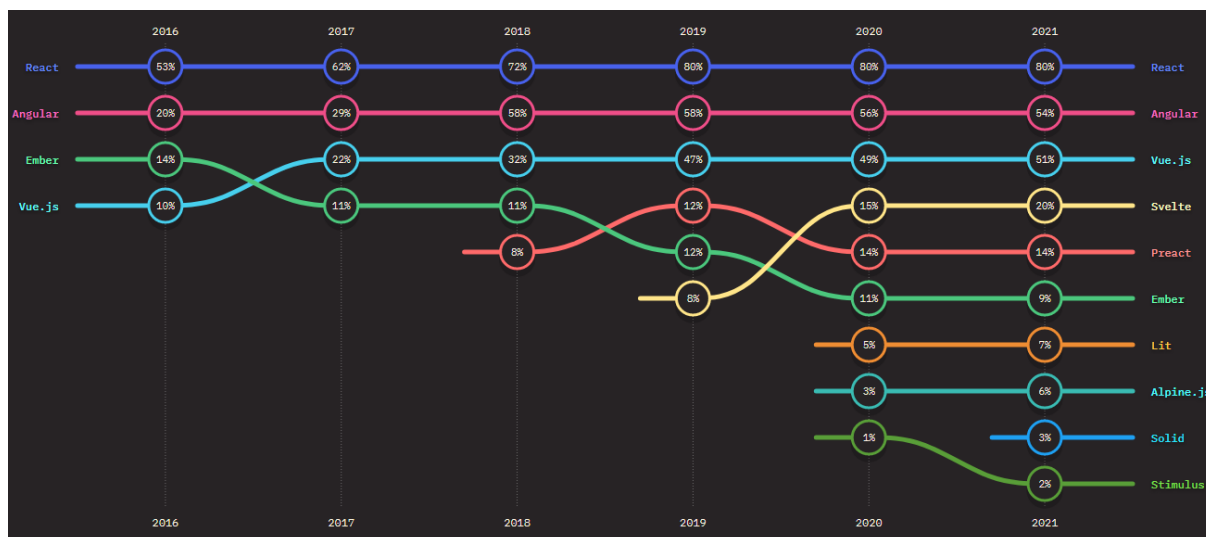


Figura 2: Porcentaje de uso de frameworks de Javascript para el desarrollo frontend [11]

Todo el frontend ha sido desarrollado enteramente en React a través del uso de componentes, hooks personalizados y algunas librerías de apoyo.

### 2.6.2. React Router

Es una de las librerías más famosas para usar junto a React. Es la más usada para hacer routing en una aplicación web, es decir, para poder crear diferentes rutas por las que ir navegando en una aplicación. Dada la forma en la que está creada, con solo usarla en una aplicación de React, se consigue de forma sencilla una SPA (Single Page Application) [12].

Una SPA es un tipo de aplicación web que ejecuta todo sobre una única página, esto se puede conseguir de manera sencilla con React Router, ya que detecta la ruta en la que se encuentra el usuario y en relación a ella desacopla el componente actual que haya en la página y lo sustituye por otro. Con esto se evitan tiempos de cargas molestos para el usuario ya que la página no se recarga constantemente cada vez que se cambia la ruta.

### 2.6.3. Redux

Para mejorar la experiencia de desarrollo se ha usado Redux junto a React. Esta es una librería que permite usar React con un estado global en el sistema. Es decir, hay un punto único y global en el que poder guardar y recuperar información desde cualquier componente. Esto

reduce en gran medida los props (properties) que habría que pasar a los componentes hijos de forma recursiva. Su uso principal en el proyecto es para mantener los datos del usuario conectado de forma global, permitiendo que cualquier componente sea capaz de recuperar dicha información en cualquier momento, sin necesidad de estar pasando sus datos entre componentes [13].

Las properties son los parámetros que le llegan a un componente de React desde su componente padre. Estos son usados para personalizar más el componente hijo. A continuación hay un ejemplo para visualizar cuál es el problema que tenemos con las properties y cuál es la solución que nos proporciona Redux.

```
export const Abuelo = () => {
  const [nombre, setNombre] = useState('Nombre abuelo')
  return <Padre nombreAbuelo={nombre} />
}

// Este componente recibe properties de su padre que no
// necesita y se lo envía a su hijo, que es el que lo necesita
export const Padre = (props) => {
  const [nombre, setNombre] = useState('Nombre padre')
  return <Hijo nombreAbuelo={props.nombreAbuelo} nombrePadre={
    nombre} />
}

// El hijo finalmente recibe las properties de su padre y su
// abuelo
export const Hijo = (props) => {
  const [nombre, setNombre] = useState('Nombre nombre hijo')
  return (
    <>
      <p>Nombre Abuelo: {props.nombreAbuelo}</p>
      <p>Nombre Padre: {props.nombrePadre}</p>
      <p>Nombre Hijo: {nombre}</p>
    </>
  )
}
```

```
)  
}
```

### Algoritmo 1: Componente padre envía properties a componentes hijos

Como se puede ver en el ejemplo anterior, hay ocasiones en la que queremos pasar información a algún componente que no se relaciona de forma directa, por tanto la única solución es pasar la información por componentes intermedios hasta que finalmente llega al componente que lo necesita. Esto se puede simplificar con Redux. A continuación se puede comprobar cómo quedaría el mismo ejemplo si se usara Redux.

```
export const Abuelo = () => {  
  
  const [nombre, setNombre] = useState('Nombre abuelo')  
  const dispatch = useDispatch();  
  dispatch( addNameToStorageAction( nombre ) ); // Por  
  simplicidad no se explicará lo que es una acción en Redux  
  
  return <Padre />  
}  
  
export const Padre = () => {  
  
  const [nombre, setNombre] = useState('Nombre padre')  
  const dispatch = useDispatch();  
  dispatch( addNameToStorageAction( nombre ) ); // Por  
  simplicidad no se explicará lo que es una acción en Redux  
  
  return <Hijo />  
}  
  
export const Hijo = () => {
```

```

const [nombre, setNombre] = useState('Nombre nombre hijo')
const { nombreAbuelo, nombrePadre } = useSelector( state =>
  state.nombres );

return (
  <>
    <p>Nombre Abuelo: {nombreAbuelo}</p>
    <p>Nombre Padre: {nombrePadre}</p>
    <p>Nombre Hijo: {nombre}</p>
  </>
)
}

```

Algoritmo 2: Componente padre envía properties a componentes hijos con Redux

Como podemos comprobar con el ejemplo de arriba, una vez usado Redux nos hemos ahorrado mucho código que simplemente servía para pasar información. En este caso era un ejemplo sencillo, pero en una gran aplicación el uso de Redux es mucho más notable. Gracias al uso de esta librería la información del usuario es accesible desde cualquier punto de la aplicación y no es necesario pasar los datos entre los muchos componentes del proyecto.

#### 2.6.4. SASS

Los estilos de la aplicación han sido dados con SASS. Este es un preprocesador CSS que amplía las funcionalidades del lenguaje. Le permite usar variables (un poco distintas a las, ahora existentes, en CSS), reglas anidadas, mixins, funciones y más, todo con una sintaxis totalmente compatible con CSS [14].

A continuación hay un ejemplo de “nesting” de SASS, dado que es la funcionalidad más usada en el proyecto. El “nesting” permite anidar selectores css, lo que permite tener un código más claro, abajo hay un ejemplo de código CSS y ese mismo código en SASS.

```

.contenedor {
  color: red;
}

```

```

.contenedor > .cuadrado {
  color: blue;
}

.contenedor > .cuadrado > .triangulo {
  color: green;
}

```

Algoritmo 3: Ejemplo código CSS

```

.contenedor {
  color: red;

  > .cuadrado {
    color: blue;

    > .triangulo {
      color: green;
    }
  }
}

```

Algoritmo 4: Ejemplo código SASS con nesting

En el segundo ejemplo, con el uso del nesting, queda un archivo con menos código y queda mucho más clara la jerarquía de los selectores.

## 2.7. Backend

Aquí vienen descritas las tecnologías más importantes respecto al desarrollo backend de la aplicación. Cabe mencionar que hay varias librerías usadas en el proyecto para la creación del backend, pero que no van a ser explicadas ya que han tenido poco impacto sobre el proyecto y no tienen la suficiente relevancia como para destacarse. Algunas de estas librerías son: dotenv,

cloudinary, express-validator, etc.

### 2.7.1. Express

Express es un framework Open-Source de Node creado en 2010 por T.J. Holowaychuk. Es el framework más usado para el desarrollo de aplicaciones web y creación de APIs como ha demostrado el *State of JS* del último año.

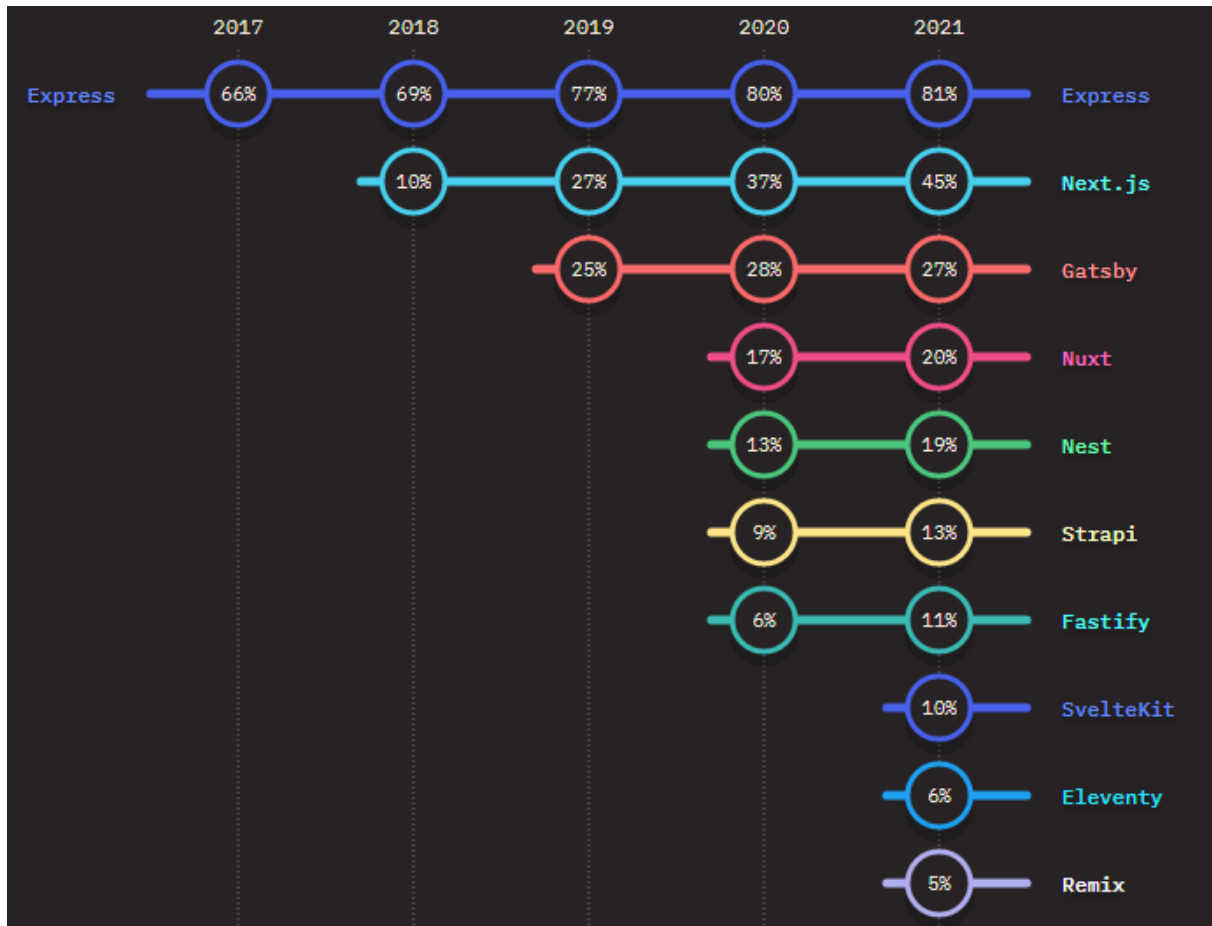


Figura 3: Porcentaje de uso de frameworks de Javascript para el desarrollo backend [11]

Todo el backend ha sido escrito enteramente en Express haciendo uso de diversas rutas creadas con este framework, además de varios middlewares, tanto propios como de terceros, para añadir funcionalidades a la API REST [15].



### 2.7.2. MongoDB

MongoDB es una de las bases de datos más populares hoy en día, en especial para el desarrollo web. Es una base de datos de código abierto, no relacional y orientada a documentos, que guarda toda la información en formato BSON (Binary JSON) [16].

BSON es una variante de JSON, que usa MongoDB gracias al aumento de velocidad y eficiencia a la hora de guardar los datos, además aumenta el número de tipos de datos que es posible usar [17].

### 2.7.3. Mongoose

Mongoose es una librería de Javascript utilizada como ODM (mapeo objeto-documento) de MongoDB. El uso de esta librería facilita la creación de Schemas (modelos de los objetos que guardaremos) y los métodos para interactuar con la base de datos [18].

Un ODM (Object Document Mapping) sirve para relacionar un modelo de objetos a una colección de la base de datos, es decir, podemos crear modelos de nuestros documentos para que todos los documentos de la base de datos relacionados a un modelo tengan la misma estructura. [19]

Si usaremos la librería de MongoDB podríamos insertar en una colección de usuarios objetos que no son correctos, pero que la base de datos no controla.

```
const user = { name: "John", password: "helloworld" };
const userImposter = { whatever: "doe", password: "okok" };

// Insertar un usuario con los atributos correctos
dbo.collection("users").insertOne(user, function (error, result) {
  if (error) throw error;
  console.log(result);
});

// Insertar un usuario con los atributos incorrectos
dbo.collection("users").insertOne(userImposter, function (error
```

```

    , result) {
  if (error) throw error;
  console.log(result);
});

```

Algoritmo 5: Ejemplo insertar dos objetos distintos en una misma colección

Como se puede ver en el ejemplo de arriba se está añadiendo a la colección dos usuarios con distintos atributos. Usando Mongoose esto se arreglaría al poder crear un modelo para la colección de usuario. Un posible modelo para la colección de usuarios podría ser el siguiente.

```

const User = mongoose.model("User", {
  username: { type: String },
  password: { type: String },
});

```

Algoritmo 6: Modelo para la colección de usuario

#### 2.7.4. Bcryptjs

Bcryptjs es tanto una librería de Javascript como una técnica de encriptación. A continuación se procede a dar una pequeña introducción a cada una de ellas [20].

- Bcrypt fue creada en 1999 por Niels Provos como una función de hash de contraseñas. Esta técnica tiene varias cosas positivas:
  - Usa salt junto al hash, lo que proporciona más seguridad y protección contra ataques de rainbow (tabla arcoiris).
  - Se pueden ajustar las rondas de salt que queremos hacer sobre la contraseña. Cuantas menos rondas se hagan el proceso se hará más rápido, pero la seguridad será menor. Cuanto más alto sea, más lento irá, pero mayor será la seguridad.
- Hay librerías para usar esta técnica de encriptación en varios lenguajes. En el caso de este proyecto se ha usado la librería *bcrypt* de NPM. En la sección de implementación aparecerá un trozo de código de las funciones usadas de la librería para encriptar las contraseñas.

## 2.8. Pruebas

### 2.8.1. Jest

Jest es uno frameworks de Javascript más famosos para el desarrollo de pruebas. Tiene un reporte de pruebas con colores, dejando claras las pruebas pasadas y las fallidas, función de cobertura (coverage) para saber qué partes del código han sido cubiertas y cuáles no, extensión oficial para Visual Studio Code y otras muchas características [21, 22].

El desarrollo de Jest fue iniciado por Meta en 2011 para probar su código al reescribir la funcionalidad de su chat de Facebook con Javascript. En 2014 se convirtió en un proyecto Open Source. Años más tarde, meta abandonó el desarrollo de Jest y este fue llevado solo por Christoph Nakazawa, ex-ingeniero de Meta. Actualmente el mantenimiento de la misma es llevado a cabo gracias a Open collective, un sitio web donde se pueden hacer donaciones para proyecto Open Source [23, 24].

### 2.8.2. Cypress

Es un framework open-source muy usado para la realización de pruebas UAT (User Acceptance Testing) o E2E (End to End), aunque recientemente, en Junio de 2022, ha añadido la posibilidad de hacer test sobre componentes, sin embargo, por ahora está en la versión beta [25]. Cypress dispone al usuario una serie de métodos para realizar las pruebas de forma fácil e intuitiva. Además dispone de una interfaz gráfica muy amigable para el usuario y que permite identificar de forma rápida y clara qué prueba ha fallado y dónde, también tiene un log sobre fallo muy claro y con consejos para solucionarlo [26].

## 2.9. Otros

Por último, cabe destacar un paquete de npm que se ha usado durante todo el desarrollo de la aplicación. Dicha librería es ESLint, este es uno de los linter más famosos de Javascript para mantener siempre un estilo de código firme y constante a través de una serie de reglas predefinidas pero personalizables. ESLint no sólo avisa al desarrollador de que hay un error estilográfico en su código, hay muchos errores que es capaz de arreglar la propia librería con solo un comando, un ejemplo de esto sería no poner un punto y como al final de una línea,

ESLint es capaz de ponerlo por el usuario [27].



# 3

## Especificación y Análisis

### 3.1. Actores del sistema

Averiguar cuáles son los actores de un proyecto es de los primeros pasos a la hora de empezar con una nueva aplicación. Los actores son todas las personas que formen parte o se encuentren en algún grupo, organización, etc, y que tengan algún tipo de relación con el proyecto, ya sea una relación directa o una indirecta.

Hay 4 actores diferenciables en el sistema: invitado, usuario, moderador y administrador. A continuación se procede a explicar cómo interactúa cada uno con el sistema.

#### ▪ Invitado

- Son los usuarios que usan la aplicación sin haber iniciado sesión previamente.
- Estos actores pueden ver todo tipo de contenido relativo a bloques de contenidos y podrá visualizar la información de los perfiles de otros usuarios.

#### ▪ Usuario

- Son los usuarios que usan la aplicación habiendo iniciado sesión.
- Este actor puede hacer todo lo que un invitado puede y no tiene permiso alguno en la aplicación. Además puede editar su información de perfil, añadir nuevos recursos a los bloques de contenidos existentes y denunciar, si es necesario, un recurso.

#### ▪ Moderador

- Son los usuarios que usan la aplicación habiendo iniciado sesión y tienen rol de moderador. Estos usuarios tienen ciertos permisos en el sistema. Para ser moderador primero ha de ser usuario.
- Este actor puede hacer todo lo que un usuario puede. Además, es capaz de mirar los recursos con denuncias y los motivos de estas, eliminar recursos de los bloques de contenidos y dar o quitar a un usuario permisos de moderador.

#### ▪ **Administrador**

- Son los usuarios que usan la aplicación habiendo iniciado sesión y tienen rol de administrador. Solo hay un administrador, siendo este el mismo creador de la propia aplicación web y tiene todos los permisos sobre el sistema.
- Este actor puede hacer todo lo que un moderador puede. Además puede crear, editar y eliminar bloques de contenido de la aplicación web.

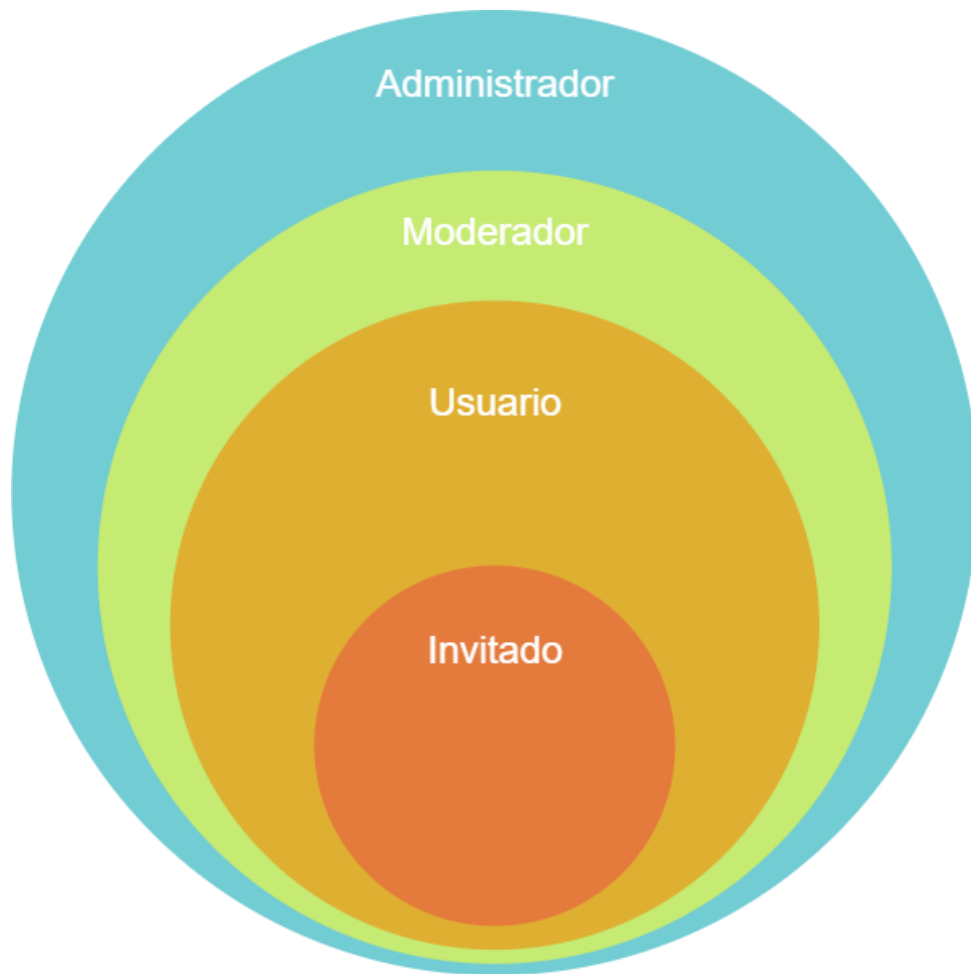


Figura 4: Diagrama de Venn que muestra la cantidad de permisos que tiene cada rol

### 3.2. Requisitos funcionales y no funcionales

Aparte de conocer las personas o partes implicadas con el proyecto o la aplicación, nos hace falta saber las características que deben cumplir la misma. Dichas especificaciones es lo que se conoce como los *requisitos del sistema*. Los requisitos son una especificación de las características que ha de tener la aplicación que se va a desarrollar. Hay dos tipos de requisitos los *requisitos funcionales* y los *requisitos no funcionales*.

Los requisitos funcionales son aquellas características o funcionalidades que quiere que posea la aplicación. Los requisitos no funcionales son características de calidad como, por ejemplo, sobre tiempo de cargas o el uso de cierta tecnología.



### 3.2.1. Requisitos funcionales

RF 1. Un invitado debe poder acceder a la página sin iniciar sesión.

RF 1. 1. Un invitado debe poder visualizar cualquier recurso de la página.

RF 1. 2. Un invitado debe poder visualizar cualquier bloque de contenido de la página.

RF 1. 3. Un invitado debe poder compartir el enlace del recurso mediante la url.

RF 1. 4. Un invitado debe poder visualizar los perfiles de los usuarios.

RF 1. 5. Un invitado debe poder compartir el enlace del perfil de usuario mediante la url.

RF 1. 6. Un invitado debe poder visualizar todos los roadmaps de la página.

RF 2. Un invitado debe poder registrarse en la aplicación.

RF 3. Un invitado debe poder iniciar sesión en la aplicación.

RF 3.1. Un invitado debe poder iniciar sesión en la aplicación como usuario normal.

RF 3.2. Un invitado debe poder iniciar sesión en la aplicación como moderador.

RF 3.3. Un invitado debe poder iniciar sesión en la aplicación como administrador.

RF 4. Un invitado debe poder filtrar los bloques de contenido por nombre.

RF 5. Un invitado debe poder ordenar los bloques de contenido.

RF 5.1. Un invitado debe poder ordenar los bloques de contenido por orden alfabético ascendente.

RF 5.2. Un invitado debe poder ordenar los bloques de contenido por orden alfabético descendente.

RF 5.3. Un invitado debe poder ordenar los bloques de contenido por número de recursos ascendente.

RF 5.4. Un invitado debe poder ordenar los bloques de contenido por número de recursos descendente.

RF 6. Un invitado debe poder ordenar los recursos de un bloque de contenido.

- RF 6.1. Un invitado debe poder ordenar los recursos de un bloque de contenido por fecha ascendente.
- RF 6.2. Un invitado debe poder ordenar los recursos de un bloque de contenido por fecha descendente.
- RF 6.3. Un invitado debe poder ordenar los recursos de un bloque de contenido por likes ascendente.
- RF 6.4. Un invitado debe poder ordenar los recursos de un bloque de contenido por likes descendente.
  
- RF 7. Un invitado debe poder acceder al perfil del usuario creador de un recurso.
  
- RF 8. Un usuario debe poder contactar con los moderadores y el administrador, a través de un correo.
  
- RF 9. Un usuario debe poder cambiar los datos de su perfil.
  - RF 9.1. Un usuario debe poder cambiarse la foto de perfil.
  - RF 9.2. Un usuario debe poder cambiarse la descripción de su perfil.
  
- RF 10. Un usuario debe poder manejar recursos en un bloque de contenido.
  - RF 10.1. Un usuario debe poder crear recursos
    - RF 10.1.1. Un usuario debe poder crear recursos con libros en un bloque de contenido.
    - RF 10.1.2. Un usuario debe poder crear recursos con vídeos de YouTube en un bloque de contenido.
    - RF 10.1.3. Un usuario debe poder crear recursos con tutoriales en un bloque de contenido.
      - RF 10.1.3.1. Un usuario debe poder crear **tutoriales propios** en un bloque de contenido.
      - RF 10.1.3.2. Un usuario debe poder crear **tutoriales externos** en un bloque de contenido.
  - RF 10.2. Un usuario debe poder borrar sus recursos creados

RF 11. Un usuario debe poder puntuar un recurso.

RF 11.1. Un usuario debe poder dar like a un recurso subido por otro usuario.

RF 11.2. Un usuario no debe poder dar like a un recurso subido por el mismo.

RF 11.3. Un usuario debe poder dar dislike a un recurso subido por otro usuario.

RF 11.4. Un usuario no debe poder dar dislike a un recurso subido por el mismo.

RF 12. Un usuario debe poder denunciar.

RF 12.1. Un usuario debe poder denunciar un recurso.

RF 12.2. Un usuario debe indicar el motivo de la denuncia.

RF 13. Un moderador o administrador debe poder cambiar los permisos de los usuarios y moderadores.

RF 13.1. Un moderador o administrador debe poder convertir a un usuario sin permisos en moderador.

RF 13.2. Un moderador o administrador debe poder convertir a un moderador en usuario sin permisos.

RF 14. Un moderador o administrador debe poder eliminar un recurso.

RF 15. Un moderador o administrador debe poder interactuar con las denuncias.

RF 15.1. Un moderador o administrador debe poder ver los recursos denunciados.

RF 15.2. Un moderador o administrador debe poder ver el número de denuncias de un recurso

RF 16. Un administrador debe poder manejar los bloques de contenido de la aplicación.

RF 16.1. Un administrador debe poder crear un nuevo bloque de contenido.

RF 16.2. Un administrador debe poder actualizar un bloque de contenido.

RF 16.3. Un administrador debe poder eliminar un bloque de contenido.

### 3.2.2. Requisitos no funcionales

RNF 1.1. La base de datos será no relacional.

RNF 1.2. Las contraseñas estarán guardadas de forma segura.

### 3.3. Casos de uso

Dado que tanto los actores como los requisitos ya han sido escritos, ahora es posible especificar los casos de uso. Un caso de uso es la descripción escrita de una acción. Los casos de uso van a estar organizados de la siguiente manera:

- Tabla con información básica sobre el caso de uso.
- Escenario principal.
- Escenarios secundarios, si procede.
- Casos de pruebas descritos en gherkin.
- Mock-up de la interfaz de usuario requerida para realizar el caso de uso.

#### 3.3.1. Requisito 1

Título	Los datos públicos serán visibles para cualquiera
Descripción	Un invitado debe poder acceder a la aplicación sin iniciar sesión y deberá de poder acceder a las páginas públicas de la aplicación como recursos, bloques de contenidos, perfiles de otros usuarios y roadmaps
Pre-condiciones	No procede
Post-condiciones	No procede
Prioridad	Alta

#### Escenarios principales

Escenario 1. Visualizar bloques de contenido.

1.1. El invitado entra en la página principal.

- 1.2. El sistema muestra la página principal.
- 1.3. El invitado visualiza los bloques de contenido.

Escenario 2. Visualizar recursos de un bloques de contenido.

