

ALEJANDRO DANIEL LEZCANO BALBUENA- 4637984

GITHUB: <https://github.com/alezcanossj/StudyApp.git>

PROTECCION DE RUTAS

Middleware

```
BackEnd > src > middleware > JS auth.controller.js > auth
1  const { sequelize } = require("../connection");
2  const jwt = require("jsonwebtoken");
3
4  const auth = async function (req, res, next) {
5    if (!req.headers.authorization) {
6      res.json({
7        success: false,
8        error: "No authorization header",
9      });
10     return;
11   } else {
12     let token = req.headers.authorization;
13
14     const userDB = await sequelize.query(
15       "SELECT * FROM users WHERE token = '" + token + "'"
16     );
17     let user = null;
18
19     if (userDB.length > 0 && userDB[0].length > 0) {
20       user = userDB[0][0];
21
22       console.log("Token del usuario: ", user);
23
24       res.locals.userId = user.id;
25
26       next();
27     } else {
28       res.json({
29         success: false,
30         error: "Token inválido",
31       });
32     }
33   }
34 }
```

User

```
$ app-routing.module.ts TS edit-topic.page.ts TS list-topic.page.ts TS home.page.ts <> list-t
BackEnd > src > route > users > JS users.route.js > <unknown> > exports
1  const userController = require("../controller/users/users.controller");
2  const Middleware = require("../middleware/auth.controller");
3
4  module.exports = function(app) {
5
6     app.get("/users/list",Middleware.auth, userController.listar);
7     app.get("/users/:id", Middleware.auth,userController.consultarPorCodigo);
8     app.post("/users/update",Middleware.auth, userController.actualizar);
9     app.delete("/users/delete/:id",Middleware.auth, userController.eliminar);
10    app.post("/users/login/", userController.login);
11    app.post("/users/logout/",Middleware.auth, userController.logout);
12  }
```

Topics

```
Backend > src > route > topics > JS topics.route.js > <unknown> > exports
1  const Middleware = require('../../middleware/auth.controller');
2  const topicsController = require('../../controller/topics/topics.controller');
3  module.exports = function(app) {
4
5      app.get("/topics/list",Middleware.auth, topicsController.listar);
6      app.post("/topics/update",Middleware.auth, topicsController.actualizar);
7      app.delete("/topics/delete/:id",Middleware.auth, topicsController.eliminar);
8      app.get("/topics/:id",Middleware.auth, topicsController.consultarPorCodigo);
9  }
```

TOPICS BACKEND

Model

```
const {DataTypes} = require('sequelize');
const {sequelize}= require('../connection');
const TopicsModel = sequelize.define('Topics', {
  id: {
    type: DataTypes.INTEGER,
    primaryKey: true,
    autoIncrement: true,
  },
  create_date: {
    type: DataTypes.DATE,
    allowNull: false,
  },
  name: {
    type: DataTypes.STRING,
    allowNull: false,
  },
  topic_id: {
    type: DataTypes.INTEGER,
    allowNull: true,
  },
  order: {
    type: DataTypes.INTEGER,
    allowNull: false,
  },
  priority: {
    type: DataTypes.INTEGER,
    allowNull: false,
  },
  color: {
    type: DataTypes.STRING,
    allowNull: false,
  },
  owner_user_id: {
    type: DataTypes.INTEGER,
    allowNull: false,
  },
},{
  tableName: 'topics',
  timestamps: false
});

module.exports={
  TopicsModel
}
```

Controller

Listar

```
backend > src > Controller > topics > JS topics.controller.js > ...
1  const {sequelize} = require("../connection");
2  const {TopicsModel}= require("../model/topic.model");
3  const TopicService= require("../service/topic.service")
4  const listar = async function(req, res) {
5      console.log("listar topicos");
6      try {
7          const topics = await TopicService.listar(req.query.filtro || '');
8          // EN topics[0] SE ENCUENTRA NUESTRO LISTADO DE SQL
9          if(topics && topics[0]){
10             res.json({
11                 success:true,
12                 topics: topics[0]
13             });
14         }else{
15             res.json({
16                 success:true,
17                 topics: []
18             });
19         }
20     } catch (error) {
21         res.json({
22             success:false,
23             error: error.message
24         });
25     }
26 };
27 const controllerPorFiltro = async function(req, res) {
```

