

Decisiones de diseño

En este apartado, antes de empezar a implementar el patrón Redux, he reestructurado el código para tener una estructura más clara y que cumpliera con las peticiones de esta PEC.

Siguiendo el planteamiento de usar módulos, el código ha quedado estructurado de la siguiente manera:

- Módulo **activities**: contiene todo lo relacionado con las actividades.
En este módulo se agrupan los siguientes componentes:
 - **Activity-crud**: componente que contiene la tabla con los botones de update/delete de las actividades.
 - **Activity-detail**: componente que contiene los detalles de una actividad.
 - **Activity-form**: componente que contiene un formulario para añadir una actividad.
 - **Activity-list**: componente que contiene un listado de las actividades.
 - **Admin**: componente que hace de pagina de admin y contiene los componentes **activity-crud** y **activity-form**.
 - **Home**: componente que hace de pagina de home y contiene los componentes **activity-list** y **activity-detail**.
 - **My-activities**: componente que hace de pagina de my-activities y contiene los componentes **activity-list** y **activity-detail**.

Además de estos componentes, el modulo contiene los models del apartado activities y los servicios relacionados con ellas.

- Módulo **login**: contiene todo lo relacionado con el inicio de sesión y el registro de usuarios.
En este módulo se agrupan los siguientes componentes:
 - **Login**: componente que hace de página de login.
 - **Register**: componente que hace de página de register.

Además de estos componentes, el modulo contiene los servicios relacionados con las tareas de iniciar sesión y registrar.

- Módulo **profile**: contiene todo lo relacionado con la gestión de datos de los usuarios.
En este módulo se agrupan los siguientes componentes:
 - **Education-form**: componente que contiene un formulario para añadir un campo de educación a los usuarios.
 - **Education-info**: componente que contiene la tabla con los botones de update/delete de los campos de educación.

- **Profile:** componente que hace de pagina de profile y contiene los componentes **profile-info**, **profile-form**, **education-info**, **education-form**.
- **Profile-form:** componente que contiene el formulario para añadir datos a un usuario.
- **Profile-info:** componente que muestra los datos relacionados con un usuario.

Además de estos componentes, el modulo contiene los models del apartado profile y los servicios relacionados con el.

- Por último, en la carpeta **shared**, tenemos los componentes, directivas, guards y servicios que se usan en todos los módulos de la aplicación y que, por tanto, se usan en diversos sitios.

Para la aplicación del patrón redux, cada módulo contiene sus actions, reducers y effects.

- En el módulo de actividades, el estado de la aplicación contiene las siguientes propiedades:
 - **Activities:** listado de las actividades obtenidas del servicio http.
 - **Loading:** true si se solicitan datos al servidor y se esta esperando respuesta.
 - **Loaded:** true si se han solicitado datos al servidor y se ha recibido correctamente la respuesta.
 - **Updating:** true si se han actualizado datos en el servidor y se esta esperando respuesta.
 - **Updated:** true si se han actualizado datos en el servidor y se ha recibido correctamente la respuesta.
 - **Error:** datos de error al realizar una acción que lanza un effect.
- En el módulo de login, el estado de la aplicación contiene las siguientes propiedades:
 - **LoggingIn:** true si se ha solicitado un inicio de sesión y se esta esperando respuesta del servidor.
 - **LoggedIn:** true si se ha solicitado un inicio de sesión y se ha recibido correctamente la respuesta de usuario valido. False si no se ha podido iniciar la sesión.
 - **Registering:** true si se ha solicitado un registro y se esta esperando respuesta del servidor.
 - **Registered:** true si se ha solicitado un registro y se ha recibido correctamente la respuesta de registro valido. False si no se ha podido registrar al usuario.
 - **Error:** datos de error al realizar una acción que lanza un effect.

- En el módulo profile, el estado de la aplicación contiene las siguientes propiedades:
 - **User**: datos del usuario obtenidos del servicio http.
 - **Loading**: true si se solicitan datos al servidor y se esta esperando respuesta.
 - **Loaded**: true si se han solicitado datos al servidor y se ha recibido correctamente la respuesta.
 - **Updating**: true si se han actualizado datos en el servidor y se esta esperando respuesta.
 - **Updated**: true si se han actualizado datos en el servidor y se ha recibido correctamente la respuesta.
 - **Error**: datos de error al realizar una acción que lanza un effect.