

FRONTEND

Formulario

React App x +

localhost:3000

Bookmarks Inventario de Siste... Utilidades

GESTION DE FISCALIAS

Ingresar:

Fiscalia* Direccion* Telefono* Departamento*

Editar:

Fiscalia* Direccion* Telefono* Departamento*

Listado de Fiscalias:

Departamento

- Todas -

Fiscalia	Direccion	Telefono	Departamento	Acciones
MP001	Avenida las Americas 18-36 Zona 13, Edificio CPS Segundo Nivel Informatica	41513253	CHIQUIMULA	X
MP002	Avenida las Americas 18-36 Zona 13, Edificio CPS 2do. Nivel Informatica	3242342	GUATEMALA	X
MP003	salud publica	678966444	HUEHUETENANGO	X

Imports

```
JS App.js 4 x # index.css
src > JS App.js > App

> GuardarFiscalia Aa .ab 2 of 2

1 import React, { useState, useEffect } from "react";
2 import Container from 'react-bootstrap/Container';
3 import Row from 'react-bootstrap/Row';
4 import Col from 'react-bootstrap/Col';
5 import Card from 'react-bootstrap/Card';
6 import Form from 'react-bootstrap/Form';
7 import Button from 'react-bootstrap/Button';
8 import Table from 'react-bootstrap/Table';
9 import { useForm } from "react-hook-form";
10 import './index.css';
11
12
13 export default function App() {
14
15
16   const { register, formState: { errors }, reset, handleSubmit } = useForm();
17
18   const [
19     register: register2,
20     setValue,
21     formState: { errors: errors2 },
22     reset: reset2,
23     handleSubmit: handleSubmit2,
24   ] = useForm({
25     mode: "onBlur",
26   });
27
28   const [data, setData] = useState("");
29
30
```

Consumos a backend nodejs para obtención de fiscalías

```
29
30
31 const [listDepartamentos, setDepartamentos] = useState([]);
32 const getDepartamentos = async () => {
33   const response = await fetch(
34     "http://localhost:4000/Departamentos"
35   );
36   const data = await response.json();
37   setDepartamentos(data);
38 };
39
40 const [listFiscalias, setFiscalias] = useState([]);
41 const getFiscalias = async () => {
42   const response = await fetch(
43     "http://localhost:4000/Departamento/0/ObtenerFiscalias"
44   );
45   const data = await response.json();
46   setFiscalias(data);
47 };
48
```

Consumos a backend java para gestión de fiscalías

Agregar Fiscalía

```
49
50 const GuardarFiscalia = data => {
51   console.log("GUARDAR::", data);
52
53   fetch('http://localhost:8080/fiscalias', {
54     method: 'POST',
55     headers: {
56       'Content-Type': 'application/json',
57     },
58     body: JSON.stringify(
59       data
60     ),
61   })
62   .then((res) => res.json())
63   .then((result) => {
64     if (!(result.status === 500)) {
65       alert("Fiscalia registrada con exito.");
66       getFiscalias();
67       console.log(result);
68       reset();
69     } else {
70       alert("Error ya existe una fiscalia con ese nombre.");
71     }
72   })
73   .catch((err) => {
74     alert("Error con el servidor.");
75   })
76
77   return false;
78 }
```

Editar Fiscalía

```
const EditarFiscalia = data => {  
  
  console.log(data);  
  
  fetch('http://localhost:8080/fiscalias', {  
    method: 'PUT',  
    headers: {  
      'Content-Type': 'application/json',  
    },  
    body: JSON.stringify(data ),  
  })  
  .then((res) => res.json())  
  .then((result) => {  
  
    if (!(result.status === 500)) {  
      alert("Fiscalia Editada con exito.");  
      getFiscalias();  
      console.log(result);  
      reset2();  
    } else {  
      alert("Error ya existe una fiscalia con ese nombre.");  
    }  
  })  
  .catch((err) => {  
    alert("Error con el servidor.");  
  })  
  
  return false;  
}
```

Eliminar Fiscalía

```
129  
130 const BorrarFiscalia = data => {  
131   console.log("BORRAR FISCALIA ::", data);  
132  
133  
134   fetch('http://localhost:8080/fiscalias', {  
135     method: 'DELETE',  
136     headers: {  
137       'Content-Type': 'application/json',  
138     },  
139     body: JSON.stringify(data),  
140   })  
141   // .then((res) => res.json())  
142   .then((result) => {  
143     getFiscalias();  
144     // console.log(result);  
145   })  
146   // .catch((err) => {
```

