



**Tecnológico de Estudios Superiores De
Ecatepec**

**Division de Ingeniería en Sistemas
Computacionales**

**TERCER PARCIAL
“AVANCE PROYECTO”**

**BASES DE DATOS PARA DISPOSITIVOS
MOVILES**

GRUPO: 5801

Profesora: Griselda Cortes Barrera

Modulo Registro

Integrantes del equipo:

- + Arguelles Castillo Misael Alejandro**
- + Badillo Zamora Cynthia Jacqueline**
- + Cruz Godínez Miguel Ángel**
- + Hurtado Guerra Eduardo**

INDICE

| | |
|--|-----------|
| INTRODUCCION | 3 |
| MODULO REGISTRO DE EMPLEADOS | 4 |
| Funcionamiento | 5 |
| VISION DE ENVIO DE CORREO ELECTRONICO | 6 |
| VISTAS DE WEBSTROM..... | 7 |
| VISTAS EN MONGO DB | 10 |
| POSTMAN | 11 |

INTRODUCCION

En este ultimo avance de proyecto, concluimos el diseño de nuestro frontEnd haciendo una vista para que los usuarios puedan acceder fácilmente. También conectamos la base datos a la web para así poder tener una consulta de los usuarios que están en nuestra base. En donde se visualizan los datos de nuestros usuarios.

En el siguiente trabajo vamos a checar más a detalle lo que es la creación de un servidor en MONGO DB, el cual es bastante interesante, ya que es ideal para entornos con pocos recursos de computación, cualquier servidor o cualquier ordenador personal sirve para montar MongoDB y tener un servidor para tus proyectos. Es una herramienta con un coste bajo, al ser una herramienta de código abierto no se paga licencia, lo único que se paga es por el soporte, en caso de necesitarlo. Tiene una gran documentación, posee una documentación muy buena, muy amplia y detallada en comparación con otras bases de datos NoSQL. Es un complemento perfecto para JavaScript, si eres desarrollador de aplicaciones utilizando este lenguaje, podrás utilizar toda la potencia de sus funciones y operadores en MongoDB.

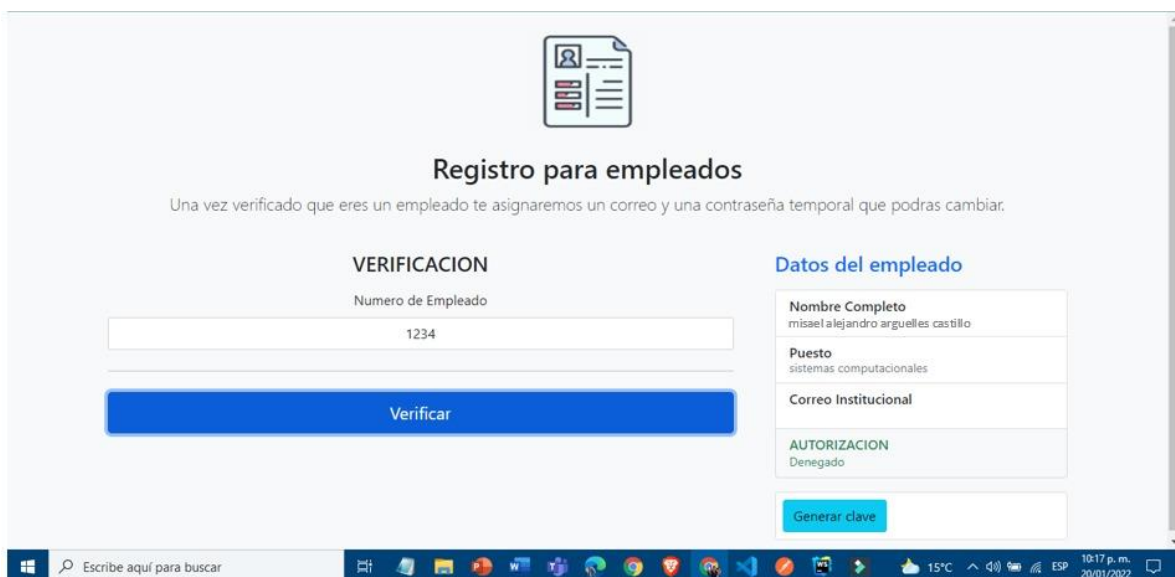
MODULO REGISTRO DE EMPLEADOS

Nuestro modulo tiene varios elementos en su interfaz

Tanto como la verificación con el numero de empleado, sus datos del empleado:

- Nombre completo
- Puesto
- Correo personal
- Y un botón para generar su correo

En esta imagen se muestra que el usuario si se encuentra en la base de datos, pero no cuenta con un correo institucional entonces lo que debe de hacer es solicitar si correo y contraseña mediante el botón de generar clave, estos datos serán enviados a su correo personal previamente registrado en la base de datos.



| Registro para empleados | |
|--|--|
| Una vez verificado que eres un empleado te asignaremos un correo y una contraseña temporal que podras cambiar. | |
| VERIFICACION | Datos del empleado |
| Numero de Empleado | Nombre Completo |
| <input type="text" value="1234"/> | misael alejandro arguelles castillo |
| <input type="button" value="Verificar"/> | Puesto |
| | sistemas computacionales |
| | Correo Institucional |
| | |
| | AUTORIZACION |
| | Denegado |
| | <input type="button" value="Generar clave"/> |

Funcionamiento

En esta parte se podrá verificar si el usuario se encuentra en la base de datos ingresando con su número de empleado, mostrándole del lado inferior izquierdo si esta autorizado para generarle una clave.

The screenshot shows a web application titled "Registro para empleados". At the top, there is a sub-header "VERIFICACION" and a text prompt: "Una vez verificado que eres un empleado te asignaremos un correo y una contraseña temporal que podras cambiar." Below this, there is a form with a label "Numero de Empleado" and a text input field containing "1234567". A blue button labeled "Verificar" is positioned below the input field. To the right of the form, under the heading "Datos del empleado", there is a table displaying employee information:

| | |
|----------------------|----------------------------------|
| Nombre Completo | badillo zamora cynthia jaqueline |
| Puesto | ing.sistemas |
| Correo Institucional | cynthiabz@gmail.com |
| AUTORIZACION | Autorizado |

Below the table is a blue button labeled "Generar clave". The Windows taskbar at the bottom shows the search bar with the text "Escribe aquí para buscar", several application icons, and system information including "14°C", "ESP", and "10:29 p. m. 20/01/2022".

En dado caso de que su consulta no se encuentre en la base de datos saldrá como denegado.

This screenshot shows the same web application as the previous one, but with a different employee number. The "Numero de Empleado" input field now contains "1234". The "Verificar" button remains blue. In the "Datos del empleado" table, the "AUTORIZACION" status is now "Denegado". The "Generar clave" button is still present. The Windows taskbar at the bottom shows the search bar with "Escribe aquí para buscar", application icons, and system information including "15°C", "ESP", and "10:17 p. m. 20/01/2022".

Si la ID del usuario es errónea, le aparecerá una leyenda del lado superior indicando que no esta en la base de datos.