- 2.1. El invitado entra en la página principal.
- 2.2. El sistema muestra la página principal.
- 2.3. El invitado hace clic en un bloque de contenido.
- 2.4. El sistema muestra la información del bloque de contenido.
- 2.5. El invitado visualiza los recursos de ese bloque de contenido.

Escenario 3. Visualizar perfiles de usuarios.

- 3.1. Entro clic en la página principal.
- 3.2. El sistema muestra la página principal.
- 3.3. El invitado hace clic en un bloque de contenido.
- 3.4. El sistema muestra la información del bloque de contenido.
- 3.5. El invitado hace clic en un recurso.
- 3.6. El sistema muestra la información del recurso.
- 3.7. El invitado hace clic en el perfil del creador del recurso.
- 3.8. El sistema muestra la información del perfil del usuario.
- 3.9. El invitado visualiza el perfil del usuario.

Escenario 4. Visualizar roadmaps

- 4.1. El usuario entra en la página principal.
- 4.2. El sistema muestra la página principal.
- 4.3. El invitado hace clic en “Road Maps” en el navbar.
- 4.4. El sistema muestra los roadmaps.
- 4.5. El invitado visualiza los distintos roadmaps.

## Escenarios de prueba

**Feature:** Un invitado debe poder acceder a la página sin iniciar sesión y visualizar datos

**Scenario:** Visualizar bloques de contenido

**Given** El invitado entra en la página principal

**Then** El invitado visualiza los bloques de contenido

**Scenario:** Visualizar recursos de un bloques de contenido

**Given** El invitado entra en la página principal

**When** El invitado hace clic en un bloque de contenido

**Then** El invitado visualiza los recursos de ese bloque de contenido

**Scenario:** Visualizar perfiles de usuarios

**Given** El invitado entra en la página principal

**When** El invitado hace clic en un bloque de contenido

**And** El invitado hace clic en un recurso

**And** El invitado hace clic en el perfil del creador del recurso

**Then** El invitado visualiza el perfil del usuario

**Scenario:** Visualizar roadmaps

**Given** El invitado entra en la página principal

**When** El invitado hace clic en roadmaps en el navbar

**Then** El invitado visualiza los distintos roadmaps

### 3.3.2. Requisito 2

Título	Crear una cuenta
Descripción	Un invitado debe poder registrarse con su nombre, correo y contraseña
Pre-condiciones	No procede
Post-condiciones	Se crea una cuenta para el invitado
Prioridad	Alta

#### Escenario principal

Escenario 1. Crear una nueva cuenta.

1. El invitado entra en la página principal.
2. El sistema muestra la página principal.
3. El invitado hace clic en “Registrarse” en el navbar.
4. El sistema muestra la página de registro.
5. El invitado introduce su nombre, correo, contraseña y confirmar contraseña.
6. El invitado hace clic en “Registrarse”.
7. El sistema muestra un pop-up de tipo “success” con el mensaje “Usuario creado satisfactoriamente”.

#### Escenarios secundarios

Alternativo 1. Fallo al crear una nueva cuenta con un nombre en uso.

- 5.a.1 El invitado introduce un nombre en uso.
- 5.a.2 El invitado hace clic en “Registrarse”.
- 5.a.3 El sistema muestra un pop-up de tipo “error” con el mensaje “Ya existe un usuario con ese nombre”.

Alternativo 2. Fallo al crear una nueva cuenta con un email en uso.

- 5.b.1. El invitado introduce un nombre y un email en uso.
- 5.b.2. El invitado hace clic en “Registrarse”.
- 5.b.3. El sistema muestra un pop-up de tipo “error” con el mensaje “Ya existe un usuario con ese email”.

Alternativo 3. Fallo al crear una nueva cuenta con un email no válido.

- 5.c.1. El invitado introduce un nombre y un email no válido.
- 5.c.2. El invitado hace clic en “Registrarse”.
- 5.c.3. El sistema muestra un pop-up de tipo “error” con el mensaje “Email no válido”.

Alternativo 4. Fallo al crear una nueva cuenta con un campo vacío.

- 5.d.1. El invitado no introduce datos.
- 5.d.2. El invitado hace clic en “Registrarse”.
- 5.d.3. El sistema muestra un pop-up de tipo “error” con el mensaje “El nombre es obligatorio”.

Alternativo 5. Fallo al crear una nueva cuenta con contraseña pero sin confirmar contraseña.

- 5.e.1. El invitado introduce un nombre, email y contraseña.
- 5.e.2. El invitado hace clic en “Registrarse”.
- 5.e.3. El sistema muestra un pop-up de tipo “error” con el mensaje “La confirmación de la contraseña es obligatoria”.

Alternativo 6. Fallo al crear una nueva cuenta con contraseñas y confirmar contraseña no coincidente.

- 5.f.1. El invitado introduce un nombre, email, contraseña y confirmar contraseña no coincidente.
- 5.f.2. El invitado hace clic en “Registrarse”.
- 5.f.3. El sistema muestra un pop-up de tipo “error” con el mensaje “Las contraseñas no coinciden”.



Alternativo 7. Fallo al crear una nueva cuenta con contraseñas y confirmar contraseña no coincidente.

5.h.1. El invitado introduce un nombre, email y contraseña con menos de 6 dígitos.

5.h.2. El invitado hace clic en “Registrarse”.

5.h.3. El sistema muestra un pop-up de tipo “error” con el mensaje “La contraseña debe ser de 6 caracteres como mínimo”.

### Escenarios de prueba

**Feature:** Un invitado debe poder registrarse en la aplicación

**Scenario:** Crear una nueva cuenta

**Given** El invitado entra en la página principal

**When** El invitado hace clic en registrarse en el navbar

**And** El invitado introduce su “nombre”, “correo”, “contraseña” y “confirmar contraseña”

**And** El invitado hace clic en registrarse

**Then** El sistema muestra un pop-up de tipo “success” con el mensaje “Usuario creado satisfactoriamente”

**Scenario:** Fallo al crear una nueva cuenta con un nombre en uso

**Given** El invitado entra en la página principal

**When** El invitado hace clic en registrarse en el navbar

**And** El invitado introduce su “nombre”, “correo”, “contraseña” y “confirmar contraseña”

**And** El invitado hace clic en registrarse

**Then** El sistema muestra un pop-up de tipo “error” con el mensaje “Ya existe un usuario con ese nombre”

**Scenario:** Fallo al crear una nueva cuenta con un email en uso

**Given** El invitado entra en la página principal  
**When** El invitado hace clic en registrarse en el navbar  
**And** El invitado introduce su "nombre", "correo", "contraseña" y "confirmar contraseña"  
**And** El invitado hace clic en registrarse  
**Then** El sistema muestra un pop-up de tipo "error" con el mensaje "Ya existe un usuario con ese email"

**Scenario:** Fallo al crear una nueva cuenta con un email no válido

**Given** El invitado entra en la página principal  
**When** El invitado hace clic en registrarse en el navbar  
**And** El invitado introduce su "nombre", "correo", "contraseña" y "confirmar contraseña"  
**And** El invitado hace clic en registrarse  
**Then** El sistema muestra un pop-up de tipo "error" con el mensaje "Email no válido"

**Scenario:** Fallo al crear una nueva cuenta con un campo vacío

**Given** El invitado entra en la página principal  
**When** El invitado hace clic en registrarse en el navbar  
**And** El invitado introduce su "nombre", "correo", "contraseña" y "confirmar contraseña"  
**And** El invitado hace clic en registrarse  
**Then** El sistema muestra un pop-up de tipo "error" con el mensaje "El nombre es obligatorio"

**Scenario:** Fallo al crear una nueva cuenta con contraseña pero sin confirmar contraseña

**Given** El invitado entra en la página principal  
**When** El invitado hace clic en registrarse en el navbar

And El invitado introduce su "nombre", "correo", "contraseña" y "confirmar contraseña"

And El invitado hace clic en registrarse

Then El sistema muestra un pop-up de tipo "error" con el mensaje "La confirmación de la contraseña es obligatoria"

Scenario: Fallo al crear una nueva cuenta con contraseñas y confirmar contraseña no coincidente

Given El invitado entra en la página principal

When El invitado hace clic en registrarse en el navbar

And El invitado introduce su "nombre", "correo", "contraseña" y "confirmar contraseña"

And El invitado hace clic en registrarse

Then El sistema muestra un pop-up de tipo "error" con el mensaje "Las contraseñas no coinciden"

Scenario: Fallo al crear una nueva cuenta con contraseñas menor de 6 dígitos

Given El invitado entra en la página principal

When El invitado hace clic en registrarse en el navbar

And El invitado introduce su "nombre", "correo", "contraseña con menos de 6 dígitos" y "confirmar contraseña"

And El invitado hace clic en registrarse

Then El sistema muestra un pop-up de tipo "error" con el mensaje "La contraseña debe ser de 6 caracteres como mínimo"

### 3.3.3. Requisito 3

Título	Iniciar sesión en la aplicación web
Descripción	Un invitado debe poder iniciar sesión en la aplicación
Pre-condiciones	El invitado debe de haber creado una cuenta con anterioridad
Post-condiciones	El invitado ha iniciado sesión y es redirigido a la pantalla principal
Prioridad	Alta

#### Escenario principal

Escenario 1. Iniciar sesión con un usuario existente.

1. El invitado entra en la página principal.
2. El sistema muestra la página principal.
3. El invitado hace clic en “Iniciar sesión” en el navbar.
4. El sistema muestra la página de inicio de sesión.
5. El invitado introduce su nombre o correo y su contraseña.
6. El invitado hace clic en “Iniciar sesión”.
7. El sistema muestra un pop-up de tipo “success” con el mensaje “Inicio de sesión exitoso”.

#### Escenarios secundarios

Alternativo 1. Fallo al iniciar sesión sin escribir nombre.

- 5.a.1 El invitado introduce un nombre vacío y contraseña.
- 5.a.2 El invitado hace clic en “Iniciar sesión”.
- 5.a.3 El sistema muestra un pop-up de tipo “error” con el mensaje “El nombre es obligatorio”.

Alternativo 2. Fallo al iniciar sesión sin escribir contraseña.

- 5.b.1 El invitado introduce un nombre y contraseña vacía.

5.b.2 El invitado hace clic en “Iniciar sesión”.

5.b.3 El sistema muestra un pop-up de tipo “error” con el mensaje “La contraseña es obligatoria”.

Alternativo 3. Fallo al iniciar sesión con nombre inexistente.

5.c.1 El invitado introduce un nombre inexistente y contraseña.

5.c.2 El invitado hace clic en “Iniciar sesión”.

5.c.3 El sistema muestra un pop-up de tipo “error” con el mensaje “El usuario con ese nombre no existe”.

Alternativo 4. Fallo al iniciar sesión con contraseña incorrecta.

5.d.1 El invitado introduce un nombre y contraseña incorrecta.

5.d.2 El invitado hace clic en “Iniciar sesión”.

5.d.3 El sistema muestra un pop-up de tipo “error” con el mensaje “La contraseña es incorrecta”.

### Escenarios de prueba

**Feature:** Un invitado debe poder registrarse en la aplicación

**Scenario:** Un invitado debe poder iniciar sesión en la aplicación

**Given** El invitado entra en la página principal

**When** El invitado hace clic en iniciar sesión en el navbar

**And** El invitado introduce su “nombre” y “contraseña”

**And** El invitado hace clic en iniciar sesión

**Then** El sistema muestra un pop-up de tipo “success” con el mensaje “Inicio de sesión exitoso”

**Scenario:** Fallo al iniciar sesión sin escribir nombre

**Given** El invitado entra en la página principal

**When** El invitado hace clic en iniciar sesión en el navbar  
**And** El invitado introduce su "" y "contraseña"  
**And** El invitado hace clic en iniciar sesión  
**Then** El sistema muestra un pop-up de tipo "error" con el mensaje "El nombre es obligatorio"

**Scenario:** Fallo al iniciar sesión sin escribir contraseña

**Given** El invitado entra en la página principal  
**When** El invitado hace clic en iniciar sesión en el navbar  
**And** El invitado introduce su "nombre" y ""  
**And** El invitado hace clic en iniciar sesión  
**Then** El sistema muestra un pop-up de tipo "error" con el mensaje "La contraseña es obligatoria"

**Scenario:** Fallo al iniciar sesión con nombre inexistente

**Given** El invitado entra en la página principal  
**When** El invitado hace clic en iniciar sesión en el navbar  
**And** El invitado introduce su "nombre inexistente" y "contraseña"  
**And** El invitado hace clic en iniciar sesión  
**Then** El sistema muestra un pop-up de tipo "error" con el mensaje "El usuario con ese nombre no existe"

**Scenario:** Fallo al iniciar sesión con contraseña incorrecta

**Given** El invitado entra en la página principal  
**When** El invitado hace clic en iniciar sesión en el navbar  
**And** El invitado introduce su "nombre" y "contraseña incorrecta"  
**And** El invitado hace clic en iniciar sesión  
**Then** El sistema muestra un pop-up de tipo "error" con el mensaje "La contraseña es incorrecta"

### 3.3.4. Requisito 4

Título	Filtrar bloques de contenido por nombre
Descripción	Un invitado debe poder filtrar los bloques de contenido por nombre
Pre-condiciones	No procede
Post-condiciones	Los bloques de contenido son filtrados correctamente
Prioridad	Baja

#### Escenario principal

Escenario 1. Filtrar los bloques de contenido por nombre

1. El invitado entra en la página principal.
2. El sistema muestra la página principal.
3. El invitado escribe en la barra de búsqueda un texto.
4. El sistema muestra los bloques de contenido que contengan el texto escrito.

#### Escenario secundario

Alternativo 1. Fallo al filtrar por nombre ya que no hay bloques de contenido coincidentes.

- 4.a.1. El sistema no muestra ningún bloque de contenido que contenga el texto escrito.

#### Escenarios de prueba

**Feature:** Un invitado debe poder filtrar los bloques de contenido por nombre

**Scenario:** Filtrar los bloques de contenido por nombre

**Given** El invitado entra en la página principal

**When** El invitado busca "nombre"

**Then** El sistema muestra un bloque de contenido

**Scenario:** Fallo al filtrar por nombre ya que no hay bloques de contenido coincidentes.

**Given** El invitado entra en la página principal

**When** El invitado busca "nombre"

**Then** El sistema muestra el mensaje "No existen bloques de contenidos"

### 3.3.5. Requisito 5

Título	Ordenar bloques de contenido
Descripción	Un invitado debe poder ordenar los bloques de contenido por nombre y número de recursos
Pre-condiciones	No procede
Post-condiciones	Los bloques de contenido están ordenados
Prioridad	Baja

### Escenarios principales

Escenario 1. Ordenar bloques de contenido por nombre ascendente.

1. El invitado entra en la página principal.
2. El sistema muestra la página principal.
3. El invitado selecciona "Alfabético (Ascendente)" como orden para los bloques de contenido.
4. El sistema muestra los bloques de contenido por el orden "Alfabético (Ascendente)".

Escenario 2. Ordenar bloques de contenido por nombre descendente.

1. El invitado entra en la página principal.
2. El sistema muestra la página principal.



3. El invitado selecciona “Alfabético (Descendente)” como orden para los bloques de contenido.
4. El sistema muestra los bloques de contenido por el orden “Alfabético (Descendente)”.

Escenario 3. Ordenar bloques de contenido por número de recursos ascendente.

1. El invitado entra en la página principal.
2. El sistema muestra la página principal.
3. El invitado selecciona “Nº Recursos (Ascendente)” como orden para los bloques de contenido.
4. El sistema muestra los bloques de contenido por el orden “Nº Recursos (Ascendente)”.

Escenario 4. Ordenar bloques de contenido por número de recursos descendente.

1. El invitado entra en la página principal.
2. El sistema muestra la página principal.
3. El invitado selecciona “Nº Recursos (Descendente)” como orden para los bloques de contenido.
4. El sistema muestra los bloques de contenido por el orden “Nº Recursos (Descendente)”.

### Escenarios de prueba

**Feature:** Un invitado debe poder ordenar los bloques de contenido

**Scenario:** Ordenar bloques de contenido por nombre ascendente

**Given** El invitado entra en la página principal

**When** El invitado ordena los bloques de contenido por “**Alfabético (Ascendente)**”

**Then** El sistema muestra los bloques de contenido ordenados por "Alfabético (Ascendente)"

**Scenario:** Ordenar bloques de contenido por nombre descendente

**Given** El invitado entra en la página principal

**When** El invitado ordena los bloques de contenido por "Alfabético (Descendente)"

**Then** El sistema muestra los bloques de contenido ordenados por "Alfabético (Descendente)"

**Scenario:** Ordenar bloques de contenido por número de recursos ascendente

**Given** El invitado entra en la página principal

**When** El invitado ordena los bloques de contenido por "Nº Recursos (Ascendente)"

**Then** El sistema muestra los bloques de contenido ordenados por "Nº Recursos (Ascendente)"

**Scenario:** Ordenar bloques de contenido por número de recursos descendente

**Given** El invitado entra en la página principal

**When** El invitado ordena los bloques de contenido por "Nº Recursos (Descendente)"

**Then** El sistema muestra los bloques de contenido ordenados por "Nº Recursos (Descendente)"

### 3.3.6. Requisito 6

Título	Ordenar recursos
Descripción	Un invitado debe poder ordenar los recursos de un bloque de contenido por fecha de creación y número de likes
Pre-condiciones	No procede
Post-condiciones	Los recursos del bloque de contenido están ordenados
Prioridad	Baja

#### Escenarios principales

Escenario 1. Ordenar los recursos de un bloque de contenido por la fecha de creación ascendente.

1. El invitado entra en la página principal.
2. El sistema muestra la página principal.
3. El invitado hace clic en el bloque de contenido “nombre del bloque de contenido”.
4. El sistema muestra la información y recursos del bloque de contenido.
5. El invitado selecciona “Fecha (Ascendente)” como orden para los recursos del bloque de contenido.
6. El sistema muestra los recursos por el orden “Fecha (Ascendente)” del bloque de contenido.

Escenario 2. Ordenar los recursos de un bloque de contenido por la fecha de creación descendente.

1. El invitado entra en la página principal.
2. El sistema muestra la página principal.
3. El invitado hace clic en el bloque de contenido “nombre del bloque de contenido”.
4. El sistema muestra la información y recursos del bloque de contenido.

5. El invitado selecciona “Fecha (Descendente)” como orden para los recursos del bloque de contenido.
6. El sistema muestra los recursos por el orden “Fecha (Descendente)” del bloque de contenido.

Escenario 3. Ordenar los recursos de un bloque de contenido por número de likes ascendente.

1. El invitado entra en la página principal.
2. El sistema muestra la página principal.
3. El invitado hace clic en el bloque de contenido “nombre del bloque de contenido”.
4. El sistema muestra la información y recursos del bloque de contenido.
5. El invitado selecciona “Likes (Ascendente)” como orden para los recursos del bloque de contenido.
6. El sistema muestra los recursos por el orden “Likes (Ascendente)” del bloque de contenido.

Escenario 4. Ordenar los recursos de un bloque de contenido por número de likes descendente.

1. El invitado entra en la página principal.
2. El sistema muestra la página principal.
3. El invitado hace clic en el bloque de contenido “nombre del bloque de contenido”.
4. El sistema muestra la información y recursos del bloque de contenido.
5. El invitado selecciona “Likes (Descendente)” como orden para los recursos del bloque de contenido.
6. El sistema muestra los recursos por el orden “Likes (Descendente)” del bloque de contenido.

## Escenarios de prueba

**Feature:** Un invitado debe poder ordenar los recursos de un bloque de contenido

**Scenario:** Ordenar recursos de un bloque de contenido por fecha de creación ascendente

**Given** El invitado entra en la página principal

**When** El invitado hace clic en el bloque de contenido " nombre del bloque de contenido "

**And** El invitado ordena los recursos por "Fecha (Ascendente) "

**Then** El sistema muestra los recursos ordenados por "Fecha (Ascendente) "

**Scenario:** Ordenar recursos de un bloque de contenido por fecha de creación descendente

**Given** El invitado entra en la página principal

**When** El invitado hace clic en el bloque de contenido " nombre del bloque de contenido "

**And** El invitado ordena los recursos por "Fecha (Descendente) "

**Then** El sistema muestra los recursos ordenados por "Fecha (Descendente) "

**Scenario:** Ordenar recursos de un bloque de contenido por número de likes ascendente

**Given** El invitado entra en la página principal

**When** El invitado hace clic en el bloque de contenido " nombre del bloque de contenido "

**And** El invitado ordena los recursos por "Likes (Ascendente) "

**Then** El sistema muestra los recursos ordenados por "Likes (Ascendente)"

**Scenario:** Ordenar recursos de un bloque de contenido por número de likes descendente

**Given** El invitado entra en la página principal

**When** El invitado hace clic en el bloque de contenido "nombre del bloque de contenido"

**And** El invitado ordena los recursos por "Likes (Descendente)"

**Then** El sistema muestra los recursos ordenados por "Likes (Descendente)"

### 3.3.7. Requisito 7

Título	Ver perfiles de otros usuarios
Descripción	Un invitado debe poder acceder al perfil del usuario creador de un recurso
Pre-condiciones	No procede
Post-condiciones	Se visualiza el perfil del usuario
Prioridad	Baja

#### Escenario principal

Escenario 1. Acceder al perfil del usuario creador de un recurso.

1. El invitado entra en la página principal.
2. El sistema muestra la página principal.
3. El invitado hace clic en el bloque de contenido "nombre del bloque de contenido".
4. El sistema muestra la información y recursos del bloque de contenido.

5. El invitado hace clic en el recurso “nombre del recurso”.
6. El sistema muestra la información del recurso.
7. El invitado hace clic en el avatar del usuario creador del recurso.
8. El sistema muestra la información del perfil del usuario creador del recurso.

### Escenario de prueba

**Feature:** Un invitado debe poder acceder al perfil del usuario creador de un recurso

**Scenario:** Acceder al perfil del usuario creador de un recurso

**Given** El invitado entra en la página principal

**When** El invitado hace clic en el bloque de contenido “**nombre del bloque de contenido**”

**And** El invitado hace clic en el recurso “**nombre del recurso**”

**And** El invitado hace clic en el avatar del usuario creador del recurso

**Then** El invitado ve la información del usuario creador del recurso

### 3.3.8. Requisito 8

Título	Contactar con el administrador y moderadores
Descripción	Un usuario debe poder contactar con los moderadores y el administrador, a través de un correo
Pre-condiciones	Iniciar sesión con un usuario sin rol de moderador o administrador
Post-condiciones	El usuario es redirigido a su aplicación predefinida de email para contactar con los moderadores y el administrador
Prioridad	Media

## Escenario principal

Escenario 1. Contactar con el administrador o los moderadores a través del correo.

1. El usuario entra en la página principal.
2. El sistema muestra la página principal.
3. El usuario hace clic en el botón de danger.
4. El sistema muestra el correo para contactar con el administrador o los moderadores.

## Escenario de prueba

**Feature:** Un usuario debe poder contactar con los moderadores y el administrador, a través de un correo

**Scenario:** Contactar con el administrador o los moderadores

**Given** El usuario entra en la página principal

**When** El usuario hace clic en el botón danger

**Then** El sistema muestra el correo para contactar con el administrador o los moderadores

### 3.3.9. Requisito 9

Título	Cambiar datos del perfil
Descripción	Un usuario debe poder cambiar la foto y descripción de su perfil
Pre-condiciones	Iniciar sesión
Post-condiciones	Se pueden visualizar los datos del usuario
Prioridad	Baja

## Escenarios principales

Escenario 1. Cambiar descripción del perfil.

1. El usuario entra en la página principal.



2. El sistema muestra la página principal.
3. El usuario hace clic en el botón de perfil.
4. El sistema muestra el perfil del usuario.
5. El usuario hace clic en el botón “Editar perfil”.
6. El sistema muestra un pop-up con la opción de “Cambiar foto” y “Cambiar descripción”.
7. El usuario hace clic en el botón “Cambiar descripción”.
8. El sistema muestra un textarea para escribir la descripción.
9. El usuario escribe en el textarea el texto “nueva descripción”.
10. El usuario hace clic en el botón “Cambiar”.
11. El sistema muestra el perfil con la información actualizada.

#### Escenario 2. Cambiar foto del perfil.

1. El usuario entra en la página principal.
2. El sistema muestra la página principal.
3. El usuario hace clic en el botón de perfil.
4. El sistema muestra el perfil del usuario.
5. El usuario hace clic en el botón “Editar perfil”.
6. El sistema muestra un pop-up con la opción de “Cambiar foto” y “Cambiar descripción”.
7. El usuario hace clic en el botón “Cambiar foto”.
8. El sistema muestra un drag and drop para subir la foto.
9. El usuario sube una foto.
10. El usuario hace clic en el botón “Cambiar”.
11. El sistema muestra el perfil con la información actualizada.

## Escenarios de prueba

**Feature:** Un usuario debe poder cambiar los datos de su perfil

**Scenario:** Cambiar descripción

**Given** El usuario entra en la página principal

**When** El usuario hace clic en el botón perfil

**And** El usuario hace clic en el botón "Editar perfil"

**And** El usuario hace clic en el botón "Cambiar descripción"

**And** El usuario escribe en el textarea "nueva descripción"

**And** El usuario hace clic en el botón "Cambiar"

**Then** El usuario ve la descripción del perfil "nueva descripción"

**Scenario:** Cambiar foto

**Given** El usuario entra en la página principal

**When** El usuario hace clic en el botón perfil

**And** El usuario hace clic en el botón "Editar perfil"

**And** El usuario hace clic en el botón "Cambiar foto"

**And** El usuario sube su nueva foto

**And** El usuario hace clic en el botón "Cambiar"

**Then** El usuario ve su nueva foto

### 3.3.10. Requisito 10

Título	Gestionar recursos
Descripción	Un usuario debe poder crear recursos en un bloque de contenido
Pre-condiciones	Iniciar sesión
Post-condiciones	Se crea un nuevo recurso en el bloque de contenido
Prioridad	Alta

#### Escenario principal

## Escenario 1. Crear recurso.

1. El usuario entra en la página principal.
2. El sistema muestra la página principal.
3. El usuario hace clic en el bloque de contenido “nombre del bloque de contenido”.
4. El sistema muestra la información y recursos del bloque de contenido.
5. El usuario hace clic en el botón “Crear un nuevo recurso”.
6. El sistema muestra la pantalla de crear un nuevo recurso.
7. El usuario escribe en el input con id “título” el texto “título del recurso”.
8. El usuario selecciona en el dropdown con id “tipo recurso” el elemento “vídeo de Youtube”.
9. El usuario escribe en el input con id “url” el texto “url del vídeo de Youtube”.
10. El sistema muestra un vídeo de Youtube.
11. El usuario escribe en el input con id “descripción” el texto “descripción del recurso”.
12. El usuario hace clic en el botón “Crear”.
13. El sistema muestra un pop-up de tipo “success” con el mensaje “El recurso ‘título del recurso’ se ha creado correctamente”.

## Escenarios secundarios

### Alternativo 1. Fallo al crear el recurso por título vacío.

- 7.a.1. El usuario escribe en el input con id “título” el texto “título del recurso con menos de 10 caracteres”.
- 7.a.2. El usuario hace clic en el botón “Crear”.
- 7.a.3. El sistema muestra un pop-up de tipo “error” con el mensaje “El nombre del recurso no puede estar vacío”.

Alternativo 2. Fallo al crear el recurso por título menor a 10 caracteres.

- 7.b.1. El usuario escribe en el input con id "título" el texto "título del recurso con menos de 10 caracteres".
- 7.b.2. El usuario hace clic en el botón "Crear".
- 7.b.3. El sistema muestra un pop-up de tipo "error" con el mensaje "El nombre del recurso debe tener entre 10 y 50 caracteres".

Alternativo 3. Fallo al crear el recurso por título mayor a 50 caracteres.

- 7.c.1. El usuario escribe en el input con id "título" el texto "título del recurso con más de 50 caracteres".
- 7.c.2. El usuario hace clic en el botón "Crear".
- 7.c.3. El sistema muestra un pop-up de tipo "error" con el mensaje "El nombre del recurso debe tener entre 10 y 50 caracteres".

Alternativo 4. Fallo al crear el recurso por tipo de recurso no elegido.

- 8.a.1. El usuario hace clic en el botón "Crear".
- 8.a.2. El sistema muestra un pop-up de tipo "error" con el mensaje "El tipo de recurso ' ' no es válido".

Alternativo 5. Fallo al crear recurso de tipo de libro con url vacía.

- 8.b.1. El usuario selecciona en el dropdown con id "tipo recurso" el elemento "Libro".
- 8.b.2. El usuario escribe en el input con id "url" el texto "".
- 8.b.3. El usuario escribe en el input con id "descripción" el texto "descripción del recurso".
- 8.b.4. El usuario hace clic en el botón "Crear".
- 8.b.5. El sistema muestra un pop-up de tipo "error" con el mensaje "La url del libro proporcionada está vacía o no es una URL válida".

Alternativo 6. Fallo al crear recurso de tipo de libro con url no válida.

