

# TECNOLOGICO DE ESTODIOS SUPERIORES DE ECATEPEC

División de ingeniería en Sistemas Computacionales



Proyecto:

Registro de visitas internas

Equipo: Codex

Materia: Base de datos para dispositivos móviles

Segunda Vuelta

Grupo: 5851

Nombre: Leyva Gabriel Miguel Angel

## Modulo Funcionando

The image shows a web application interface for registering a visit and a MongoDB Compass window displaying the data stored in a database.

**Web Application: Registrar Visita**

Agendar Cita de Usuario Interno

Empleado de area:  Area:  Asunto:

Fecha:  Hora:

**MongoDB Compass: citas.empleados**

Documents

DOCUMENTS 3 STORAGE SIZE 20.5KB AVG. SIZE 125B INDEXES 1 TOTAL SIZE 36.9KB AVG. SIZE 36.9KB

Documents Aggregations Schema Explain Plan Indexes Validation

FILTER { field: 'value' } OPTIONS FIND RESET

ADD DATA VIEW

Displaying documents 1 - 3 of 3 REFRESH

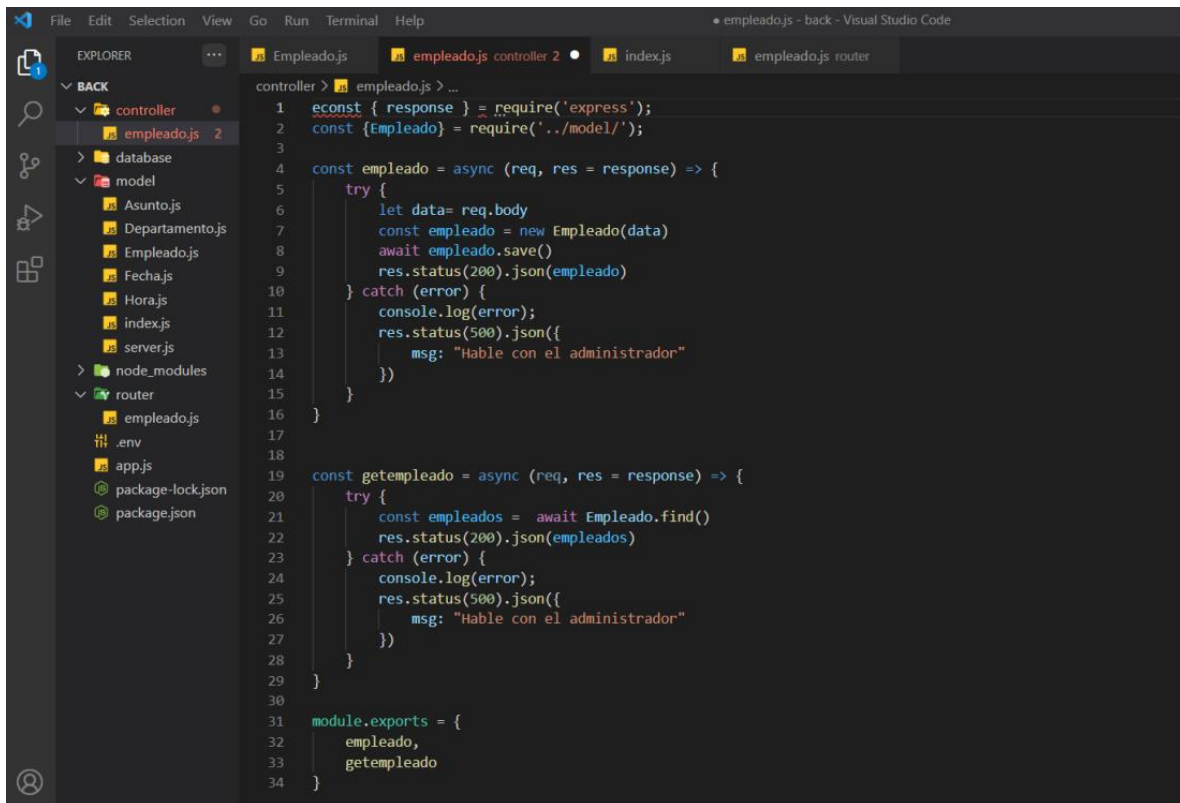
```
{
  "_id": ObjectId("61f0e54d85363666711bb105"),
  "empleado": "hola",
  "area": "rr",
  "asunto": "prueba",
  "fecha": "22/01/2022",
  "hora": "13:00",
  "__v": 0
}
```

```
{
  "_id": ObjectId("61f0e5d485363666711bb107"),
  "empleado": "prueba",
  "area": "conta",
  "asunto": "pr1",
  "fecha": "22/01/2022",
  "hora": "11:00",
  "__v": 0
}
```