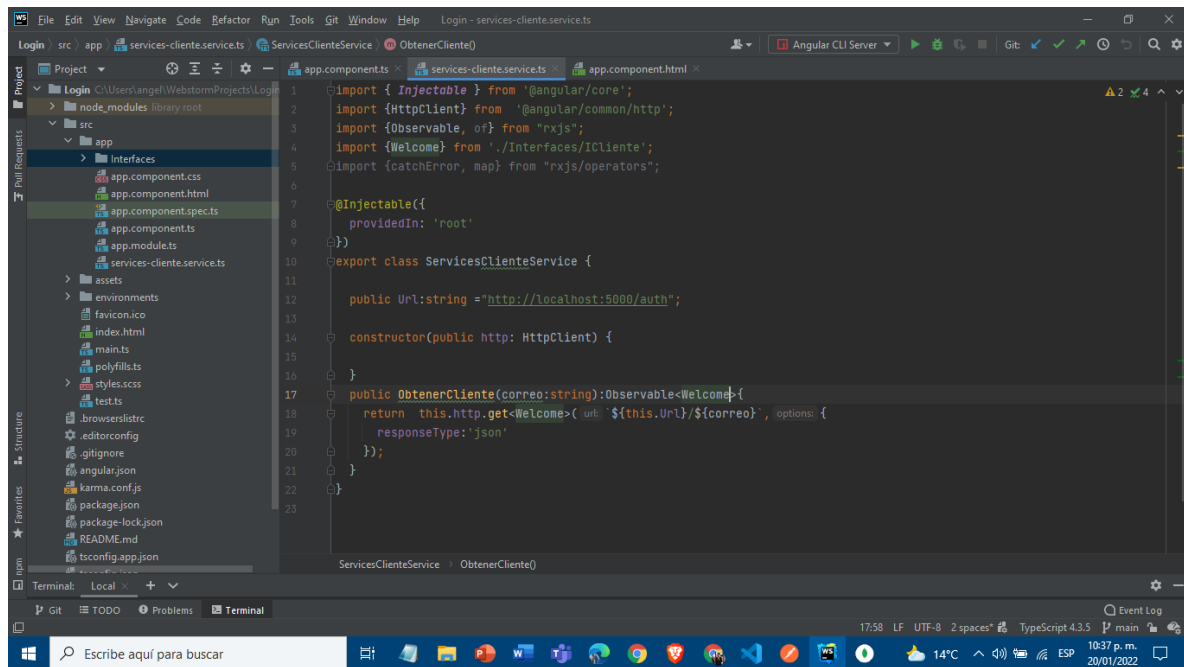
The screenshot shows a web browser window with a URL bar indicating 'localhost:4200'. A modal dialog box is displayed at the top center, stating 'localhost:4200 dice Usuario No Existe' with an 'Aceptar' button. Below the dialog, the main content area is titled 'Registro para empleados' with a subtitle: 'Una vez verificado que eres un empleado te asignaremos un correo y una contraseña temporal que podras cambiar.' The form is divided into two sections: 'VERIFICACION' on the left and 'Datos del empleado' on the right. The 'VERIFICACION' section has a label 'Numero de Empleado' above a text input field containing '123456' and a blue 'Verificar' button. The 'Datos del empleado' section contains a form with fields for 'Nombre Completo', 'Puesto', and 'Correo Institucional', followed by an 'AUTORIZACION' section with a 'Generar clave' button.

VISION DE ENVIO DE CORREO ELECTRONICO

Aquí se muestra un ejemplo de la llegada del correo institucional asignado al usuario junto con su contraseña proporcional que podrá ser cambiada.

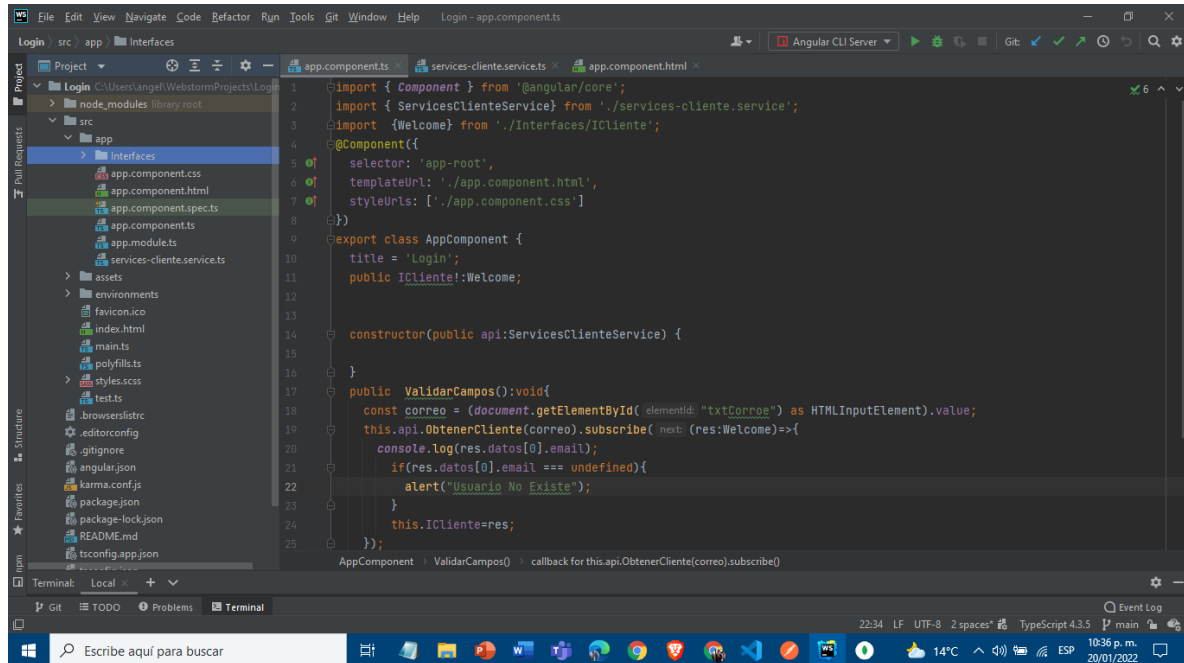
The screenshot shows an email interface with the title 'Correo Institucional' and a 'Recibidos x' button. The email content is addressed 'para mí' and includes the following text: 'Saludos, Misael Alejandro Arguelles Castillo', 'Tu Correo Institucional es: misael4512@ejemplo.edu.mx', and 'Tu contraseña es: Misael98_@'. At the bottom, there are two buttons: 'Responder' with a left-pointing arrow and 'Reenviar' with a right-pointing arrow.

