

```
1 import React, { useState, useEffect } from 'react';
2 import { toast } from 'react-toastify';
3 import Tarjeta from '../Tarjeta';
4 import ConfirmaEliminacion from '../ConfirmaEliminacion';
5 import MiModal from '../MiModal';
6 import Lista from './Lista';
7 import Formulario from './Formulario';
8 import useFetch from '../../hooks/useFetch';
9 import MensajeError from '../MensajeError';
10 import { FontAwesomeIcon } from '@fortawesome/react-fontawesome';
11 import { faUsers } from '@fortawesome/free-solid-svg-icons';
12 import {API_GRUPOS_LABORATORIO as API_URL, METODO_LISTAR, METODO_CREAR, METODO_EDITAR,
13   METODO_BORRAR, API_TOKEN, CREAR, EDITAR} from '../../Constantes';
14 const opcionesListar = {};
15 export default function GruposLaboratorio(props) {
16   // estados
17   const[datosModal, setDatosModal] = useState({mostrar:false});
18   const[datosConfirmaEliminacion, setDatosConfirmaEliminacion] = useState({mostrar:false});
19   const[accion, setAccion] = useState(null);
20   const[formulario, setFormulario] = useState({
21     cuatrimestre_id: 1
22     , laboratorio_id: 1
23     , direccion_id: 1
24     , carrera_id: 1
25     , asignatura_id: 1
26     , docente_id: 1
27     , dias_asignados: []
28     , grupo: "GPONVO"
29     , lunes_inicio: "07:00:00"
30     , lunes_fin: "08:40:00"
31     , martes_inicio: "07:00:00"
32     , martes_fin: "08:40:00"
33     , miercoles_inicio: "07:00:00"
34     , miercoles_fin: "08:40:00"
35     , jueves_inicio: "07:00:00"
36     , jueves_fin: "08:40:00"
37     , viernes_inicio: "07:00:00"
38     , viernes_fin: "08:40:00"
39     , sabado_inicio: "07:00:00"
40     , sabado_fin: "08:40:00"
41     , domingo_inicio: "07:00:00"
42     , domingo_fin: "08:40:00"
43     , estatus: "Activo"
44   });
45   const[url_lista, setUrlLista] = useState(API_URL);
46   const[url_edicion, setUrlEdicion] = useState(API_URL);
47   const[url_borra, setUrlBorra] = useState(API_URL);
48   const[id, setId] = useState(0);
49   const[opcionesCreacion, setOpcionesCreacion] = useState({body:null});
50   const[detonarCreacion, setDetonarCreacion] = useState(false);
51   const[opcionesEdicion, setOpcionesEdicion] = useState({body:null});
52   const[detonarEdicion, setDetonarEdicion] = useState(false);
53   const[opcionesBorrado, setOpcionesBorrado] = useState({body:null});
54   const[detonarBorrado, setDetonarBorrado] = useState(false);
55
56   const estatusLista = useFetch(url_lista, opcionesListar, API_TOKEN, METODO_LISTAR, true);
57   const estatusCrear = useFetch(API_URL, opcionesCreacion, API_TOKEN, METODO_CREAR,
```

```
detonarCreacion);
58   const estatusEditar = useFetch(url_edicion, opcionesEdicion, API_TOKEN, METODO_EDITAR,
detonarEdicion);
59   const estatusBorrar = useFetch(url_borra, opcionesBorrado, API_TOKEN, METODO_BORRAR,
detonarBorrado);
60
61   // efectos
62   useEffect(() => {
63     //si se encuentra cargando no hacer nada, aún no hay respuesta
64     if (estatusCrear.loading)
65       return;
66     //analizar código de error
67     switch(estatusCrear.status) {
68       case 201:
69         toast.success(<MensajeCreadoEditado titulo="Registro creado" result=
{estatusCrear.result} />, {autoClose: 2500});
70         setDatosModal({mostrar:false});
71         setUrlLista(`${API_URL}?time=${(new Date()).getTime()}`);
72         setDetonarCreacion(false);
73         break;
74       default:
75         toast.error(<MensajeError result={estatusCrear.result} />, {autoClose: 2500});
76     }
77   }, [estatusCrear])
78
79   useEffect(() => {
80     //si se encuentra cargando no hacer nada, aún no hay respuesta
81     if (estatusEditar.loading)
82       return;
83     //analizar código de error
84     switch(estatusEditar.status) {
85       case 200:
86         toast.success(<MensajeCreadoEditado titulo="Registro actualizado" result=
{estatusEditar.result} />, {autoClose: 2500});
87         setDatosModal({mostrar:false});
88         setUrlLista(`${API_URL}?time=${(new Date()).getTime()}`);
89         setDetonarEdicion(false);
90         break;
91       default:
92         toast.error(<MensajeError result={estatusEditar.result} />, {autoClose: 2500});
