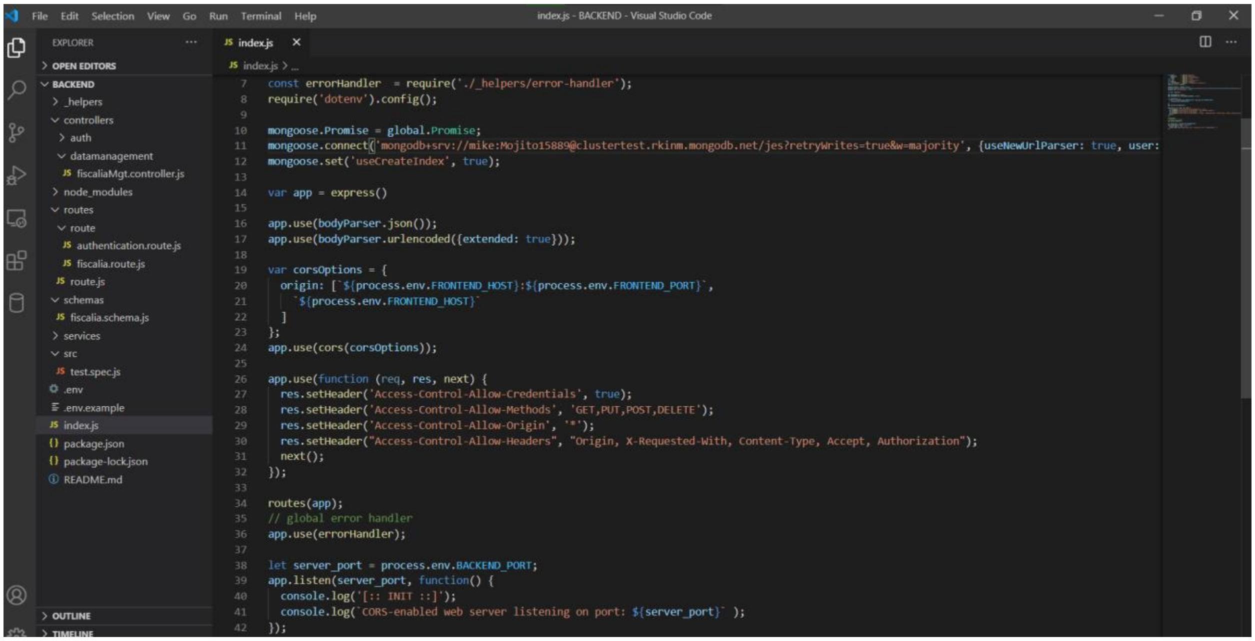
Miguel Angel Omar Ruano Roca

2742187980403

PRUEBA JES

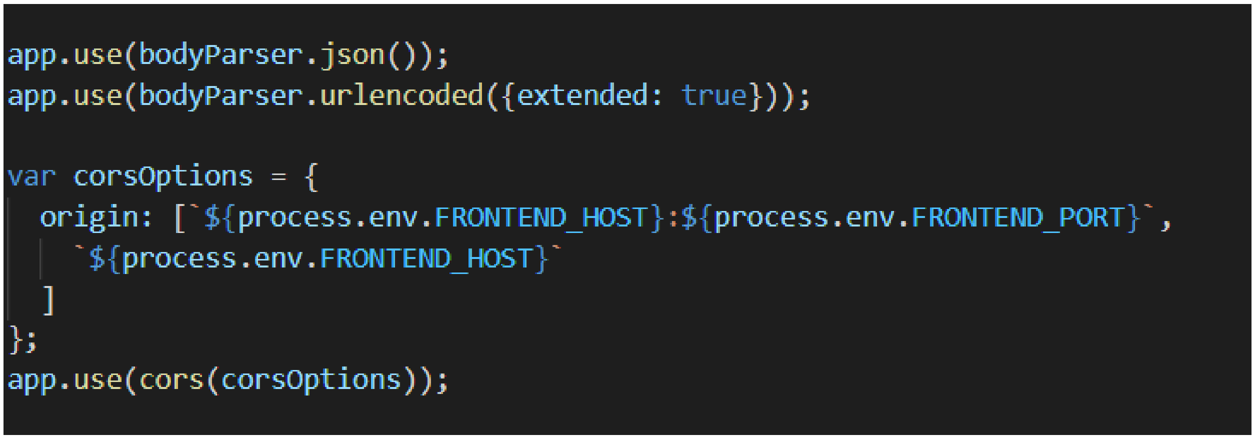
Backend

Mongo + ExpressJS



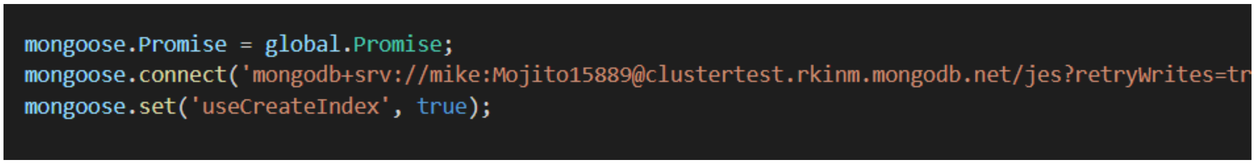
```javascript
const errorHandler  = require('./_helpers/error-handler');
require('dotenv').config();

mongoose.Promise = global.Promise;
mongoose.connect('mongodb+srv://mike:Mojito15889@clustertest.rkinm.mongodb.net/jes?retryWrites=true&w=majority', {useNewUrlParser: true, user:
mongoose.set('useCreateIndex', true);

var app = express()

app.use(bodyParser.json());
app.use(bodyParser.urlencoded({extended: true}));

var corsOptions = {
  origin: [`${process.env.FRONTEND_HOST}:${process.env.FRONTEND_PORT}`,
    `${process.env.FRONTEND_HOST}`
  ]
};
app.use(cors(corsOptions));

app.use(function (req, res, next) {
  res.setHeader('Access-Control-Allow-Credentials', true);