VISTAS DE WEBSTROM



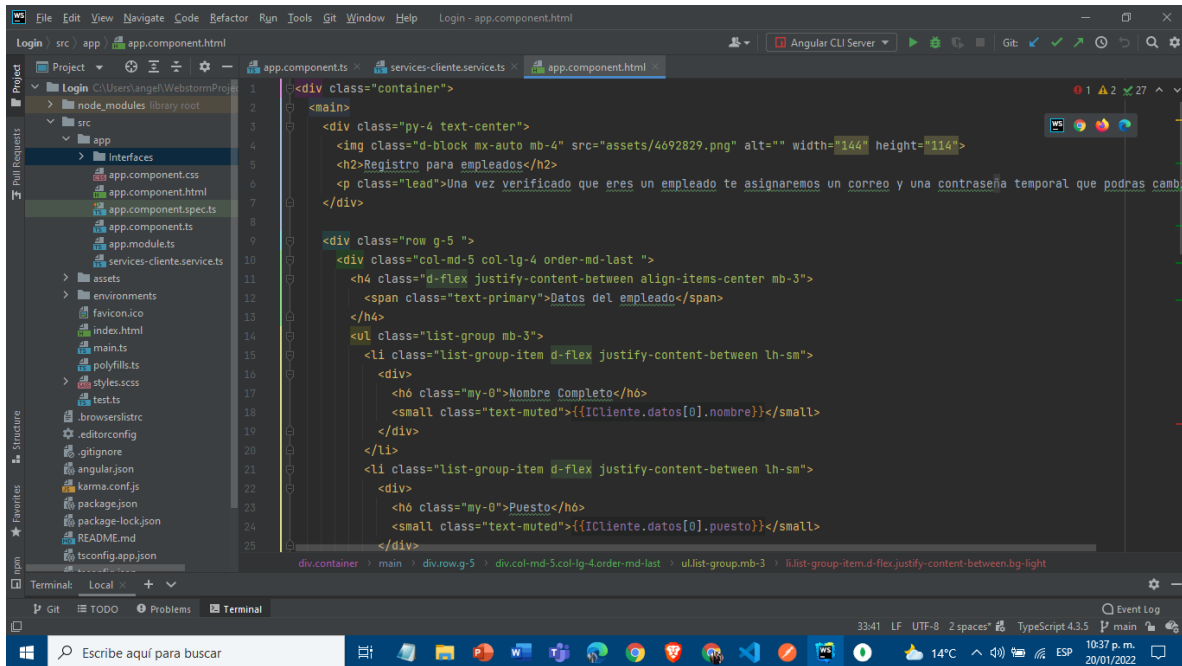
This screenshot shows the Visual Studio Code editor with the file `ServicesClienteService.ts` open. The file is part of the `src/app` directory. The code defines an Angular service named `ServicesClienteService` that implements the `ICliente` interface. It uses `HttpClient` to make a GET request to `http://localhost:5000/auth` with the email as a query parameter. The service is decorated with `@Injectable` and has a `providedIn: 'root'` configuration.

```
1 import { Injectable } from '@angular/core';
2 import { HttpClient } from '@angular/common/http';
3 import { Observable, of } from 'rxjs';
4 import { Welcome } from '../Interfaces/ICliente';
5 import { catchError, map } from 'rxjs/operators';
6
7 @Injectable({
8   providedIn: 'root'
9 })
10 export class ServicesClienteService {
11
12   public Url:string = "http://localhost:5000/auth";
13
14   constructor(public http: HttpClient) {
15
16   }
17
18   public ObtenerCliente(correo:string):Observable<Welcome>{
19     return this.http.get<Welcome>({ url: `${this.Url}/${correo}`, options: {
20       responseType: 'json'
21     }});
22   }
23 }
```

