



### **Máster Universitario en Ingeniería Informática**

**Curso** : 2020 - 2021

**Asignatura** : Redes Sociales, Colaboración en Red

**Práctica** : SCNI Videoclub

**Alumno** : Daniel Serena Sanz, Víctor García-Bermejo Mazorra

**Madrid, 2020**

## Contenido

Contenido .....	2
Introducción .....	3
Recursos .....	3
Repositorio .....	3
Hosting .....	3
Aplicación base.....	4
Análisis.....	4
Diseño.....	4
Pantallas .....	5
Index.....	5
Login / Create .....	5
Perfil .....	6
Nueva película .....	6
Videoclub.....	8
Base de datos .....	10
Market .....	10
Usuarios.....	11
Conclusión .....	11
Referencias.....	12

## Introducción

Se entiende como aplicación web, aquellas herramientas que los usuarios pueden utilizar accediendo a un servidor web a través Internet o una intranet mediante un navegador. Uno de los motivos más importantes por los que se utilizan, es debido a la actuación como un cliente ligero por parte del navegador independientemente del sistema operativo en el que se utilice y por ello las aplicaciones web ganan una gran compatibilidad con todos los sistemas.

El sistema clásico que siguen las aplicaciones web es sencillo: el navegador como cliente realiza una petición al servidor donde recopila la información y los datos necesarios para crear una página con una lógica por detrás, y la envía al cliente en forma de respuesta. En este momento el navegador interpreta la respuesta y la muestra al usuario por pantalla.

A lo largo del tiempo se han ido mejorando las aplicaciones web. Actualmente los problemas a tratar dentro de este mundo son la latencia ya que la mayor parte de los cálculos y cambios se hacen en el servidor y el cliente se ve obligado a descargar constantemente recursos; y la escalabilidad del servidor en función de la cantidad de usuarios accediendo a los recursos y la cantidad de datos que deba manejar el servidor.

Una de las soluciones que se plantean en esta asignatura para resolver estos problemas, son el uso de aplicaciones web progresivas [1] en las que, todo lo que es la presentación se encarga el lado cliente, se ejecutan en segundo plano e incluso se pueden instalar a partir de los navegadores.

En esta práctica usaremos Nuxt[2] utilizando VUEX y Nodejs (npm)[3], con la idea de generar una aplicación web que gestione y genere páginas y el estado de las sesiones utilizando componentes. En el primer apartado que compondrá la aplicación base, haremos un breve análisis donde se marcarán unos pequeños objetivos del sistema a implementar. Seguidamente en el apartado de diseño se propondrá un diseño del flujo que seguirán los usuarios que utilicen la aplicación. En el apartado de pantallas se dará una breve explicación de los elementos más relevantes de cada página. Para finalizar se presentarán la base de datos con las estructuras donde se guardarían los datos y el documento se cerrará con unas conclusiones.

## Recursos

### Repositorio

En este enlace se muestra el código desarrollado, la metodología de trabajo llevada a cabo así como todos los recursos y requisitos para correr el sistema.

[https://github.com/danielss24/SNCI\\_VUE](https://github.com/danielss24/SNCI_VUE)

### Hosting

En este enlace se muestra la aplicación web desplegada.

<https://sncimarket.web.app/>



## Aplicación base

### Análisis

Para la realización de esta práctica, se plantean una serie de objetivos donde se destacarían los siguientes:

- Creación de una aplicación web que gestione y genere páginas y el estado de las sesiones utilizando diversos componentes alojados tanto en el cliente como en el servidor.
- Observación del funcionamiento de servicios y micro servicios cloud relacionados con las plataformas de Google. Para ello se utilizan diversos elementos de desarrollo que propone google como su sistema de autenticación o sus bases de datos (Firestore).
- Comprobación en tiempo real de la actualización de los cambios en varios clientes a la vez.

