Práctica evaluable 3: Frameworks Javascript

Luis Vidal Rico

45927898ª

➢ Requisitos adicionales implementados

- **Vue-CLI**: Se ha usado Vue-CLI tanto en la instalación del proyecto como en la instalación del router.
- **"Ver detalles" y "Eliminar"**: En la página se permite eliminar Posts y comentarios de los usuarios. También se permite ver en detalle los posts y las comunidades.
- **"Editar"**: Se permite editar el perfil de un usuario desde su perfil.
- **Framework de componentes visuales**: Se ha usado Tailwindcss para la realización del css de la aplicación.
- **Estado**: Se ha usado Vuex para mantener el estado del usuario loggeado en la aplicación.
- **Router**: Se ha usado el router de vue para gestionar las urls de la aplicación.

➢ Inicialización del proyecto

Para iniciar el servidor, basta con entrar a la carpeta 'server', modificar el .env y introducirle los datos de la base de datos Postgres elegida. Una vez hecho *npm install* con ejecutar *npm start ya debería estar todo listo.*

Para instalar el cliente, entrar en la carpeta 'client' y ejecutar *npm i* y *npm run serve*. Una vez hecho esto, recomiendo registrarse, crear una comunidad y crear un post.

Así, los usuarios que no están registrados, en la página principal verán todos los posts de la aplicación ordenados por la fecha de creación.

Los usuarios registrados solo verán los posts de las comunidades a las que siguen.

En las páginas en las que se listan elementos (la página de home, de un post, de comunidad) se ha implementado un scroll infinito, por lo que si el usuario llega al último post cargado, se cargarán más posts (si los hay) de la BD.

➢ Importante:

Me acabo de dar cuenta de que el inicio de sesión/logout está algo buggeado, por lo que si se inicia sesión y alguna funcionalidad no funciona como debería, hay que pulsar F5 solamente.