  res.setHeader('Access-Control-Allow-Methods', 'GET,PUT,POST,DELETE');
  res.setHeader('Access-Control-Allow-Origin', '*');
  res.setHeader("Access-Control-Allow-Headers", "Origin, X-Requested-With, Content-Type, Accept, Authorization");
  next();
});

routes(app);
// global error handler
app.use(errorHandler);

let server_port = process.env.BACKEND_PORT;
app.listen(server_port, function() {
  console.log('[:: INIT ::]');
  console.log(`CORS-enabled web server listening on port: ${server_port}` );
});
```

Agregar CORS



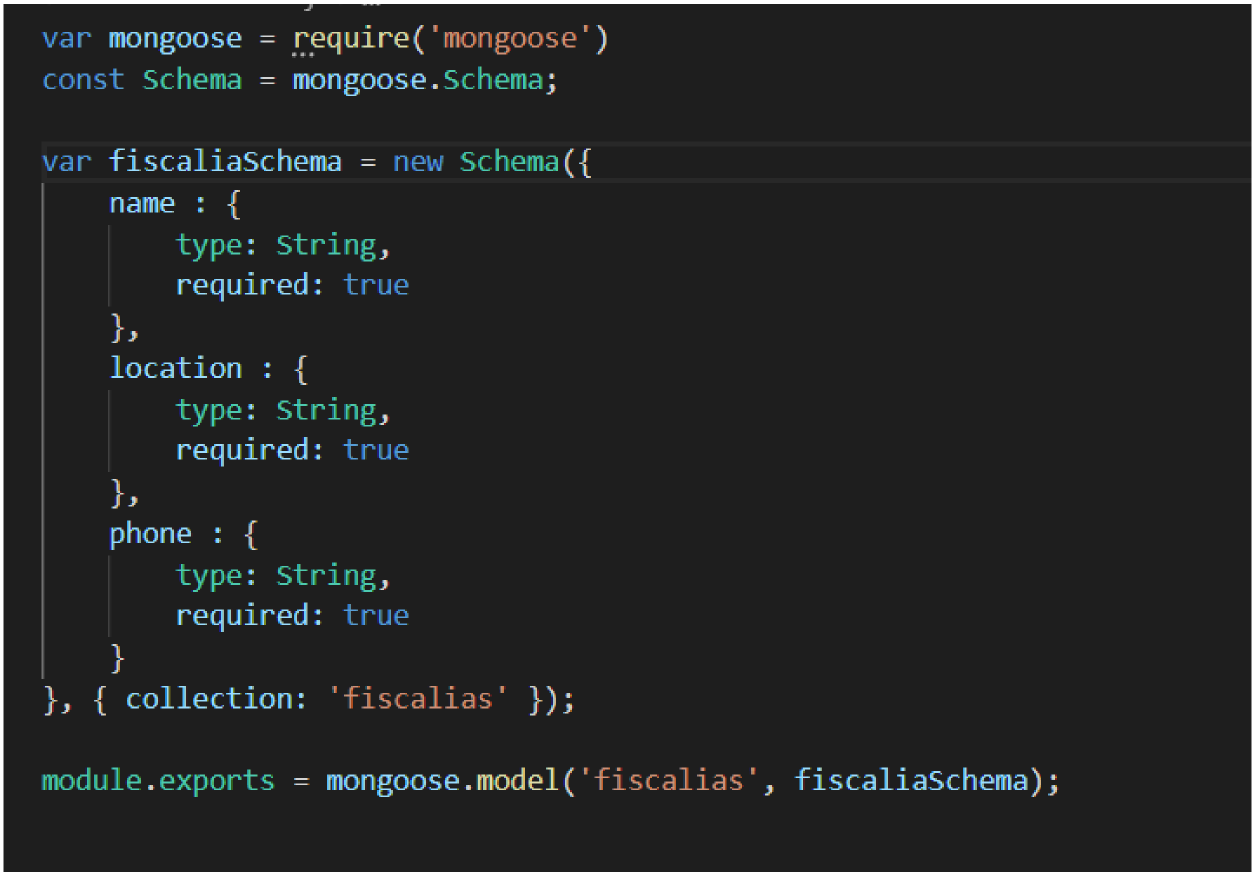
```javascript
app.use(bodyParser.json());
app.use(bodyParser.urlencoded({extended: true}));

var corsOptions = {
  origin: [`${process.env.FRONTEND_HOST}:${process.env.FRONTEND_PORT}`,
    `${process.env.FRONTEND_HOST}`
  ]
};
app.use(cors(corsOptions));
```