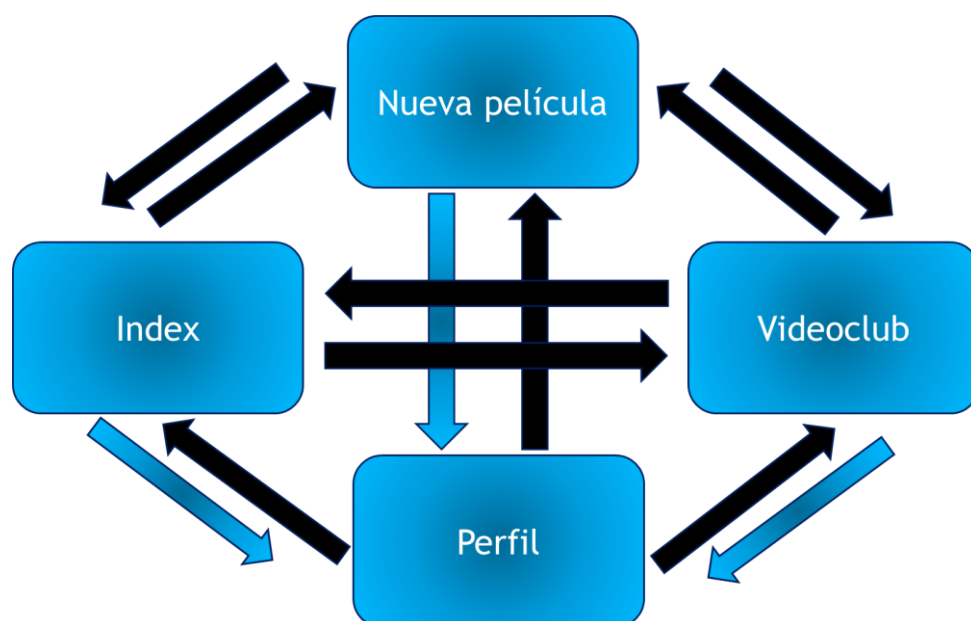
Para cumplir estos objetivos, se decidió crear una aplicación web cuyo tema principal sería una versión simplificada de lo que sería la página web de un videoclub mediante Nuxt. Los requisitos propuestos para cumplir las operaciones anteriores son los siguientes:

- Autenticación de usuarios utilizando la que utiliza Firebase de Google. Los usuarios pueden iniciar sesión registrarse o seguir utilizando la página sin hacerlo pero con menos funcionalidades.
- Creación de nuevas películas que se guardarán en la base de datos de Google (Firestore).
- Actualización inmediata en todos los clientes abiertos sin necesidad de actualizar la aplicación web.

Adicionalmente, debido al uso de Vuex y sus componentes la página tendrá una interfaz responsive y adaptativa al tamaño de las pantallas de los usuarios.

### Diseño

Para el diseño de la plataforma se planteó una interacción sencilla entre las diferentes páginas de la misma. Acorde a esto, creamos el siguiente flujo:



La aplicación web, constaría así de cuatro páginas que serían la página principal o Index, el Videoclub, Nueva película y la página de Perfil. Tal y como muestran las flechas propuestas en el diagrama, se puede distinguir dos tipos de interacciones entre las páginas:

- En el caso de las flechas negras, indican un acceso entre las páginas sin necesidad de haber iniciado sesión. Cabe destacar que, se han marcado las interacciones de salida de la página del perfil, como que no tienen necesidad de tener la sesión iniciada puesto que en las páginas entrantes, no existe ningún tipo de restricción de entrada.
- En el caso de las flechas azules, indican la necesidad de haber iniciado sesión antes de poder acceder a una página.

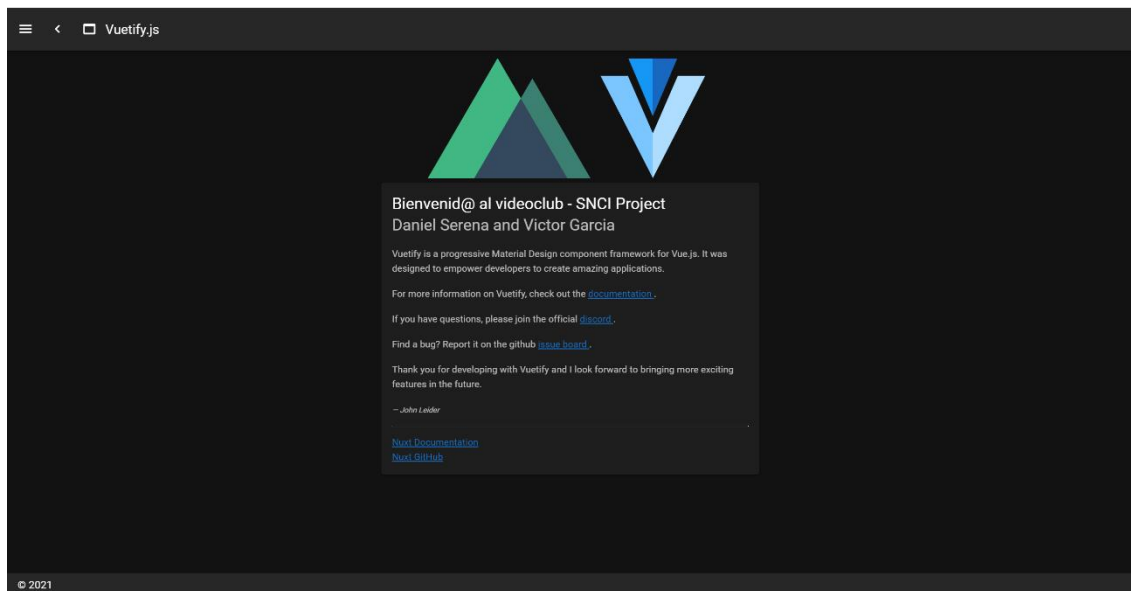
Por este segundo caso y conociendo que no hay una página de inicio de sesión, en el menú lateral izquierdo de todas las páginas, estará un componente que servirá para iniciar sesión dentro de la página de dos formas, la primera mediante un correo de google y la segunda mediante un correo en el momento. Si el usuario no existe, tras la solicitud del email, se creara una nueva cuenta pidiendo al nuevo usuario su nombre y una contraseña.