Consultar

```
const consultarPorCodigo = async function(req, res) {  
  console.log("consultar Topics por codigo");  
  try {  
    const topics = await TopicService.consultarPorCodigo(req.params.id);  
  
    // EN topics[0] SE ENCUENTRA NUESTRO LISTADO DE SQL  
    if(topics){  
      res.json({  
        success:true,  
        topics: topics  
      });  
    }else{  
      res.json({  
        success:true,  
        topics: []  
      });  
    }  
  } catch (error) {  
    console.log(error);  
    res.json({  
      success:false,  
      error: error.message  
    });  
  }  
};
```

Actualizar y eliminar

```
4
5  const actualizar = async function(req, res) {
6      let topicsRetorno=null; //GUARDARA EL TOPICO QUE SE VA A INCLUIR O EDITAR
7      const data =req.body; // SE OBTIENE LOS DATOS DEL CUERPO DE LA PETICION
8
9      try {
10
11          topicsRetorno = await TopicService.actualizar(data);
12
13          res.json({
14              success: true,
15              topics: topicsRetorno
16          });
17      } catch (error) {
18          res.json({
19              success:false,
20              error: error.message
21          });
22      }
23  };
24
25  const eliminar = async function( req, res) {
26      console.log( "eliminar topics" ) ;
27      try {
28          await TopicService.eliminar(req.params.id)
29          res.json({
30              success: true
31          });
32      } catch (error) {
33          res.json({
34              success: false,
35              error: error.message
36          });
37      }
38  };
39
40  module.exports = {
41      listar, actualizar, eliminar, consultarPorCodigo
42  };
43
```

Service

Listar

```
const listar = async function (textoBuscar) {
  console.log("listar topics ACA");
  try {
    const topics = await sequelize.query(`SELECT topics.*, users.name as Owner
      FROM topics
      left join users on users.id=topics.owner_user_id
      WHERE
      UPPER(topics.name) LIKE UPPER('%${textoBuscar}%')
      ORDER BY topics.id`);
    // EN topics[0] SE ENCUENTRA NUESTRO LISTADO DE SQL
    if (topics && topics[0]) {
      return topics;
    } else {
      return [];
    }
  } catch (error) {
    console.log(error);
    throw error;
  }
};
```

Consultar

```
5  const consultarPorCodigo = async function (id) {
6    console.log("consultar topics por codigo");
7    try {
8      const topics = await TopicsModel.findByPk(id);
9
10     // EN topics SE ENCUENTRA NUESTRO topics BUSCADO
11     if (topics) {
12       return topics;
13     } else {
14       return [];
15     }
16   } catch (error) {
17     console.log(error);
18     throw error;
19   }
20 };
21
```

Actualizar y Eliminar

```
2  const actualizar = async function (data) {
3    console.log("actualizar topics");
4    let topicsRetorno = null; //GUARDARA EL TEMA QUE SE VA A INCLUIR O EDITAR
5    const id = data.id; // ID PASADO
6
7    let topicExiste = null;
8
9    try {
10     if (id) {
11       topicExiste = await TopicsModel.findByPk(id);
12     }
13     if (topicExiste) {
14       topicsRetorno = await TopicsModel.update(data,{where: {id: id}});
15     } else {
16       topicsRetorno = await TopicsModel.create(data);
17     }
18     return topicsRetorno;
19   } catch (error) {
20     throw error;
21   }
22 };
23
24 const eliminar = async function (id) {
25   console.log("eliminar topics ");
26   try {
27     const topics = await TopicsModel.findByPk(id);
28     if (topics) {
29       await TopicsModel.destroy({
30         where: { id: id },
31       });
32     }
33     return true;
34   } else {
35     return false;
36   }
37 } catch (error) {
38   throw error;
39 }
40 };
```