- 8.c.1. El usuario selecciona en el dropdown con id “tipo recurso” el elemento “Libro”.
- 8.c.2. El usuario escribe en el input con id “url” el texto “url de una página web no válida”.
- 8.c.3. El usuario escribe en el input con id “descripción” el texto “descripción del recurso”.
- 8.c.4. El usuario hace clic en el botón “Crear”.
- 8.c.5. El sistema muestra un pop-up de tipo “error” con el mensaje “La url del libro proporcionada está vacía o no es una URL válida”.

Alternativo 7. Crear recurso de tipo libro correctamente.

- 8.d.1. El usuario selecciona en el dropdown con id “tipo recurso” el elemento “Libro”.
- 8.d.2. El usuario escribe en el input con id “url” el texto “url de una página web”.
- 8.d.3. El usuario escribe en el input con id “descripción” el texto “descripción del recurso”.
- 8.d.4. El usuario hace clic en el botón “Crear”.
- 8.d.5. El sistema muestra un pop-up de tipo “success” con el mensaje “El recurso ‘título del recurso’ se ha creado correctamente”.

Alternativo 8. Fallo al crear el recurso de tipo tutorial porque no se ha elegido el tipo de tutorial.

- 8.e.1. El usuario selecciona en el dropdown con id “tipo recurso” el elemento “Tutorial”.
- 8.e.2. El usuario hace clic en el botón “Crear”.
- 8.e.3. El sistema muestra un pop-up de tipo “error” con el mensaje “El tipo de tutorial no puede estar vacío”.

Alternativo 9. Fallo al crear el recurso de tipo tutorial externo por url vacía

- 8.f.1. El usuario selecciona en el dropdown con id “tipo recurso” el elemento “Tutorial”.
- 8.f.2. El usuario selecciona en el dropdown con id “tipo tutorial” el elemento “Tutorial de otra página web”.
- 8.f.3. El usuario hace clic en el botón “Crear”.
- 8.f.4. El sistema muestra un pop-up de tipo “error” con el mensaje “La url del tutorial proporcionada está vacía o no es una URL válida”.

Alternativo 10. Fallo al crear el recurso de tipo tutorial externo por url no válida.

- 8.g.1. El usuario selecciona en el dropdown con id “tipo recurso” el elemento “Tutorial”.
- 8.g.2. El usuario selecciona en el dropdown con id “tipo tutorial” el elemento “Tutorial de otra página web”.
- 8.g.3. El usuario hace clic en el botón “Crear”.
- 8.g.4. El sistema muestra un pop-up de tipo “error” con el mensaje “La url del tutorial proporcionada está vacía o no es una URL válida”.

Alternativo 11. Crear el recurso de tipo tutorial propio.

- 8.h.1. El usuario selecciona en el dropdown con id “tipo recurso” el elemento “Tutorial”.
- 8.h.2. El usuario selecciona en el dropdown con id “tipo tutorial” el elemento “Tutorial propio”.
- 8.h.3. El usuario hace clic en el botón “Crear”.
- 8.h.4. El sistema muestra un pop-up de tipo “success” con el mensaje “El recurso ‘título del recurso’ se ha creado correctamente”.

Alternativo 12. Crear el recurso de tipo tutorial externo.

- 8.i.1. El usuario selecciona en el dropdown con id “tipo recurso” el elemento “Tutorial”.

- 8.i.2. El usuario selecciona en el dropdown con id “tipo tutorial” el elemento “Tutorial de otra página web”.
- 8.i.3. El usuario escribe en el input con id “url” el texto “url del tutorial”.
- 8.i.4. El usuario hace clic en el botón “Crear”.
- 8.i.5. El sistema muestra un pop-up de tipo “success” con el mensaje “El recurso ‘título del recurso’ se ha creado correctamente”.

Alternativo 13. Fallo al crear el recurso de tipo vídeo de Youtube por url vacía.

- 9.a.1. El usuario escribe en el input con id “url” el texto “url del vídeo de Youtube vacía”.
- 9.a.2. El sistema no muestra un vídeo de Youtube.
- 9.a.3. El usuario escribe en el input con id “descripción” el texto “descripción del recurso”.
- 9.a.4. El usuario hace clic en el botón “Crear”.
- 9.a.5. El sistema muestra un pop-up de tipo “error” con el mensaje “La url del vídeo de youtube proporcionada está vacía o no es una URL válida”.

Alternativo 14. Fallo al crear el recurso de tipo vídeo de Youtube por url inválida.

- 9.b.1. El usuario escribe en el input con id “url” el texto “url del vídeo de Youtube vacía”.
- 9.b.2. El sistema no muestra un vídeo de Youtube.
- 9.b.3. El usuario escribe en el input con id “descripción” el texto “descripción del recurso”.
- 9.b.4. El usuario hace clic en el botón “Crear”.
- 9.b.5. El sistema muestra un pop-up de tipo “error” con el mensaje “La url del vídeo de youtube proporcionada está vacía o no es una URL válida”.

Alternativo 15. Crear recurso de tipo vídeo de Youtube con url de un vídeo de móvil.

- 9.c.1. El usuario escribe en el input con id “url” el texto “url del vídeo de Youtube de móvil”.

- 9.c.2. El sistema muestra un vídeo de Youtube.
- 9.c.3. El usuario escribe en el input con id “descripción” el texto “descripción del recurso”.
- 9.c.4. El usuario hace clic en el botón “Crear”.
- 9.c.5. El sistema muestra un pop-up de tipo “success” con el mensaje “El recurso ‘título del recurso’ se ha creado correctamente”.

Alternativo 16. Fallo al crear el recurso por descripción vacía.

- 11.a.1. El usuario escribe en el input con id “descripción” el texto “”.
- 11.a.2. El usuario hace clic en el botón “Crear”.
- 11.a.3. El sistema muestra un pop-up de tipo “error” con el mensaje “La descripción del recurso no puede estar vacía”.

### Escenarios de prueba

**Feature:** Un usuario debe poder crear recursos

**Scenario:** Crear vídeo de Youtube

**Given** El usuario entra en la página principal

**When** El usuario hace clic en el bloque de contenido “nombre del bloque de contenido”

**And** El usuario hace clic en el botón “Crear un nuevo recurso”

**And** El usuario escribe en el input con id “título” el texto “título del recurso”

**And** El usuario selecciona en el dropdown con id “tipo recurso” el elemento “vídeo de Youtube”

**And** El usuario escribe en el input con id “url” el texto “url del vídeo de Youtube”

**And** El sistema muestra un vídeo de Youtube



And El usuario escribe en el input con id "descripción" el texto "descripción del recurso"

And El usuario hace clic en el botón "Crear"

Then El sistema muestra un pop-up de tipo "success" con el mensaje "El recurso 'título del recurso' se ha creado correctamente"

Scenario: Fallo al crear recurso por nombre vacío

Given El usuario entra en la página principal

When El usuario hace clic en el bloque de contenido "nombre del bloque de contenido"

And El usuario hace clic en el botón "Crear un nuevo recurso"

And El usuario escribe en el input con id "título" el texto "título del recurso con menos de 10 caracteres"

And El usuario hace clic en el botón "Crear"

Then El sistema muestra un pop-up de tipo "error" con el mensaje "El nombre del recurso no puede estar vacío"

Scenario: Fallo al crear recurso por título con menos de 10 caracteres

Given El usuario entra en la página principal

When El usuario hace clic en el bloque de contenido "nombre del bloque de contenido"

And El usuario hace clic en el botón "Crear un nuevo recurso"

And El usuario escribe en el input con id "título" el texto "título del recurso con menos de 10 caracteres"

And El usuario hace clic en el botón "Crear"

Then El sistema muestra un pop-up de tipo "error" con el mensaje "El nombre del recurso debe tener entre 10 y 50"

caracteres”

**Scenario:** Fallo al crear recurso por título con más de 50 caracteres

**Given** El usuario entra en la página principal

**When** El usuario hace clic en el bloque de contenido ”nombre del bloque de contenido”

**And** El usuario hace clic en el botón ”Crear un nuevo recurso”

**And** El usuario escribe en el input con id ”título” el texto ”título del recurso con más de 50 caracteres”

**And** El usuario hace clic en el botón ”Crear”

**Then** El sistema muestra un pop-up de tipo ”error” con el mensaje ””

**Scenario:** Fallo al crear recurso por tipo de recurso no elegido

**Given** El usuario entra en la página principal

**When** El usuario hace clic en el bloque de contenido ”nombre del bloque de contenido”

**And** El usuario hace clic en el botón ”Crear un nuevo recurso”

**And** El usuario escribe en el input con id ”título” el texto ”título del recurso”

**And** El usuario hace clic en el botón ”Crear”

**Then** El sistema muestra un pop-up de tipo ”error” con el mensaje ”El tipo de recurso ”” no es válido”

**Scenario:** Fallo al crear recurso por url libro vacío

**Given** El usuario entra en la página principal

**When** El usuario hace clic en el bloque de contenido ”nombre del bloque de contenido”

And El usuario hace clic en el botón "Crear un nuevo recurso"

And El usuario escribe en el input con id "título" el texto "título del recurso"

And El usuario selecciona en el dropdown con id "tipo recurso" el elemento "Libro"

And El usuario escribe en el input con id "url" el texto ""

And El usuario escribe en el input con id "descripción" el texto "descripción del recurso"

And El usuario hace clic en el botón "Crear"

Then El sistema muestra un pop-up de tipo "error" con el mensaje "La url del libro proporcionada está vacía o no es una URL válida"

Scenario: Fallo al crear recurso por url libro no válido

Given El usuario entra en la página principal

When El usuario hace clic en el bloque de contenido "nombre del bloque de contenido"

And El usuario hace clic en el botón "Crear un nuevo recurso"

And El usuario escribe en el input con id "título" el texto "título del recurso"

And El usuario selecciona en el dropdown con id "tipo recurso" el elemento "Libro"

And El usuario escribe en el input con id "url" el texto "url de una página web"

And El usuario escribe en el input con id "descripción" el texto "descripción del recurso"

And El usuario hace clic en el botón "Crear"

Then El sistema muestra un pop-up de tipo "error" con el mensaje "La url del libro proporcionada está vacía o no es

una URL válida”

**Scenario:** Recurso de tipo libro creado correctamente

**Given** El usuario entra en la página principal

**When** El usuario hace clic en el bloque de contenido ”nombre del bloque de contenido”

**And** El usuario hace clic en el botón ”Crear un nuevo recurso”

**And** El usuario escribe en el input con id ”título” el texto ”título del recurso”

**And** El usuario selecciona en el dropdown con id ”tipo recurso” el elemento ”Libro”

**And** El usuario escribe en el input con id ”url” el texto ”url de una página web”

**And** El usuario escribe en el input con id ”descripción” el texto ”descripción del recurso”

**And** El usuario hace clic en el botón ”Crear”

**Then** El sistema muestra un pop-up de tipo ”success” con el mensaje ”El recurso `título del recurso` se ha creado correctamente”

**Scenario:** Fallo al crear recurso por tipo de tutorial vacío

**Given** El usuario entra en la página principal

**When** El usuario hace clic en el bloque de contenido ”nombre del bloque de contenido”

**And** El usuario hace clic en el botón ”Crear un nuevo recurso”

**And** El usuario escribe en el input con id ”título” el texto ”título del recurso”

**And** El usuario selecciona en el dropdown con id ”tipo recurso” el elemento ”Tutorial”

And El usuario hace clic en el botón "Crear"

Then El sistema muestra un pop-up de tipo "error" con el mensaje "El tipo de recurso tutorial no puede estar vacío"

Scenario: Fallo al crear recurso por url tutorial vacía

Given El usuario entra en la página principal

When El usuario hace clic en el bloque de contenido "nombre del bloque de contenido"

And El usuario hace clic en el botón "Crear un nuevo recurso"

And El usuario escribe en el input con id "título" el texto "título del recurso"

And El usuario selecciona en el dropdown con id "tipo recurso" el elemento "Tutorial"

And El usuario selecciona en el dropdown con id "tipo tutorial" el elemento "Tutorial de otra página web"

And El usuario hace clic en el botón "Crear"

Then El sistema muestra un pop-up de tipo "error" con el mensaje "La url del tutorial proporcionada está vacía o no es una URL válida"

Scenario: Fallo al crear recurso por url tutorial no válido

Given El usuario entra en la página principal

When El usuario hace clic en el bloque de contenido "nombre del bloque de contenido"

And El usuario hace clic en el botón "Crear un nuevo recurso"

And El usuario escribe en el input con id "título" el texto "título del recurso"

And El usuario selecciona en el dropdown con id "tipo recurso" el elemento "Tutorial"

And El usuario selecciona en el dropdown con id "tipo tutorial" el elemento "Tutorial de otra página web"

And El usuario hace clic en el botón "Crear"

Then El sistema muestra un pop-up de tipo "error" con el mensaje "La url del tutorial proporcionada está vacía o no es una URL válida"

Scenario: Recurso de tipo tutorial propio creado correctamente

Given El usuario entra en la página principal

When El usuario hace clic en el bloque de contenido "nombre del bloque de contenido"

And El usuario hace clic en el botón "Crear un nuevo recurso"

And El usuario escribe en el input con id "título" el texto "título del recurso"

And El usuario selecciona en el dropdown con id "tipo recurso" el elemento "Tutorial"

And El usuario selecciona en el dropdown con id "tipo tutorial" el elemento "Tutorial propio"

And El usuario hace clic en el botón "Crear"

Then El sistema muestra un pop-up de tipo "success" con el mensaje "El recurso 'título del recurso' se ha creado correctamente"

Scenario: Recurso de tipo tutorial externo creado correctamente

Given El usuario entra en la página principal

When El usuario hace clic en el bloque de contenido "nombre del bloque de contenido"

And El usuario hace clic en el botón "Crear un nuevo recurso"

And El usuario escribe en el input con id "título" el texto "título del recurso"

And El usuario selecciona en el dropdown con id "tipo recurso" el elemento "Tutorial"

And El usuario selecciona en el dropdown con id "tipo tutorial" el elemento "Tutorial de otra página web"

And El usuario escribe en el input con id "url" el texto "url del tutorial"

And El usuario hace clic en el botón "Crear"

Then El sistema muestra un pop-up de tipo "success" con el mensaje "El recurso `título del recurso` se ha creado correctamente"

Scenario: Fallo al crear recurso por url vídeo vacío

Given El usuario entra en la página principal

When El usuario hace clic en el bloque de contenido "nombre del bloque de contenido"

And El usuario hace clic en el botón "Crear un nuevo recurso"

And El usuario escribe en el input con id "título" el texto "título del recurso"

And El usuario selecciona en el dropdown con id "tipo recurso" el elemento "vídeo de Youtube"

And El usuario escribe en el input con id "url" el texto "url del vídeo de Youtube vacía"

And El sistema no muestra un vídeo de Youtube

And El usuario escribe en el input con id "descripción" el texto "descripción del recurso"

And El usuario hace clic en el botón "Crear"

Then El sistema muestra un pop-up de tipo "error" con el mensaje "La url del vídeo de youtube proporcionada está"

vacía o no es una URL válida”

**Scenario:** Fallo al crear recurso por url vídeo no válido

**Given** El usuario entra en la página principal

**When** El usuario hace clic en el bloque de contenido "nombre del bloque de contenido"

**And** El usuario hace clic en el botón "Crear un nuevo recurso"

**And** El usuario escribe en el input con id "título" el texto "título del recurso"

**And** El usuario selecciona en el dropdown con id "tipo recurso" el elemento "vídeo de Youtube"

**And** El usuario escribe en el input con id "url" el texto "url del vídeo de Youtube no válida"

**And** El sistema no muestra un vídeo de Youtube

**And** El usuario escribe en el input con id "descripción" el texto "descripción del recurso"

**And** El usuario hace clic en el botón "Crear"

**Then** El sistema muestra un pop-up de tipo "error" con el mensaje "La url del vídeo de youtube proporcionada está vacía o no es una URL válida"

**Scenario:** Recurso de tipo vídeo creado correctamente con url móvil

**Given** El usuario entra en la página principal

**When** El usuario hace clic en el bloque de contenido "nombre del bloque de contenido"

**And** El usuario hace clic en el botón "Crear un nuevo recurso"

**And** El usuario escribe en el input con id "título" el texto "título del recurso"



And El usuario selecciona en el dropdown con id "tipo recurso" el elemento "vídeo de Youtube"

And El usuario escribe en el input con id "url" el texto "url del vídeo de Youtube de móvil"

And El sistema muestra un vídeo de Youtube

And El usuario escribe en el input con id "descripción" el texto "descripción del recurso"

And El usuario hace clic en el botón "Crear"

Then El sistema muestra un pop-up de tipo "success" con el mensaje "El recurso 'título del recurso' se ha creado correctamente"

Scenario: Fallo al crear recurso por descripción del recurso vacío

Given El usuario entra en la página principal

When El usuario hace clic en el bloque de contenido "nombre del bloque de contenido"

And El usuario hace clic en el botón "Crear un nuevo recurso"

And El usuario escribe en el input con id "título" el texto "título del recurso"

And El usuario selecciona en el dropdown con id "tipo recurso" el elemento "vídeo de Youtube"

And El usuario escribe en el input con id "url" el texto "url del vídeo de Youtube"

And El sistema muestra un vídeo de Youtube

And El usuario escribe en el input con id "descripción" el texto ""

And El usuario hace clic en el botón "Crear"

Then El sistema muestra un pop-up de tipo "error" con el mensaje "La descripción del recurso no puede estar vacía"

Título	Gestionar recursos
Descripción	Un usuario debe poder eliminar sus recursos creados
Pre-condiciones	Iniciar sesión
Post-condiciones	Se borra su recurso creado
Prioridad	Alta

## Escenario principal

Escenario 1. Cambiar descripción del perfil.

1. El usuario entra en la página principal.
2. El sistema muestra la página principal.
3. El usuario hace clic en el botón de perfil.
4. El sistema muestra el perfil del usuario.
5. El usuario borra el recurso con título “”.
6. El sistema muestra un pop-up de tipo “success” con el mensaje “El recurso ha sido eliminado”.

## Escenario de prueba

**Feature:** Un usuario debe poder manejar recursos en un bloque de contenido

**Scenario:** Eliminar un recurso

**Given** El usuario entra en la página principal

**And** El usuario hace clic en el botón perfil

**And** El usuario borra el recurso “nombre del recurso”

**Then** El sistema muestra un pop-up de tipo “success” con el mensaje “El recurso ha sido eliminado”

### 3.3.11. Requisito 11

Título	Puntuar recursos
Descripción	Un usuario debe poder puntuar un recurso a través de “me gusta” y “no me gusta”
Pre-condiciones	Iniciar sesión y no ser el creador del recurso
Post-condiciones	El recurso es puntuado
Prioridad	Media

#### Escenarios principales

Escenario 1. El usuario puntua positivamente un recurso.

1. El usuario entra en la página principal.
2. El sistema muestra la página principal.
3. El usuario hace clic en un bloque de contenido.
4. El sistema muestra la información del bloque de contenido.
5. El usuario hace clic en un recurso del bloque de contenido.
6. El sistema muestra la información del bloque de contenido.
7. El usuario hace clic en el botón de like.
8. El sistema muestra el cambio de puntuación.

Escenario 2. El usuario puntua negativamente un recurso.

1. El usuario entra en la página principal.
2. El sistema muestra la página principal.
3. El usuario hace clic en un bloque de contenido.
4. El sistema muestra la información del bloque de contenido.
5. El usuario hace clic en un recurso del bloque de contenido.
6. El sistema muestra la información del bloque de contenido.
7. El usuario hace clic en el botón de dislike.
8. El sistema muestra el cambio de puntuación.

## Escenarios de prueba

**Feature:** Un usuario debe poder puntuar un recurso

**Scenario:** Dar like a un recurso

**Given** El usuario entra en la página principal

**When** El usuario hace clic en el bloque de contenido "nombre del bloque de contenido"

**And** El usuario hace clic en el recurso "nombre del recurso"

**And** El usuario hace clic en el botón de like

**Then** El sistema muestra el cambio de puntuación

**Scenario:** Dar dislike a un recurso

**Given** El usuario entra en la página principal

**When** El usuario hace clic en el bloque de contenido "nombre del bloque de contenido"

**And** El usuario hace clic en el recurso "nombre del recurso"

**And** El usuario hace clic en el botón de dislike

**Then** El sistema muestra el cambio de puntuación

### 3.3.12. Requisito 12

Título	Crear denuncias
Descripción	Un usuario debe poder denunciar
Pre-condiciones	Iniciar sesión
Post-condiciones	El recurso es denunciado
Prioridad	Alta

### Escenario principal

Escenario 1. El usuario denuncia un recurso.

1. El usuario entra en la página principal.

2. El sistema muestra la página principal.
3. El usuario hace clic en un bloque de contenido.
4. El sistema muestra la información del bloque de contenido.
5. El usuario hace clic en un recurso del bloque de contenido.
6. El sistema muestra la información del bloque de contenido.
7. El usuario hace clic en el botón “Denunciar”.
8. El sistema muestra un pop-up.
9. El usuario escribe en el textarea el texto “texto de la denuncia”.
10. El usuario hace clic en el botón “Aceptar”.
11. El sistema muestra un pop-up de tipo “success” con el mensaje “El recurso ha sido denunciado correctamente”.

### Escenario secundario

Alternativo 1. Fallo al crear denuncia por motivo vacío.

- 9.a.1 El usuario escribe en el textarea el texto “”.
- 9.a.2 El usuario hace clic en el botón “Aceptar”.
- 9.a.3 El sistema muestra un pop-up de tipo “error” con el mensaje “Hay que poner algún motivo para la denuncia”.

### Escenarios de prueba

**Feature:** Un usuario debe poder denunciar un recurso

**Scenario:** Crear denuncia

**Given** El usuario entra en la página principal

**When** El usuario hace clic en el bloque de contenido “nombre del bloque de contenido”

**And** El usuario hace clic en el recurso “nombre del recurso”

**And** El usuario hace clic en el botón “Denunciar”

And El usuario escribe en el textarea el texto "Texto de la denuncia"

And El usuario hace clic en el botón "Aceptar"

Then El sistema muestra un pop-up de tipo "success" con el mensaje "El recurso ha sido denunciado correctamente"

Scenario: Fallo al denunciar por motivo vacío

Given El usuario entra en la página principal

When El usuario hace clic en el bloque de contenido "nombre del bloque de contenido"

And El usuario hace clic en el recurso "nombre del recurso"

And El usuario hace clic en el botón "Denunciar"

And El usuario escribe en el textarea el texto ""

And El usuario hace clic en el botón "Aceptar"

Then El sistema muestra un pop-up de tipo "error" con el mensaje "Hay que poner algún motivo para la denuncia"

### 3.3.13. Requisito 13

Título	Cambiar permisos de los usuarios
Descripción	Un moderador o administrador debe poder cambiar los permisos de los usuarios y moderadores. Si el usuario tenía rol de moderador, e le quitará y viceversa
Pre-condiciones	Iniciar sesión con una cuenta con el rol de administrador o moderador
Post-condiciones	El rol del usuario elegido es cambiado
Prioridad	Baja

### Escenarios principales

Escenario 1. El administrador o moderador da permisos a un usuario.

1. El administrador o moderador entra al perfil del usuario.

2. El sistema muestra el perfil del usuario.
3. El administrador o moderador hace clic en el botón “Convertir en moderador”.
4. El sistema muestra que el rol del usuario es “MODER”.

Escenario 2. El administrador o moderador da permisos a un usuario.

1. El administrador o moderador entra al perfil del usuario.
2. El sistema muestra el perfil del usuario.
3. El administrador o moderador hace clic en el botón “Quitar de moderador”.
4. El sistema muestra que el rol del usuario es “USER”.

### Escenarios de prueba

**Feature:** Un usuario debe poder denunciar un recurso

**Scenario:** Convertir a un usuario en un moderador

**Given** El administrador o moderador entra al perfil del usuario

**And** El administrador o moderador hace clic en el botón “Convertir en moderador”

**Then** El sistema muestra que el rol del usuario es “MODER”

**Scenario:** Convertir a un moderador en un usuario

**Given** El administrador o moderador entra al perfil del usuario

**And** El administrador o moderador hace clic en el botón “Quitar de moderador”

**Then** El sistema muestra que el rol del usuario es “USER”

### 3.3.14. Requisito 14

Título	Eliminar recursos
Descripción	Un moderador o administrador debe poder eliminar un recurso
Pre-condiciones	Iniciar sesión con un usuario con rol de administrador o moderador
Post-condiciones	El recurso es eliminado
Prioridad	Alta

#### Escenarios principales

Escenario 1. El moderador borra un recurso.

1. El moderador entra en la página principal.
2. El sistema muestra la página principal.
3. El moderador hace clic en el bloque de contenido “nombre del bloque de contenido”.
4. El sistema muestra la información y recursos del bloque de contenido.
5. El moderador borra el recurso con nombre “nombre del recurso”.
6. El sistema muestra un pop-up de tipo “success” con el mensaje “El recurso ha sido eliminado”

Escenario 2. El administrador borra un recurso.

1. El administrador entra en la página principal.
2. El sistema muestra la página principal.
3. El administrador hace clic en el bloque de contenido “nombre del bloque de contenido”.
4. El sistema muestra la información y recursos del bloque de contenido.
5. El administrador borra el recurso con nombre “nombre del recurso”.
6. El sistema muestra un pop-up de tipo “success” con el mensaje “El recurso ha sido eliminado”



## Escenarios de prueba

**Feature:** Un moderador o administrador debe poder eliminar un recurso

**Scenario:** Un moderador borra un recurso

**Given** El moderador entra en la página principal

**When** El moderador hace clic en el bloque de contenido " nombre del bloque de contenido "

**And** El moderador borra el recurso con nombre " nombre del recurso "

**Then** El sistema muestra un pop-up de tipo " success " con el mensaje " El recurso ha sido eliminado "

**Scenario:** Un administrador borra un recurso

**Given** El administrador entra en la página principal

**When** El administrador hace clic en el bloque de contenido " nombre del bloque de contenido "

**And** El administrador borra el recurso con nombre " nombre del recurso "

**Then** El sistema muestra un pop-up de tipo " success " con el mensaje " El recurso ha sido eliminado "

### 3.3.15. Requisito 15

Título	Ver denuncias
Descripción	Un moderador o administrador debe poder interactuar con las denuncias
Pre-condiciones	Iniciar sesión con un usuario con rol de administrador o moderador
Post-condiciones	Se pueden visualizar las denuncias
Prioridad	Alta

## Escenarios principales

Escenario 1. El moderador ve las denuncias.

1. El moderador entra en la página principal.
2. El sistema muestra la página principal.
3. El moderador hace clic en el botón de danger.
4. El sistema muestra el listado de recursos denunciados y el número de denuncias.
5. El moderador hace clic en el recurso denunciado “nombre recurso”.
6. El sistema muestra los motivos de las denuncias de ese recurso.
7. El moderador hace clic en un motivo de denuncia.
8. El sistema muestra el recurso denunciado.

Escenario 2. El administrador ve las denuncias.

1. El administrador entra en la página principal.
2. El sistema muestra la página principal.
3. El administrador hace clic en el botón de danger.
4. El sistema muestra el listado de recursos denunciados y el número de denuncias.
5. El administrador hace clic en el recurso denunciado “nombre recurso”.
6. El sistema muestra los motivos de las denuncias de ese recurso.
7. El administrador hace clic en un motivo de denuncia.
8. El sistema muestra el recurso denunciado.

## Escenarios de prueba

**Feature:** Un moderador o administrador debe poder ver las denuncias

**Scenario:** Un moderador ve las denuncias

**Given** El moderador entra en la página principal

**When** El moderador hace clic en el botón de danger

**And** Veo los recursos denunciados

**And** El moderador hace clic en el recurso denunciado "nombre recurso"

**And** Veo los motivos de la denuncia

**And** El moderador hace clic en un motivo de denuncia

**Then** El sistema muestra el recurso denunciado

**Scenario:** Un administrador ve las denuncias

**Given** El administrador entra en la página principal

**When** El administrador hace clic en el botón de danger

**And** Veo los recursos denunciados

**And** El administrador hace clic en el recurso denunciado "nombre recurso"

**And** Veo los motivos de la denuncia

**And** El administrador hace clic en un motivo de denuncia

**Then** El sistema muestra el recurso denunciado

### 3.3.16. Requisito 16

Título	Crear bloques de contenido
Descripción	Un administrador debe poder crear
Pre-condiciones	Iniciar sesión
Post-condiciones	El bloque de contenido es creado correctamente
Prioridad	Alta

#### Escenario principal

Escenario 1. El administrador crea un bloque de contenido.

1. El administrador entra en la página principal.
2. El sistema muestra la página principal.
3. El administrador hace clic en el botón “Crear un nuevo bloque de contenido”.
4. El sistema muestra la pantalla para crear un nuevo bloque de contenido.
5. El administrador escribe en el input con id “título” el texto “título del bloque de contenido”.
6. El administrador escribe en el textarea el texto “descripción”.
7. El administrador sube una foto.
8. El administrador hace clic en el botón “Crear bloque de contenido”.
9. El sistema muestra un pop-up de tipo “success” con el mensaje “El bloque de contenido se ha creado correctamente”.