Adicionalmente, la funcionalidad de alquilar películas de la página del supermercado no será utilizable a no ser que el usuario esté registrado y haya iniciado sesión. Sin embargo, el resto de funcionalidades que tiene la aplicación web como valorar las películas o añadir películas a la base de datos sí que serán usables sin necesidad es estar dentro de una sesión.

## Pantallas

### Index

La página principal es una página básica de presentación del equipo.



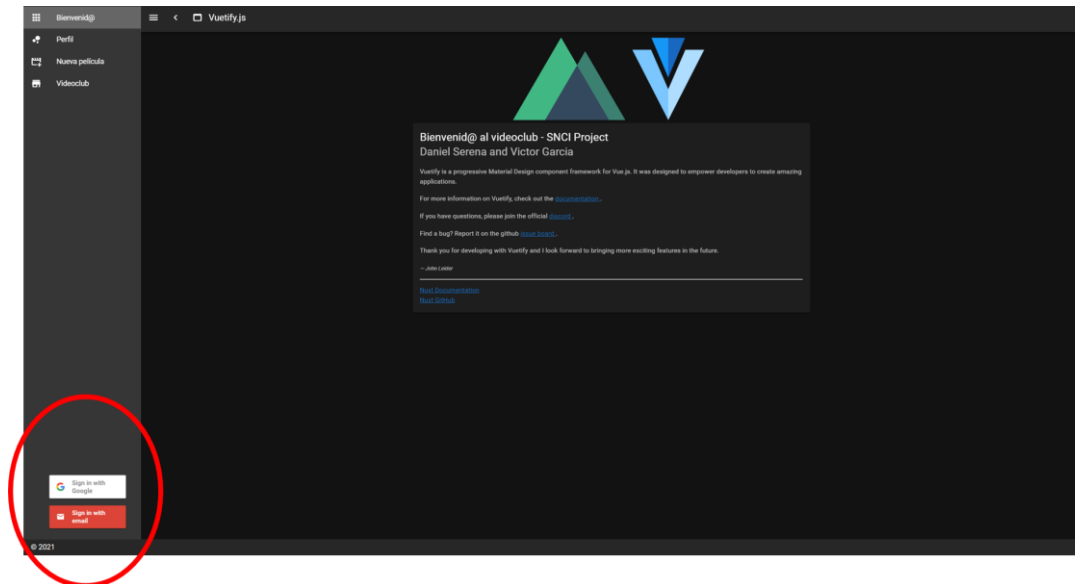
### Login / Create

En un principio se contaba con una página de login y otra de create que posteriormente se unificaron en login.

Se ha implementado la interfaz de registro de Firebase, dicha interfaz incorpora autenticaciones mediante tokens como Google, GitHub y correo-contraseña, entre otros. Esta interfaz en caso

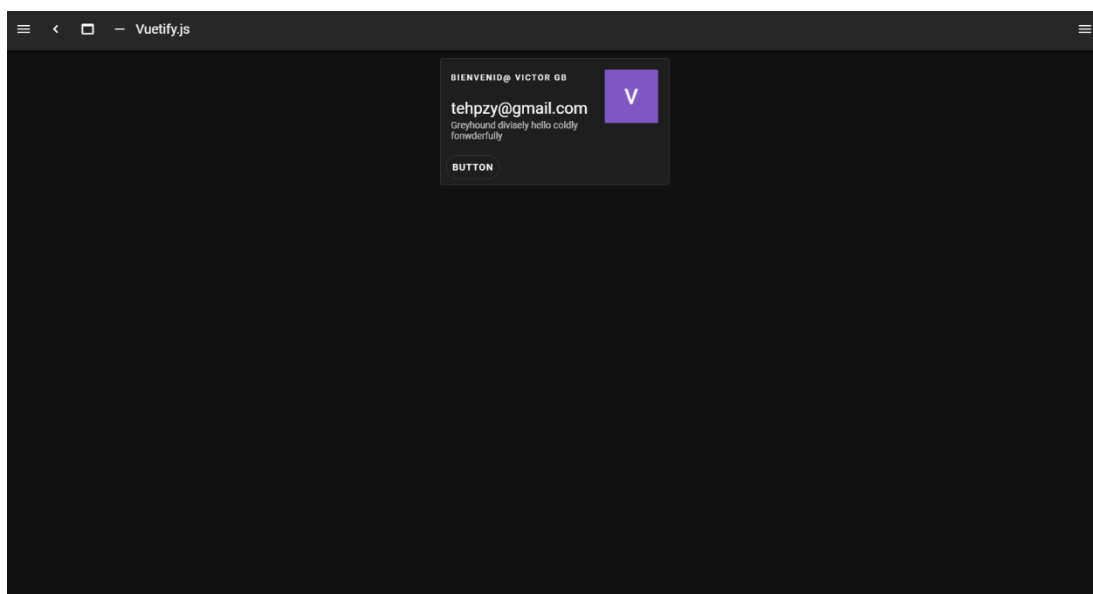
de no estar registrado con el usuario y contraseña indicados procedería a su registro y por contrario si existiese un usuario con dicho email llevaría a cabo el inicio de sesión.