LOGIN Y LOGOUT IMPLEMENTADO EN BACKEND:

```
const login = async function( req, res) {
  console.log ("login usuarios");
  try {
    // Buscar en la base de datos el usuario con el correo electrónico y contraseña
    const usersDB = await sequelize.query("SELECT * FROM users WHERE email='"+ req.body.email + "' AND password = '" + req.body.password + "'");
    console.log ("users",usersDB);
    let user = null;
    // Verificar si se encontraron resultados en la consulta a la base de datos y asignar el primer resultado
    if (usersDB.length > 0 && usersDB[0].length > 0) {
      user = usersDB[0][0];
      // Asignar el primer registro encontrado a la variable "user"
      if(user.token){
        res.json({
          success: false,
          error: 'usuario ya autenticado'
        });
      }
      let token= jwt.sign({
        codigo: user.id,
        name: user.name,
        last_name: user.last_name,
        email:user.email,
        avatar: user.avatar
      },'passwd');
      const usersDBUpdate = await sequelize.query("UPDATE users set token= '"+ token + "' WHERE id = "+user.id);
      //si se encuentra en la base de datos se mostrara true y usuario.
      res.json({
        success: true,
        token:token,
        user_id: user.id
      });
    } else{
      res.json({
        success: false,
        error: 'usuario no encontrado'
      });
    }
  }
  // Si no se encuentra el usuario en la base de datos,
```

```
136
137   } catch (error) {
138     // Si ocurre un error, devolver
139     console.log(error);
140     res.json ({
141       success: false,
142       error: error .message
143     });
144   }
145
146 };
147 const logout = async function( req, res) {
148
149   try {
150     const userDb= await sequelize.query("UPDATE users set token=null where id= " +res.locals.userId+ "");
151     res.json({
152       success:true
153     })
154   } catch (error) {
155     res.json({
156       success:false,
157       error: error.message
158     })
159   }
160
161 };
162 module.exports = {
163   listar, actualizar, eliminar,consultarPorCodigo,login,logout
164 };
```


GET users.list

POST users.login

POST users.logout

GET topics.list

POST topics.update

+ ...

No Environment

StudyApp / users.login

POST

http://localhost:3000/users/login...

Send

ParamsAuthorizationHeaders (8)BodyPre-request ScriptTestsSettings

noneform-datax-www-form-urlencodedrawbinaryGraphQLJSON

Beautifuly

```
1 {  
2   "email": "aledanielxiaomi@gmail.com",  
3   "password": "ale2@15xD"  
4 }
```

BodyCookiesHeaders (8)Test Results

Status: 200 OKTime: 69 msSize: 552 BSave Response

PrettyRawPreviewVisualizeJSON

```
1 {  
2   "success": true,  
3   "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.  
    eyJjb2RpZ281OjEsIm5hbWUwMTBjbGVqYw5kcm8iLCJsYXN0ZXhpbWUwMTBMZXBpYW5vIiwiaWF0Ij06MjAyMjA5MDQzMjE1OTU0NDUzOTU4Nm0uODMAMHk7q2eM40vsQnL1DHMQ~o1bYBNvMaeeSo",  
4   "user_id": 1  
5 }
```

POST users.logout

No Environment

Study_App / users.logout

POST

http://localhost:3000/users/logout

Send

Params

Authorization

Headers (8)

Body

Pre-request Script

Tests

Settings

Cookies

Headers

7 hidden

KEY	VALUE	DESCRIPTION		Bulk Edit	Presets
<input checked="" type="checkbox"/> Authorization	eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJjb2RpZ28iOiE...				
Key	Value	Description			

body

Cookies

Headers (8)

Test Results

Save Response

Pretty

Raw

Preview

Visualize


JSON


```
1 {
2   "success": true
3 }
```


LOGIN FRONT FUNCIONAL

< Inbox

ALEJANDRO LEZCANO
LOGIN


 Email Email


Contraseña Contraseña 



< Inbox

ALEJANDRO LEZCANO
LOGIN

 Email aledanielxiaomi@gmail.com

Contraseña ale2015xD 



TOPICS EN POSTMAN CON AUTHORIZACION

Listar

POST users.login

POST users.logout

GET topics.list

+

...

No Environment

Study_App / topics.list

Save

...

Send

Params

Authorization

Headers (7)

Body

Pre-request Script

Tests

Settings

Cookies

Headers

6 hidden

	KEY	VALUE	DESCRIPTION	...	Bulk Edit	Presets
<input checked="" type="checkbox"/>	Authorization	eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJjb2RpZ28iOiJl...				
	Key	Value	Description			

Body

Cookies

Headers (8)

Test Results

Status: 200 OK

Time: 62 ms

Size: 629 B

Save Response

Pretty

Raw

Preview

Visualize

JSON

...

...

```
4      {
5        "id": 1,
6        "create_date": "2023-05-19T04:00:00.000Z",
7        "name": "Matematica",
8        "topic_id": null,
9        "order": 1,
10       "priority": 1,
11       "color": "#54f000",
12       "owner_user_id": 1,
13       "owner": "Alejandro"
14     },
15     ]
16   }
```

Runner

Trash

...