### **Escenarios secundarios**

Alternativo 1. Fallo al crear bloque de contenido por nombre vacío.

- 5.a.1 El administrador escribe en el input con id “título” el texto “”.
- 5.a.2 El administrador escribe en el textarea el texto “descripción”.
- 5.a.3 El administrador sube una foto.
- 5.a.4 El administrador hace clic en el botón “Crear bloque de contenido”.
- 5.a.5 El sistema muestra un pop-up de tipo “error” con el mensaje “El nombre no puede estar vacío”.

Alternativo 2. Fallo al crear bloque de contenido por descripción vacía.

- 6.a.1 El administrador escribe en el textarea el texto “”.
- 6.a.2 El administrador sube una foto.
- 6.a.3 El administrador hace clic en el botón “Crear bloque de contenido”.
- 6.a.4 El sistema muestra un pop-up de tipo “error” con el mensaje “La descripción no puede estar vacía”.

Alternativo 3. Fallo al crear bloque de contenido por foto no subida.

7.a.1 El administrador hace clic en el botón “Crear bloque de contenido”.

7.a.2 El sistema muestra un pop-up de tipo “error” con el mensaje “Debe de seleccionar una imagen”.

### Escenarios de prueba

**Feature:** Un administrador debe poder crear un bloque de contenido

**Scenario:** Un administrador crea un bloque de contenido

**Given** El administrador entra en la página principal

**When** El administrador hace clic en el botón “Crear un nuevo bloque de contenido”

**And** El administrador escribe en el input con id “título” el texto “título del bloque de contenido”

**And** El administrador escribe en el textarea el texto “descripción”

**And** El administrador sube una foto

**And** El administrador hace clic en el botón “Crear bloque de contenido”

**Then** El sistema muestra un pop-up de tipo “success” con el mensaje “El bloque de contenido se ha creado correctamente”

**Scenario:** Un administrador falla al crear un bloque de contenido por nombre vacío

**Given** El administrador entra en la página principal

**When** El administrador hace clic en el botón “Crear un nuevo bloque de contenido”

**And** El administrador escribe en el input con id “título” el texto “ ”

And El administrador escribe en el textarea el texto " descripción "

And El administrador sube una foto

And El administrador hace clic en el botón "Crear bloque de contenido "

Then El sistema muestra un pop-up de tipo "error" con el mensaje "El nombre no puede estar vacío "

Scenario: Un administrador falla al crear un bloque de contenido por descripción vacía

Given El administrador entra en la página principal

When El administrador hace clic en el botón "Crear un nuevo bloque de contenido "

And El administrador escribe en el input con id "título" el texto "título del bloque de contenido "

And El administrador escribe en el textarea el texto " "

And El administrador sube una foto

And El administrador hace clic en el botón "Crear bloque de contenido "

Then El sistema muestra un pop-up de tipo "error" con el mensaje "La descripción no puede estar vacía "

Scenario: Un administrador falla al crear un bloque de contenido foto no subida

Given El administrador entra en la página principal

When El administrador hace clic en el botón "Crear un nuevo bloque de contenido "

And El administrador escribe en el input con id "título" el texto "título del bloque de contenido "

And El administrador escribe en el textarea el texto " descripción "

**And** El administrador hace clic en el botón "Crear bloque de contenido"

**Then** El sistema muestra un pop-up de tipo "error" con el mensaje "Debe de seleccionar una imagen"

Título	Editar bloques de contenido
Descripción	Un administrador debe poder editar
Pre-condiciones	Iniciar sesión
Post-condiciones	El bloque de contenido es editado correctamente
Prioridad	Baja

### Escenario principal

Escenario 1. El administrador actualiza un bloque de contenido.

1. El administrador entra en la página principal.
2. El sistema muestra la página principal.
3. El administrador actualiza el bloque de contenido "nombre del bloque de contenido".
4. El sistema muestra la pantalla para actualizar el bloque de contenido.
5. El administrador escribe en el input con id "título" el texto "título del bloque de contenido".
6. El administrador escribe en el textarea el texto "descripción".
7. El administrador sube una foto.
8. El administrador hace clic en el botón "Actualizar bloque de contenido".
9. El sistema muestra un pop-up de tipo "success" con el mensaje "El bloque de contenido se ha actualizado correctamente".

### Escenarios secundarios

Alternativo 1. Fallo al actualizar bloque de contenido por nombre vacío.

- 5.a.1 El administrador escribe en el input con id "título" el texto "".
- 5.a.2 El administrador escribe en el textarea el texto "descripción".
- 5.a.3 El administrador sube una foto.
- 5.a.4 El administrador hace clic en el botón "Crear bloque de contenido".
- 5.a.5 El sistema muestra un pop-up de tipo "error" con el mensaje "El nombre no puede estar vacío".

Alternativo 2. Fallo al actualizar bloque de contenido por descripción vacía.

- 6.a.1 El administrador escribe en el textarea el texto "".
- 6.a.2 El administrador sube una foto.
- 6.a.3 El administrador hace clic en el botón "Crear bloque de contenido".
- 6.a.4 El sistema muestra un pop-up de tipo "error" con el mensaje "La descripción no puede estar vacía".

### Escenarios de prueba

**Feature:** Un administrador debe poder actualizar un bloque de contenido

**Scenario:** Un administrador actualizar un bloque de contenido

**Given** El administrador entra en la página principal

**When** El administrador hace clic en el botón "Crear un nuevo bloque de contenido"

**And** El administrador escribe en el input con id "título" el texto "título del bloque de contenido"

**And** El administrador escribe en el textarea el texto "descripción"

**And** El administrador sube una foto

**And** El administrador hace clic en el botón "Crear bloque de contenido"



**Then** El sistema muestra un pop-up de tipo "success" con el mensaje "El bloque de contenido se ha actualizado correctamente"

**Scenario:** Un administrador falla al actualizar un bloque de contenido por nombre vacío

**Given** El administrador entra en la página principal

**When** El administrador hace clic en el botón "Crear un nuevo bloque de contenido"

**And** El administrador escribe en el input con id "título" el texto ""

**And** El administrador escribe en el textarea el texto "descripción"

**And** El administrador sube una foto

**And** El administrador hace clic en el botón "Crear bloque de contenido"

**Then** El sistema muestra un pop-up de tipo "error" con el mensaje "El nombre no puede estar vacío"

**Scenario:** Un administrador falla al actualizar un bloque de contenido por descripción vacía

**Given** El administrador entra en la página principal

**When** El administrador hace clic en el botón "Crear un nuevo bloque de contenido"

**And** El administrador escribe en el input con id "título" el texto "título del bloque de contenido"

**And** El administrador escribe en el textarea el texto ""

**And** El administrador sube una foto

**And** El administrador hace clic en el botón "Crear bloque de contenido"

**Then** El sistema muestra un pop-up de tipo "error" con el mensaje "La descripción no puede estar vacía"

Título	Eliminar de bloques de contenido
Descripción	Un administrador debe poder eliminar
Pre-condiciones	Iniciar sesión
Post-condiciones	El bloque de contenido es eliminado correctamente
Prioridad	Baja

### Escenario principal

Escenario 1. El administrador eliminar un bloque de contenido.

1. El administrador entra en la página principal.
2. El sistema muestra la página principal.
3. El administrador borra el bloque de contenido "nombre del bloque de contenido". **Then** El sistema muestra un pop-up de tipo "success" con el mensaje "El bloque de contenido se ha eliminado correctamente".

### Escenarios de prueba

**Feature:** El administrador debe poder eliminar un bloque de contenido

**Scenario:** Un administrador borra un bloque de contenido

**Given** El administrador entra en la página principal

**When** El administrador borra el bloque de contenido "nombre del bloque de contenido"

**Then** El sistema muestra un pop-up de tipo "success" con el mensaje "El bloque de contenido se ha eliminado correctamente"



# 4

## Diseño del sistema

### 4.1. Diseño arquitectónico de la aplicación

Existen diversas arquitecturas software para crear una aplicación web. Una arquitectura software es un conjunto de patrones usables en la creación de una aplicación y que proporcionan una manera más fácil y clara de interactuar con el código. El tipo de arquitectura elegida para este proyecto es la arquitectura cliente-servidor, que es una de las arquitecturas más usadas en el desarrollo web. Es un modelo de diseño de software en el que hay 2 partes diferenciadas, el cliente y el servidor [28].

El cliente se encarga de operaciones muy sencillas y de mostrar la información necesaria (HTML, CSS y Javascript) sobre la página web al usuario. En caso de tener que hacer alguna acción que requiera cierto nivel de cómputo, se le envía una petición para que lo haga el servidor. Se deja al servidor hacer estas acciones más pesadas para que el usuario vea una página más rápida y eficiente.

El servidor es el encargado de proveer recursos o servicios (páginas html, imágenes, ...) y de hacer cálculos u operaciones complejas para el cliente. Cualquier operación será ejecutada más rápido en un servidor que en un cliente y hará que la experiencia de usuario sea, en general, mejor.

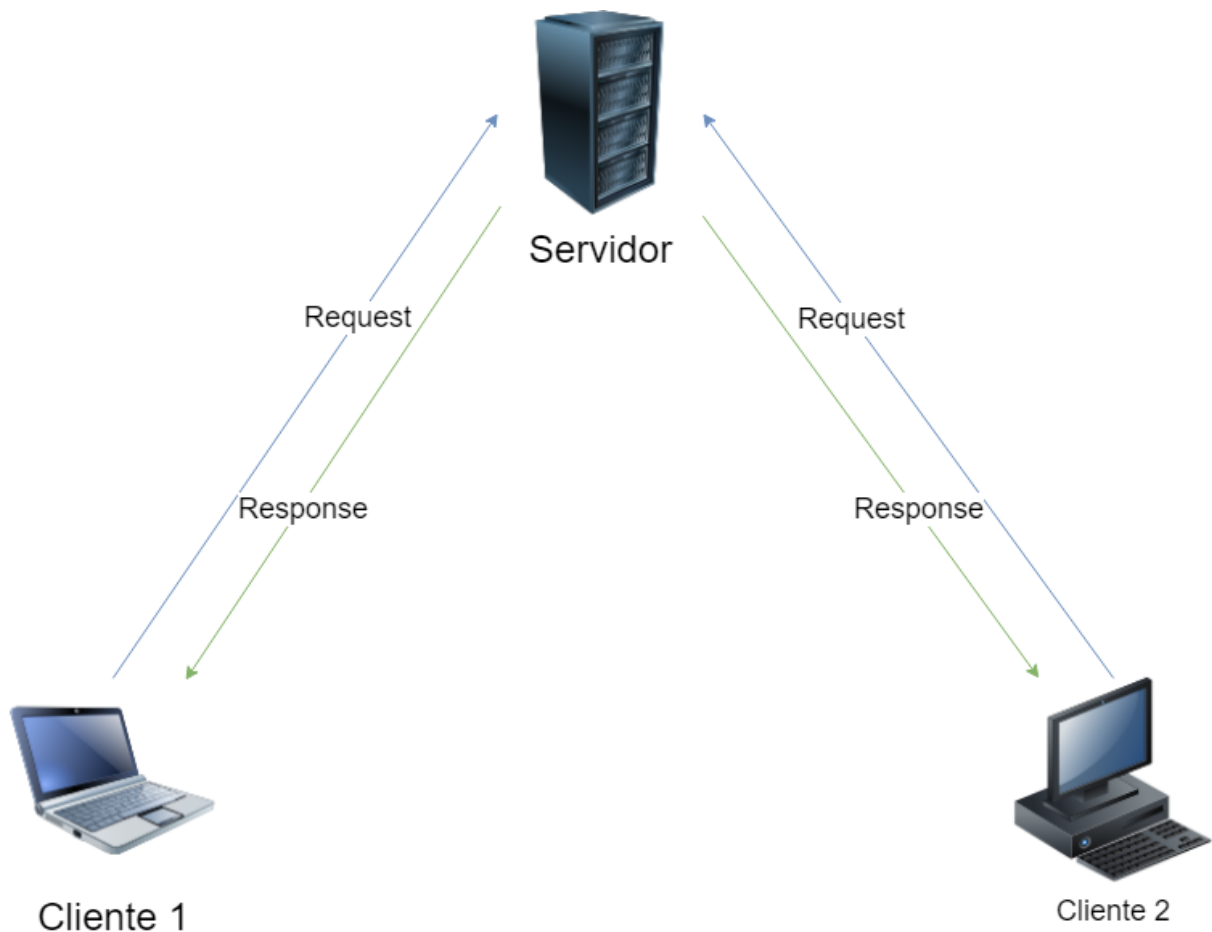


Figura 5: Representación de arquitectura Cliente-Servidor

Actualmente la forma más usada de comunicación entre el cliente y el servidor (y la usada en este proyecto) es la arquitectura API REST (aunque hay alternativas como GraphQL).

REST (Representational state transfer) es un tipo de arquitectura software definida por Roy Fielding en su tesis doctoral en el año 2000. La arquitectura REST se centra en la escalabilidad de interacciones entre componentes, interfaces uniformes, almacenamiento de caché para reducir las cargas, etc [29].

Algunas de las restricciones REST podrían ser la falta de estado, es decir, no es necesario tener información de una petición HTTP anterior para ejecutar la actual, el uso de JSON o XML para la transferencia de datos y una interfaz uniforme, intentando usar los métodos de petición HTTP más comunes como, GET, PUT, DELETE y POST, entre otros.

## 4.2. Diseño de la Base de Datos

La aplicación usa una Base de Datos en MongoDB, al ser esta una base de datos no relacional, no es posible hacer un diagrama entidad-relación sin tener que adaptarlo. Dada la naturaleza de las bases de datos no relaciones el diagrama ha sido realizado como un diagrama UML, ya que es quizás más adecuado para plasmar este tipo de bases de datos.

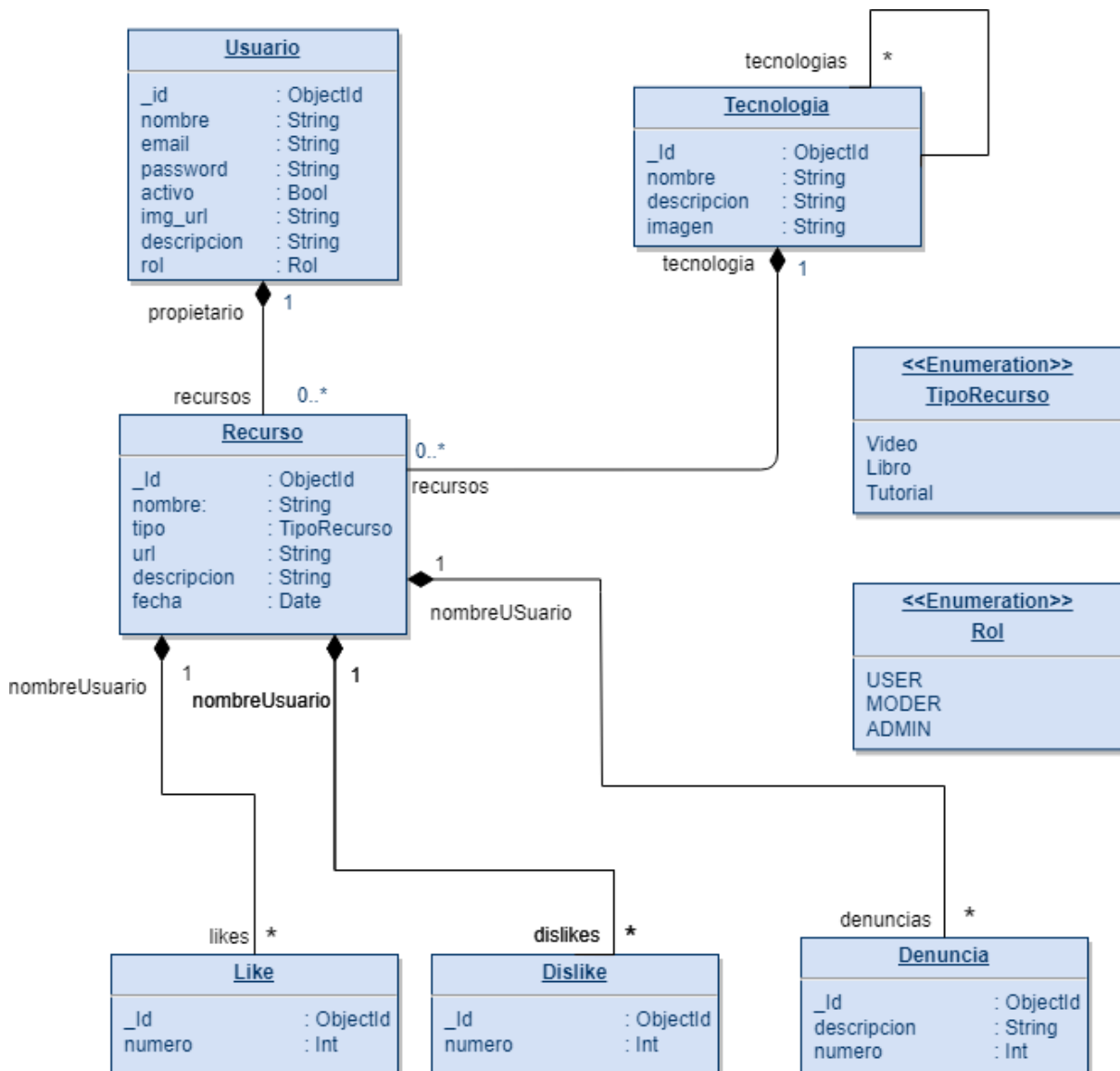


Figura 6: Diagrama de la Base de Datos

### Enumeración Rol

Enumeración con los tipos de roles que puede tener un usuario. En la aplicación se contemplan 3 roles distintos:

- **USER:** Usuario registrado en la aplicación pero que no tiene ningún tipo de permiso.
- **MODER:** El usuario con rol de moderador tiene ciertos permisos sobre la aplicación.
- **ADMIN:** El usuario con rol de administrador tiene todos los permisos sobre la aplicación.

### **Enumeración Tipo Recurso**

Enumeración con los tipos de recurso que se pueden crear. En la aplicación se contemplan 3 tipos de recursos distintos:

- **VÍDEO:** Un recurso puede ser un vídeo de Youtube que se quiera recomendar.
- **LIBRO:** Un recurso puede ser un libro que se quiera recomendar.
- **TUTORIAL:** Un recurso puede ser un tutorial, este puede ser:
  - **Externo:** Un tutorial escrito en otra página.
  - **Propio:** Un recurso puede contener un tutorial que es escrito en la propia página con Markdown.

### **Clase Usuario**

Representa a un usuario registrado de la aplicación web y viene dado por los siguientes atributos:

- **\_id:** Id creado por defecto por MongoDB para todos sus objetos. Este id ayuda a identificar cada objeto de forma inequívoca. Este id es de tipo “ObjectId”, propio de MongoDB.
- **nombre:** Atributo de tipo “String” que representa el nombre del usuario registrado. Es un atributo único, dos personas no pueden tener el mismo nombre de usuario.
- **email:** Atributo de tipo “String” que representa el email del usuario registrado. Es un atributo único, no se puede crear dos cuentas con el mismo email.
- **password:** Atributo de tipo “String” que representa la contraseña del usuario registrado. Las contraseñas son encriptadas con hash y salt.

- **activo:** Atributo de tipo “booleano” que representa si un usuario está activo o no. Un usuario que no es activo es aquel que ha decidido borrar su cuenta, pero no se borra de la base de datos ya que hay objetos que no pueden existir sin un usuario.
- **img\_url:** Atributo de tipo “String” que representa la url de la imagen del perfil del usuario. Esta imagen está alojada en cloudinary [30].
- **descripción:** Atributo de tipo “String” que representa la descripción del perfil del usuario.
- **rol:** Atributo de tipo “Enum[Rol]” que apunta a la enumeración Rol y representa el rol que tiene el usuario en la aplicación.
- **recursos:** Atributo de tipo “Array[Recurso]” que apunta a la clase Recurso, indica una colección de recursos creados por el usuario. Puede tener ninguno o muchos.

### **Clase Tecnología**

Representa una tecnología (Javascript, Python, ...) o un bloque de contenido (Machine learning, Algoritmia, ...) de la aplicación web y viene dado por los siguientes atributos:

- **\_id:** Id creado por defecto por MongoDB para todos sus objetos. Este id ayuda a identificar cada objeto de forma inequívoca. Este id es de tipo “ObjectId”, propio de MongoDB.
- **nombre:** Atributo de tipo “String” que indica el nombre de la tecnología o bloque de contenido.
- **tecnologías:** Atributo de tipo “Array[Tecnología]” que apunta a la clase Tecnología, indica las subtecnologías o asociadas a esta. Puede tener ninguno o muchos.
- **descripción:** Atributo de tipo “String” que indica la descripción de la tecnología o bloque de contenido.
- **imagen:** Atributo de tipo “String” que representa la url de la imagen de la tecnología o bloque de contenido. Esta imagen está alojada en cloudinary.
- **recursos:** Atributo de tipo “Array[Recurso]” que apunta a la clase Recurso, indica los recursos creados en esta tecnología o bloque de contenido. Puede tener ninguno o muchos.



## **Clase Recurso**

Representa los recursos creados por un usuario en cierta tecnología o bloque de contenido en la aplicación web y viene dado por los siguientes atributos:

- **\_id:** Identificador creado por defecto por MongoDB para todos sus objetos. Este id ayuda a identificar cada objeto de forma inequívoca. Este id es de tipo “ObjectId”, propio de MongoDB.
- **nombre:** Atributo de tipo “String” que indica el nombre del recurso. Debe tener entre 10 y 50 caracteres.
- **tipo:** Atributo de tipo “Enum[TipoRecurso]” que apunta a la enumeración TipoRecurso, que indica el tipo de recurso.
- **url:** Atributo de tipo “String” que indica la url del vídeo, libro o tutorial. Puede no haber url.
- **descripción:** Atributo de tipo “String” que indica la descripción del recurso.
- **fecha:** Atributo de tipo “Date” que indica la fecha de creación del recurso.
- **propietario:** Atributo de tipo “Usuario” que apunta a la clase Usuario, indicando quien es el creador del recurso. El usuario es obligatorio, dado que un recurso tiene que estar creado por algún usuario.
- **tecnología:** Atributo de tipo “Array[Tecnología]” que apunta a la clase Tecnología, indicando a qué tecnología o bloque de contenido pertenece este recurso. La tecnología es obligatoria, dado que un recurso tiene que pertenecer a alguna tecnología.
- **likes:** Atributo de tipo “Array[Like]” que apunta a la clase Like, indica los likes que ha recibido un recurso. Puede tener ninguno o muchos.
- **dislikes:** Atributo de tipo “Array[Dislike]” que apunta a la clase Dislike, indica los dislikes que ha recibido un recurso. Puede tener ninguno o muchos.
- **denuncias:** Atributo de tipo “Array[Denuncia]” que apunta a la clase Denuncia, indica las denuncias que ha recibido un recurso. Puede tener ninguna o muchas.

### **Clase Like**

- **número:** Atributo de tipo “Int” que indica el número de likes que se le ha dado a un recurso.
- **recurso:** Atributo de tipo “Recurso” que apunta a la clase Recurso, indica el recurso al cual se le ha dado like. Un like siempre tiene que tener asociado un recurso.

### **Clase Dislike**

- **número:** Atributo de tipo “Int” que indica el número de dislikes que se le ha dado a un recurso.
- **recurso:** Atributo de tipo “Recurso” que apunta a la clase Recurso, indica el recurso al cual se le ha dado dislike. Un dislike siempre tiene que tener asociado un recurso.

### **Clase Denuncia**

- **número:** Atributo de tipo “Int” que indica el número de denuncias que ha recibido un recurso.
- **descripción:** Atributo de tipo “String” que indica el motivo por el que se va a denunciar un recurso.
- **recurso:** Atributo de tipo “Recurso” que apunta a la clase Recurso, indica el recurso al cual se ha denunciado. Una denuncia siempre tiene que tener asociada un recurso.

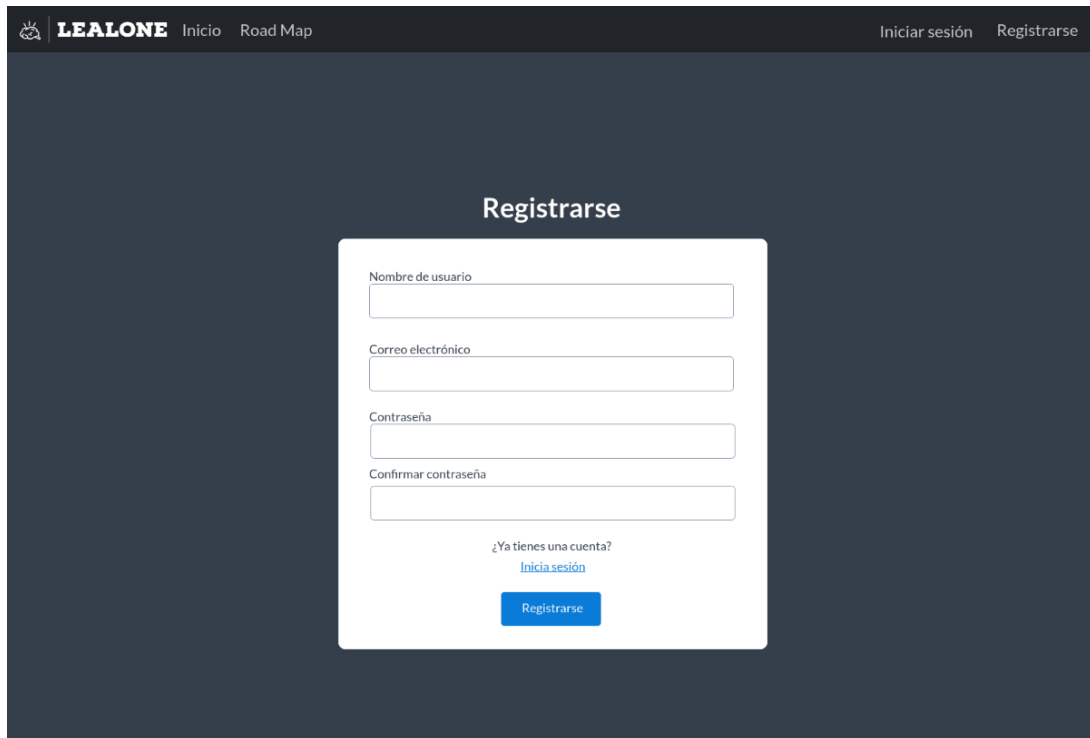
## **4.3. Mock-ups**

Un mock-ups es un prototipo de la aplicación que se realiza antes del desarrollo de la misma. Es un primer borrador de cómo va a ser la aplicación. Hay varios tipos de mock-ups pero en este proyecto se ha decidido abordar unos de los más visuales y prácticos, ya que no solo se ha realizado el diseño de las distintas páginas de la aplicación, sino que también es posible ver cómo interactúan los distintos elementos entre sí.

El mock-up sirve para ver cómo podría quedar la aplicación una vez acabada, lo cual es muy útil para enseñárselo al cliente en etapas tempranas del desarrollo. Si el cliente acepta el

mock-up se puede empezar la fase de implementación, con una idea clara de como se quiere que se vean e interactúen los diferentes elementos de la interfaz, además de tener la confianza de saber que ese diseño ha sido validado por el cliente.

En el siguiente [enlace](#) es posible ver el prototipo interactivo de la página. Además de eso, a continuación también hay imágenes de las distintas pantallas de la aplicación.