Finalmente se ha incorporado en el menú lateral de la aplicación permitiendo una mayor facilidad a la hora de iniciar sesión desde cualquier página.



## Perfil

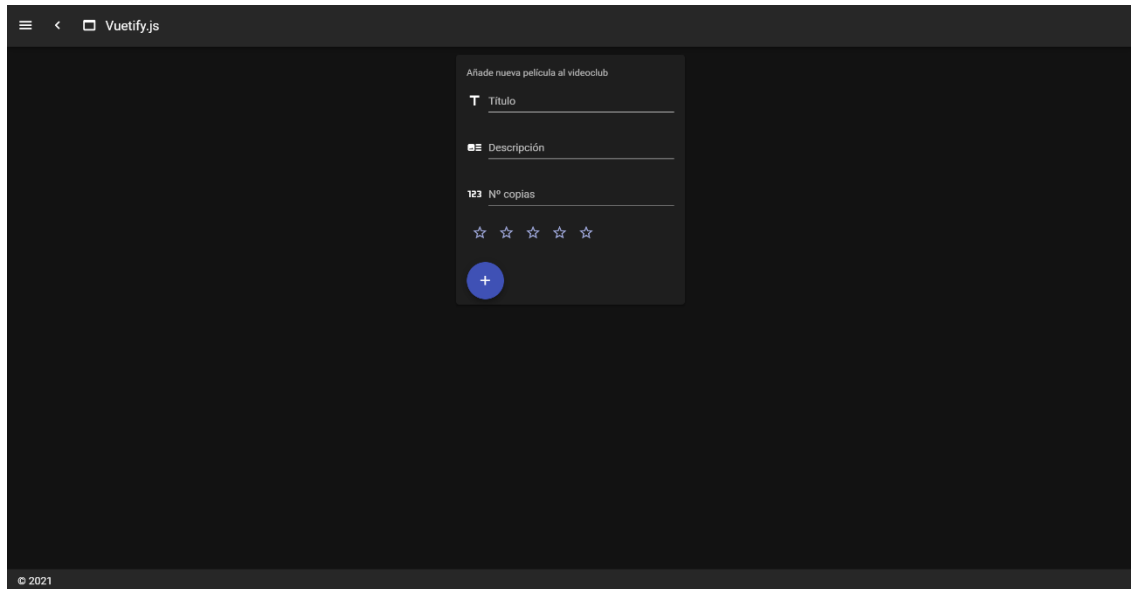
En la página de perfil se muestra los datos del usuario como su correo o su nombre de cara al resto de usuarios, en caso de iniciar sesión con los proveedores por defecto de Firebase se importará una imagen de contacto, si se inicia sesión con email y contraseña, esta imagen será una por defecto.



## Nueva película

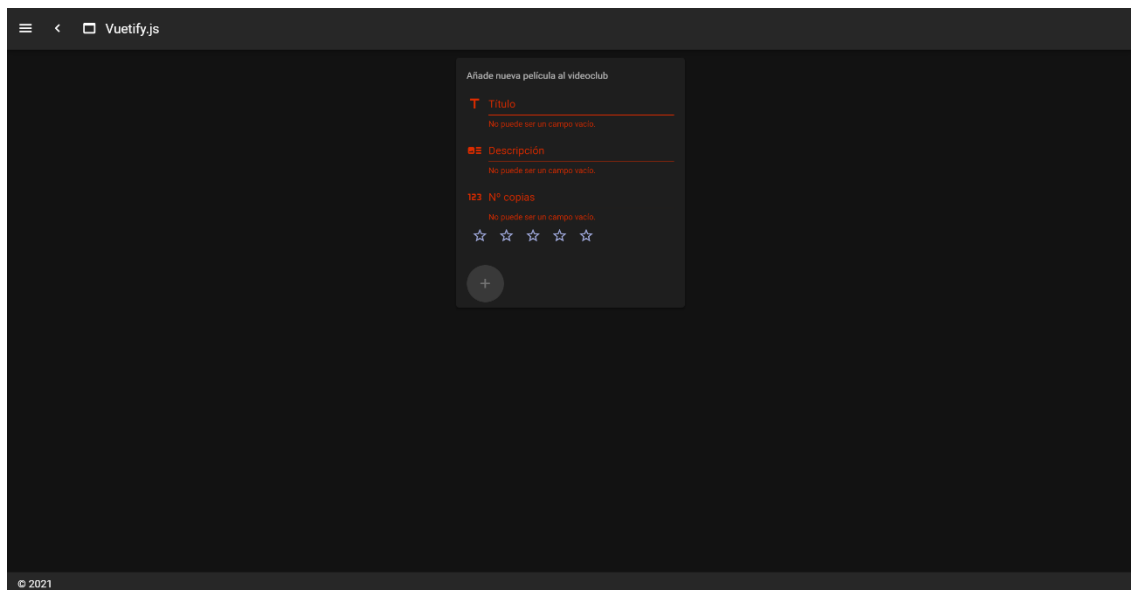
En esta página se muestra un formulario con los campos de la película que se quiere añadir y con un campo de valoración, en este caso se ha incluido para probar su funcionamiento, no obstante, este campo puede ser modificado más adelante por parte de los usuarios. En este