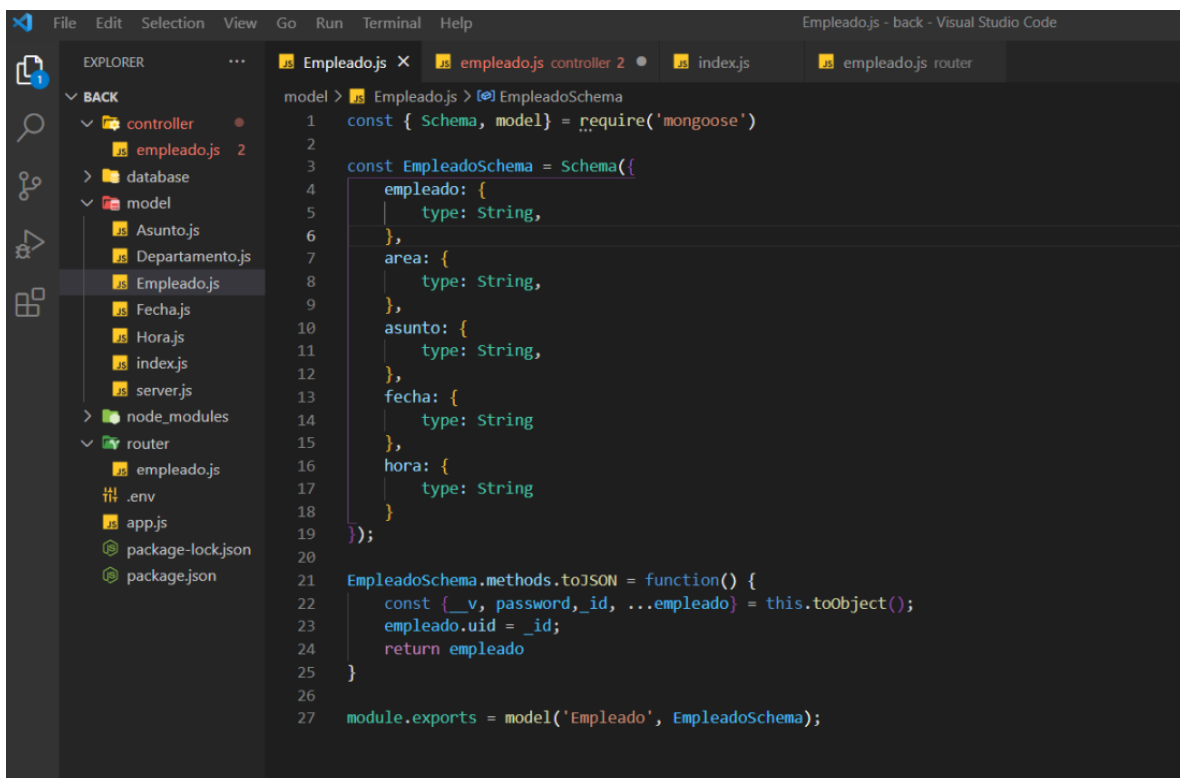
```
{
  "_id": ObjectId("61f0e85385363666711bb10a"),
  "empleado": "Manuel",
  "area": "Contabilidad",
  "asunto": "prueba1",
  "fecha": "25/01/2022",
  "hora": "15:00",
  "__v": 0
}
```