Creación de Formularios

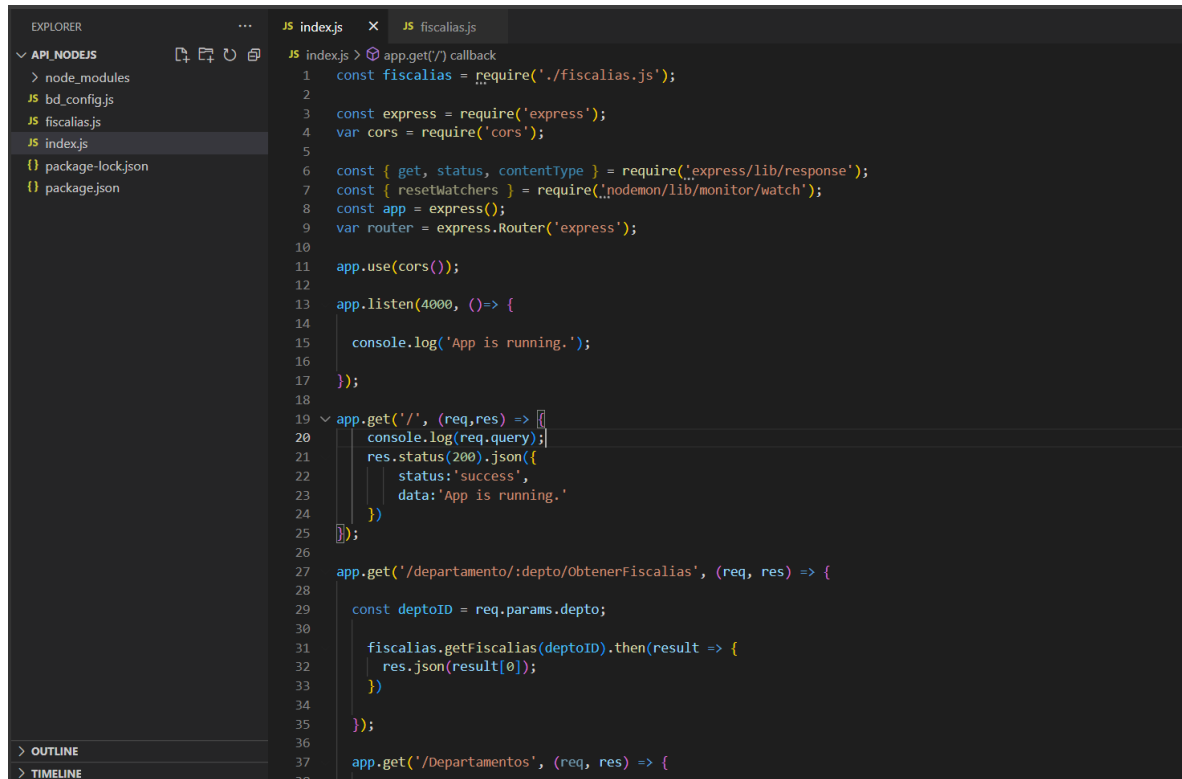
```
172 <div className="form-container">
173   <Container>
174     <Row>
175       <Col>
176         <Card.Title>GESTION DE FISCALIAS</Card.Title>
177       </Col>
178     </Row>
179     <hr></hr>
180     <Row>
181       <Col>
182         <Card.Subtitle>Ingresar:</Card.Subtitle>
183       </Col>
184     </Row>
185     <form key={1} onSubmit={handleSubmit(GuardarFiscalia)}>
186       <Row>
187         <Col>
188           <label>Fiscalia*</label>
189           <input {...register("fiscalia", { required: true })} placeholder="" />
190           {errors.FISCALIA?.type === 'required' && "campo obligatorio"}
191         </Col>
192         <Col xs={4}>
193           <label>Direccion*</label>
194           <input {...register("direccion", { required: true })} placeholder="" />
195           {errors.DIRECCION?.type === 'required' && "campo obligatorio"}
196         </Col>
197         <Col>
198           <label>Telefono*</label>
199           <input {...register("telefono", { required: true })} placeholder="" />
200           {errors.TELEFONO?.type === 'required' && "campo obligatorio"}
```

```
<form key={2} onSubmit={handleSubmit2(EditarFiscalia)}>
  <Row>
    <Col>
      <label>Fiscalia*</label>
      <select {...register2("id_fiscalia", { required: true })} onChange={SeleccionarFiscalia}>
        <option></option>
        {listFiscalias.map((option, index) => {
          return (
            <option key={index} value={option.id_fiscalia}>
              {option.fiscalia}
            </option>
          );
        })}
      </select>
      {errors2.ID_FISCALIA?.type === 'required' && "campo obligatorio"}
    </Col>
    <Col xs={4}>
      <label>Direccion*</label>
      <input {...register2("direccion", { required: true })} placeholder="" />
      {errors2.DIRECCION?.type === 'required' && "campo obligatorio"}
    </Col>
    <Col>
      <label>Telefono*</label>
      <input {...register2("telefono", { required: true })} placeholder="" />
      {errors2.TELEFONO?.type === 'required' && "campo obligatorio"}
    </Col>
    <Col>
      <label>Departamento*</label>
      <select {...register2("id_departamento", { required: true })}>
```

```
<Row>
  <Col>
    <select onChange={FiltrarFiscalias}>
      <option value="0"> - Todas - </option>
      {listDepartamentos.map((option2, index) => {
        return (
          <option key={index} value={option2.id_departamento}>
            {option2.departamento}
          </option>
        );
      })}
    </select>
  </Col>
</Row>
<Col>
  <label></label>
</Col>
<Row>
  <Col>
    <Table striped bordered hover>
      <thead>
        <tr>
          <th>Fiscalia</th>
          <th>Direccion</th>
          <th>Telefono</th>
          <th>Departamento</th>
          <th>Acciones</th>
        </tr>
      </thead>
      <tbody>
        {listFiscalias.map((fis) => {
```