LEALONE Inicio Road Map Iniciar sesión Registrarse

### Registrarse

Nombre de usuario

Correo electrónico

Contraseña

Confirmar contraseña

¿Ya tienes una cuenta?  
[Inicia sesión](#)

[Registrarse](#)

Figura 7: Página en la que el usuario puede crear una cuenta

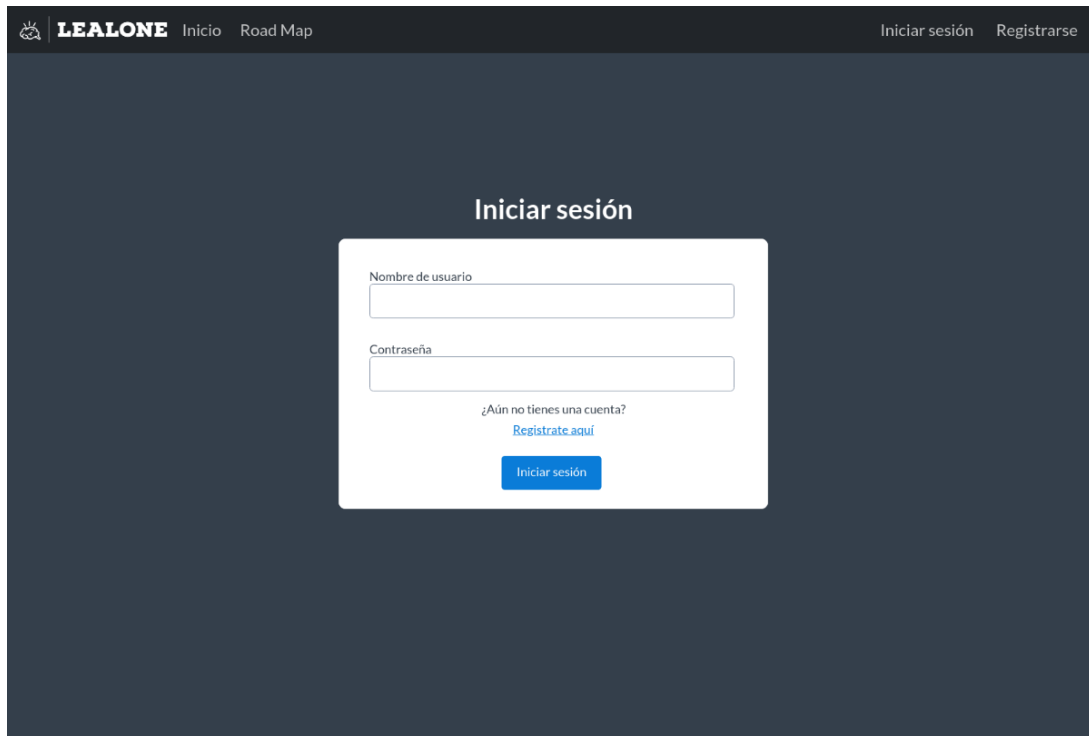


Figura 8: Página en la que el usuario puede iniciar sesión

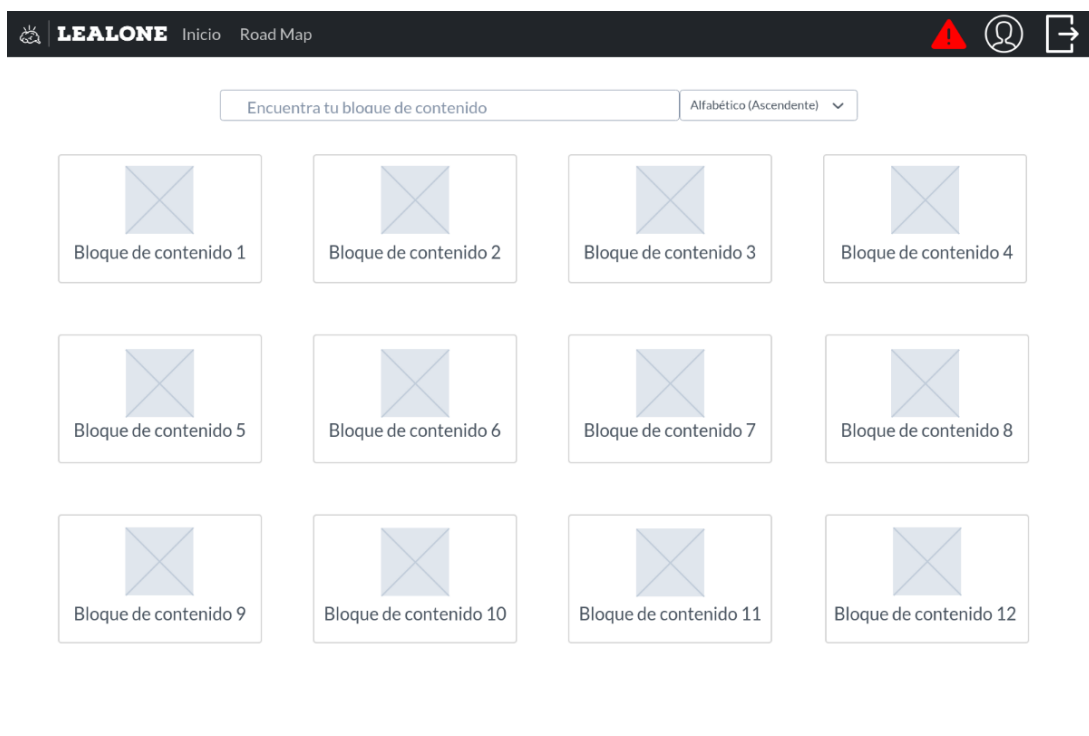


Figura 9: Página en la que el usuario puede ver los bloques de contenido

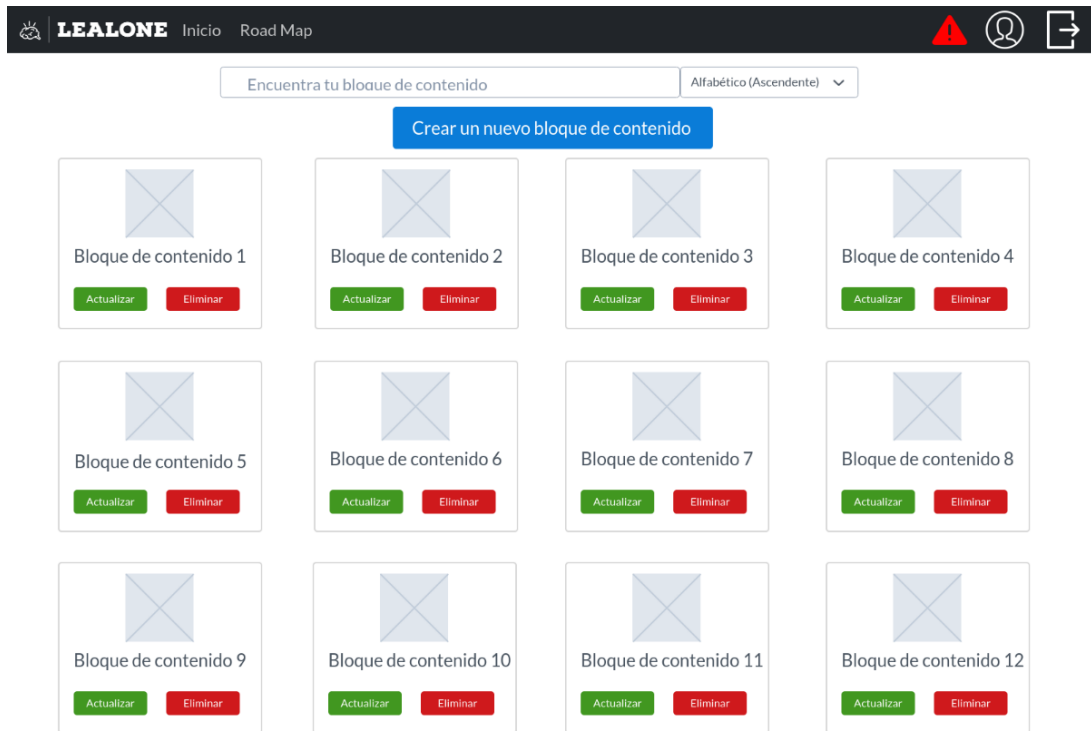


Figura 10: Página en la que el administrador puede ver los bloques de contenido

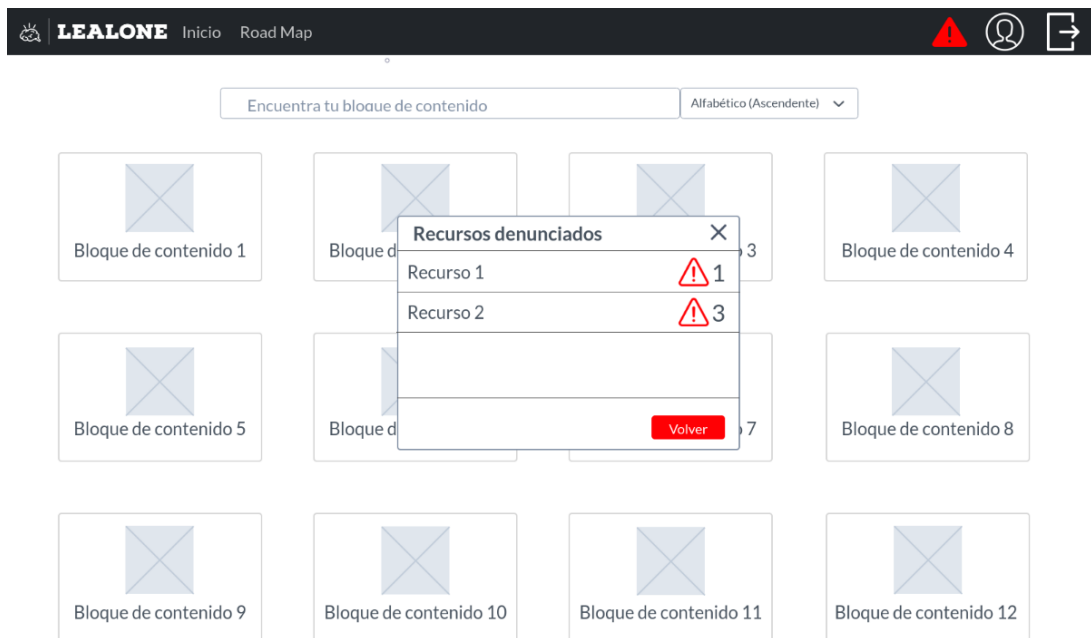


Figura 11: Página en la que el administrador y los moderadores pueden ver las denuncias

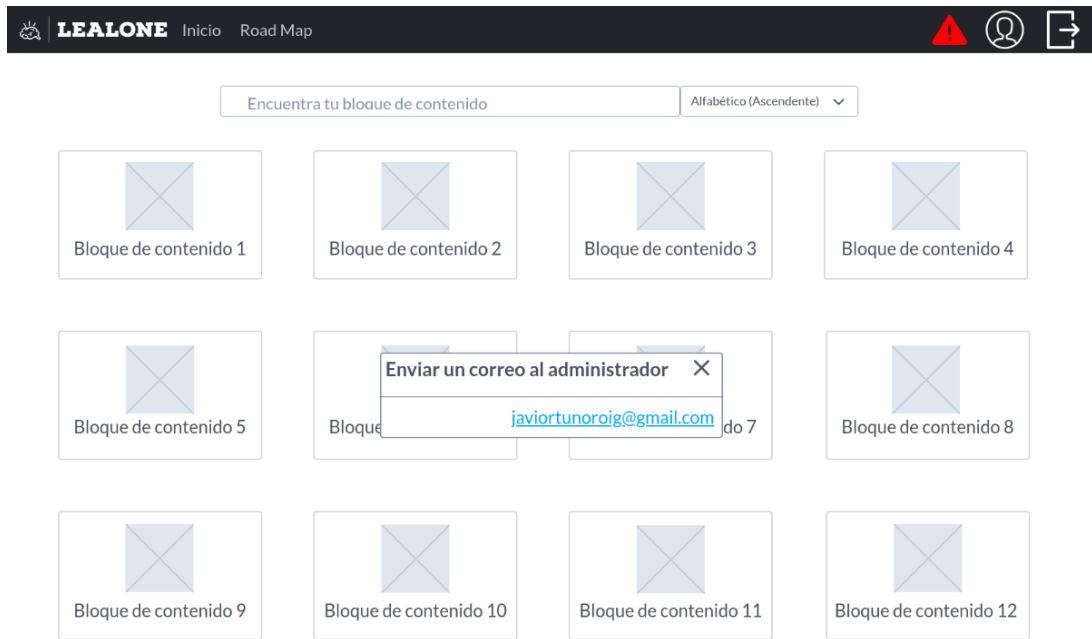


Figura 12: Página en la que el usuario puede contactar con el administrador

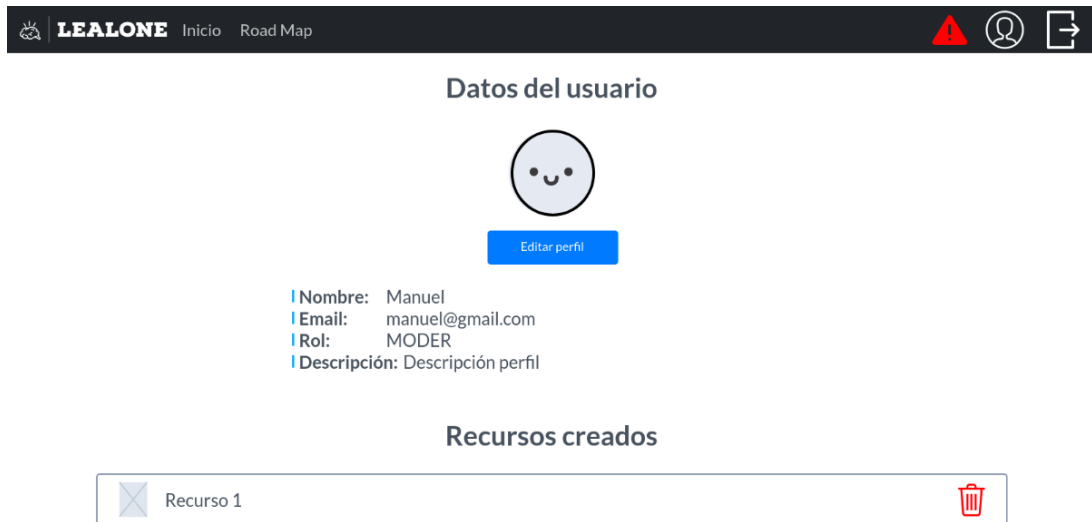


Figura 13: Página en la que el usuario puede ver su perfil

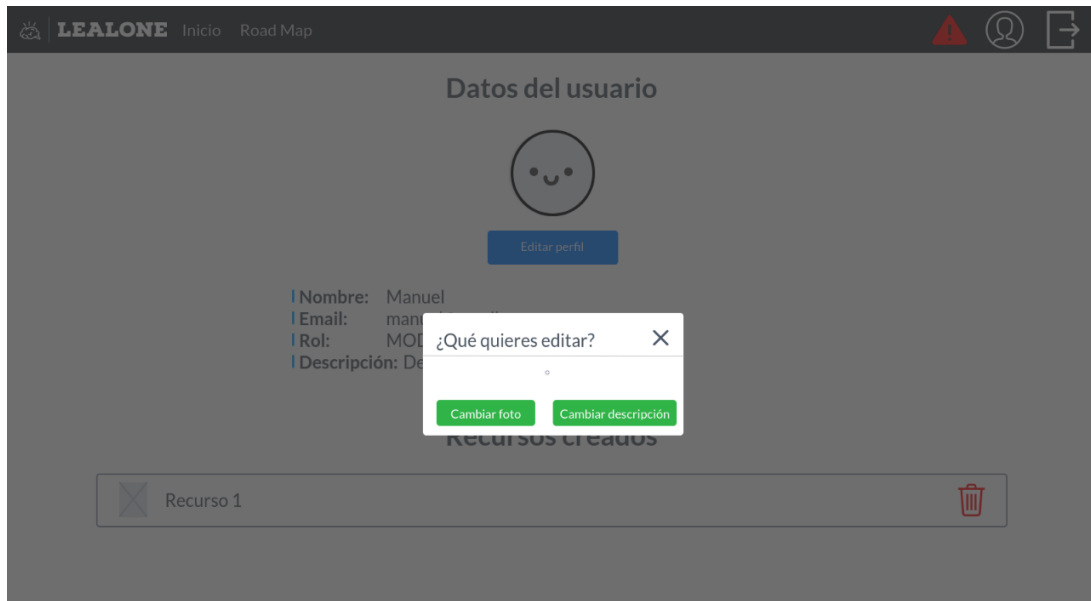


Figura 14: Página en la que el usuario puede elegir lo que quiere actualizar de su perfil



Figura 15: Página en la que el usuario puede actualizar su foto de perfil

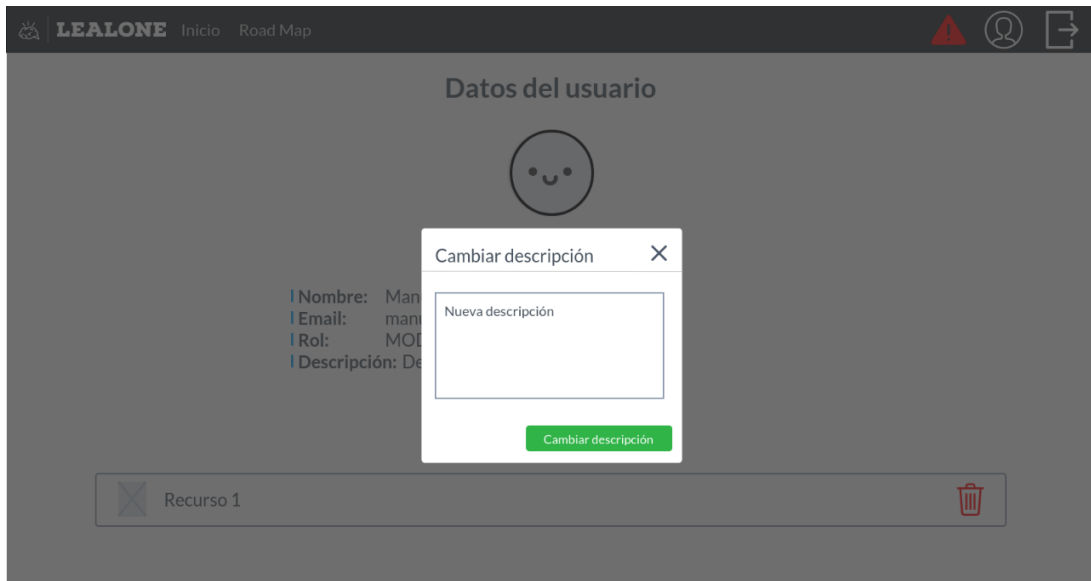


Figura 16: Página en la que el usuario puede actualizar la descripción de su perfil

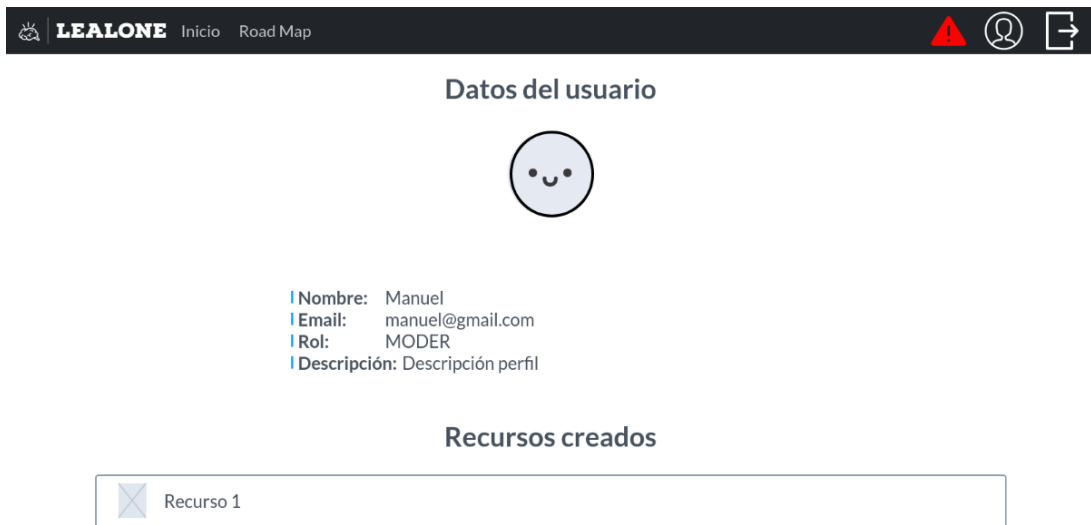


Figura 17: Página en la que el usuario puede ver el perfil de otro usuario



### Nombre del bloque de contenido



Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed tincidunt congue ligula in rutrum. Morbi nec lacus condimentum, hendrerit mi eu, feugiat.

### Librerías o Frameworks basados en "nombre del bloque de contenido"



Bloque de contenido

Crear recurso

### Recursos de "Bloque de contenido"

Fecha (Descendente) ▾

 Recurso 1	 0  3
 Recurso 2	 5  2

Figura 18: Página en la que el usuario puede ver la información de un bloque de contenido

### Crea un nuevo bloque de contenido

Nombre del bloque de contenido:

Descripción:

Imagen del bloque de contenido:

Arrastra una imagen o pinche aquí

Crear bloque de contenido

Figura 19: Página en la que el administrador puede crear un bloque de contenido

### Actualizar el nuevo bloque de contenido

Nombre del bloque de contenido:

Descripción:

Imagen del bloque de contenido:

Arrastra una imagen o pinche aquí

Actualizar el bloque de contenido

Figura 20: Página en la que el administrador puede actualizar un bloque de contenido



Figura 21: Página en la que el usuario puede crear un recurso

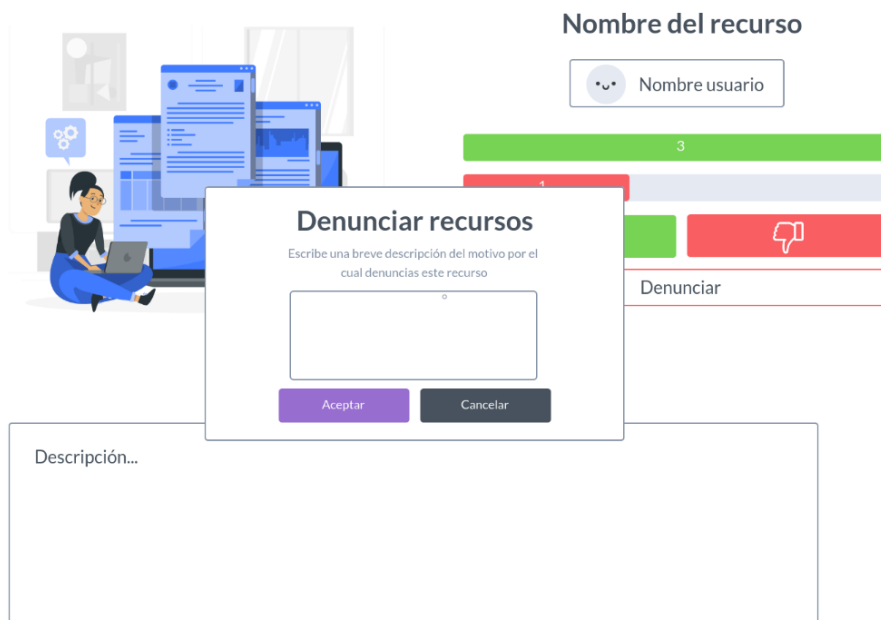


Figura 22: Página en la que el usuario puede crear una denuncia para un recurso

## ¿Que recurso quiere crear para Algoritmia?

Escriba el título del recurso

Elija el tipo del recurso

Figura 23: Página en la que el usuario puede crear un recurso

## ¿Que recurso quiere crear para Algoritmia?

Escriba el título del recurso

Elija el tipo del recurso

Seleccione una opcion	▼
video de Youtube	
Libro	
Escribir un tutorial	

Figura 24: Página en la que el usuario puede elegir qué recurso crear

### ¿Que recurso quiere crear para Algoritmia?

Escriba el título del recurso

Elija el tipo del recurso

Video de Youtube ▼

[Crear](#)

[Texto](#) [Markdown](#)

Descripción del recurso

### Configuración del recurso de tipo Video

URL del video de youtube

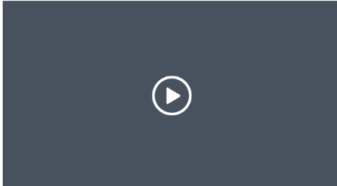
  


Figura 25: Página en la que el usuario puede crear un recurso de tipo vídeo

### ¿Que recurso quiere crear para Algoritmia?

Escriba el título del recurso

Elija el tipo del recurso

Libro ▼

[Crear](#)

[Texto](#) [Markdown](#)

Descripción del recurso

### Configuración del recurso de tipo Libro

URL donde se puede comprar o leer (si es online) del libro

Figura 26: Página en la que el usuario puede crear un recurso de tipo libro

### ¿Que recurso quiere crear para Algoritmia?

Escriba el título del recurso

Elija el tipo del recurso

Descripción del recurso

### ¿Es un tutorial propio o es de otra página web?

URL donde se puede comprar o leer (si es online) del libro

Figura 27: Página en la que el usuario puede crear un recurso de tipo tutorial propio

### ¿Que recurso quiere crear para Algoritmia?

Escriba el título del recurso

Elija el tipo del recurso

Descripción del recurso

### ¿Es un tutorial propio o es de otra página web?

Tipo de tutorial

URL donde se puede comprar o leer (si es online) del libro

Figura 28: Página en la que el usuario puede crear un recurso de tipo tutorial externo



### ¿Qué es un roadmap?

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed tincidunt congue ligula in rutrum. Morbi nec lacus condimentum, hendrerit mi eu, feugiat.

### ¿De donde son los roadmap?

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed tincidunt congue ligula in rutrum. Morbi nec lacus condimentum, hendrerit mi eu, feugiat.

Roadmap 1

Roadmap 2

Roadmap 3

Roadmap 4

Roadmap 5

Figura 29: Página en la que el usuario puede visualizar la información de los roadmaps

# 5

## Implementación

### 5.1. Preparación para el proyecto

Antes de empezar a crear todo el proyecto hay una serie de instalaciones y configuraciones que se han tenido que hacer.

#### **MongoDB Atlas**

El proyecto usa MongoDB como base de datos, por lo tanto uno de los primeros pasos es crear una cuenta en MongoDB Atlas, en la nube (aunque también es posible instalar MongoDB en un servidor) y, posteriormente, crear un clúster. Un clúster es donde creamos nuestras bases de datos y documentos.

También se ha descargado MongoDB Compass como programa cliente de visualización de datos, es decir, el sitio donde se puede observar el estado actual de la base de datos, crear reglas para los datos, probar consultas, etc.

#### **Node**

Node es imprescindible para crear y desarrollar la aplicación. Es el motor de ejecución usado en este proyecto para instalar librerías y ejecutar el código. Además, la instalación de Node se realiza a la par que NPM, este es el sistema de gestión de paquetes elegido para instalar librerías que ayuden en el desarrollo del proyecto.

#### **Cloudinary**

Dado que los usuarios pueden tener una foto de perfil y que los bloques de contenido también tiene una, es necesario usar algún servicio de hospedaje (hosting) multimedia. En este caso, el hosting elegido ha sido Cloudinary, el cual permite de forma sencilla subir y descargar fotos y vídeos a su nube a través de su librería para Javascript.



## Git y Github

Un proyecto de esta magnitud está en constante cambio, por ello es indispensable usar algún software de control de versiones como puede ser Git. Esto permite tener control sobre las distintas versiones del proyecto, identificar el cambio que hay entre una versión y otra a través de mensajes escritos por el usuario entre un cambio y otro, etc [31].

A parte de Git, se ha usado Github para alojar el código en internet, además de la potencia que le añade a Git. Con Github se pueden ver los cambios entre versiones y cualquier otra información de git de forma más visual y clara, es posible crear issues y pull request, hacer integración continua o crear tableros Kanban [32].

## Heroku

Una vez que el proyecto está acabado, es necesario alojar la aplicación web en algún sitio web de internet, para esto se ha elegido Heroku. Con Heroku es posible publicar la aplicación en internet de forma gratuita, usar plugins para realizar procesos complejos, usar variables de entorno, SSH, etc [33].

## 5.2. Estructura del proyecto

El proyecto está dividido en dos grandes partes diferenciables: el frontend y el backend.

El frontend ha sido creado con el comando oficial de React “npx create-react-app nombre\_del\_proyecto”. Con este comando se creará un proyecto base de React, preparado para empezar a desarrollar la aplicación y para hacer pruebas, sin necesidad de ninguna configuración extra. Con el comando “npm run start” un servidor es abierto en el puerto 3000 en el que es posible ver el estado actual de la aplicación, además detecta automáticamente los cambios en los archivos y se actualizará el servidor.

El backend ha sido creado y configurado desde cero. Principalmente se sustenta en 2 pilares, el servidor creado con Express y la base de datos con MongoDB y mongoose.

Toda la configuración del servidor ha sido creado en un archivo en el que se declara el objeto “Server” con toda la lógica necesaria para configurar y crear el servidor.

```
constructor () {  
  this.port = process.env.PORT || 8080;  
  this.app = express();
```

```

this . middleware ( ) ;
this . routes ( ) ;

// Base de datos
dbConnection ( ) ;
}

```

Algoritmo 7: Parte principal del objeto “Servidor”

En el algoritmo anterior se puede observar cómo es elegido el puerto del servidor, la creación del mismo y cómo se le indican los componentes de middleware y de enrutado.