> \_MONGOSH

## Código de conexión a la base de datos



```
1  const { response } = require('express');
2  const { Empleado } = require('../model/');
3
4  const empleado = async (req, res = response) => {
5    try {
6      let data = req.body
7      const empleado = new Empleado(data)
8      await empleado.save()
9      res.status(200).json(empleado)
10   } catch (error) {
11     console.log(error);
12     res.status(500).json({
13       msg: "Hable con el administrador"
14     })
15   }
16 }
17
18
19 const getempleado = async (req, res = response) => {
20   try {
21     const empleados = await Empleado.find()
22     res.status(200).json(empleados)
23   } catch (error) {
24     console.log(error);
25     res.status(500).json({
26       msg: "Hable con el administrador"
27     })
28   }
29 }
30
31 module.exports = {
32   empleado,
33   getempleado
34 }
```



```
1  const { Schema, model } = require('mongoose')
2
3  const EmpleadoSchema = Schema({
4    empleado: {
5      type: String,
6    },
7    area: {
8      type: String,
9    },
10    asunto: {
11      type: String,
12    },
13    fecha: {
14      type: String
15    },
16    hora: {
17      type: String
18    }
19  });
20
21 EmpleadoSchema.methods.toJSON = function() {
22   const { _v, password, _id, ...empleado } = this.toObject();
23   empleado.uid = _id;
24   return empleado
25 }
26
27 module.exports = model('Empleado', EmpleadoSchema);
```

The screenshot shows the Visual Studio Code interface with the Explorer sidebar on the left. The Explorer is expanded to show the 'router' folder, which contains 'empleado.js'. The main editor area displays the code for 'empleado.js' with the following content:

```
router > empleado.js > ...
1  const { Router } = require('express');
2  const { empleado, getempleado } = require('../controller/empleado');
3
4  // ---
5  const routes = Router();
6
7  // post empleado
8  routes.post('/', empleado);
9  // Get
10 routes.get('/', getempleado);
11
12 module.exports = routes;
```

The screenshot shows the Visual Studio Code interface with the Explorer sidebar on the left. The Explorer is expanded to show the 'database' folder, which contains 'config.js'. The main editor area displays the code for 'config.js' with the following content:

```
database > config.js > ...
1  const mongoose = require('mongoose');
2
3  const dbConection = async() =>{
4      try{
5          await mongoose.connect(process.env.MONGODB_ATLAS)
6          console.log("Base de datos online");
7      }catch(error){
8          console.log(error);
9          throw new Error('Error al iniciar conexion')
10     }
11 }
12
13 module.exports = {
14     dbConection
15 }
```

The screenshot shows the Visual Studio Code interface with the Explorer sidebar on the left. The Explorer sidebar is expanded to show the 'model' directory, which contains several files: Asunto.js, Departamento.js, Empleado.js, Fecha.js, Hora.js, index.js, and server.js. The 'index.js' file is selected and its content is displayed in the main editor area. The code in index.js is as follows:

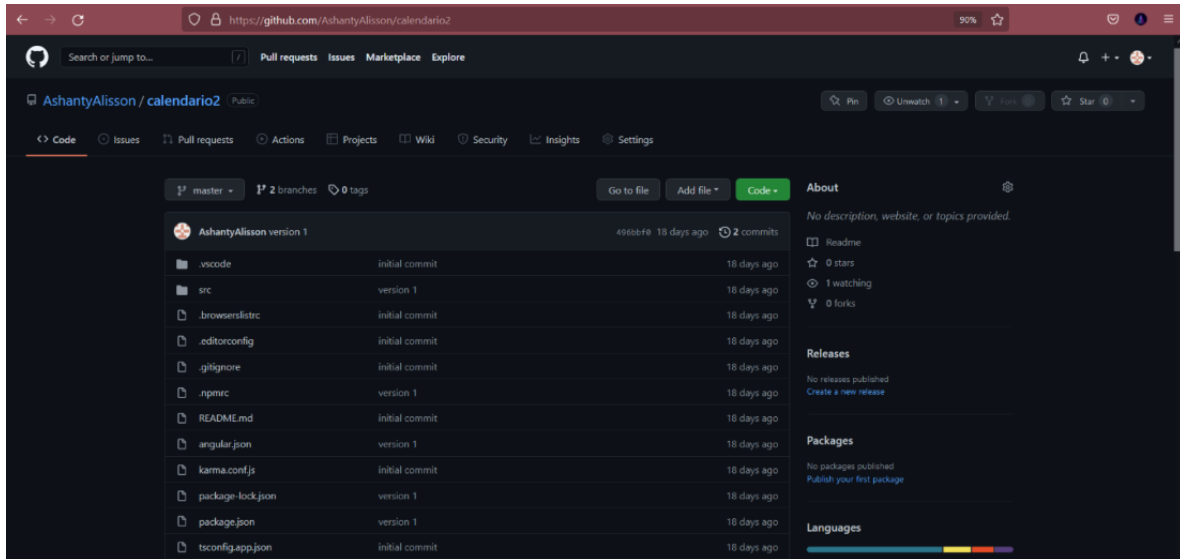
```
1 const Empleado = require('./Empleado')
2 const Server = require('./server')
3
4 module.exports = {
5   Server,
6   Empleado,
7 }
8
```

The screenshot shows the Visual Studio Code interface with the Explorer sidebar on the left. The Explorer sidebar is expanded to show the '.env' file, which is selected and its content is displayed in the main editor area. The content of the .env file is as follows:

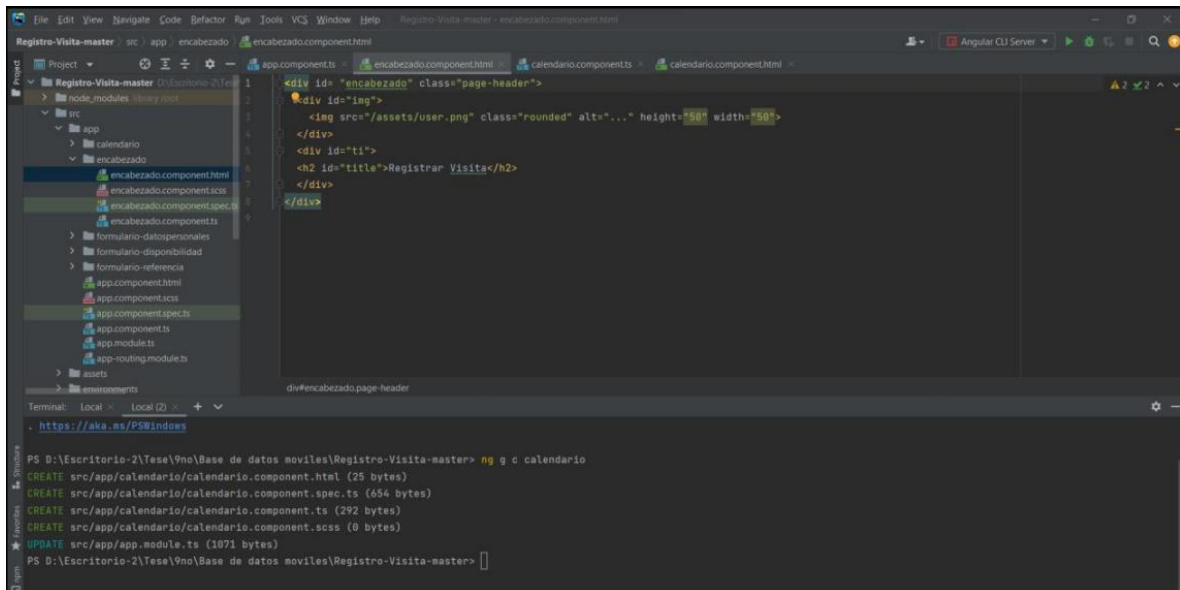
```
1 PORT=5000
2 MONGODB_ATLAS=mongodb+srv://Alisson:2Arc3098@citass.4y6sj.mongodb.net/citas
```