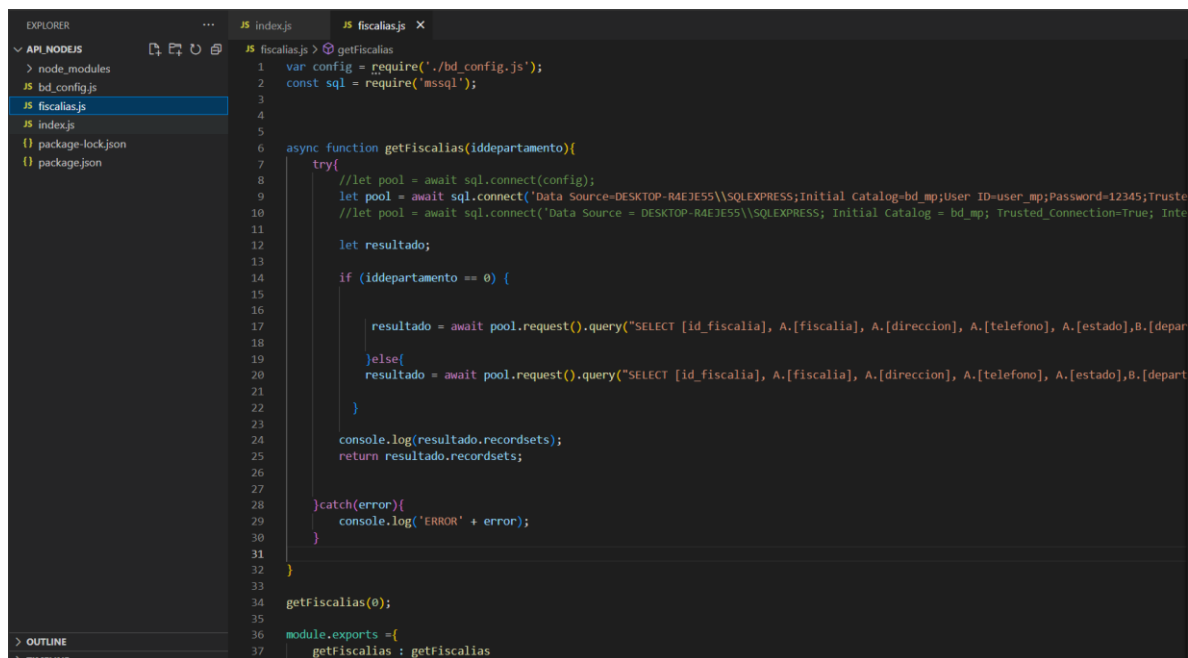
BACKEND - NODEJS

Configuración de aplicación y rutas



```
1  const fiscalias = require('./fiscalias.js');
2
3  const express = require('express');
4  var cors = require('cors');
5
6  const { get, status, contentType } = require('express/lib/response');
7  const { resetWatchers } = require('nodemon/lib/monitor/watch');
8  const app = express();
9  var router = express.Router('express');
10
11 app.use(cors());
12
13 app.listen(4000, () => {
14
15   console.log('App is running. ');
16
17 });
18
19 app.get('/', (req, res) => {
20   console.log(req.query);
21   res.status(200).json({
22     status: 'success',
23     data: 'App is running.'
24   });
25 });
26
27 app.get('/departamento/:depto/ObtenerFiscalias', (req, res) => {
28
29   const deptoID = req.params.depto;
30
31   fiscalias.getFiscalias(deptoID).then(result => {
32     res.json(result[0]);
33   });
34
35 });
36
37 app.get('/Departamentos', (req, res) => {
```

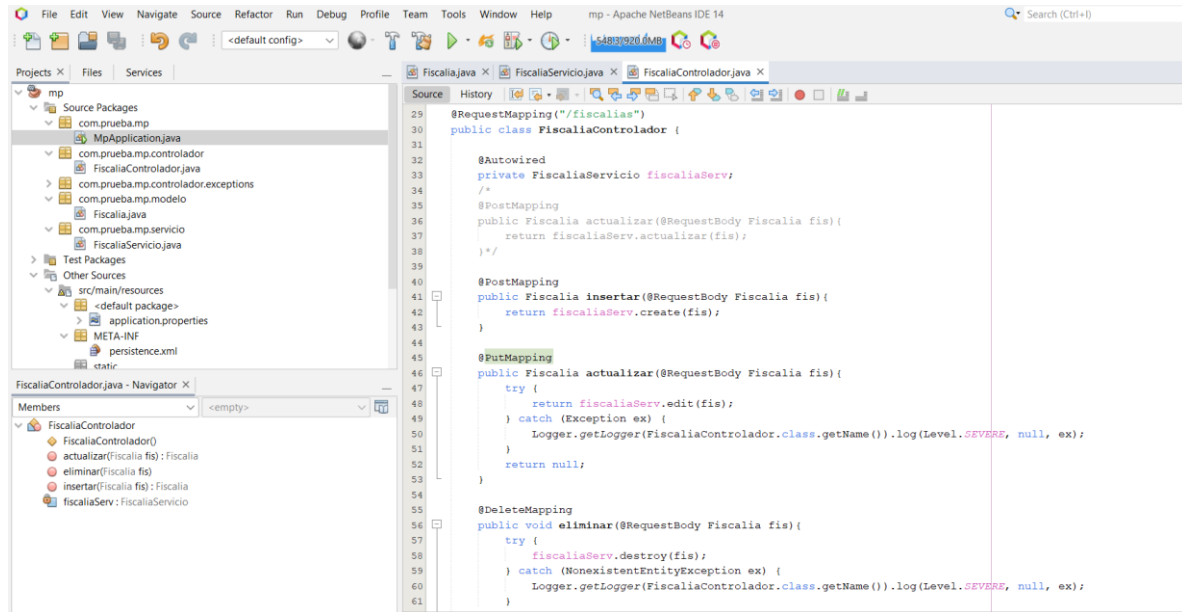
Obtención de Fiscalías por departamento



```
1  var config = require('./bd_config.js');
2  const sql = require('mssql');
3
4
5
6  async function getFiscalias(iddepartamento){
7    try{
8      //let pool = await sql.connect(config);
9      let pool = await sql.connect('Data Source=DESKTOP-R4EJE55\\SQLEXPRESS;Initial Catalog=bd_mp;User ID=user_mp;Password=12345;Truste
10     //let pool = await sql.connect('Data Source = DESKTOP-R4EJE55\\SQLEXPRESS; Initial Catalog = bd_mp; Trusted_Connection=True; Inte
11
12     let resultado;
13
14     if (iddepartamento == 0) {
15
16
17       resultado = await pool.request().query("SELECT [id_fiscalia], A.[fiscalia], A.[direccion], A.[telefono], A.[estado],B.[depar
18
19     }else{
20       resultado = await pool.request().query("SELECT [id_fiscalia], A.[fiscalia], A.[direccion], A.[telefono], A.[estado],B.[depart
21
22     }
23
24     console.log(resultado.recordsets);
25     return resultado.recordsets;
26
27   }catch(error){
28     console.log('ERROR' + error);
29   }
30 }
31
32
33
34 getFiscalias(0);
35
36 module.exports = {
37   getFiscalias : getFiscalias
```