## Middleware

Son funciones que son ejecutadas antes de que una petición sea procesada. Un ejemplo práctico es un middleware de Express el cual convierte el cuerpo de una petición en un objeto JSON para su posterior lectura al tratar la petición y crear una respuesta.

```

middleware ( ) {

// CORS
this . app . use ( cors ( ) ) ;

// Directorio con recursos público
this . app . use ( express . static ( path . join ( __dirname , '../
public' ) ) ) ;

// Lectura y parseo del body
this . app . use ( express . json ( ) ) ;

// Permite el uso de x-www-form-urlencoded
this . app . use ( express . urlencoded ( { extended : true } ) ) ;

// Validación de caracteres , quitando $ para evitar

```

### inyección de código en MongoDB

```
this.app.use( validarCaracteresMongoDB );  
}
```

#### Algoritmo 8: Middelwares del servidor

### Enrutado

Las rutas del servidor son las rutas a las que llegarán las peticiones del cliente. Una ruta tiene la dirección de la misma, middleware (si procede) y una función que será ejecutada para cuando llegue una petición a la ruta y que devolverá una respuesta.

```
routes() {  
  
  // Rutas normales  
  this.app.use( '/api/auth', require( '../routes/auth' ) );  
  this.app.use( '/api/tecnologias', require( '../routes/  
tecnologia' ) );  
  this.app.use( '/api/recursos', require( '../routes/recurso'  
  ) );  
  
  // Ruta para limpiar datos de la base de datos de prueba  
  if ( process.env.node_env === 'test' ) this.app.use( '/api/  
reset', require( '../routes/reset' ) )  
  
  // En caso de hacer una petición GET a una ruta inexistente  
  se ejecutará la siguiente función  
  this.app.get( '/*', function( req, res ) {  
  
    res.sendFile( path.join( __dirname, '../public/index.html'  
  ), function( err ) {  
    if ( err ) res.status( 500 ).send( err )  
  });  
}
```

```
    });  
  }  
}
```

### Algoritmo 9: Rutas del servidor

#### Nodemon

Aparte de la creación del servidor hay otro elemento importante, una librería que ha facilitado enormemente el desarrollo de la aplicación. Nodemon es una librería que está a la espera de que un archivo sea editado y en caso de que se edite, relanzará el servidor que ahora incluirá los nuevos datos [34].

#### Mongoose

Por último, se ha creado un clúster en MongoDB Atlas, que para el uso que se le ha dado, es gratuito. Después de crear un clúster solo queda conectar el servidor con la base de datos, para ello se ha usado Mongoose a través de las siguientes líneas.

```
require( 'dotenv' ).config();  
const mongoose = require( 'mongoose' );  
  
// Se conecta a una BD u otra dependiendo de si queremos usar  
// los datos verdaderos o los datos de pruebas para los test.  
const connectionString = process.env.node_env === 'test' ?  
  process.env.DB_CNN_TEST : process.env.DB_CNN;  
  
const dbConnection = async () => {  
  try {  
    // Se abre la conexión con la base de datos  
    await mongoose.connect( connectionString );  
  } catch ( e ) {  
    throw new Error( 'Error a la hora de conectar a la BD' );  
  }  
}
```

```
};
```

## Algoritmo 10: Configuración de la base de datos

### 5.2.1. Seguridad

Por último, se ha de resaltar tres medidas de seguridad tomadas en el proyecto para asegurar protección frente a ataques a la base de datos.

#### Inyecciones NoSQL

Uno de los ataques más famosos son las inyecciones SQL. Este tipo de ataque nació en 1998 por una publicación de Jeff Forristal, que exponía por primera vez una vulnerabilidad de inyección SQL. Las inyecciones SQL es un método para ejecutar código no deseado al realizar una consulta SQL. Con este método es posible borrar tablas y bases de datos, crear nuevos valores u obtener toda la información de la base de datos.

De forma análoga, años más tarde, con la creación de las bases de datos noSQL, también se crearon inyecciones noSQL. Para poder evitar este tipo de ataque se ha creado un middleware que revisar los datos enviados por el usuario para reemplazar los “\$” de las peticiones por otro tipo de caracteres. Los dolares son peligrosos porque permiten ejecutar sentencias “\$where()”, por ejemplo.

```
const validarCaracteresMongoDB = ( req , res , next ) => {  
  
  const { body , query } = req;  
  
  let bodyString = JSON.stringify( body );  
  let queryString = JSON.stringify( query );  
  
  while ( bodyString.includes( '$' ) ) {  
  
    bodyString = bodyString.replace( '$', '-' );  
  
  }  
}
```

```

while ( queryString.includes( '$' ) ) {

    queryString = queryString.replace( '$', '-' );

}

req.body = JSON.parse( bodyString );
req.query = JSON.parse( queryString );

next();

};

```

Algoritmo 11: Middleware que sanea las peticiones

## Bcrypt

Como se ha mencionado anteriormente en el apartado de tecnologías, se ha usado la técnica bcrypt para encriptar las contraseñas de los usuarios, aumentando la seguridad de sus cuentas en el caso de que se reciba un ataque a la base de datos.

Esta técnica de encriptación no solo usa hash sino que también salt. Para su implementación en el proyecto se ha usado la librería [bcryptjs](#), la cual proporciona métodos predefinidos que se encargan de encriptar los datos y también de comparar los datos encriptados. Como se puede comprobar a continuación el código necesario para encriptar y comparar los datos es muy sencillo.

```

const usuarioNuevo = new Usuario( { nombre, email, password } );
const salt = bcrypt.genSaltSync();
usuarioNuevo.password = bcrypt.hashSync( password, salt );
await Usuario.create( usuarioNuevo );

```

Algoritmo 12: Encriptación de datos

```
const validPassword = bcrypt.compareSync( password , usuario .  
password );
```

Algoritmo 13: Comparación de un dato con otro encriptado

## JWT

Json Web Token (JWT) es un estándar abierto que define una forma autocontenida y compacta para transmitir información de forma segura en formato JSON. Este método es seguro ya que es firmado digitalmente con HMAC, RSA o ECDSA, que son diferentes técnicas criptográficas.

Los JWT son útiles para autenticación de usuarios. Una vez que el usuario haya iniciado sesión, cada solicitud posterior incluirá el JWT, lo que le permitirá acceder a las rutas, servicios y recursos que están permitidos con ese token.

En el lado del frontend se ha guardado el token del usuario cada vez que este iniciaba sesión o se creaba una cuenta nueva. Este token es enviado siempre que se vaya a hacer una consulta al backend.

En el backend se ha creado un middleware que recoger el token del usuario y hace varias comprobaciones, si todo está correcto deja pasar la petición a la ruta correspondiente, sino, devuelve un error. Este middleware solo se usa en las rutas protegidas.

```
const checkJWT = ( req , res , next ) => {  
  
  const bearerToken = req.headers.authorization ;  
  if ( !bearerToken ) return responseErrorJWT( res , 401 , false ,  
    'No se ha enviado el token de acceso , intente iniciar  
    sesión' );  
  
  const token = bearerToken.split( ' ' )[1];  
  if ( !token ) return responseErrorJWT( res , 401 , false , 'No  
    se ha enviado el token de acceso con el formato correcto ,  
    intente iniciar sesión' );
```

```

console.log( `Token: ${token}` );

try {

    const payload = jwt.verify( token , process.env.JWT_SECRET ,
    {} );
    Usuario.findById( payload.id )
        .then( usuario => {

                if ( !usuario ) return responseErrorJWT( res , 401 ,
false , 'Error con el payload del token , intente iniciar
sesión ' );

                if ( usuario.token !== token ) return responseErrorJWT(
res , 401 , false , 'El token no coincide con el del usuario ,
intente iniciar sesión ' );

                next();

            }
        ) );

} catch ( error ) {

    console.log( error );

    if ( error.name === 'TokenExpiredError' ) return
responseErrorJWT( res , 401 , false , 'El token ha expirado ' );
    if ( error.name === 'JsonWebTokenError' ) return
responseErrorJWT( res , 401 , false , 'El token no es válido ' )
;

}

```



```
};
```

Algoritmo 14: Middleware que comprueba el token enviado en la petición

### 5.3. Metodología de trabajo

A lo largo del proyecto se ha trabajado con distintas iteraciones o sprints de 2 a 3 semanas cada uno. Al principio de cada sprint se ha realizado un análisis de cuáles debían de ser los requisitos o tareas a implementar en dicho sprint.

Al tener el análisis del trabajo a realizar en las siguientes semanas, se procedía a añadirlo en el backlog (o pila de tareas). Este backlog es una de las características que proporciona Github llamada "Projects". En cada repositorio se pueden crear varios proyectos, estos proyectos pueden ser usados como un backlog dentro del propio repositorio, pudiendo prescindir así de herramientas externas como Trello. El backlog proporcionado por Github detecta automáticamente las incidencias (issues) creadas en el proyecto, las cuales pasan al estado "To-do". Por tanto, cuando estaba claro las tareas que había que realizar se creaban distintas issues en el proyecto, asignándole títulos, descripciones, etiquetas claras e hitos claros.

Conforme se empezaba a trabajar en una issue se movía al estado "Doing" y una vez cerrada la issue automáticamente pasaba al estado "Done".

Para mantener el trabajo más organizado, las issues han sido resueltas a través de pull request, permitiendo así saber qué cambios han sido realizados para cada issue del proyecto. Una vez que todas las issues estaban en el estado "Done", se procedía a empezar un nuevo sprint, repitiendo este proceso hasta el final del proyecto.

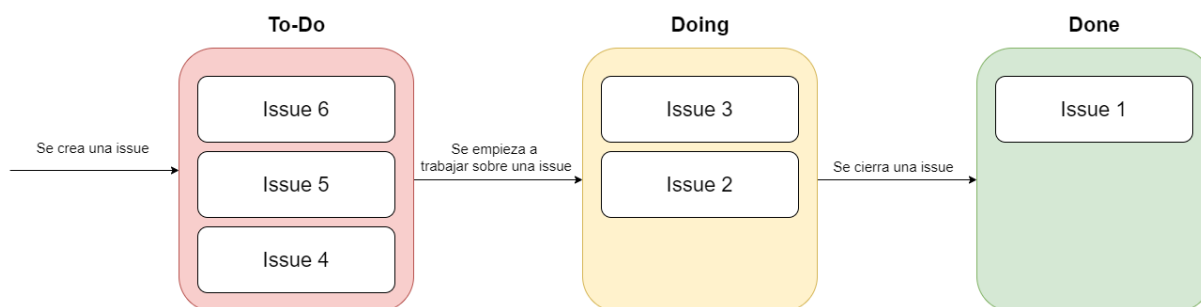


Figura 30: Ejemplo de funcionamiento del proyecto

## 5.4. Diagrama de despliegue

Los diagramas de despliegue son un poco diferente a otros que muestran la parte lógica del sistema. Por el contrario, este diagrama se centra más en representar la parte física. En palabras simples, el diagrama de despliegue especifica qué componentes software se van a ejecutar en qué hardware en particular. Esto se podrá visualizar a través de nodos y artefactos que se componen y comunican entre sí.

Un nodo es dispositivo o componente hardware donde se ejecuta algún programa, ejecutable, librería, etc. Estos programas son llamado, artefactos.

Con este tipo de diagramas se puede tener una vista sencilla y clara sobre como está desplegada nuestra aplicación en internet y como se comunica con otros sistemas, como por ejemplo, el backend comunicándose con la base de datos.

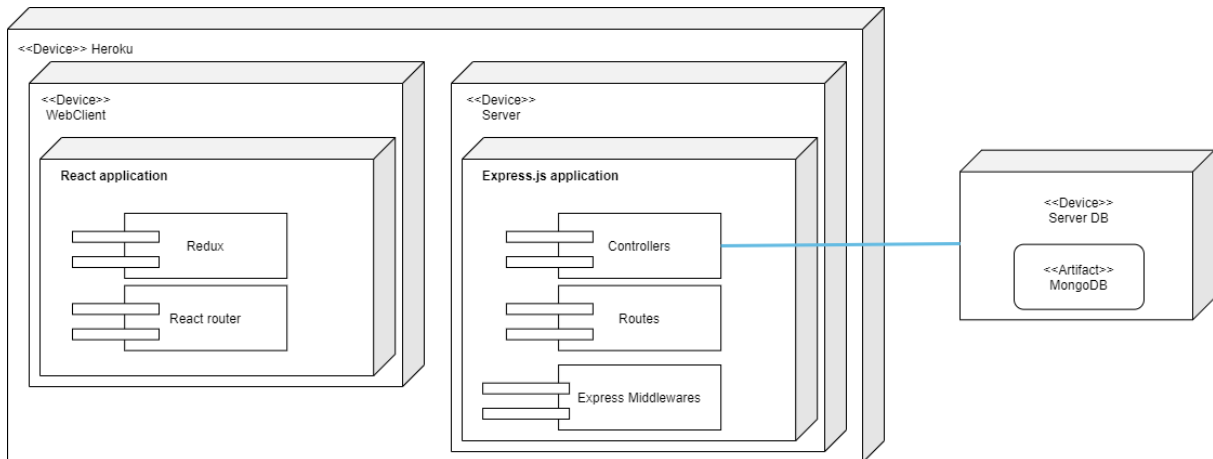


Figura 31: Diagrama de despliegue del proyecto



# 6

## Pruebas

Una de las tareas más importantes a la hora de hacer cualquier aplicación son las pruebas. Las pruebas consiste en comprobar que distintos aspectos de la aplicación funcionen correctamente, esto permite seguir añadiendo funcionalidades o refactorizar código, con la confianza de que si alguna otra parte del código dejar de funcionar seremos capaces de identificarlo rápidamente gracias al conjunto de prueba realizado. Esto conlleva una serie de beneficios:

- Reduce el coste de mantenimiento del proyecto.
- Mejora la calidad del código gracias a la detección temprana de errores.
- Es posible trabajar más ágil gracias a la confianza que tenemos para cambiar partes del código.
- Si se sigue una metodología TDD (Test Driven Development) se conseguirá tener un software robusto y confiable desde etapas tempranas.

Existen varios tipos de pruebas, explicaré, resumidamente, algunas de las más importantes:

- **Pruebas unitarias:** Comprueba la correcta funcionalidad de una unidad de código.
- **Pruebas de integración:** Comprueba que todos los elementos unitarios funcionen correctamente en conjunto.
- **Pruebas de sistemas:** Comprueba la integración del sistema globalmente, verificando el funcionamiento correcto de las interfaces entre los distintos subsistemas que lo componen y con el resto de sistemas de información con los que se comunica.
- **Pruebas de implantación:** Comprueba el correcto funcionamiento del hardware y software en un entorno de producción.

- **Pruebas de aceptación:** Comprueba que se cumplen los distintos requisitos de la aplicación. Son las más cercanas al cliente.

## Niveles de prueba

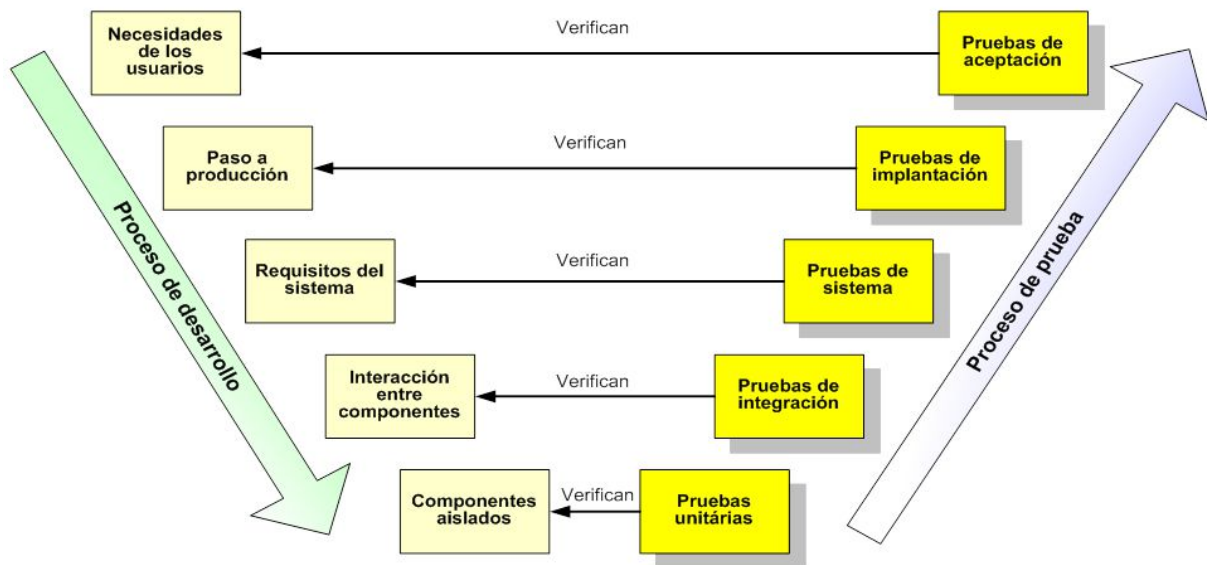


Figura 32: Tipos de pruebas y procesos que verifican

Para este proyecto se han realizado pruebas de integración para los distintos endpoints de la API que ofrece el backend y pruebas unitarias del frontend.

### 6.1. Backend - Pruebas de integración

Para hacer estas pruebas se ha hecho uso de dos librerías: Jest y Supertest. La primera, como se ha mencionado anteriormente, es un framework que permite hacer pruebas sobre nuestro código. Por otro lado, Supertest nos proporciona herramientas fáciles y efectivas para hacer peticiones a la API y probar la respuesta de dichas peticiones. La configuración necesaria para las pruebas es mínima, simplemente se crea un archivo “.test.js”, añadir un “beforeEach”

para poder establecer datos falsos antes de cada prueba y un “afterAll” para cerrar la conexión con la base de datos al final de las pruebas.

```
const supertest = require( 'supertest' );
const api = supertest( app );

beforeEach( async ( ) => {

  await Recurso.deleteMany( {} );
  await Tecnologia.deleteMany( {} );
  await Usuario.deleteMany( {} );

  await Usuario.insertMany( usuarios );
  await Tecnologia.insertMany( tecnologias );
  await Recurso.insertMany( recursos );

});

afterAll( async ( ) => {

  await mongoose.disconnect();
  await mongoose.connection.close();
  await serverCreated.close();

});
```

#### Algoritmo 15: Configuración de pruebas con Jest y Supertest

En los test con Supertest lo primero es hacer una llamada a la API y posteriormente se comprueban los datos de la respuesta para ver si son los esperados. Un test de integración se podría ver de la siguiente manera.

```
test( 'Un usuario no puede iniciar sesión con un usuario que no existe', async ( ) => {
```

```

const response = await api.post( '/api/auth' )
  .send({
    nombre: 'norma',
    password: '654321'
  });

expect( response.status ).toBe( 400 );
expect( response.body.ok ).toBe( false );
expect( response.body.msg )
  .toBe( 'El usuario con ese nombre no existe' );
});

```

Algoritmo 16: Test que comprueba que al iniciar sesión con un usuario inexistente salte cierto mensaje de error

## 6.2. Frontend - Pruebas de aceptación

Para realizar las pruebas de aceptación se ha hecho uso de Cypress. Con Cypress es posible hacer pruebas en las que se imita el comportamiento del usuario y se hacen comprobaciones sobre dichos comportamientos.

Junto a Cypress se ha usado un plugin del mismo, para poder especificar las pruebas en Gherkin, creando así una especificación de pruebas en lenguaje natural y que sea más fácil de entender para el cliente u otros interesados que no entiendan mucho sobre informática [35].

Se ha elegido Cypress para la realización de estas pruebas en vez de otras soluciones como Playwright por la amigable interfaz gráfica que nos ofrece ya que podemos ver un timeline con todas las acciones que se han hecho a la hora de ejecutar cada sentencia de cada prueba, por lo que hace muy sencillo encontrar los fallos que ocurran.

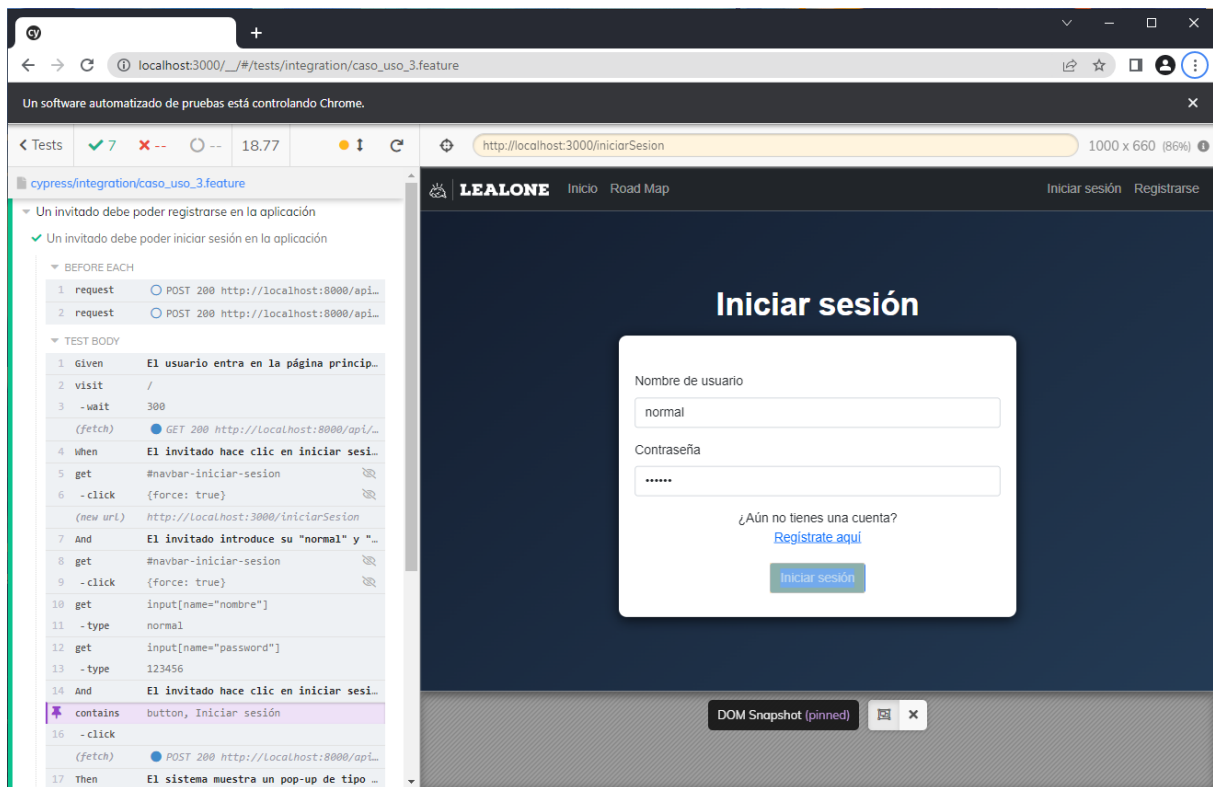


Figura 33: Interfaz de Cypress

Las especificación en Gherkin coinciden con la parte de escenarios de pruebas de los casos de uso.

**Feature:** Un usuario debe poder contactar con los moderadores y el administrador, a través de un correo

**Scenario:** Contactar con el administrador o los moderadores

**Given** El usuario entra en la página principal

**When** El invitado hace clic en iniciar sesión en el navbar

**And** El invitado introduce su "normal" y "123456"

**And** El invitado hace clic en iniciar sesión

**And** El sistema muestra un pop-up de tipo "success" con el mensaje "Inicio de sesión exitoso"

**And** Se pulsa el pop-up

**And** El usuario hace clic en el botón danger

**Then** El sistema muestra el correo para contactar con el



administrador o los moderadores

Algoritmo 17: Especificación de la prueba para el caso de uso 8

Una vez especificada la prueba en Gherkin, usamos Cypress y el plugin mencionado antes para simular el comportamiento del navegador.

```
import {When, Then} from 'cypress-cucumber-preprocessor/steps';

When( 'El usuario hace clic en el botón danger', () => {
  cy.get( '#navbar-incidencias' )
    .click({ force: true }) )
}

Then( 'El sistema muestra el correo para contactar con el
administrador o los moderadores', () => {
  cy.get( '.modal-footer > a' )
    .should( 'have.text', 'javiortunoroig@gmail.com' ) );
}
```

Algoritmo 18: Test con Cypress para el caso de uso 8

# 7

## Conclusiones y Líneas Futuras

### 7.1. Conclusiones

Sin duda alguna en los últimos años ha habido un crecimiento exponencial en la tecnología en todo el mundo. En poco años ha habido un avance increíble tanto en hardware como en software, al punto de que los desarrolladores software se ha convertido en uno de los trabajos más demandados. Pero, el crecimiento del software también conlleva un gran crecimiento en cuanto a nuevos conocimientos que están en constante cambio: lo que sabías sobre un lenguaje hace 5 años es posible que ya no sirva de nada.

Ante este dilema, han surgido diversas plataformas y medios distintos para poder aprender todo este nuevo conocimiento. Plataformas de ejercicios de código, sitios web de e-learning, páginas con retos, de tutoriales, etc. El problema es que hay tanto contenido que es difícil saber cuál elegir. Este proyecto está orientado a intentar darle una solución a ese problema dando orientación sobre los mejores contenidos de internet, recomendados de los propios desarrolladores a otros que estén en proceso de aprendizaje.

Para la creación del proyecto se ha decidido usar una pila de programación muy usada, el “MERN Stack”. En el MERN Stack se usa:

- React, la librería más usada para la creación de interfaces.
- MongoDB, una de las bases de datos noSQL más usada actualmente, orientada a documentos y almacenando la información en formato BSON.
- Express, siendo el framework de Javascript más usado para el desarrollo de backend y APIs; es un framework muy sencillo, pero muy potente.

- Node, el entorno de ejecución de Javascript usado por la grandísima mayoría de desarrolladores.

El resultado final ha sido una aplicación web con un diseño simple, adaptable (responsive) y funcional, en la que desarrolladores pueden publicar sus mejores recursos en los bloques de contenido creados, recomendar nuevos bloques de contenido, puntuar recursos de otros desarrolladores y ver roadmaps para el bloque de contenido o tecnología que quiere aprender.

## 7.2. Líneas Futuras

Hay diversas características y funcionalidades que se podrían añadir a la aplicación web en un futuro.

### **Soporte para varios idiomas**

Actualmente la aplicación está solo disponible en español, por tanto sería una gran funcionalidad a añadir el poder cambiar entre varios idiomas. Como mínimo que esté el inglés, dado que es el idioma más globalizado actualmente.

### **Comentarios en los recursos publicados**

Actualmente la única forma de opinar sobre un recurso es a través de likes o dislikes, pero esto puede llegar a quedarse corto. Quizás un usuario quiere preguntar “¿Por qué es bueno este recurso?” ya que el recurso como tal no tiene ninguna descripción puesta. Por tanto, poder comentar en los recursos de otros desarrolladores hará la página más interactiva y dará lugar a otro medio para opinar sobre los recursos.

### **Modo oscuro**

En los últimos años se ha visto una tendencia en el software a permitir varias paletas de colores para la misma aplicación. Desde entonces, se podría decir que se han creado dos posicionamientos: las personas que les gusta el modo claro y las que les gusta el modo oscuro. Esto llega al punto de que hay extensiones del navegador que cambian los estilos de una página para ponerle estilos oscuros.

Por tanto, últimamente son más las aplicaciones que deciden dar a elegir al usuario que tipo de colores quiere en la aplicación, ejemplo de esto son grandes aplicaciones como Facebook,

Instagram, Twitter y el propio Whatsapp.

Entonces una buena característica a añadir a futuro sería poder elegir entre modo oscuro y modo claro, y no tener solo el actual modo claro.

### **Eliminar cuenta**

En la aplicación solo es posible crear y actualizar la cuenta del usuario, pero no es posible eliminarla. La aplicación ya está preparada para hacer este cambio en un futuro. En la figura 6, con la representación de la base de datos, se puede observar que los usuarios tienen un atributo activo, que se puso con la idea de poder borrar la cuenta en un futuro. Realmente borrar la cuenta, no es tan fácil como borrar al usuario, esto podría crear problemas con los recursos y tecnologías que había creado este usuario.

Por tanto, la solución que se suele tomar es poner un atributo booleano que indica si el usuario ha eliminado o no la cuenta. Cuando el usuario elimine la cuenta el valor del atributo cambiará y el servidor puede hacer uso de este atributo para no mostrar ninguna información de este usuario.

### **Mejorar la información ofrecida por los perfiles**

Una buena mejora sería poder añadir más información al perfil, por ejemplo para poner qué lenguajes dominas y cuántos años llevas usándolos. Esto podría servir para ver si un usuario está recomendando un recurso de una tecnología que domina o no.



