Práctica 9

Componentes:

**ConfigureStore:**

Se ha configurado un store con los reducers: excursiones, comentarios, cabeceras y actividades con los middlewares thunk y logger.

**Action Types:**

Se han establecido para cada uno de los reducers las acciones: Loading, Add y Failed que corresponden con Cargando, Cargado (Luego se añade como payload en la aplicación) y Fallido en caso de no poderse cargar.

**Reducers:**

Excursiones: Tiene en su estado 3 variables, el estado de carga (isLoading), falso si no se está cargando y verdadero en caso contrario, mensaje de error (errMess), un array con las excursiones cargadas (excursiones, si las ha cargado) y la acción a realizar (action).

Según la acción devuelve los otros 3 estados de diferentes maneras:

Si la acción es ADD: isLoading es falso (ya ha cargado las excursiones), errMess es nulo (no hay mensaje de error) y excursiones es el array que ha cargado, la carga/mensaje de la acción.

Si la acción es LOADING: isLoading es verdadero (está en proceso de carga), errMess es nulo y excursiones es un array vacío.

Si la acción es FAILED: isLoading es falso (no se ha podido cargar nada), errMess es la carga/mensaje de la acción.

Todas las opciones van precedidas por el estado, para no machacar el resto de las variables. Similar en el resto de reducers: comentarios, cabeceras y actividades.

**Action Creators:**

Para cada reducer realiza el proceso de carga de contenidos devolviendo el array cargado y añadiéndolo como carga para que el reducer se lo envíe al store. En caso de fallo devuelve como carga el mensaje de error, en todos los casos también envía el tipo de acción a realizar.

En el resto de los componentes se han modificado para que cojan y modifiquen las variables del store, a través del sistema anterior, en vez de localmente. Finalmente se ha modificado App.js para que se configure el store y toda la aplicación tenga acceso a el al estar definida dentro del ‘proveedor del almacén de estado’.