formulario se ha implementado una serie de reglas que se deben cumplir para poder añadir una película, una vez cumplimentado correctamente y añadido se muestra un feedback de confirmación.



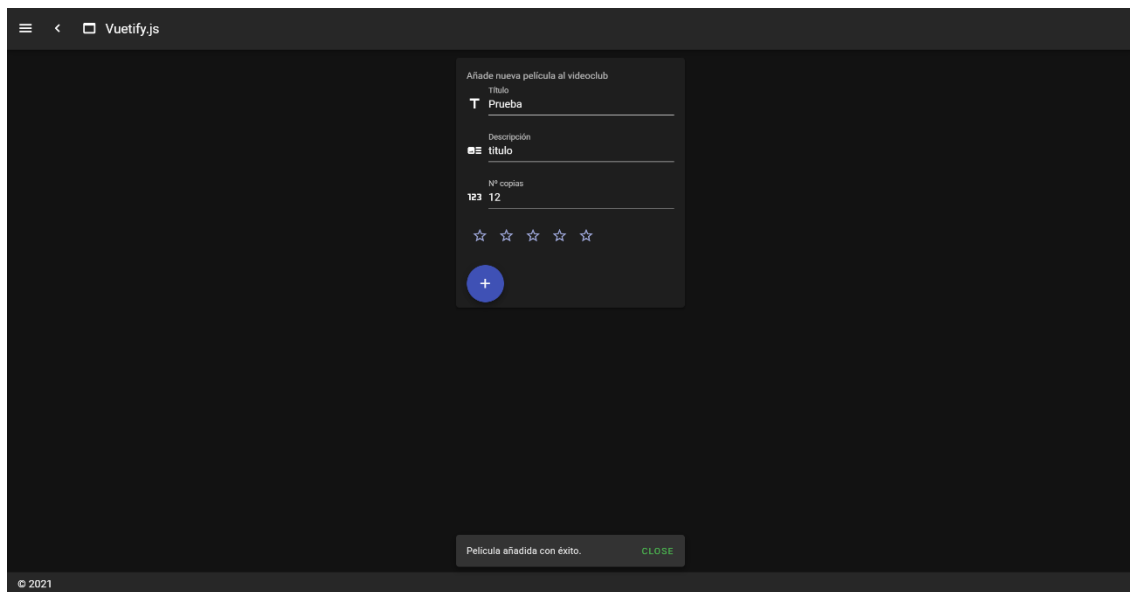
A screenshot of a web application interface in a dark theme. The browser's address bar shows 'Vuetify.js'. The main content area features a modal form titled 'Añade nueva película al videoclub'. The form contains four input fields: 'Título' (with a 'T' icon), 'Descripción' (with a 'D' icon), 'Nº copias' (with a '123' icon), and a five-star rating system. Below the stars is a blue circular button with a white plus sign. The footer of the application shows '© 2021'.

*Formulario básico.*



A screenshot of the same web application interface, but with error messages displayed below each input field. The error messages are: 'No puede ser un campo vacío.' for the 'Título' field, 'No puede ser un campo vacío.' for the 'Descripción' field, and 'No puede ser un campo vacío.' for the 'Nº copias' field. The blue plus button is now disabled and appears as a light gray circle. The footer still shows '© 2021'.