# Formulario

## GitHub

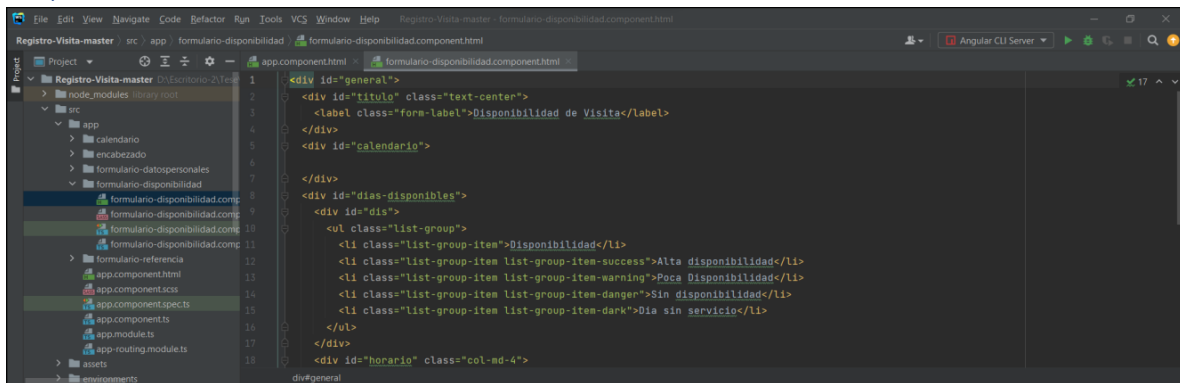


## Encabezado



```
<div id="encabezado" class="page-header">
  <div id="img">
    
  </div>
  <div id="ti">
    <h2 id="title">Registrar Visita</h2>
  </div>
</div>
```

## Disponibilidad



```
<div id="general">
  <div id="titulo" class="text-center">
    <label class="form-label">Disponibilidad de Visita</label>
  </div>
  <div id="calendario">

</div>
<div id="dias-disponibles">
  <div id="dis">
    <ul class="list-group">
      <li class="list-group-item">Disponibilidad</li>
      <li class="list-group-item list-group-item-success">Alta disponibilidad</li>
      <li class="list-group-item list-group-item-warning">Poca Disponibilidad</li>
      <li class="list-group-item list-group-item-danger">Sin disponibilidad</li>
      <li class="list-group-item list-group-item-dark">Dia sin servicio</li>
    </ul>
  </div>
  <div id="horario" class="col-md-4">
    <label for="inputState" class="form-label">Horario disponible</label>
    <select id="inputState" class="form-select">
      <option selected>Choose...</option>
      <option>...</option>
    </select>
  </div>
</div>
```

The screenshot shows the Visual Studio Code interface. On the left, the File Explorer displays the project structure, with 'formulario-referencia.component.html' highlighted under the 'src/app' directory. The main editor window shows the HTML code for this component, which includes a header section with a title and two input fields for 'Sucursal' and 'Departamento'. The code uses Angular's `form` directive and `validation` classes for form handling.