Agregar MongoPromise



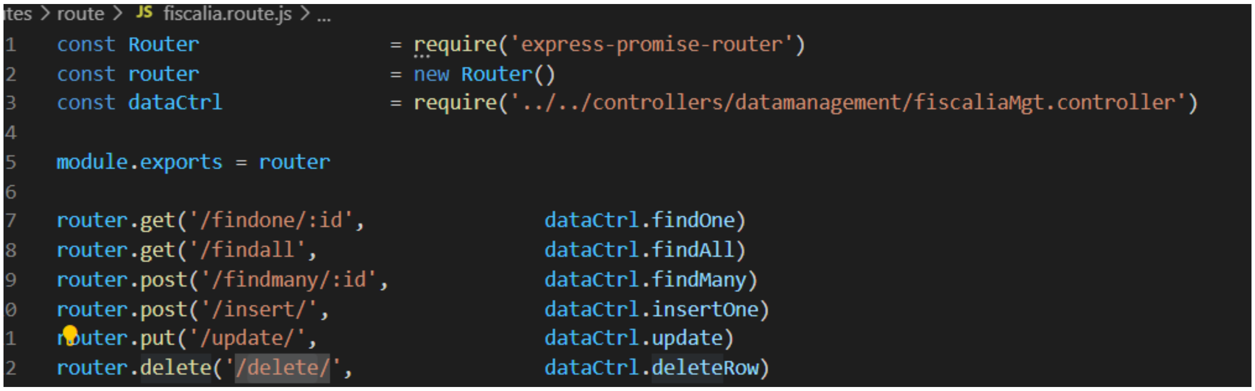
```
mongoose.Promise = global.Promise;
mongoose.connect('mongodb+srv://mike:Mojito15889@clustertest.rkinm.mongodb.net/jes?retryWrites=tr
mongoose.set('useCreateIndex', true);
```

Agregar Esquema



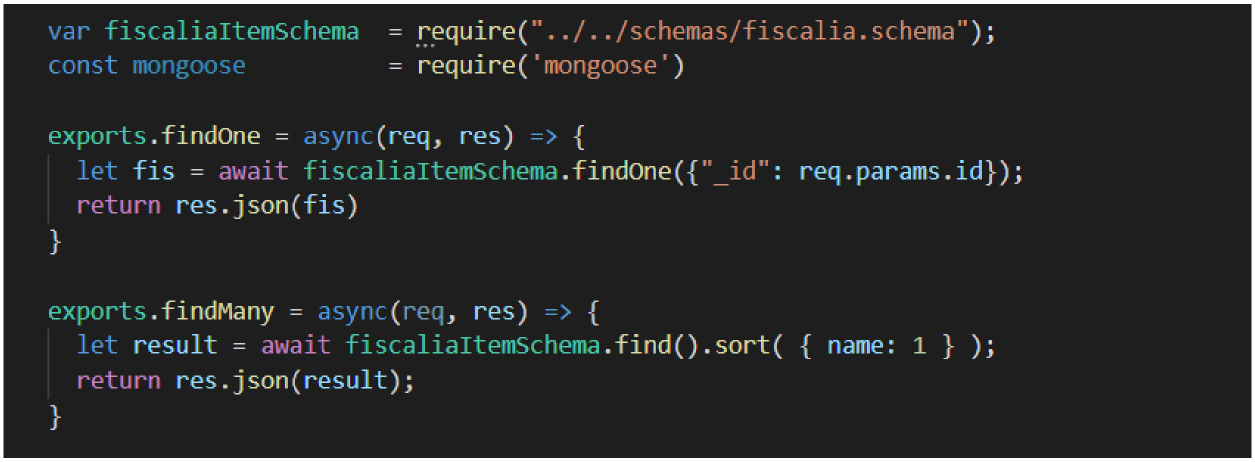
```
var mongoose = require('mongoose')
const Schema = mongoose.Schema;

var fiscaliaSchema = new Schema({
    name : {
        type: String,
        required: true
    },
    location : {
        type: String,
        required: true
    },
    phone : {
        type: String,
        required: true
    }
}, { collection: 'fiscalias' });

module.exports = mongoose.model('fiscalias', fiscaliaSchema);
```