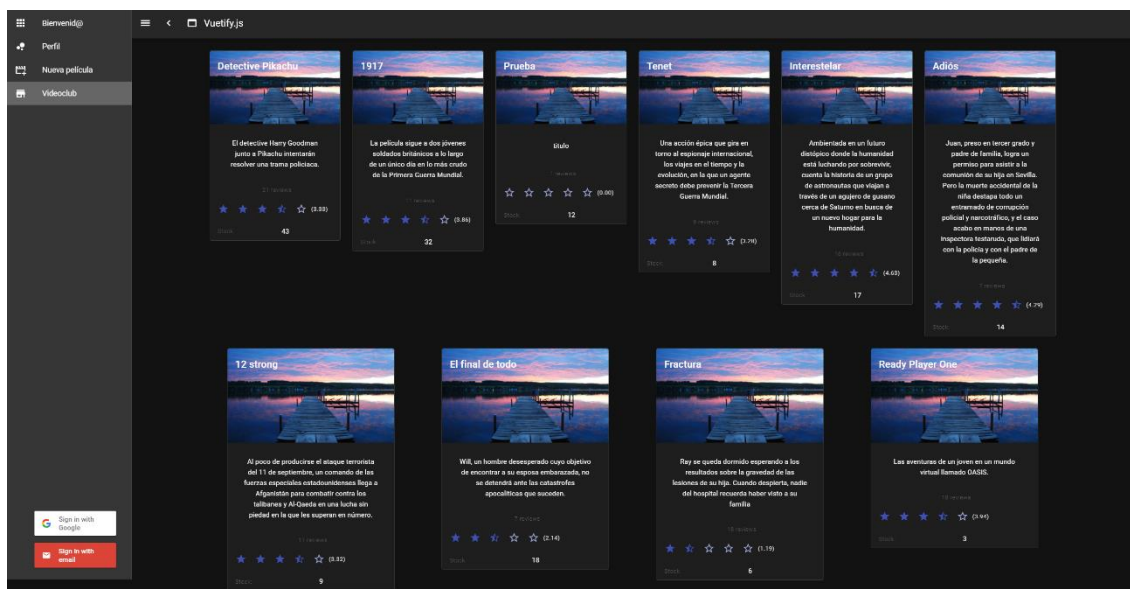
*Formulario con control de errores.*



*Formulario con feedback de añadir película correctamente.*

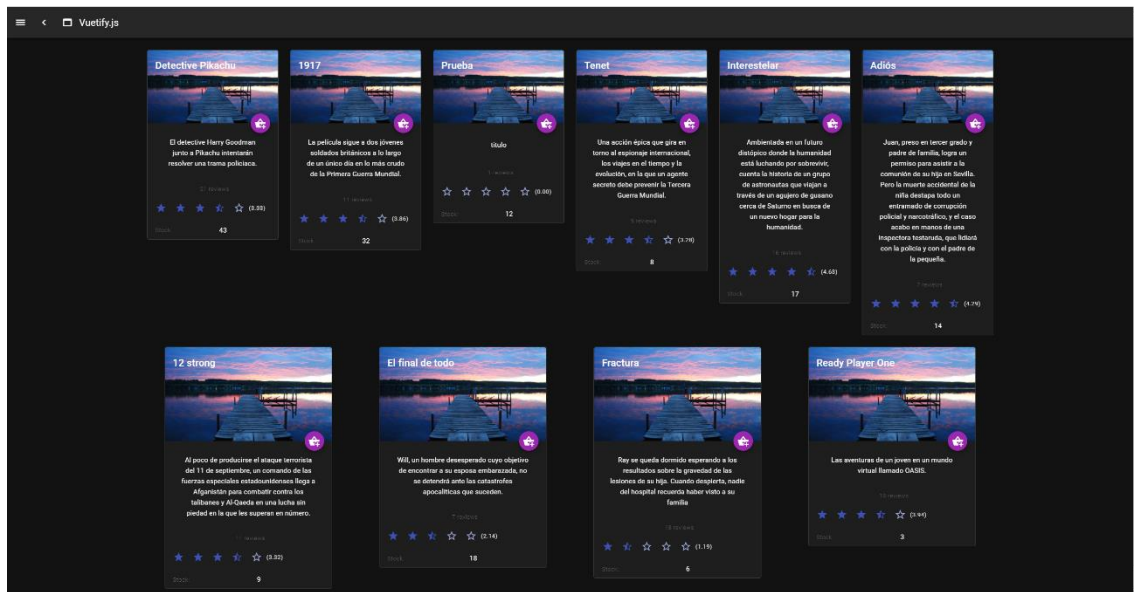
## Videoclub

En la página de videoclub, se muestran todas las películas registradas en el sistema. Esta página es la más completa y la que recoge la funcionalidad requerida para el proyecto, como por ejemplo la sincronización de datos y la actualización de estos en tiempo real. En caso de que se añada una película, el usuario que este visualizando esta página vera una nueva película al final de la lista, lo mismo ocurre cuando se produce una valoración de cualquier película, los usuarios que estén visualizando esta página verán modificado el valor de dicha película.