# Referencias

- [1] *By The Numbers: MOOCs in 2021*. en-US. Dic. de 2021. URL: <https://www.classcentral.com/report/mooc-stats-2021/> (visitado 16-05-2022).
- [2] *World Wide Web Consortium (W3C)*. URL: <https://www.w3.org/> (visitado 16-05-2022).
- [3] *HTML*. es. Page Version ID: 143470132. Mayo de 2022. URL: <https://es.wikipedia.org/w/index.php?title=HTML&oldid=143470132> (visitado 16-05-2022).
- [4] *CSS*. es. Page Version ID: 140722926. Ene. de 2022. URL: <https://es.wikipedia.org/w/index.php?title=CSS&oldid=140722926> (visitado 16-05-2022).
- [5] *1.2. Breve historia de CSS (Introducción a CSS)*. es. URL: <https://uniwebsidad.com/libros/css/capitulo-1/breve-historia-de-css> (visitado 16-05-2022).
- [6] *JavaScript*. es. Page Version ID: 143460295. Mayo de 2022. URL: <https://es.wikipedia.org/w/index.php?title=JavaScript&oldid=143460295> (visitado 16-05-2022).
- [7] Marzo 23rd y 2017 | Menú Profesional. *Historia del lenguaje Javascript*. Mar. de 2017. URL: <https://utel.edu.mx/blog/menu-profesional/historia-del-lenguaje-javascript/> (visitado 16-05-2022).
- [8] *Node.js*. en. Page Version ID: 1087989453. Mayo de 2022. URL: <https://en.wikipedia.org/w/index.php?title=Node.js&oldid=1087989453> (visitado 16-05-2022).
- [9] *Node.js*. *Node.js*. es. URL: <https://nodejs.org/es/> (visitado 27-05-2022).
- [10] *Gherkin Syntax - Cucumber Documentation*. URL: <https://cucumber.io/docs/gherkin/> (visitado 27-05-2022).
- [11] *The State of JS 2021: Front-end Frameworks*. en. URL: <https://2021.stateofjs.com/en-US/libraries/front-end-frameworks/> (visitado 16-05-2022).
- [12] *Declarative routing for React apps at any scale | React Router*. en. URL: <https://reactrouter.com/> (visitado 27-05-2022).
- [13] *Read Me · Redux en Español*. URL: <https://es.redux.js.org/> (visitado 27-05-2022).
- [14] *Sass: Syntactically Awesome Style Sheets*. URL: <https://sass-lang.com/> (visitado 27-05-2022).

- [15] *Express - Infraestructura de aplicaciones web Node.js*. es. URL: <https://expressjs.com/es/> (visitado 27-05-2022).
- [16] *MongoDB: La Plataforma De Datos Para Aplicaciones*. es. URL: <https://www.mongodb.com/es> (visitado 27-05-2022).
- [17] *JSON And BSON*. en-us. URL: <https://www.mongodb.com/json-and-bson> (visitado 27-05-2022).
- [18] *Mongoose ODM v6.3.4*. URL: <https://mongoosejs.com/> (visitado 27-05-2022).
- [19] Deepanshu Dhruw. *ORM and ODM — A Brief Introduction*. en. Nov. de 2020. URL: <https://medium.com/spidernitt/orm-and-odm-a-brief-introduction-369046ec57eb> (visitado 27-05-2022).
- [20] *bcrypt*. en. URL: <https://www.npmjs.com/package/bcrypt> (visitado 27-05-2022).
- [21] *Jest. Delightful JavaScript Testing*. es-ES. URL: <https://jestjs.io/es-ES/> (visitado 04-06-2022).
- [22] *Meta Open Source está transfiriendo Jest a la Fundación OpenJS*. es. Mayo de 2022. URL: <https://ciberninjas.com/jest-transferido-fundacion-openjs/> (visitado 04-06-2022).
- [23] *Open Collective*. URL: <https://opencollective.com/> (visitado 04-06-2022).
- [24] *Jest - Open Collective*. URL: <https://opencollective.com/jest> (visitado 04-06-2022).
- [25] *Announcing Cypress 10 with Component Testing Beta!* Jun. de 2022. URL: <https://www.cypress.io/blog/2022/06/01/cypress-10-release/> (visitado 04-06-2022).
- [26] *JavaScript End to End Testing Framework | cypress.io testing tools*. URL: <https://www.cypress.io/> (visitado 27-05-2022).
- [27] *ESLint - Pluggable JavaScript linter*. en. URL: <https://eslint.org/> (visitado 27-05-2022).
- [28] *Cliente-servidor*. es. Page Version ID: 142472606. Mar. de 2022. URL: <https://es.wikipedia.org/w/index.php?title=Cliente-servidor&oldid=142472606> (visitado 27-05-2022).

- [29] *Representational state transfer*. en. Page Version ID: 1089945865. Mayo de 2022. URL: [https://en.wikipedia.org/w/index.php?title=Representational\\_state\\_transfer&oldid=1089945865](https://en.wikipedia.org/w/index.php?title=Representational_state_transfer&oldid=1089945865) (visitado 27-05-2022).
- [30] *Image and Video Upload, Storage, Optimization and CDN*. URL: <https://cloudinary.com/> (visitado 27-05-2022).
- [31] *Git*. URL: <https://git-scm.com/> (visitado 04-06-2022).
- [32] *Build software better, together*. en. URL: <https://github.com> (visitado 04-06-2022).
- [33] *Heroku | Login*. URL: <https://id.heroku.com/login> (visitado 04-06-2022).
- [34] *nodemon*. URL: <https://nodemon.io/> (visitado 04-06-2022).
- [35] Jonas Amundsen. *cypress-cucumber-preprocessor*. original-date: 2017-12-23T11:56:45Z. Jun. de 2022. URL: <https://github.com/badeball/cypress-cucumber-preprocessor> (visitado 04-06-2022).
- [36] Kamran Ahmed. *Developer Roadmaps*. URL: <https://roadmap.sh> (visitado 28-05-2022).





# Apéndice A

## Manual de Usuario

### A.1. Página principal

Al entrar en la página por primera vez el usuario se encuentra en la página principal en donde se puede ver todos los bloques de contenidos disponibles de la aplicación. Aquí, como invitado, es posible entrar en algún bloque de contenido y visualizar sus recursos. También desde el navbar es posible registrarse e iniciar sesión haciendo clic en los botones “Iniciar sesión” y “Registrarse”.

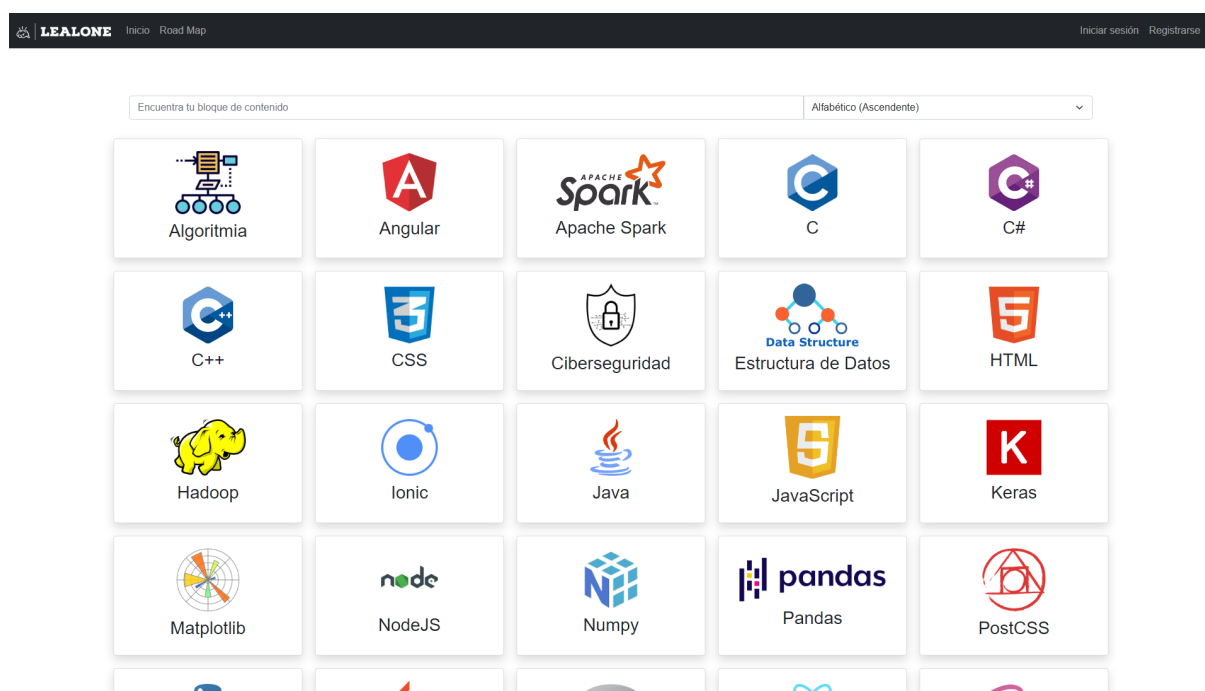


Figura 34: Página principal

### A.2. Registrarse

Si el usuario es nuevo, lo primero es crearse una nueva cuenta en la aplicación para tener las ventajas que esto supone. Para crearse una cuenta nueva hay que escribir:

- Nombre del usuario (no puede estar ya en uso).

- Email del usuario (no puede estar ya en uso).
- Contraseña de la cuenta.
- Escribir otra vez la misma contraseña para asegurarse que la contraseña es la deseada.

Al hacer clic en “Registrarse” se creará la cuenta y el usuario será redirigido a la página principal. También es posible ir a la página de inicio de sesión haciendo clic en “Inicia sesión”.

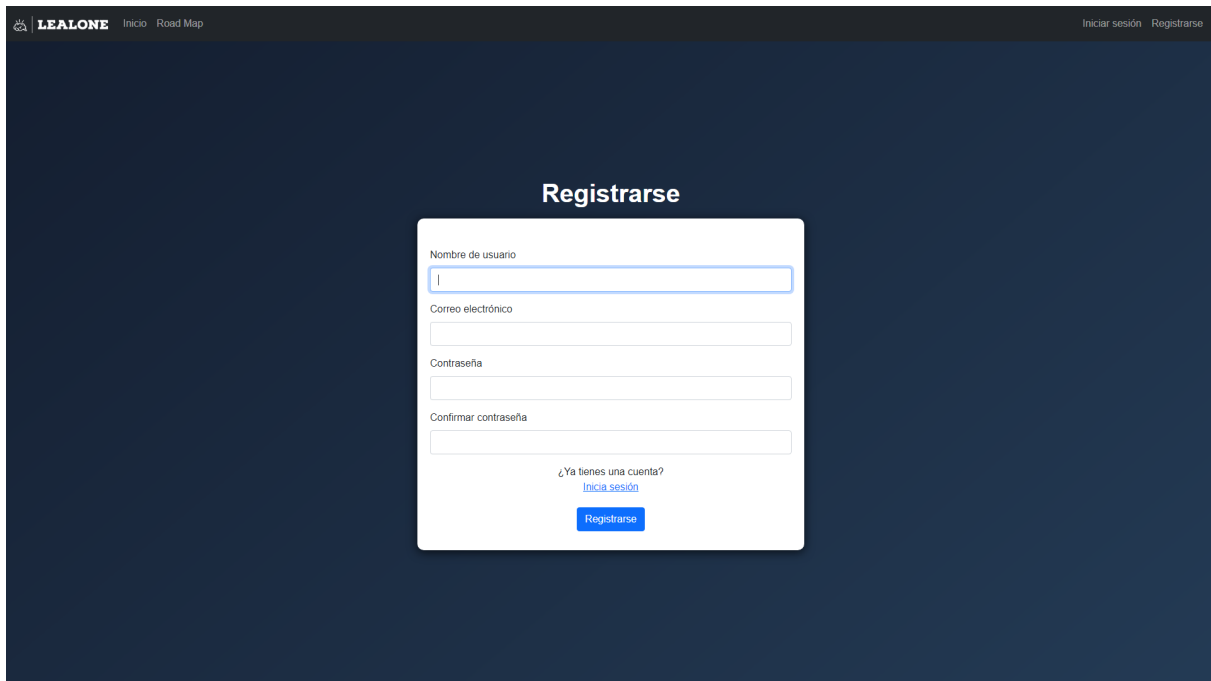


Figura 35: Página de registro

### A.3. Iniciar sesión

Si el invitado ya ha creado una cuenta anteriormente entonces puede iniciar sesión en la aplicación. Para ello debe de poner:

- Nombre del usuario (no puede estar ya en uso).
- Contraseña de la cuenta.

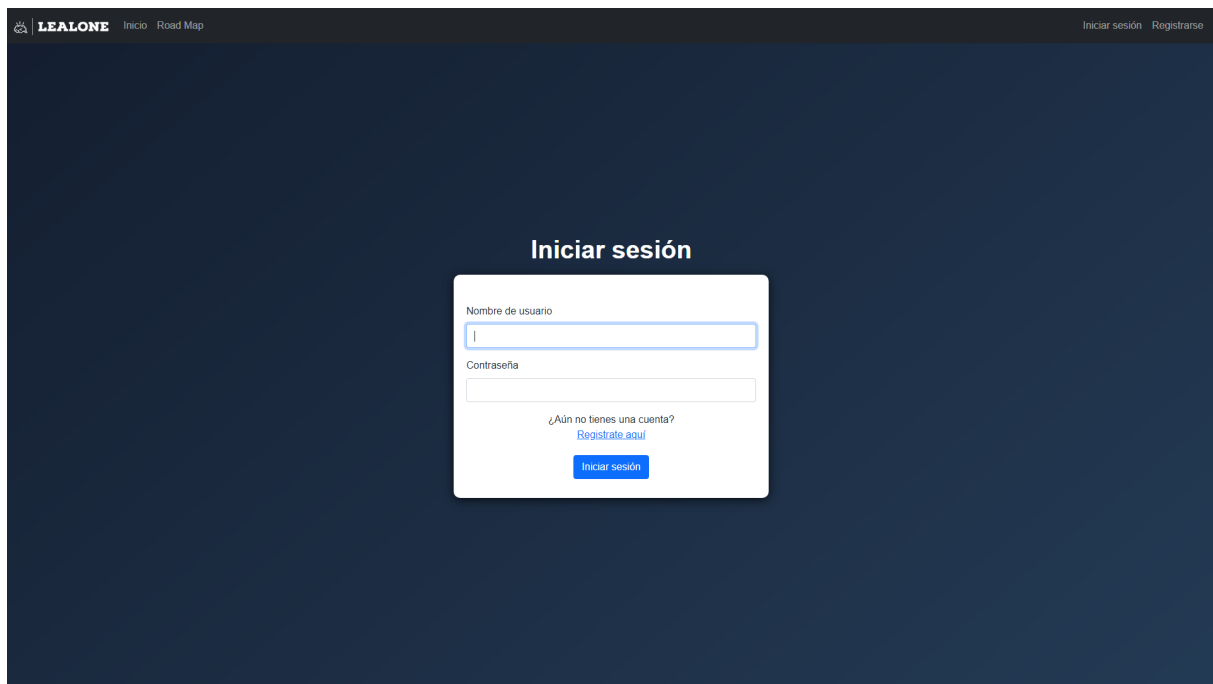


Figura 36: Página de inicio de sesión

Al hacer clic en “Iniciar sesión” el usuario inicia sesión en la aplicación y será redirigido a la página principal. También es posible ir a la página de registro haciendo clic en “Regístrate aquí”.

#### A.4. Roadmaps

Cualquier usuario puede acceder a los roadmaps desde el navbar, haciendo clic en “Road Map”. Un roadmap es una ruta a seguir para lograr un objetivo. En este caso, hay una página que tiene varios roadmaps hechos para aprender lenguajes de programación, librerías, etc. Para este proyecto, dirigido a aprender lenguajes y librerías nuevas, esta es una página muy interesante y por ello se le ha dedicado una sección entera con todos los roadmaps de la página y un enlace a la misma [36].



Figura 37: Página donde están todos los Roadmaps

## A.5. Navbar para usuarios que han iniciado sesión

Un usuario que ha iniciado sesión dispondrá de un navbar distinto a los invitados. Hay 3 botones adicionales que aparecen al iniciar sesión:

- Cerrar sesión: Este botón cierra la sesión del usuario y redirige a la pantalla de inicio.



Figura 38: Botón para cerrar sesión

- Acceder al perfil del usuario, donde se puede visualizar toda la información relativa al mismo y editar dicha información.



Figura 39: Acceder al perfil privado del usuario

- Este botón tiene 2 funciones dependiendo de los permisos del usuario.



Figura 40: Botón para ver las denuncias / contactar al administrador

- Si el usuario no tiene ningún permiso, este botón le permitirá al usuario contactar con el administrador a través de un correo.

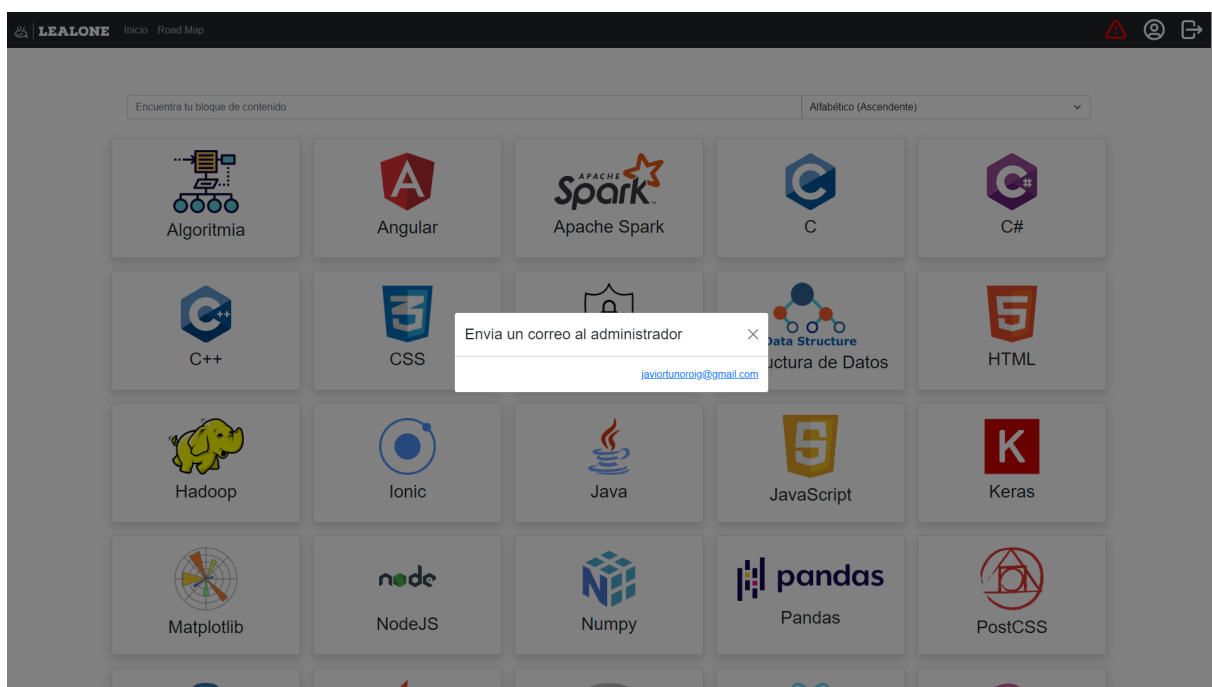


Figura 41: Pop-up con el correo del administrador para contactar con él

- Si el usuario tiene permisos, este botón le permitirá ver los recursos que han sido denunciados, el número de denuncias de cada recurso y los motivos de dichas denuncias.

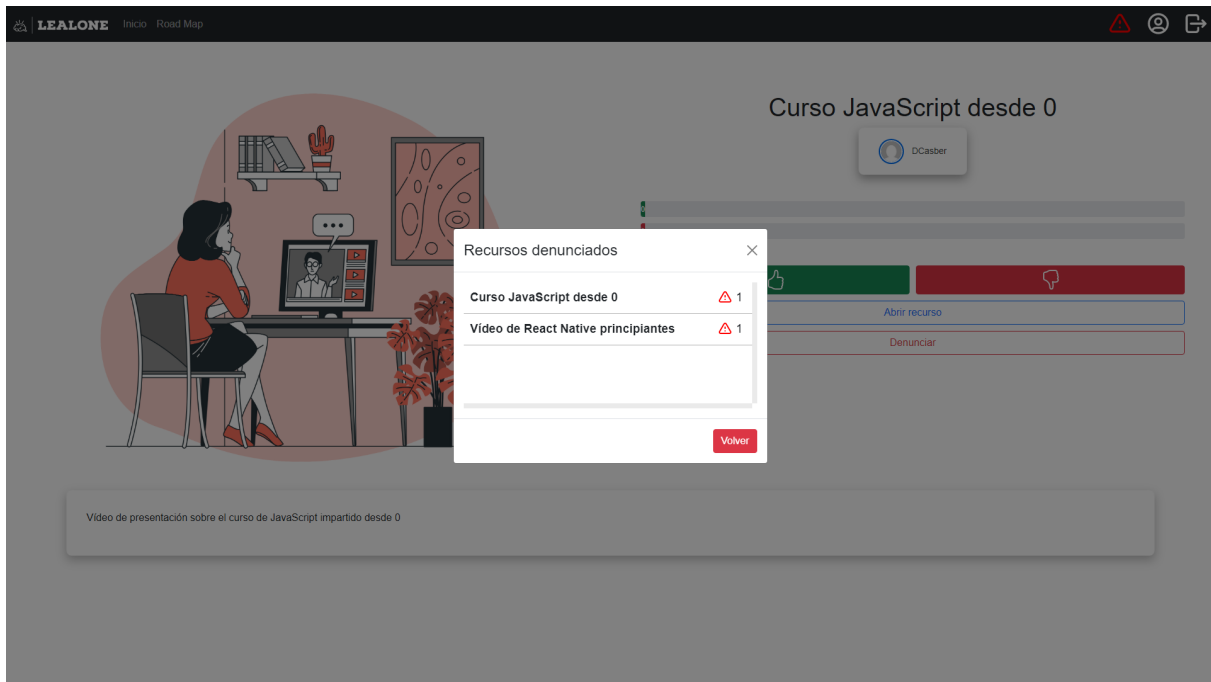


Figura 42: Pop-up que muestra todos los recursos que han sido denunciados y cuantas denuncias tienen

## A.6. Bloques de contenido

Cualquier usuario podrá ver información relativa a los bloques de contenido creados en la página como puede ser una imagen, nombre, descripción, sub-bloques de contenido y los recursos sobre dicho bloque de contenido. En esta pantalla un usuario podrá acceder a un recurso ya creado y ordenar estos recursos por el número de likes o la fecha de creación. También, si el usuario ha iniciado sesión puede crear un recurso haciendo clic en “Crear un nuevo recurso”.



Figura 43: Página con la información sobre un bloque de contenido

## A.7. Crear y ver recursos

Al hacer clic en “Crear un nuevo recurso”, se abre una ventana donde el usuario puede:

- Escribir el nombre del recurso, de entre 10 y 50 caracteres.
- Escribir una descripción en formato Markdown que no puede estar vacía. Se puede pre-visualizar el formato Markdown haciendo clic en “Markdown”.



LEALONE Inicio Road Map

### Configuración del recurso de tipo Video

¿Que recurso quiere crear para JavaScript?

Escriba el título del recurso

Un buen video sobre react ✓

Elija el tipo del recurso


Video de Youtube

Crear

URL del video de youtube

[https://www.youtube.com/watch?v=T\\_j60n1zgu0&t=4s](https://www.youtube.com/watch?v=T_j60n1zgu0&t=4s) ✓

Esta URL pertenece a un video de Youtube



Escriba una descripción del video de Youtube

Texto Markdown

Este es muy buen video para aprender React

Cosas positivas:

- Gratuito
- El profesor es un experto
- Empezar desde cero

Figura 44: Previsualización de la descripción de un recurso en markdown

- El usuario debe elegir si quiere recomendar un vídeo, libro o tutorial. Si elige tutorial tendrá que decir si es externo o propio.
  - En caso de elegir vídeo, libro o tutorial externo, debe de poner la url del recurso que quiere compartir. En caso de ser un vídeo de Youtube se podrá ver una previsualización del mismo para comprobar que es el correcto.
  - En el caso de ser un tutorial propio, el usuario no debe de escribir ninguna url. El usuario escribirá el tutorial en el textarea de la descripción.

Para crear el recurso el usuario debe de hacer clic en “Crear”.

LEALONE Inicio Road Map

Configuración del recurso de tipo Video

¿Que recurso quiere crear para JavaScript?

Escriba el título del recurso

Un buen video sobre react ✓

Elija el tipo del recurso


Video de Youtube

Crear

URL del video de youtube

https://www.youtube.com/watch?v=T\_j60n1zgu0&t=4s ✓

Esta URL pertenece a un video de Youtube



Escriba una descripción del video de Youtube

Texto Markdown

Este es muy buen video para aprender React

Cosas positivas:

- \* Gratuito
- \* El profesor es un experto
- \* Empieza desde cero

Figura 45: Página para crear un nuevo recurso

El usuario también puede ver los recursos haciéndoles clic desde el bloque de contenido donde se ha creado o desde su perfil, si quiere ver sus propios recursos. Al entrar en un recurso se muestra la siguiente información:

- Nombre del recurso.
- Descripción del recurso.
- Nombre del creador del recurso.
- Número de likes.
- Número de dislikes.

Desde esta página también se puede:

- Dar like a un recurso haciendo clic en el botón con un pulgar hacia arriba.
- Dar dislike a un recurso haciendo clic en el botón con un pulgar hacia abajo.
- Denunciar un recurso haciendo clic en “Denunciar”.

- Abrir la url del recurso, si no es un tutorial propio, haciendo clic en “Ver recurso”.
- Acceder al perfil del usuario que ha creado el recurso haciendo clic sobre su nombre.

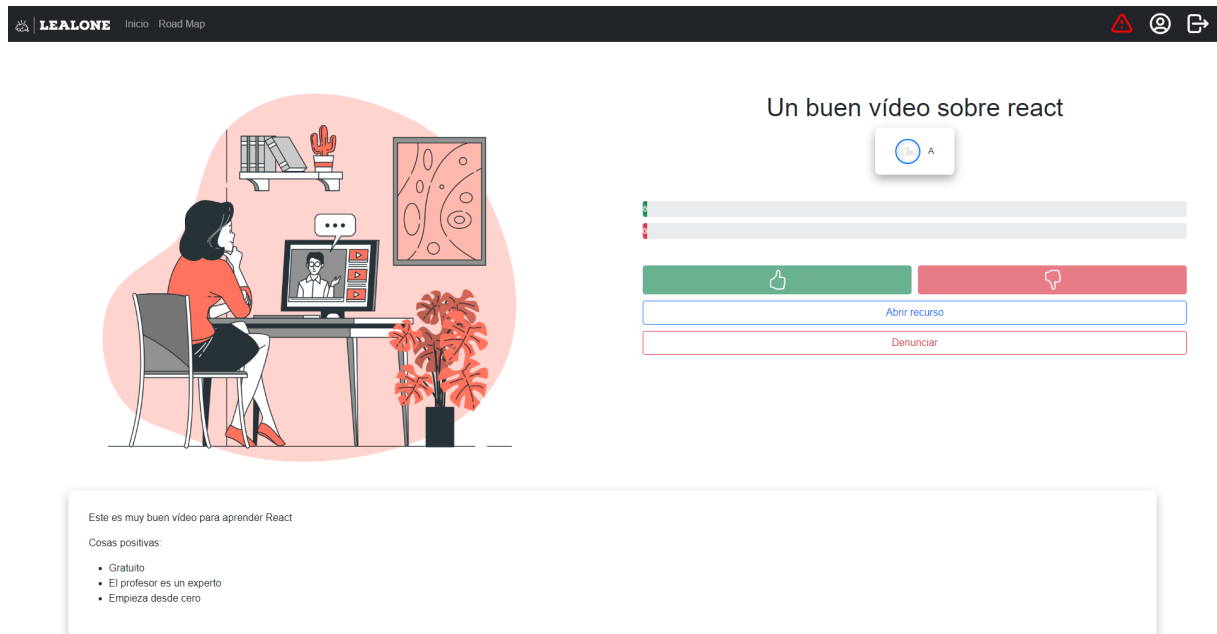


Figura 46: Página donde se puede visualizar la información de un recurso

## A.8. Denunciar

Dentro de la página de un recurso, un usuario puede denunciarlo por algún motivo de peso, como puede ser un título, descripción o url inapropiado. Para esto el usuario accede al recurso que quiere denunciar y hace clic en “Denunciar”. Al hacer esto se abrirá un pop-up donde se pone el motivo de la denuncia y para crearla solo hace falta darle a “Aceptar”

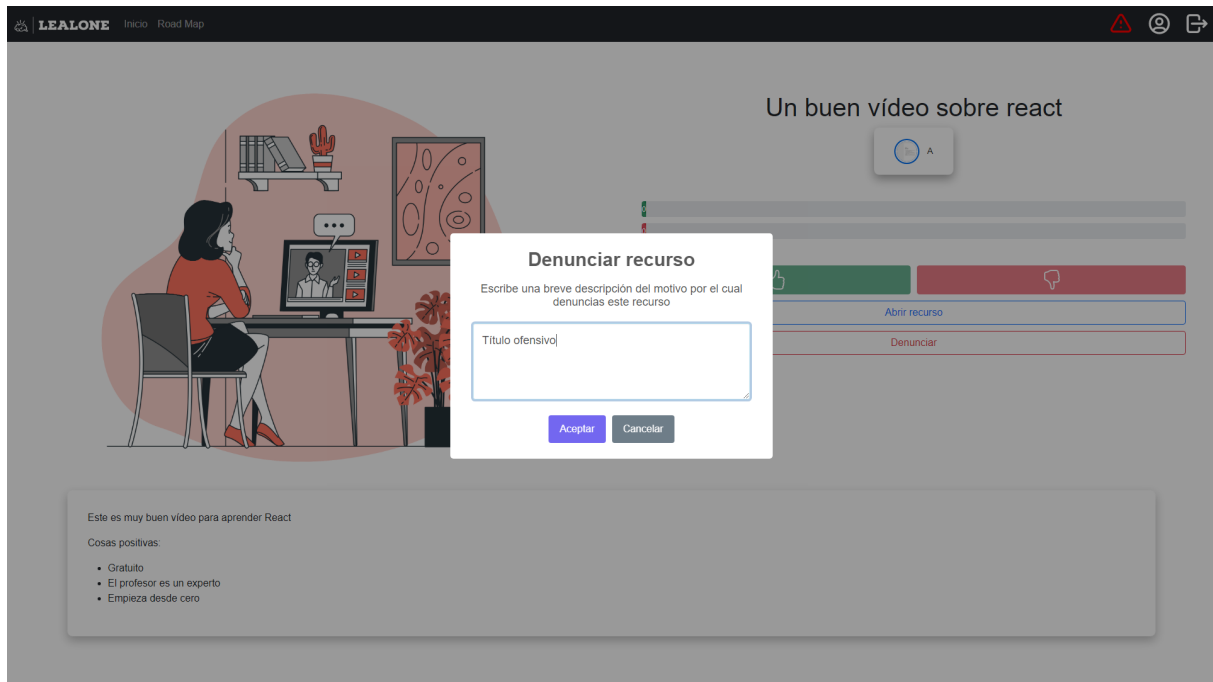


Figura 47: Pop-up para crear una denuncia

### A.8.1. Perfiles

Hay dos tipos de perfiles, el primero es el perfil privado donde puede ver la información de tu cuenta, como puede ser el nombre, descripción, recursos, etc, además de poder editar dichos datos. El segundo es el perfil público, el cual muestra la información del usuario igual que el privado, solo que no es posible editar dicha información.