BACKEND - JAVA

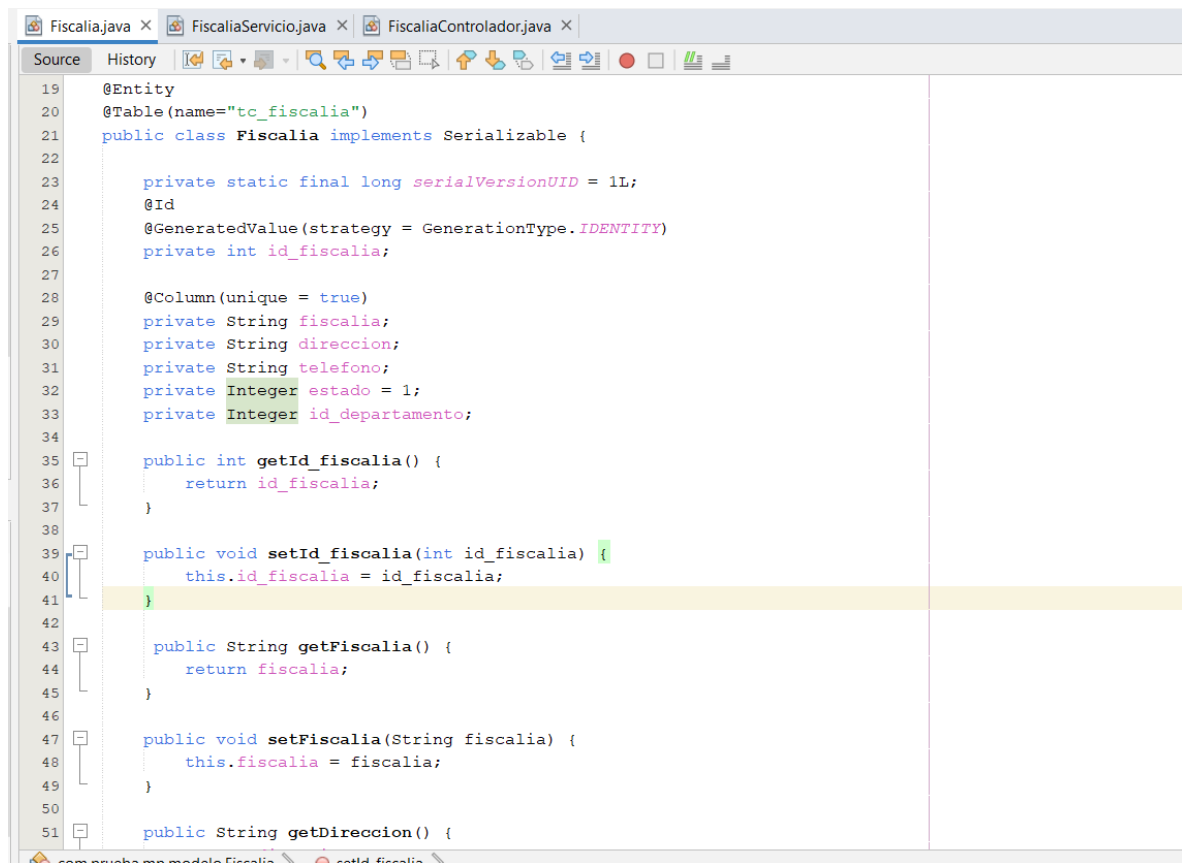
Definición de Rutas



The screenshot shows the NetBeans IDE with the `FiscaliaControlador.java` file open. The code defines several REST endpoints using Spring MVC annotations:

```
29 @RequestMapping("/fiscalias")
30 public class FiscaliaControlador {
31
32     @Autowired
33     private FiscaliaServicio fiscaliaServ;
34     /*
35     @PostMapping
36     public Fiscalia actualizar(@RequestBody Fiscalia fis) {
37         return fiscaliaServ.actualizar(fis);
38     }
39
40     @PostMapping
41     public Fiscalia insertar(@RequestBody Fiscalia fis) {
42         return fiscaliaServ.create(fis);
43     }
44
45     @PutMapping
46     public Fiscalia actualizar(@RequestBody Fiscalia fis) {
47         try {
48             return fiscaliaServ.edit(fis);
49         } catch (Exception ex) {
50             Logger.getLogger(FiscaliaControlador.class.getName()).log(Level.SEVERE, null, ex);
51         }
52         return null;
53     }
54
55     @DeleteMapping
56     public void eliminar(@RequestBody Fiscalia fis) {
57         try {
58             fiscaliaServ.destroy(fis);
59         } catch (NoSuchEntityException ex) {
60             Logger.getLogger(FiscaliaControlador.class.getName()).log(Level.SEVERE, null, ex);
61         }
62     }
63 }
```

Modelo de persistencia – Fiscalia.java



The screenshot shows the NetBeans IDE with the `Fiscalia.java` file open. The code defines the `Fiscalia` entity using JPA annotations:

```
19 @Entity
20 @Table(name="tc_fiscalia")
21 public class Fiscalia implements Serializable {
22
23     private static final long serialVersionUID = 1L;
24     @Id
25     @GeneratedValue(strategy = GenerationType.IDENTITY)
26     private int id_fiscalia;
27
28     @Column(unique = true)
29     private String fiscalia;
30     private String direccion;
31     private String telefono;
32     private Integer estado = 1;
33     private Integer id_departamento;
34
35     public int getId_fiscalia() {
36         return id_fiscalia;
37     }
38
39     public void setId_fiscalia(int id_fiscalia) {
40         this.id_fiscalia = id_fiscalia;
41     }
42
43     public String getFiscalia() {
44         return fiscalia;
45     }
46
47     public void setFiscalia(String fiscalia) {
48         this.fiscalia = fiscalia;
49     }
50