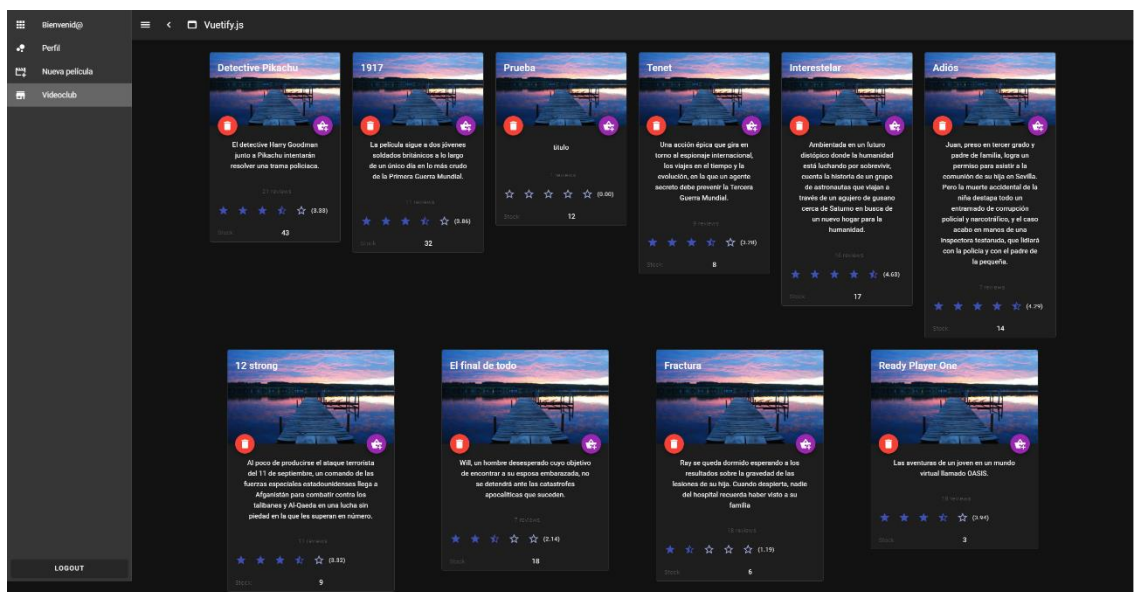


En esta vista del videoclub se observa como lo verían los usuarios no registrados.

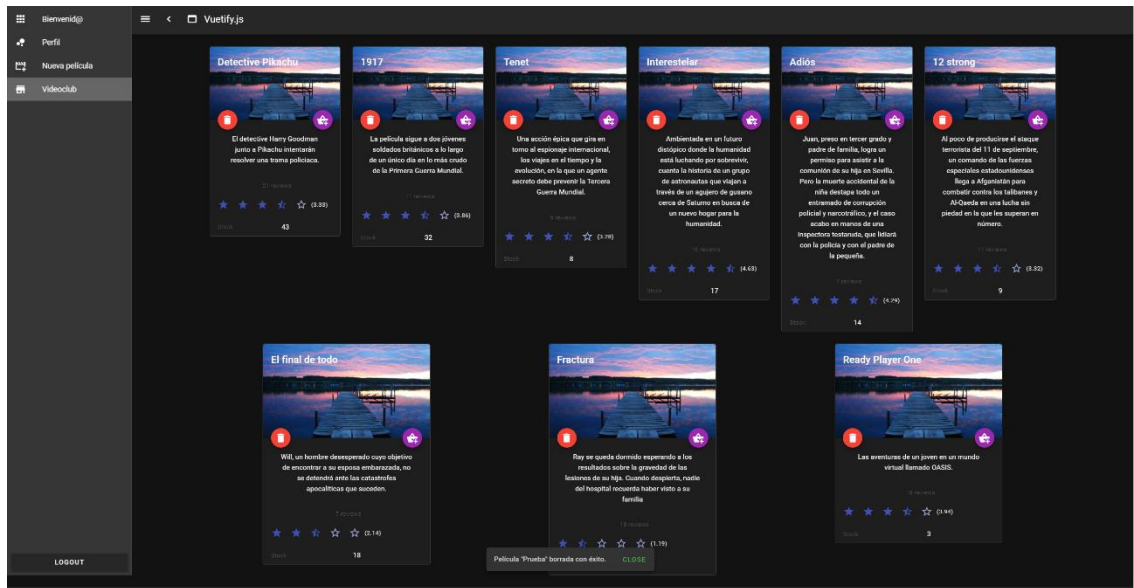




Una vez registrados en la aplicación, se podría observar un botón de añadir al carrito para poder alquilar cualquiera de las películas en caso de haber stock, esta implementación no se ha llevado a cabo.



Se ha implementado un sistema básico de roles el cual permite al usuario [daniel@daniel.com](mailto:daniel@daniel.com), poder borrar las películas mediante el botón de papelera, solo visible a dicho usuario cuando se registre en la aplicación.



Finalmente, si se lleva a cabo un borrado de cualquier película se mostrará un snackbar en el margen inferior de la pantalla como confirmación.

## Base de datos

Para la base de datos, hemos recurrido a una base de datos no relacional, en este caso usamos Firebase con el servicio de Firestore. Esta plataforma nos permite hacer una serie de llamadas a la base de datos para añadir, actualizar y previsualizar las diferentes colecciones y documentos.