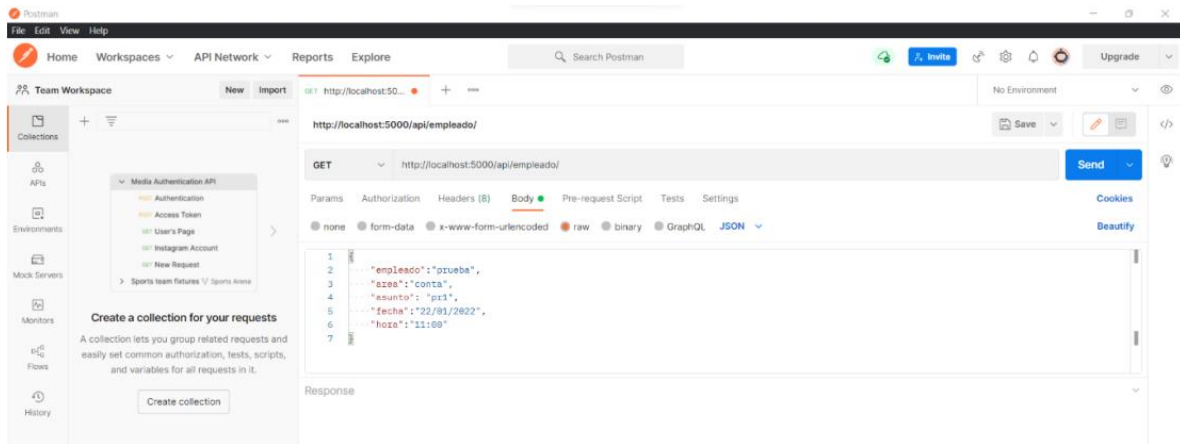
```
<!-- Header -->
<div id="num2" class="text-center">
  <div class="row g-3">
    <div id="titulo">
      <label class="form-label">Datos de Referencia/</label>
    </div>
    <div id="sucursal" class="col-md-3">
      <label for="validationSucursal" class="form-label">Sucursal/</label>
      <select style="..." class="form-select" id="validationSucursal" required>
        <option selected disabled value="">Choose...</option>
      </select>
    </div>
    <div id="departamento" class="col-md-3">
      <label for="validationDepartamento" class="form-label">Departamento que visita/</label>
      <select style="..." class="form-select" id="validationDepartamento" required>
        <option selected disabled value="">Choose...</option>
      </select>
    </div>
  </div>
</div>
```

```
<div id="num2" class="text-center">
<form class="row g-3">
  <div id="titutlo">
    <label class="form-label">Datos de Referencia</label>
  </div>
  <div id="sucursal" class="col-md-3">
    <label for="validationSucursal" class="form-label">Sucursal</label>
    <select style="width: 400px" class="form-select" id="validationSucursal" required>
      <option selected disabled value="">Choose...</option>
      <option>...</option>
    </select>
  </div>
  <div id="departamento" class="col-md-3">
    <label for="validationDepartamento" class="form-label">Departamento que visita</label>
    <select style="width: 400px" class="form-select" id="validationDepartamento" required>
      <option selected disabled value="">Choose...</option>
      <option>...</option>
    </select>
  </div>
  <div id="personal" class="col-md-3">
    <label for="validationPersonal" class="form-label">Personal</label>
    <select style="width: 400px;" class="form-select" id="validationPersonal" required>
      <option selected disabled value="">Choose...</option>
      <option>...</option>
    </select>
  </div>
  <div id="asunto" class="col-md-3">
    <label for="validationAsunto" class="form-label">Asunto</label>
    <select class="form-select" id="validationAsunto" required>
      <option selected disabled value="">Choose...</option>
      <option>...</option>
    </select>
  </div>
</form>
</div>
```



```
    </select>
  </div>
  <div id="des" class="col-mb-3">
    <label for="exampleFormControlTextarea1" class="form-label">Descripcion de su
visita</label>
    <textarea class="form-control" id="exampleFormControlTextarea1" rows="3"></textarea>
  </div>
</form>
</div>
```

## Postman



## MongoDB