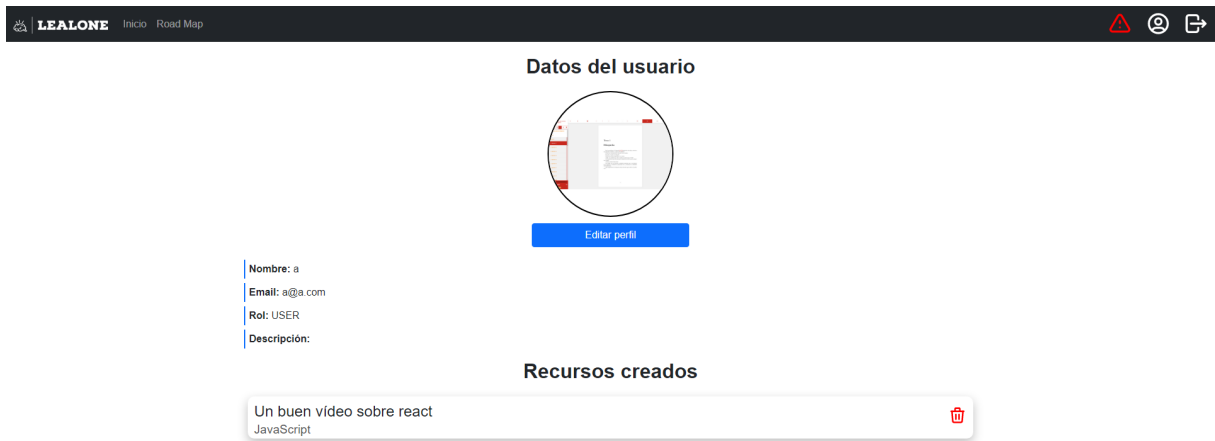


Figura 48: Página del perfil privado de un usuario

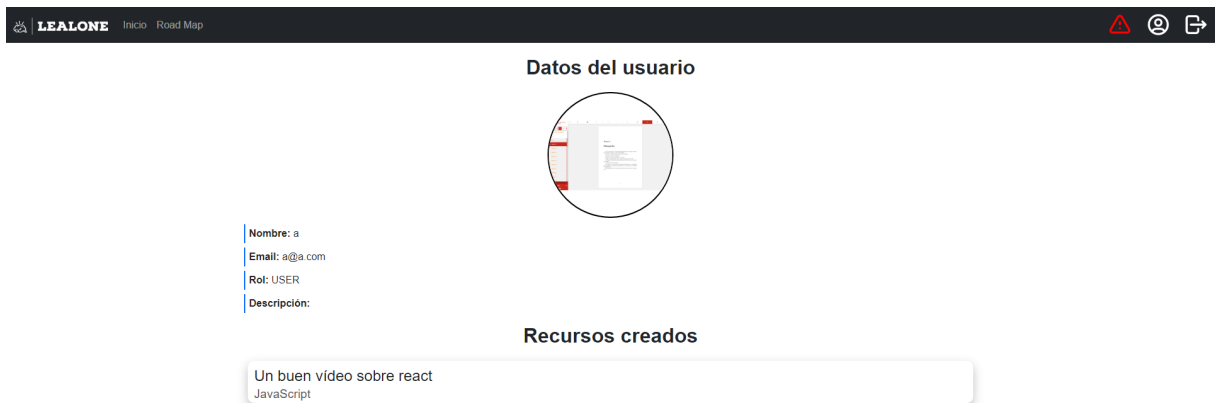


Figura 49: Página del perfil público de un usuario

Para acceder al perfil privado solo hay que hacer clic al botón del navbar que muestra a una persona.



Figura 50: Botón en el navbar para acceder al perfil privado de un usuario

Para acceder al perfil público se puede hacer mediante url o entrando desde un recurso, haciendo clic en el nombre del usuario que ha creado el recurso.

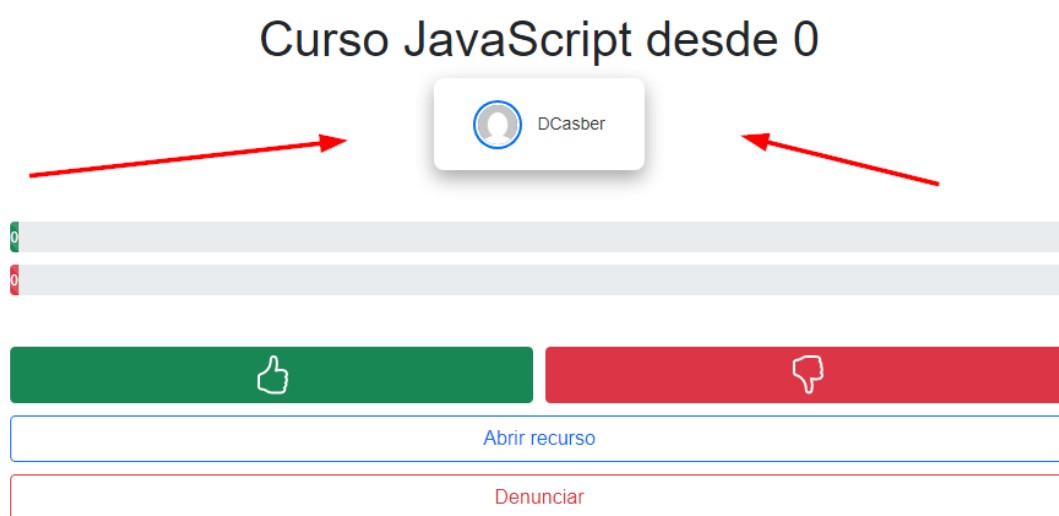


Figura 51: Botón para acceder al perfil público del usuario creador de un recurso

## A.9. Editar perfil

Un usuario puede editar su foto y descripción del perfil entrando en su perfil privado y haciendo clic en “Editar perfil”. También puede eliminar los recursos que haya creado haciendo clic en el icono de la papelera que aparece junto a cada recurso.

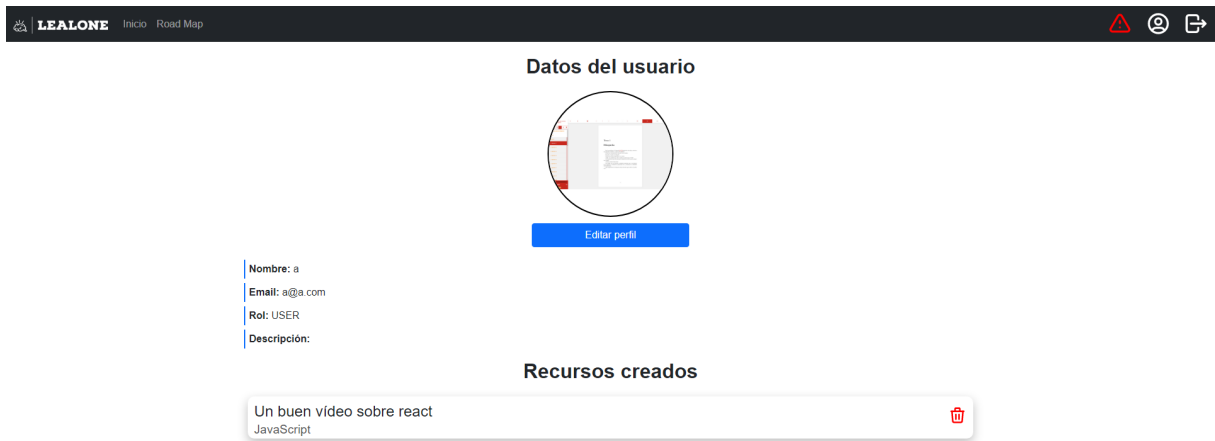


Figura 52: Página del perfil privado de un usuario

Para editar el perfil hay que hacer clic en “Editar perfil”, aparecerá un pop-up que nos dará dos opciones:

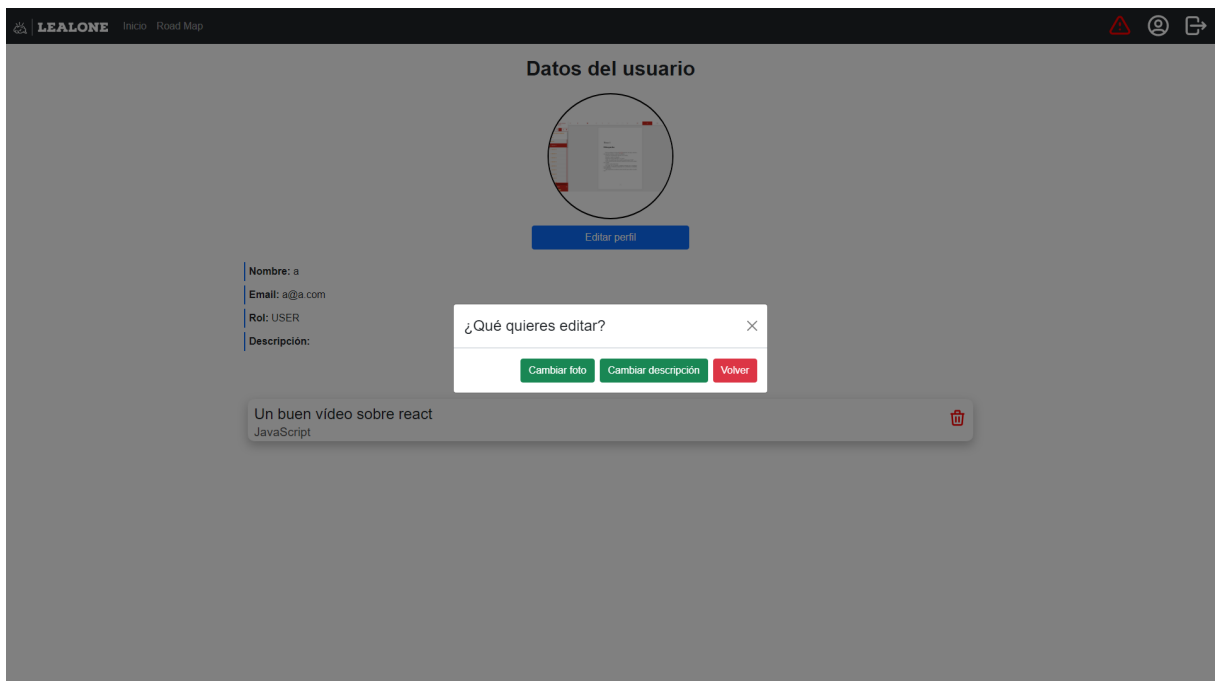


Figura 53: Pop-up con las opciones para editar el perfil

- Editar la foto de perfil, haciendo clic en “Cambiar foto”. Luego, el usuario hace clic en “Arrastra una imagen o pinche aquí” para seleccionar la imagen de su ordenador o también puede directamente arrastrarla al recuadro. Finalmente, cuando el usuario haga clic en “Cambiar” se habrá cambiado la foto de perfil.

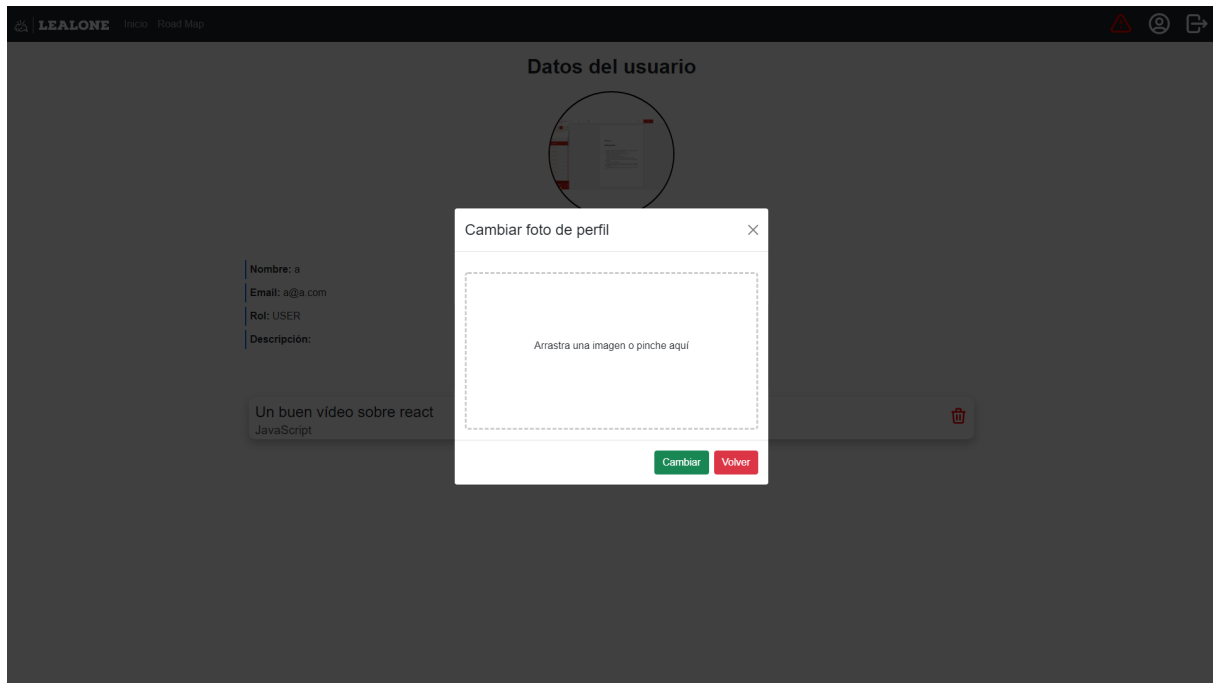


Figura 54: Pop-up con la opción para cambiar la imagen del perfil

- Editar la descripción de perfil, haciendo clic en “Cambiar descripción”. Luego el usuario escribe la nueva descripción en el textarea. Finalmente, cuando el usuario haga clic en “Cambiar” se habrá cambiado la descripción del perfil.



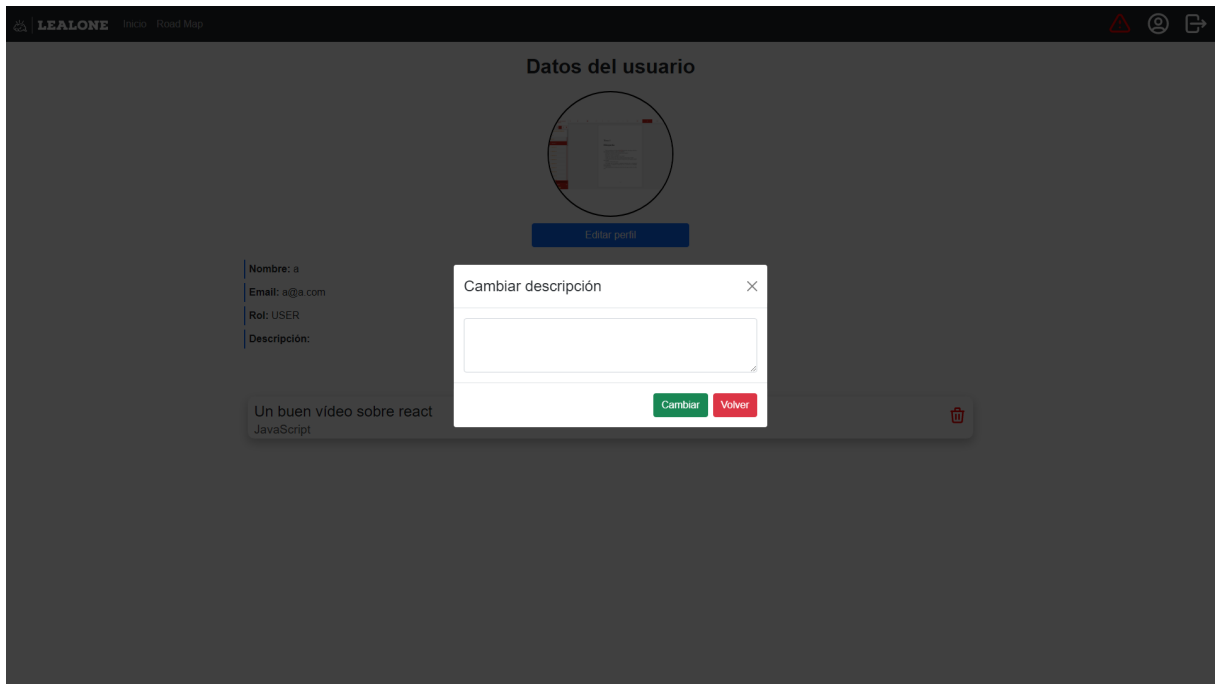


Figura 55: Pop-up con la opción para cambiar la descripción del perfil

## A.10. Crear, actualizar y eliminar bloques de contenido

El administrador podrá crear, actualizar y eliminar bloques de contenido. Para ello debe de acceder a la página principal. Esta es diferente de la que ven otros usuarios, ya que tiene un botón para crear un nuevo bloque de contenido y en cada bloque de contenido hay un botón para actualizar y eliminar los bloques de contenido ya existentes.

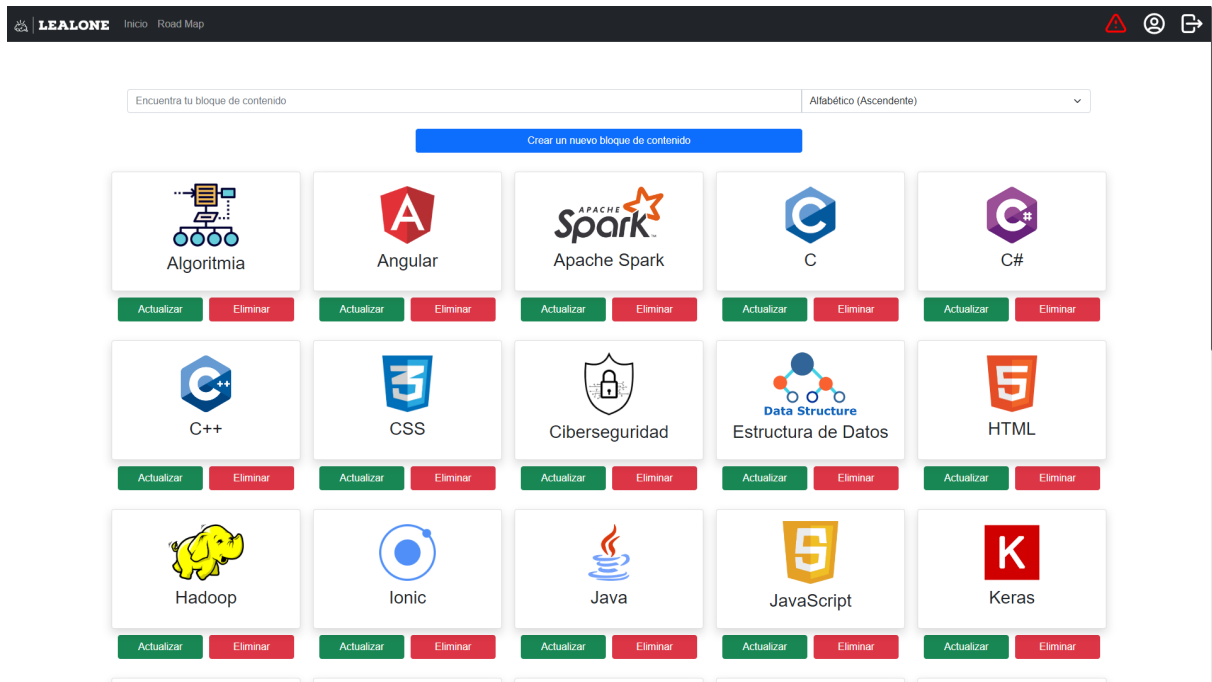


Figura 56: Página principal desde la vista de un administrador

#### A.10.1. Crear un bloque contenido

El administrador, desde la pantalla principal, debe hacer clic en “Crear un nuevo bloque de contenido”. Al hacerlo será redirigido a otra página donde podrá configurar la siguiente información sobre el bloque de contenido:

- Nombre del bloque de contenido.
- Descripción del bloque de contenido.
- La imagen del bloque de contenido, ya sea arrastrándola o seleccionándola desde el explorador que se abre al hacer clic en “Arrastra una imagen o pinche aquí”.

Al hacer clic en “Crear bloque de contenido”, el bloque se habrá creado y será público para cualquier otro usuario.

LEALONE Inicio Road Map

### Crea un nuevo bloque de contenido

Nombre del bloque de contenido:

Descripción:

Imagen del bloque de contenido:

Arrastra una imagen o pinche aquí

Crear bloque de contenido

Figura 57: Página para que un administrador pueda crear un nuevo bloque de contenido

#### A.10.2. Actualizar bloque de contenido

En la pantalla principal, el administrador, tiene dos botones disponibles por cada bloque de contenido, si quiere actualizar uno, debe de buscarlo y hacer clic en el botón “Actualizar”. Al hacerlo aparecerá la misma página que al crear un bloque de contenido, solo que ahora aparecerá por defecto toda la información del bloque de contenido ya escrita, como en la siguiente imagen.



Figura 58: Opciones adicionales que tiene un administrador sobre los bloques de contenido

Para actualizar los datos, simplemente se debe cambiar la información deseada y hacer clic en “Actualizar el bloque de contenido”.



Figura 59: Página para que un administrador pueda actualizar un nuevo bloque de contenido

### A.10.3. Eliminar bloque de contenido

En la pantalla principal, el administrador, tiene dos botones disponibles por cada bloque de contenido, si quiere borrar uno, simplemente debe de buscarlo y hacer clic en el botón “Eliminar”.

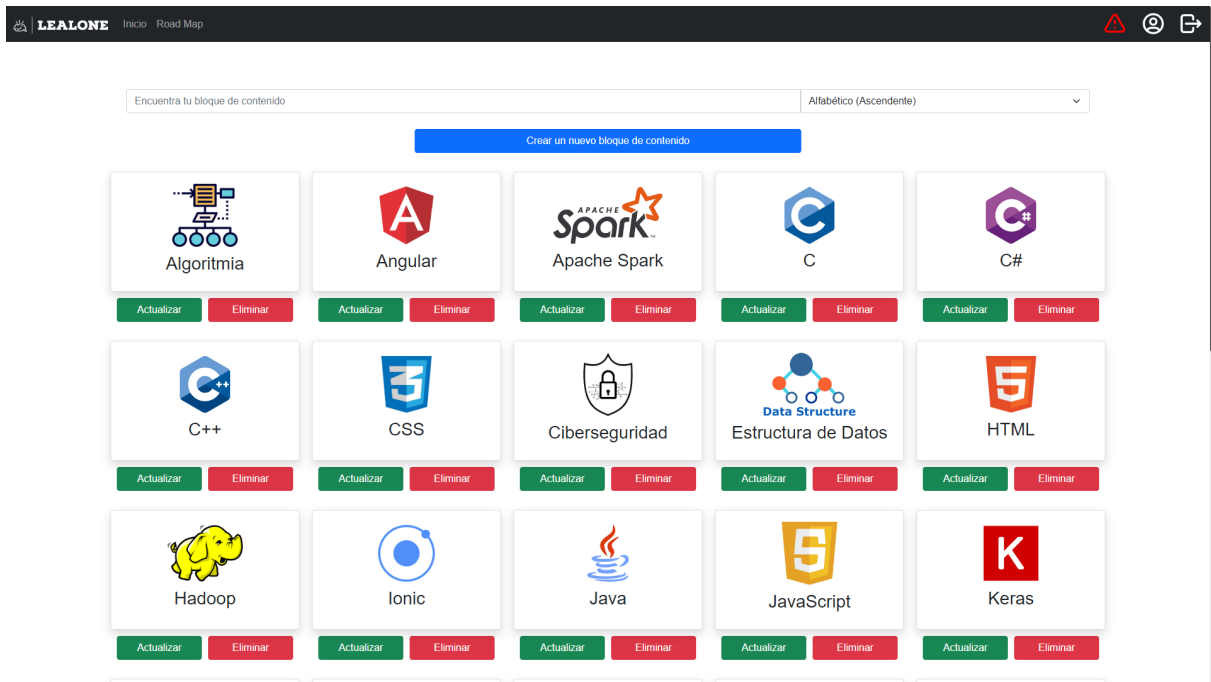


Figura 60: Página principal desde el punto de vista de un administrador

# Apéndice B

## Manual de Instalación

El proyecto está alojado en Heroku, y es accesible sin necesidad de instalar nada, accediendo al siguiente [enlace](#). Hay que tener en cuenta que por la forma en la que trabaja Heroku, cuando la aplicación lleva 30 minutos sin usarse, Heroku deja el servidor suspendido para no malgastar recursos, por tanto, la primera vez que se acceda a la página es posible que tarde unos cuantos segundos en cargar.

Aunque el programa es accesible desde la web, en caso de que se quiera ejecutar de forma local habría que seguir las siguientes instrucciones:

1. Instalar Node, preferiblemente la versión 14.19.1 o al menos que sea 14.x.x. El proyecto ha sido desarrollado con esta versión y es probable que en caso de usar otra versión de problemas de versiones por la librería de SASS.
2. Una vez que el proyecto esté descomprimido, entrar en la carpeta raíz del mismo.
3. Abrir dos consolas de comandos distintas, ambas situadas en la carpeta raíz del proyecto.

Terminal 1, paso 1. Escribir el comando “cd backend” para entrar en el proyecto del backend.

Terminal 1, paso 2. Escribir el comando “npm install” para instalar las dependencias, este paso puede tardar varios minutos.

Terminal 1, paso 3. Escribir el comando “npm run dev” para ejecutar el servidor en modo desarrollo con Nodemon.

Terminal 1, paso 4. En caso de que todo haya ido correctamente debería de aparecer algo parecido a esto.

```
D:\Universidad\4 - Anualidad\1º Cuatrimestre\TFG\learn-alone\backend>npm run dev
> learn-alone-backend@1.0.0 dev D:\Universidad\4 - Anualidad\1º Cuatrimestre\TFG\learn-alone\backend
> nodemon index.js

[nodemon] 2.0.7
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,json
[nodemon] starting `node index.js`
DB Online
Servidor corriendo en puerto 8000
```

Figura 61: Resultado del comando “npm run dev”

Terminal 2, paso 1. Escribir el comando “cd frontend” para entrar en el proyecto del frontend.

Terminal 2, paso 2. Escribir el comando “npm install” para instalar las dependencias, este paso puede tardar varios minutos.

Terminal 2, paso 3. Escribir el comando “npm start” para ejecutar el cliente con React.

Terminal 2, paso 4. En caso de que todo haya ido correctamente debería de aparecer algo parecido a esto.

```
Compiled successfully!

You can now view frontend in the browser.

Local:            http://localhost:3000
On Your Network:  http://192.168.56.1:3000

Note that the development build is not optimized.
To create a production build, use yarn build.
```

Figura 62: Resultado del comando “npm start”

Terminal 2, paso 5. Además, automáticamente se abrirá una pestaña en el navegador en la dirección “http://localhost:3000/” con el cliente desplegado. Puede tomar unos segundos la primera vez que se arranca el cliente y **es indispensable levantar el cliente antes que el servidor.**



UNIVERSIDAD  
DE MÁLAGA

| [uma.es](http://uma.es)

E.T.S de Ingeniería Informática  
Bulevar Louis Pasteur, 35  
Campus de Teatinos  
29071 Málaga

E.T.S. DE INGENIERÍA INFORMÁTICA