Agregar Ruteo



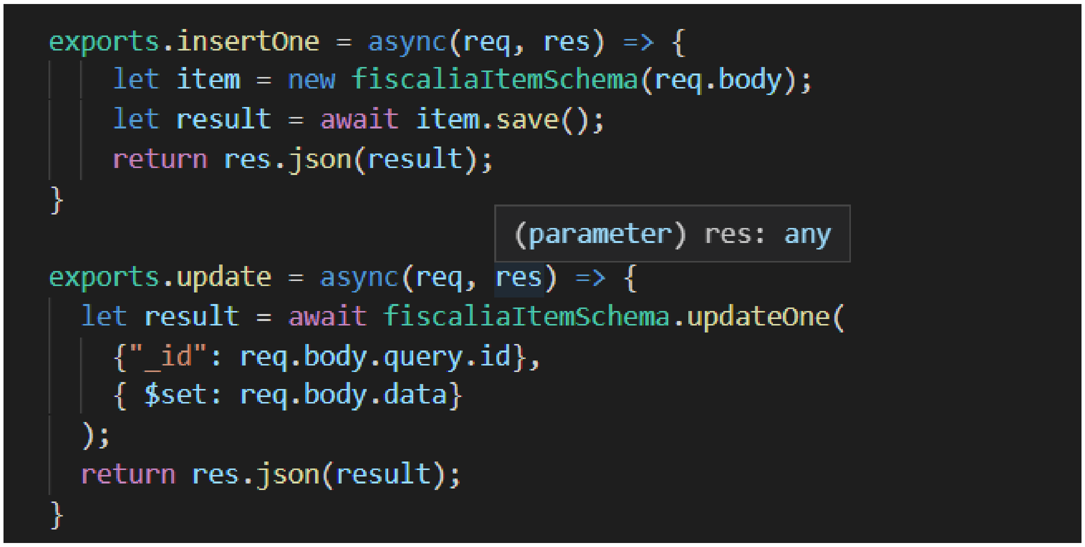
```
const Router                    = require('express-promise-router')
const router                    = new Router()
const dataCtrl                  = require('../../controllers/datamanagement/fiscaliaMgt.controller')

module.exports = router

router.get('/findone/:id',               dataCtrl.findOne)
router.get('/findall',                   dataCtrl.findAll)
router.post('/findmany/:id',             dataCtrl.findMany)
router.post('/insert/',                  dataCtrl.insertOne)
router.put('/update/',                   dataCtrl.update)
router.delete('/delete/',                dataCtrl.deleteRow)
```

Buscar datos en MongoDB



```
var fiscaliaItemSchema  = require("../../schemas/fiscalia.schema");
const mongoose          = require('mongoose')

exports.findOne = async(req, res) => {
  let fis = await fiscaliaItemSchema.findOne({"_id": req.params.id});
  return res.json(fis)
}

exports.findMany = async(req, res) => {
  let result = await fiscaliaItemSchema.find().sort( { name: 1 } );
  return res.json(result);
}
```

Insertar y Actualizar

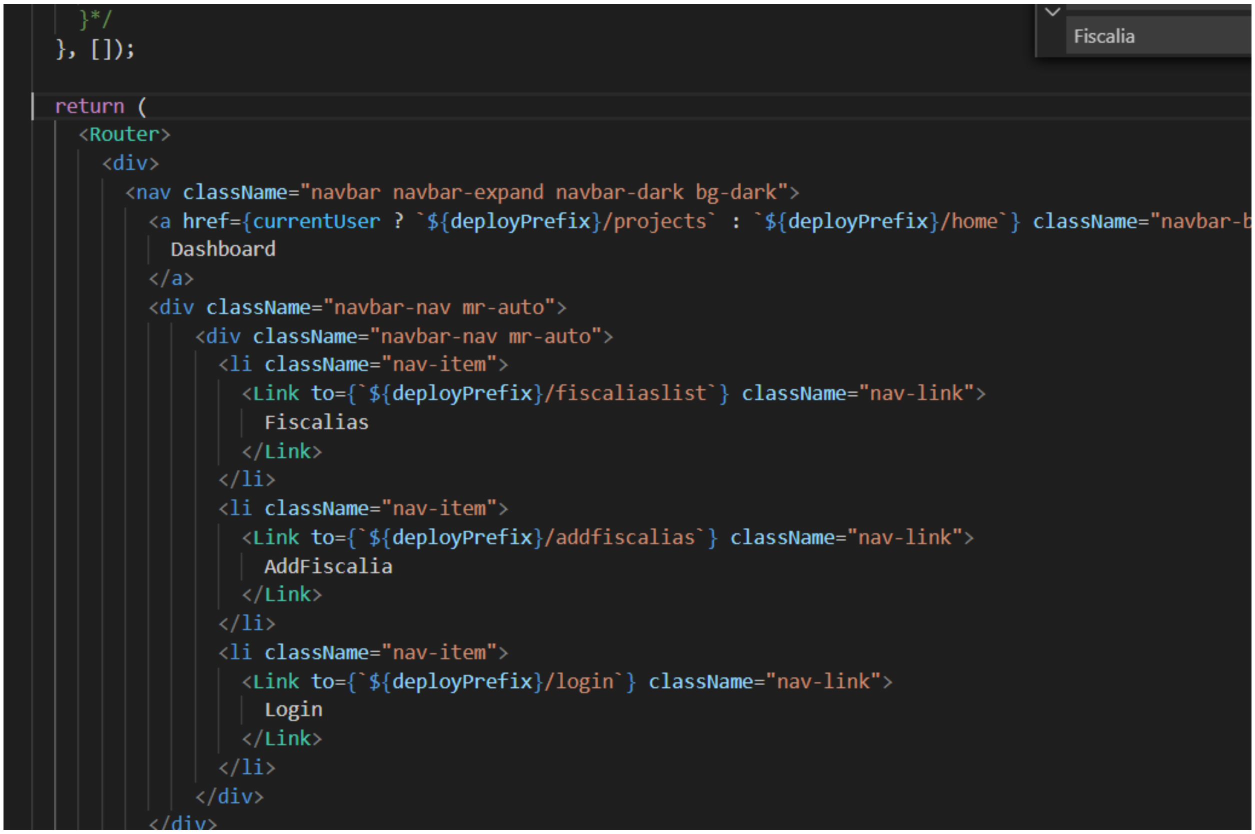


```
exports.insertOne = async(req, res) => {
    let item = new fiscaliaItemSchema(req.body);
    let result = await item.save();
    return res.json(result);
}

exports.update = async(req, res) => {
  let result = await fiscaliaItemSchema.updateOne(
    {"_id": req.body.query.id},
    { $set: req.body.data}
  );
  return res.json(result);
}
```