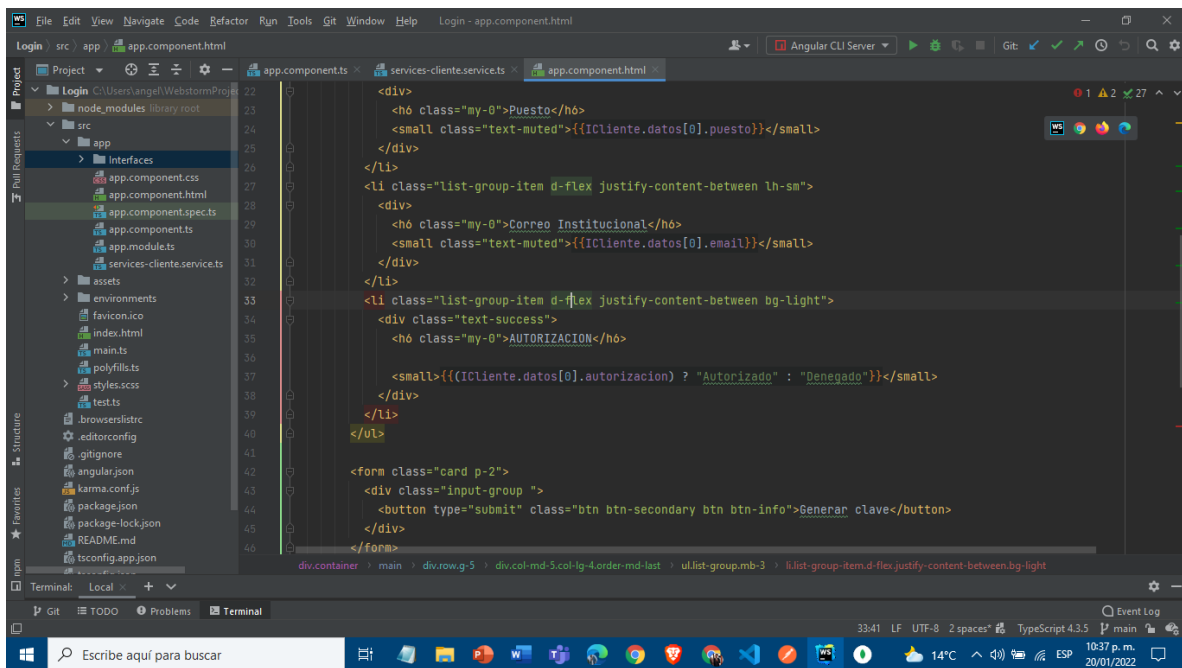


This screenshot shows the Visual Studio Code editor with the file `app.component.ts` open. The file is part of the `src/app` directory. The code defines an Angular component named `AppComponent` that implements the `ICliente` interface. It uses the `ServicesClienteService` to make a GET request to `http://localhost:5000/auth` with the email as a query parameter. The component has a `selector: 'app-root'` and a `templateUrl: './app.component.html'`. It also has a `styleUrls: ['./app.component.css']`. The component is decorated with `@Component` and has a `providedIn: 'root'` configuration.