Buscar

POST users.login

POST users.logout

GET topics.list

GET topics.search.code

+

...

No Environment

Study_App / topics.search.code

Save

...

Send

Params

Authorization

Headers (7)

Body

Pre-request Script

Tests

Settings

Cookies

Headers

6 hidden

	KEY	VALUE	DESCRIPTION	...	Bulk Edit	Presets
<input checked="" type="checkbox"/>	Authorization	eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJjb2RpZ28iOiJl...				
	Key	Value	Description			

Body

Cookies

Headers (8)

Test Results

Status: 200 OK

Time: 16 ms

Size: 438 B

Save Response

Pretty

Raw

Preview

Visualize

JSON

...

...

```
1  {
2    "success": true,
3    "topics": {
4      "id": 1,
5      "create_date": "2023-05-19T04:00:00.000Z",
6      "name": "Matematica",
7      "topic_id": null,
8      "order": 1,
9      "priority": 1,
10     "color": "#54f000",
11     "owner_user_id": 1
12   }
13 }
```

Runner

Trash

...

Crear

Study_App / topics.update

POST http://localhost:3000/topics/update

Params Authorization Headers (9) Body Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```
1 {
2   "id": null,
3   "create_date": "2023-03-17",
4   "name": "LENGUAJE 5",
5   "topic_id": null,
6   "order": 1,
7   "priority": 1,
8   "color": "#53563",
9   "owner_user_id": 1
}
```

Body Cookies Headers (8) Test Results

Status: 200 OK Time: 65 ms Size: 437 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "success": true,
3   "topics": {
4     "id": 7,
5     "create_date": "2023-03-17T03:00:00.000Z",
6     "name": "LENGUAJE 5",
7     "topic_id": null,
8     "order": 1,
9     "priority": 1,
10    "color": "#53563",
11    "owner_user_id": 1
12  }
}
```