Frontend

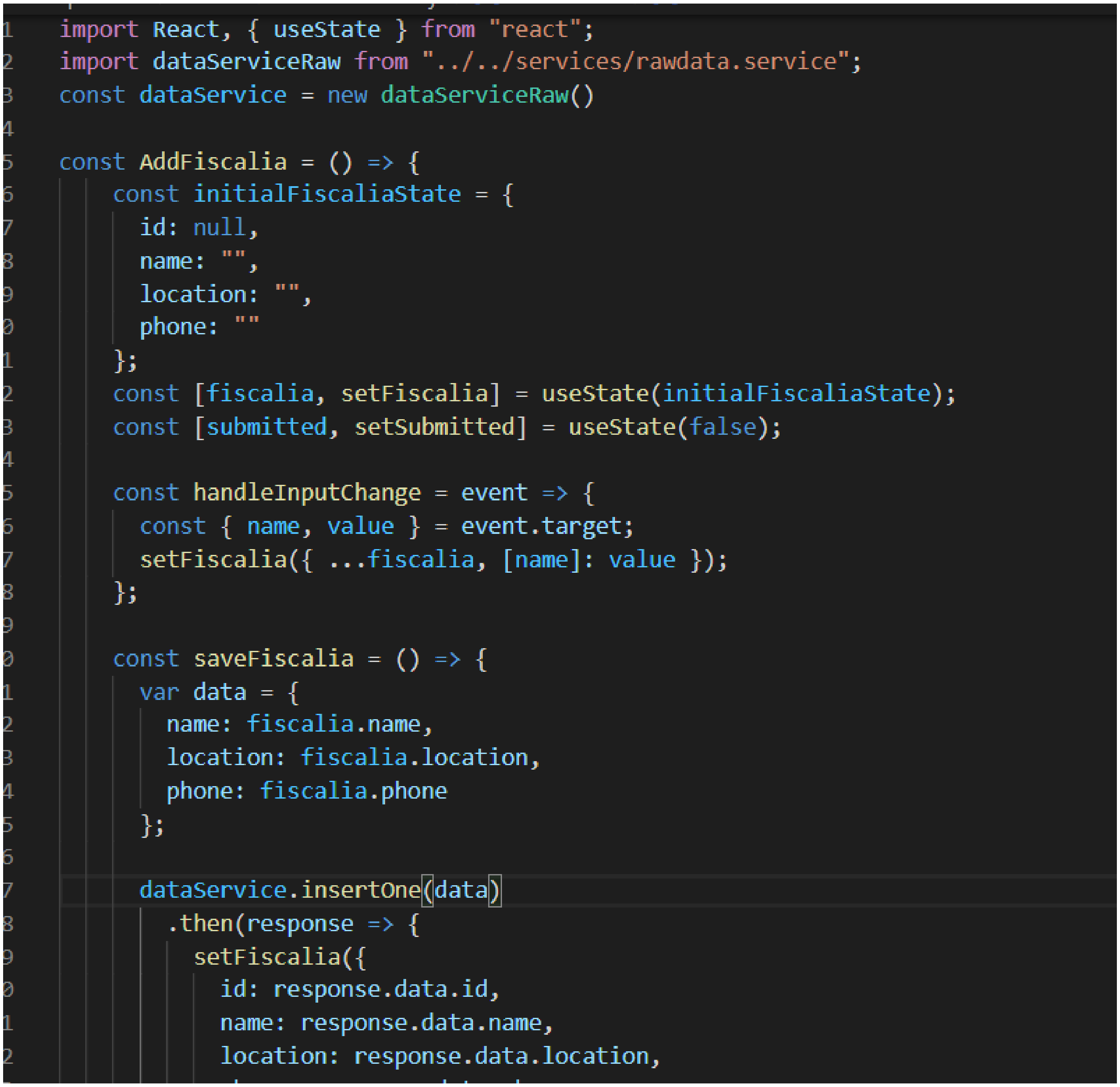
React

Pagina principal



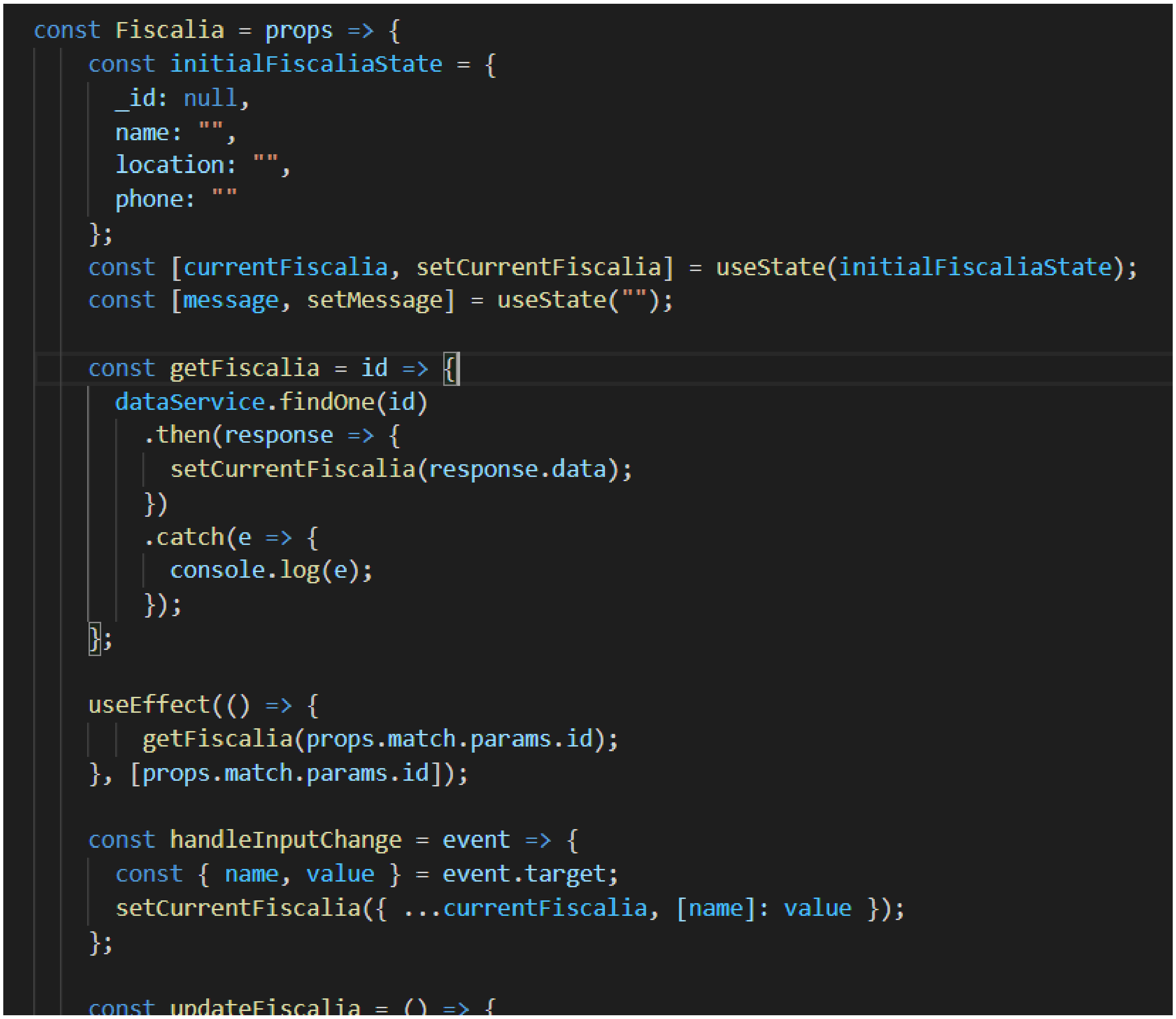
```
    }*/
  }, []);

  return (
    <Router>
      <div>
        <nav className="navbar navbar-expand navbar-dark bg-dark">
          <a href={currentUser ? `${deployPrefix}/projects` : `${deployPrefix}/home`} className="navbar-b
            Dashboard
          </a>
          <div className="navbar-nav mr-auto">
              <div className="navbar-nav mr-auto">
                <li className="nav-item">
                  <Link to={`${deployPrefix}/fiscaliaslist`} className="nav-link">
                    Fiscalias
                  </Link>
                </li>
                <li className="nav-item">
                  <Link to={`${deployPrefix}/addfiscalias`} className="nav-link">
                    AddFiscalia
                  </Link>
                </li>
                <li className="nav-item">
                  <Link to={`${deployPrefix}/login`} className="nav-link">
                    Login
                  </Link>
                </li>
              </div>
          </div>
```