93     }
94   }, [estatusEditar])
95
96   useEffect(() => {
97     //si se encuentra cargando no hacer nada, aún no hay respuesta
98     if (estatusBorrar.loading)
99       return;
100    //analizar código de error
101    switch(estatusBorrar.status) {
102      case 200:
103        toast.success(estatusBorrar.result.message, {autoClose: 2500});
104        setUrlLista(`${API_URL}?time=${(new Date()).getTime()}`);
105        setDetonarBorrado(false);
106        break;
107      default:
108        toast.error(<MensajeError result={estatusBorrar.result} />, {autoClose: 2500});
109    }
110   }, [estatusBorrar])
111
```

```
112
113 function handleAgrega() {
114     setAccion(CREAR);
115     setDatosModal({
116         mostrar: true
117         , titulo: 'Agregar grupo de laboratorio'
118     });
119 }
120
121 function handleEdita(registro) {
122     setId(registro.id);
123     setAccion(EDITAR);
124     setFormulario({
125         laboratorio_id: registro.laboratorio_id
126         , cuatrimestre_id: registro.cuatrimestre_id
127         , direccion_id: registro.direccion_id
128         , carrera_id: registro.carrera_id
129         , asignatura_id: registro.asignatura_id
130         , docente_id: registro.docente_id
131         , dias_asignados: registro.dias_asignados
132         , grupo: registro.grupo
133         , lunes_inicio: registro.lunes_inicio
134         , lunes_fin: registro.lunes_fin
135         , martes_inicio: registro.martes_inicio
136         , martes_fin: registro.martes_fin
137         , miercoles_inicio: registro.miercoles_inicio
138         , miercoles_fin: registro.miercoles_fin
139         , jueves_inicio: registro.jueves_inicio
140         , jueves_fin: registro.jueves_fin
141         , viernes_inicio: registro.viernes_inicio
142         , viernes_fin: registro.viernes_fin
143         , sabado_inicio: registro.sabado_inicio
144         , sabado_fin: registro.sabado_fin
145         , domingo_inicio: registro.domingo_inicio
146         , domingo_fin: registro.domingo_fin
147         , estatus: registro.estatus
148     });
149     setDatosModal({
150         mostrar: true
151         , titulo: 'Editar grupo de laboratorio'
152     });
153 }
154
155 function handleBorra(registro) {
156     setUrlBorra(`${API_URL}/${registro.id}`);
157     setDatosConfirmaEliminacion({
158         mostrar: true
159         , titulo: 'CONFIRMA'
160         , body: '¿Realmente deseas eliminar el elemento?'
161     });
162 }
163
164 function handleSubmit(formulario) {
165     switch(accion) {
166         case CREAR:
167             toast.info('Enviando datos (creación)...', {autoClose: 1500});
168             setDetonarCreacion(true);
169             setOpcionesCreacion({body: JSON.stringify(formulario)});
```

```
170         break;
171     case EDITAR:
172         toast.info('Enviando datos (edición)...', {autoClose: 1500});
173         setUrlEdicion(`${API_URL}/${id}`)
174         setDetonarEdicion(true);
175         setOpcionesEdicion({body:JSON.stringify(formulario)});
176         break;
177     default:
178     }
179 }
180
181 function handleConfirmaEliminacion() {
182     setDatosConfirmaEliminacion({mostrar:false});
183     setDetonarBorrado(true);
184     setOpcionesBorrado({body:null});
185 }
186
187 return (
188     <Tarjeta titulo={<><FontAwesomeIcon icon={faUsers} /> Grupos de laboratorio</>>
189     <ConfirmaEliminacion datos={datosConfirmaEliminacion} handleConfirmaEliminacion=
190 {handleConfirmaEliminacion} />
191     <MiModal datos={datosModal}>
192     <Formulario formulario={formulario} handleSubmit={handleSubmit} />
193     </MiModal>
194     <Lista handleAgrega={handleAgrega} handleEdita={handleEdita} handleBorra={handleBorra}
195 estatus={estatuslista} />
196     </Tarjeta>
197 );
198 }
199
200 function MensajeCreadoEditado(props) {
201     const {data:{id, grupo, dias_asignados}} = props.result;
202     return (
203         <>
204         <div>
205             <strong>{props.titulo}</strong>
206         </div>
207         <div>
208             [{id}] {grupo} ({dias_asignados.join(', ')})
209         </div>
210     </>
211 );
212 }
```