Actualizar

Study_App / topics.update

POST http://localhost:3000/topics/update

Params Authorization Headers (9) Body Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```
1 {
2   "id": 5,
3   "create_date": "2023-03-17",
4   "name": "LENGUAJE 5",
5   "topic_id": null,
6   "order": 2,
7   "priority": 1,
8   "color": "#53563",
9   "owner_user_id": 1
}
```

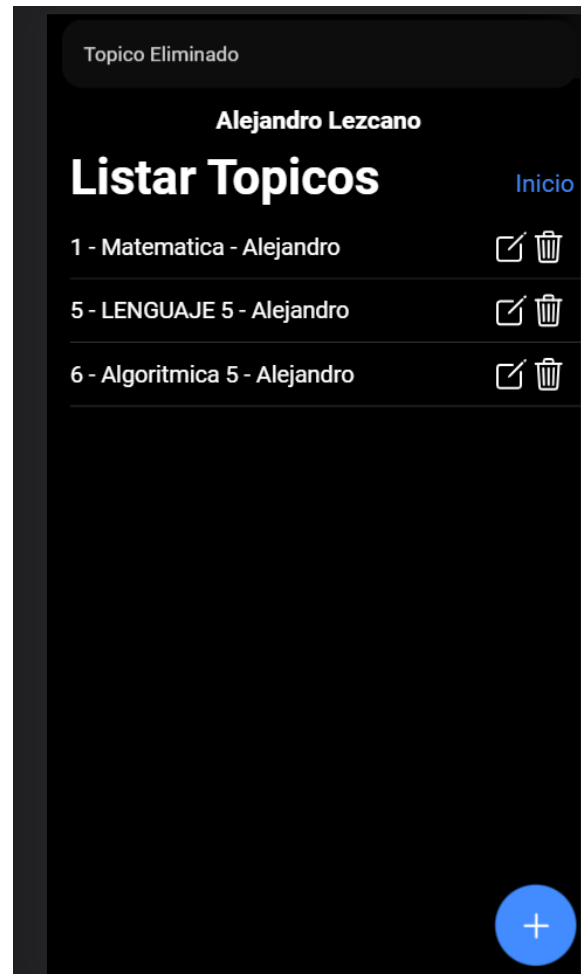
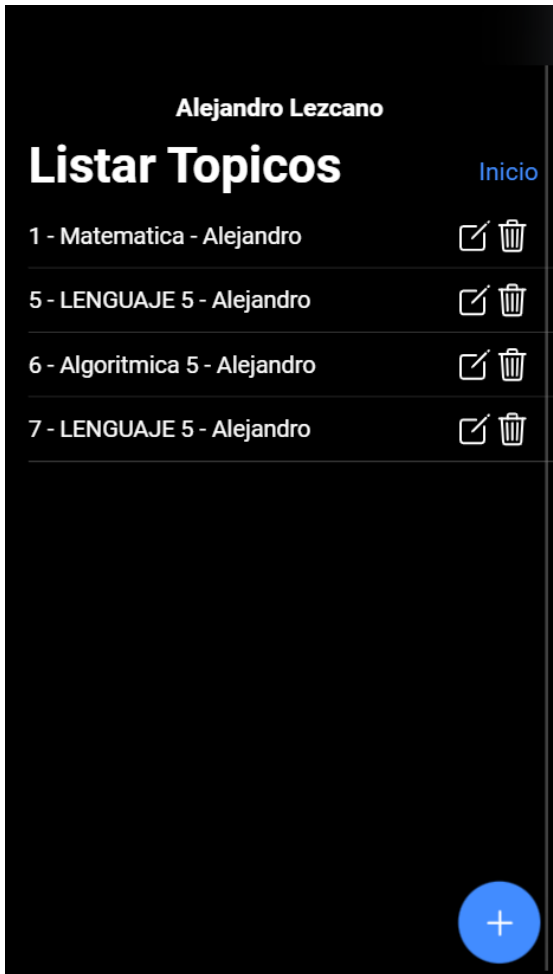
Body Cookies Headers (8) Test Results

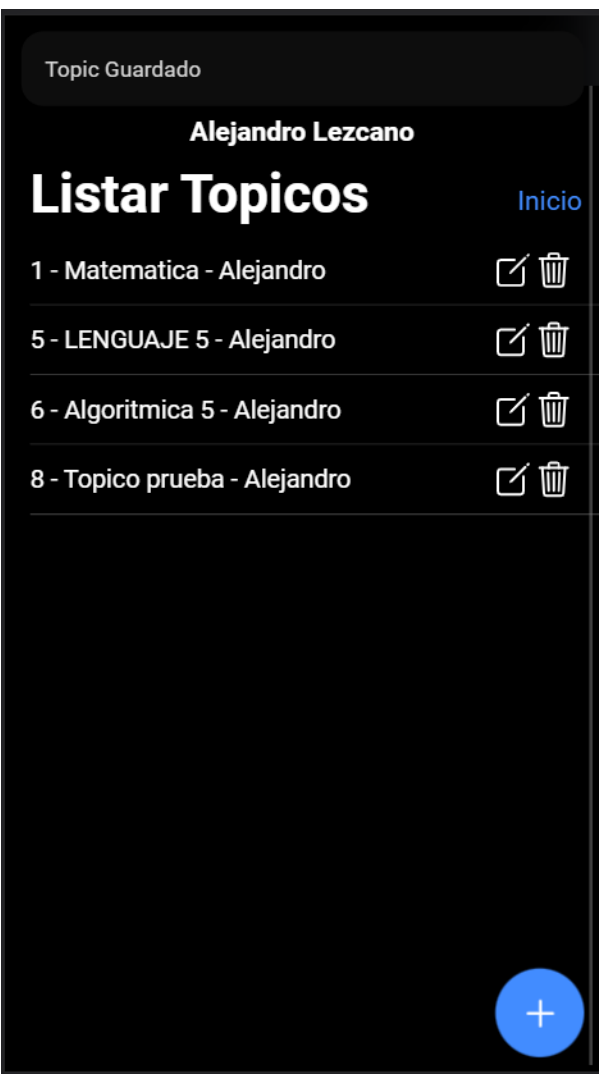
Status: 200 OK Time: 69 ms Size: 437 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "success": true,
3   "topics": {
4     "id": 5,
5     "create_date": "2023-03-17T03:00:00.000Z",
6     "name": "LENGUAJE 5",
7     "topic_id": null,
8     "order": 2,
9     "priority": 1,
10    "color": "#53563",
11    "owner_user_id": 1
12  }
}
```

TOPICOS FRONT END





Topic Guardado

Alejandro Lezcano

Listar Topics

[Inicio](#)

1 - Matematica - Alejandro



5 - LENGUAJE 5 - Alejandro



6 - Algoritmica 5 - Alejandro



8 - Topico prueba Editado - Alejandro