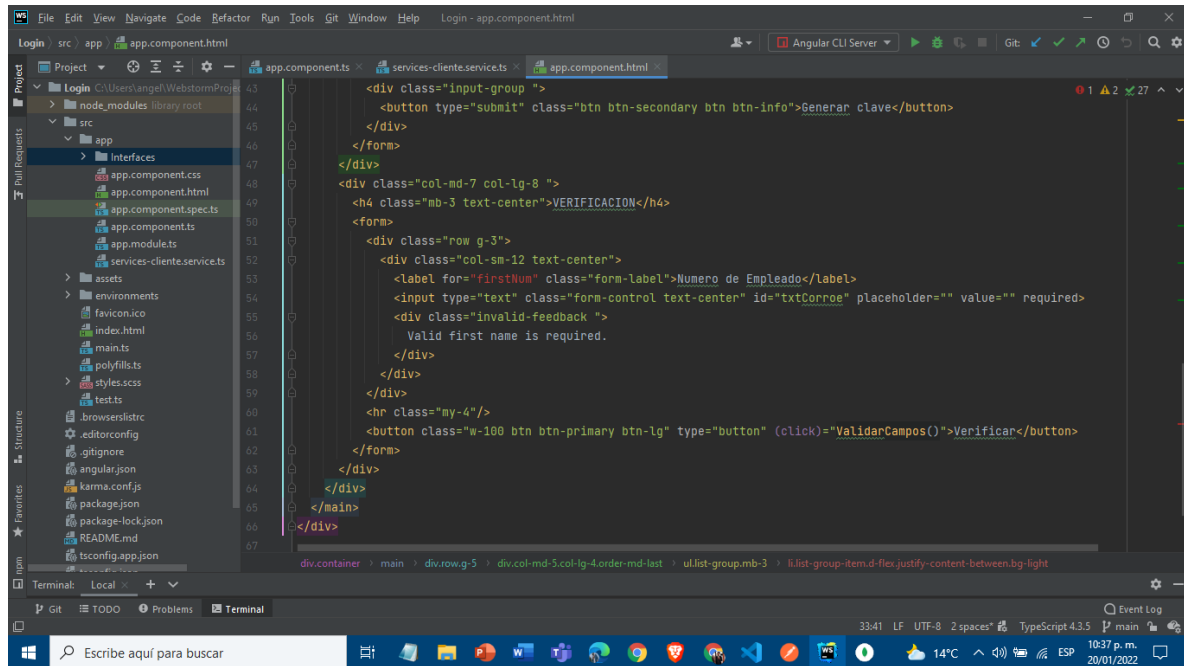
```
1 import { Component } from '@angular/core';
2 import { ServicesClienteService } from './services-cliente.service';
3 import { Welcome } from '../Interfaces/ICliente';
4
5 @Component({
6   selector: 'app-root',
7   templateUrl: './app.component.html',
8   styleUrls: ['./app.component.css']
9 })
10 export class AppComponent {
11   title = 'Login';
12   public ICliente!:Welcome;
13
14   constructor(public api:ServicesClienteService) {
15
16   }
17
18   public ValidarCampos():void{
19     const correo = (document.getElementById( "txtCorreo" ) as HTMLInputElement).value;
20     this.api.ObtenerCliente(correo).subscribe( next: (res:Welcome)=>{
21       console.log(res.datos[0].email);
22       if(res.datos[0].email === undefined){
23         alert("Usuario No Existe");
24       }
25       this.ICliente=res;
26     });
27   }
28 }
```



```
1 <div class="container">
2   <main>
3     <div class="py-4 text-center">
4       
5       <h2>Registro para empleados</h2>
6       <p class="lead">Una vez verificado que eres un empleado te asignaremos un correo y una contraseña temporal que podras cambi
7     </div>
8
9     <div class="row g-5">
10      <div class="col-md-5 col-lg-4 order-md-last">
11        <h4 class="d-flex justify-content-between align-items-center mb-3">
12          <span class="text-primary">Datos del empleado</span>
13        </h4>
14        <ul class="list-group mb-3">
15          <li class="list-group-item d-flex justify-content-between lh-sm">
16            <div>
17              <h6 class="my-0">Nombre Completo</h6>
18              <small class="text-muted">{{ICliente.datos[0].nombre}}</small>
19            </div>
20          </li>
21          <li class="list-group-item d-flex justify-content-between lh-sm">
22            <div>
23              <h6 class="my-0">Puesto</h6>
24              <small class="text-muted">{{ICliente.datos[0].puesto}}</small>
25            </div>
26          </li>
27        </ul>
28      </div>
29    </div>
30  </main>
31</div>
```



```
22 <div>
23   <h6 class="my-0">Puesto</h6>
24   <small class="text-muted">{{ICliente.datos[0].puesto}}</small>
25 </div>
26 </li>
27 <li class="list-group-item d-flex justify-content-between lh-sm">
28   <div>
29     <h6 class="my-0">Correo Institucional</h6>
30     <small class="text-muted">{{ICliente.datos[0].email}}</small>
31   </div>
32 </li>
33 <li class="list-group-item d-flex justify-content-between bg-light">
34   <div class="text-success">
35     <h6 class="my-0">AUTORIZACION</h6>
36
37     <small>{{(ICliente.datos[0].autorizacion) ? "Autorizado" : "Denegado"}}</small>
38   </div>
39 </li>
40 </ul>
41
42 <form class="card p-2">
43   <div class="input-group">
44     <button type="submit" class="btn btn-secondary btn btn-info">Generar clave</button>
45   </div>
46 </form>
```