Add Fiscalia con HOOKS



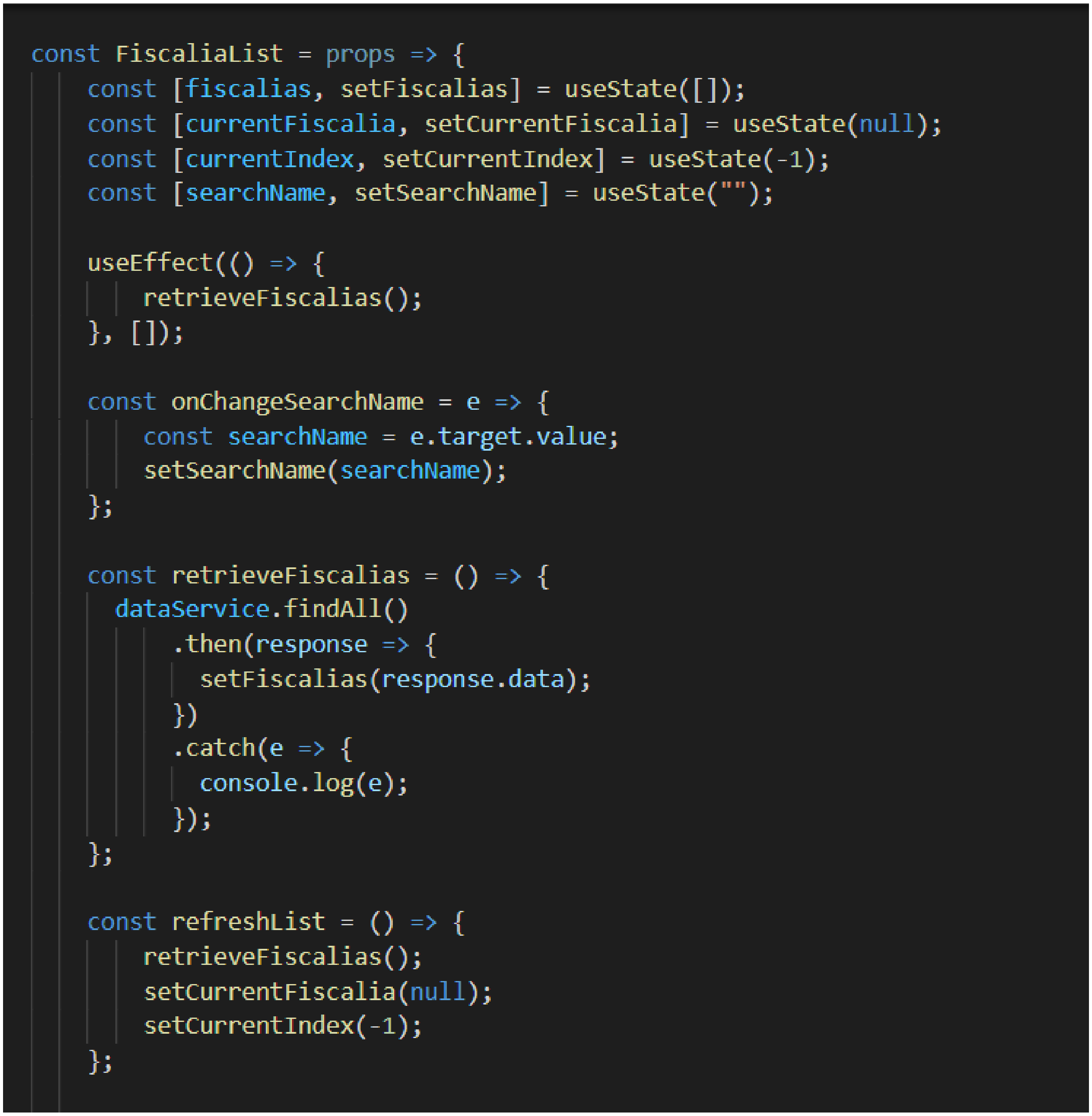
```
import React, { useState } from "react";
import dataServiceRaw from "../../services/rawdata.service";
const dataService = new dataServiceRaw()

const AddFiscalia = () => {
    const initialFiscaliaState = {
      id: null,
      name: "",
      location: "",
      phone: ""
    };
    const [fiscalia, setFiscalia] = useState(initialFiscaliaState);
    const [submitted, setSubmitted] = useState(false);

    const handleInputChange = event => {
      const { name, value } = event.target;
      setFiscalia({ ...fiscalia, [name]: value });
    };

    const saveFiscalia = () => {
      var data = {
        name: fiscalia.name,
        location: fiscalia.location,
        phone: fiscalia.phone
      };

      dataService.insertOne(data)
        .then(response => {
          setFiscalia({
            id: response.data.id,
            name: response.data.name,
            location: response.data.location,
```