51     public String getDireccion() {
52         return direccion;
53     }
54 }
```

Funciones de Persistencias – FiscaliaServicios.java

Crear

```
Fiscalia.java x FiscaliaServicio.java x FiscaliaControlador.java x
Source History
25
26 public FiscaliaServicio(EntityManagerFactory emf) {
27     this.emf = emf;
28 }
29 private EntityManagerFactory emf = null;
30
31 public EntityManager getEntityManager() {
32     return emf.createEntityManager();
33 }
34
35 public Fiscalia create(Fiscalia fiscalia) {
36     EntityManager em = null;
37     try {
38         em = getEntityManager();
39         em.getTransaction().begin();
40         em.persist(fiscalia);
41         em.getTransaction().commit();
42     } finally {
43         if (em != null) {
44             em.close();
45         }
46     }
47
48     return fiscalia;
49 }
50
```

Editar

```
public Fiscalia edit(Fiscalia fiscalia) throws NonexistentEntityException, Exception {
    EntityManager em = null;
    try {
        em = getEntityManager();
        em.getTransaction().begin();
        fiscalia = em.merge(fiscalia);
        em.getTransaction().commit();
    } catch (Exception ex) {
        String msg = ex.getLocalizedMessage();
        if (msg == null || msg.length() == 0) {
            int id = fiscalia.getId_fiscalia();
            if (findFiscalia(id) == null) {
                throw new NonexistentEntityException("The fiscalia with id " + id + " no longer exists.");
            }
        }
        throw ex;
    } finally {
        if (em != null) {
            em.close();
        }
    }

    return fiscalia;
}
```


Eliminar

```
76 public void destroy(Fiscalia fis) throws NonexistentEntityException {
77     EntityManager em = null;
78     try {
79         em = getEntityManager();
80         em.getTransaction().begin();
81         Fiscalia fiscalia;
82         try {
83             fiscalia = em.getReference(Fiscalia.class, fis.getId_fiscalia());
84             fiscalia.getId_fiscalia();
85         } catch (EntityNotFoundException enfe) {
86             throw new NonexistentEntityException("The fiscalia with id " + fis.getId_fiscalia() + " no longer exists."
87         )
88         em.remove(fiscalia);
89         em.getTransaction().commit();
90     } finally {
91         if (em != null) {
92             em.close();
93         }
94     }
95 }
96 }
```