VISTAS EN MONGO DB

The screenshot shows the MongoDB Compass interface. On the left, the 'Local' sidebar displays the database structure: '5 DBS' and '13 COLLECTIONS'. The 'HOST' is 'localhost:27017', the 'CLUSTER' is 'Standalone', and the 'EDITION' is 'MongoDB 4.4.1 Community'. The 'Filter your data' section shows a tree view with 'Direcciones', 'RegistroEmpleados' (expanded), 'CorreosNuevos', 'Empleados', 'Empleados-Verificar', 'Informacion', 'Verificar', 'users' (selected), and 'admin'. The main panel shows the 'RegistroEmpleados.users' collection with 4 documents. The 'Documents' tab is active, showing a list of documents with their keys and values. The first document is:

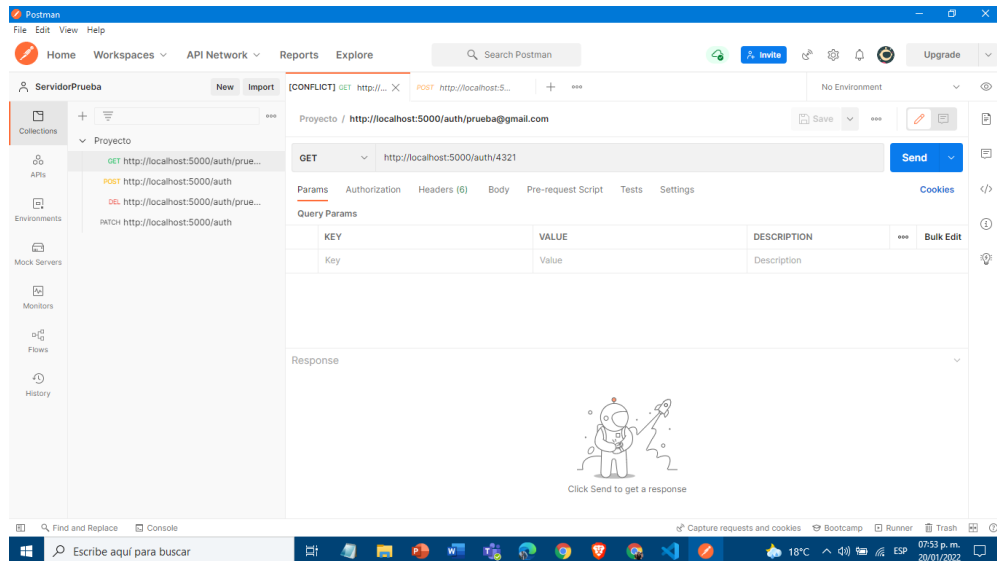
```
{ "_id": ObjectId("61e7abe912778a5bc849e677"), "nombre": "jose", "puesto": "sistemas computacionales", "email": "prueba@gmail.com", "password": "$2b$10$3XSHL7jeDvZ2hsunhcxtHTG6yPvrvzy1blsppqA39eqH138fpw", "autorizacion": false, "NEmpleado": "1234", "_v": 0 }
```

The screenshot shows the MongoDB Compass interface, similar to the first one. The 'Filter your data' section shows the same tree view. The main panel shows the 'RegistroEmpleados.users' collection with 4 documents. The 'Documents' tab is active, showing a list of documents with their keys and values. The second document is:

```
{ "_id": ObjectId("61ea34ee3f06490be54c93fe"), "nombre": "hurtado guerra eduardo", "puesto": "ing.sistemas", "email": "eduardo@gmail.com", "password": "$2b$10$8bT5W482TFf19U9Cs2WMSJe2lCQhR26AVUJ79s1L1u08ICXKh.2.Gy", "autorizacion": true, "NEmpleado": "5657", "_v": 0 }
```

POSTMAN

- Obtención de datos de MongoDB desde el api de Postman
- Conectividad apiRest
- Prueba del servidor con datos del usuario
- Testeo del servicio



- Visualización de los datos almacenados desde la web