Obtener Fiscalia



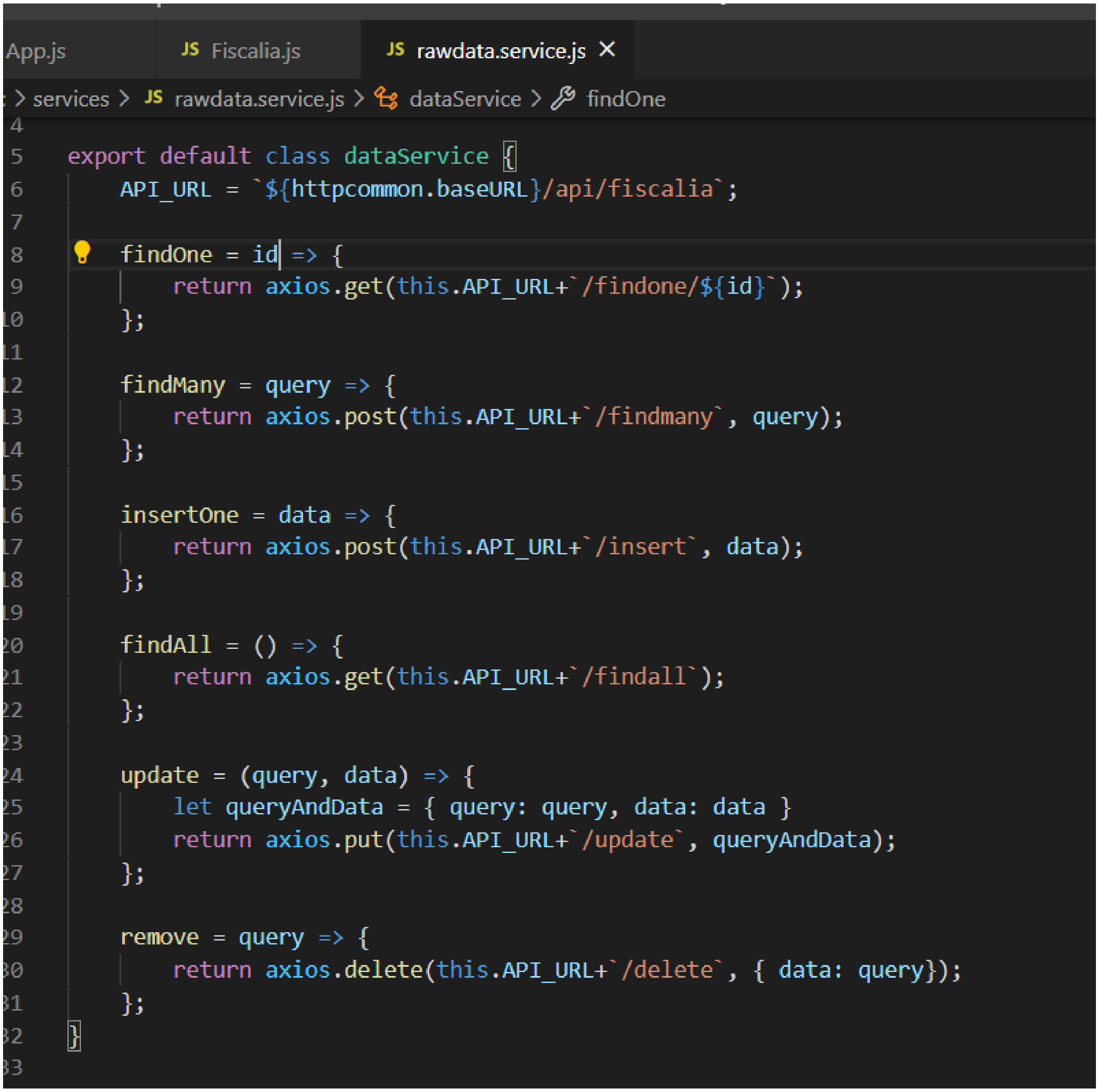
```
const Fiscalia = props => {
    const initialFiscaliaState = {
      _id: null,
      name: "",
      location: "",
      phone: ""
    };
    const [currentFiscalia, setCurrentFiscalia] = useState(initialFiscaliaState);
    const [message, setMessage] = useState("");

    const getFiscalia = id => {
      dataService.findOne(id)
        .then(response => {
          setCurrentFiscalia(response.data);
        })
        .catch(e => {
          console.log(e);
        });
    };

    useEffect(() => {
        getFiscalia(props.match.params.id);
    }, [props.match.params.id]);

    const handleInputChange = event => {
      const { name, value } = event.target;
      setCurrentFiscalia({ ...currentFiscalia, [name]: value });
    };

    const updateFiscalia = () => {
```

Listar Fiscalia



```
const FiscaliaList = props => {
    const [fiscalias, setFiscalias] = useState([]);
    const [currentFiscalia, setCurrentFiscalia] = useState(null);
    const [currentIndex, setCurrentIndex] = useState(-1);
    const [searchName, setSearchName] = useState("");

    useEffect(() => {
        retrieveFiscalias();
    }, []);

    const onChangeSearchName = e => {
        const searchName = e.target.value;
        setSearchName(searchName);
    };

    const retrieveFiscalias = () => {
      dataService.findAll()
          .then(response => {
            setFiscalias(response.data);
          })
          .catch(e => {
            console.log(e);
          });
    };

    const refreshList = () => {
        retrieveFiscalias();
        setCurrentFiscalia(null);
        setCurrentIndex(-1);
    };
```

Servicio



```
export default class dataService {
    API_URL = `${httpcommon.baseURL}/api/fiscalia`;

    findOne = id => {
        return axios.get(this.API_URL+`/findone/${id}`);
    };

    findMany = query => {
        return axios.post(this.API_URL+`/findmany`, query);
    };

    insertOne = data => {
        return axios.post(this.API_URL+`/insert`, data);
    };

    findAll = () => {
        return axios.get(this.API_URL+`/findall`);
    };

    update = (query, data) => {
        let queryAndData = { query: query, data: data }
        return axios.put(this.API_URL+`/update`, queryAndData);
    };

    remove = query => {
        return axios.delete(this.API_URL+`/delete`, { data: query});
    };
}
```