## Market

Contamos con una colección base para las películas, la cual contiene una serie de documentos con los datos de cada una de las películas, como se muestra en la siguiente imagen:

<div> <div> <div>🏠</div> <div>&gt;</div> <div>market</div> <div>&gt;</div> <div>1sejhMaVQcmtL...</div> </div> </div>		
<div> <div>📁</div> <div>sncimarket</div> </div>	<div> <div>📁</div> <div>market</div> </div>	<div> <div>📄</div> <div>1sejhMaVQcmtxxLigFzt</div> </div>
<div> <div>+</div> <div>Iniciar colección</div> </div>	<div> <div>+</div> <div>Agregar documento</div> </div>	<div> <div>+</div> <div>Iniciar colección</div> </div>
<div> <div>market</div> <div>&gt;</div> </div>	<div> <div>1sejhMaVQcmtxxLigFzt</div> <div>&gt;</div> <div> <div>6sW1s7Frutx1SohG17NQ</div> <div>0GQ7STwXblyQ4TFV0EyI</div> <div>a9Cw6cc9T89sgwdB1JLe</div> <div>itzNsLS3WMjVDH5VRqvv</div> <div>kMP3L2ucVzoPYPCR8Fwm</div> <div>ryxZ39fwilub1e0ujTUq</div> <div>tM8GJ854d1dNfSKzb2K9</div> <div>uxE23e8AvepBVo2NLy4k</div> <div>x5HeI4QkKhECAuRx974x</div> </div> </div>	<div> <div>+</div> <div>Agregar campo</div> </div> <div> <div>contVal: 11</div> <div>description: "El detective Harry Goodman junto a Pikachu intentarán resolver una trama policiaca."</div> <div>name: "Detective Pikachu"</div> <div>quantity: "43"</div> <div>rating: 25</div> </div>

En esta colección base se introducen las películas que posteriormente se guardan en Vuex para su fácil acceso, en Vuex se incluye como campo el id del propio elemento para facilitar su acceso a Firestore y poder modificar fácilmente sus valores.

## Usuarios

Por último, contamos con una base de datos o un registro de usuario pero en este caso solo en Vuex, esto nos permite tener todos los datos del servicio de autenticación de Firebase en Vuex y con ello la estructura de usuario que deseamos, como se puede ver a continuación:

Buscar por dirección de correo electrónico, número de teléfono o UID de usuario					Agregar usuario	↺	⋮
Identificador	Proveedores	Fecha de creación	Fecha de acceso	UID de usuario	↑		
tehpzy@gmail.com		9 may. 2021	9 may. 2021	WgWbY5z4WHSwvR6U0h1GunlgO...			
pepe@pepe.com		29 abr. 2021	29 abr. 2021	IQDaOYwRWjdRf2cIKJ0f5MdXM3...			
jose@jose.com		29 abr. 2021	29 abr. 2021	n4cUS19wtZeuKBFDl0kgWOKMJe...			
daniel@daniel.com		22 abr. 2021	6 may. 2021	nuQdEBjeVjRVvXsYXzDV3uEOXS12			
danielsersan@gmail.com		1 may. 2021	7 may. 2021	vnUNC6kFjNTNM4ToTn5D0pJNx0...			
juan@juan.es		3 may. 2021	3 may. 2021	yU9WlYvQi2eb3FudViz1rnLX6vt2			

Filas por página: 50 1 – 6 of 6 < >

Con esta estructura de usuario podemos acceder al campo que queramos desde cualquier punto de la aplicación, como por ejemplo si el usuario ha iniciado sesión o no, estado necesario para proceder al alquiler cuando se implemente en un futuro.

## Conclusión

En esta práctica hemos creado una aplicación web con la herramienta de npm y la tecnología de Nuxt. Se han propuesto diversos problemas a paliar y en base a esos problemas se ha generado una aplicación web. Esta, es un servicio de alquiler de películas donde destacan las funciones de crear ítems, valorarlos y alquilarlos. Lo más particular de esta aplicación web es que en el uso de estas funciones, la página se actualiza de forma directa para todos los usuarios ya que cada cliente está suscrito a cambios en la base de datos y cuando suceden se visualizan por cómo funcionan las aplicaciones que utilizan Vuex.

Respecto al trabajo futuro, la página es muy sencilla y se debería perfeccionar algunas funcionalidades o añadir otras como control de precios en los alquileres. Por otro lado, respecto a los problemas planteados en la introducción, sería una buena idea comprobar tiempos de actualización en base a una gran cantidad de usuarios utilizando a la vez la aplicación web.

## Referencias

- [1] [https://es.wikipedia.org/wiki/Aplicaci%C3%B3n\\_web](https://es.wikipedia.org/wiki/Aplicaci%C3%B3n_web)
- [2] <https://www.xataka.com/basics/que-es-una-aplicacion-web-progresiva-o-pwa>
- [3] <https://nuxtjs.org/>
- [4] <https://github.com/nvm-sh/nvm>
- [5] <https://firebase.google.com/?hl=es>
- [6] <https://firebase.google.com/docs/reference/js>