



Universidad Tecnológica de la Mixteca.
Ingeniería en Computación.
Programación Web II.

Reporte sobre cómo cambiar la contraseña.

Medina Navarrete Rheveri Antonio.
Sexto Semestre.
Matricula: 2019020365.

14 de Marzo del 2022.

Problemática

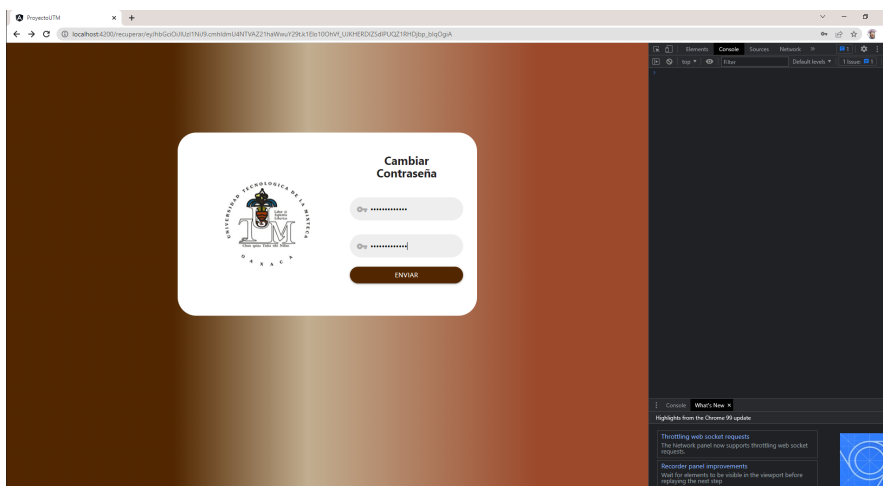
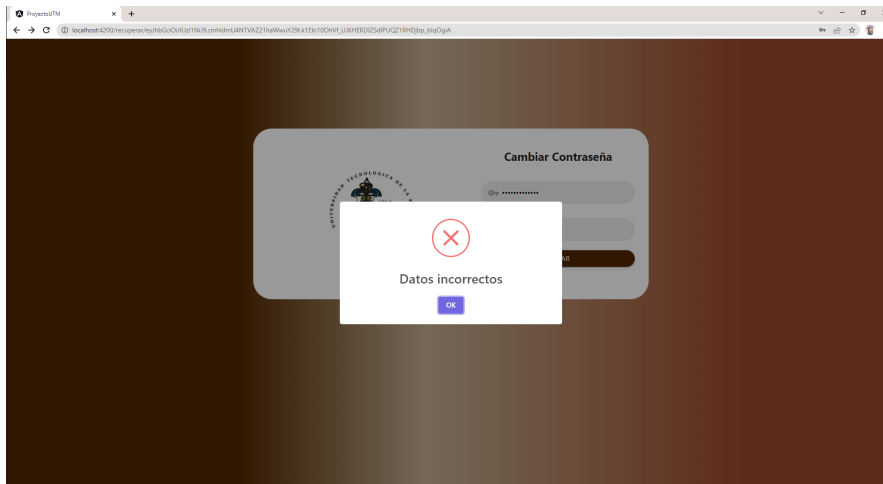
En este trabajo se encuentran dos problemáticas, la primera es crear el componente para cambiar la contraseña de un profesor la cual debe contar con un documento html con sus debidos campos y en el archivo .ts con sus debidas funciones y trabajando todo correctamente. Y la segunda problemática es que al loguearse se debe poder lograr con cualquier tipo de contraseña sin importar qué símbolos hayamos usado para la contraseña.

Desarrollo

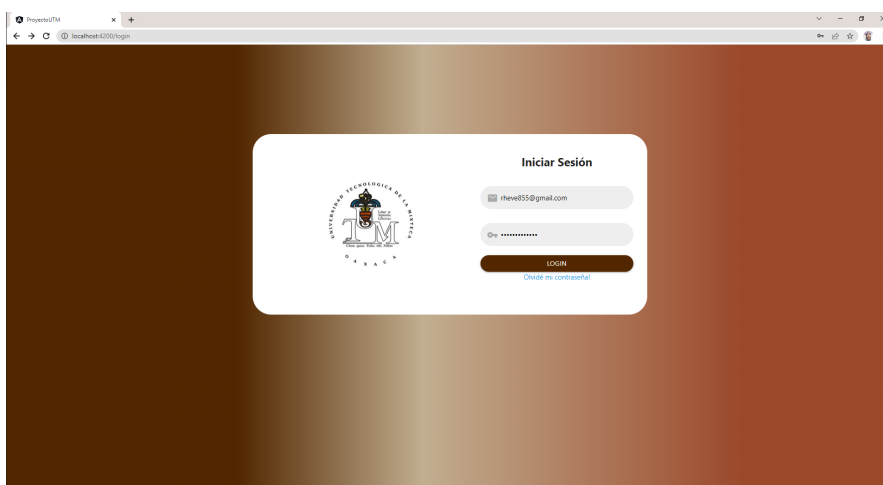
Para la primera problemática ya tenemos un complemento llamado “recuperar”, en el .ts ya tenemos en la función “ngOnInit()” un método que destokeniza el correo y recupera toda la información del profesor, ahora con esta información se puede cambiar la contraseña por lo que creí una función en el al recibir dos contraseñas y si son iguales se debe realizar el cambio de contraseña si no son iguales entonces debe salir un error. Si las contraseñas son iguales entonces debe hacer el cambio por lo que llamé a un servicio el cual recibe el correo y la contraseña a través de un modelo llamado “Usuario” que tiene dos variables, el correo y la contraseña. este servicio a la vez llama a una dirección que está un controlador que cree llamado “cambiarContraseña” en la que con el correo busca el Profesor y a la contraseña la cifra para que al actualizar la cuenta del Profesor y se cambie la contraseña, la contraseña ya esté cifrada. Cuando la contraseña son iguales y se logró cambiar correctamente la página te redirecciona a la página de logueo.

La segunda problemática es el que al iniciar sesión se pueda con cualquier símbolo en la contraseña por lo que solo se debe cambiar el controlador en lugar de un “get” que sea un “post” con esto podrá recibir un archivo “.json”, el archivo contendrá el correo y la contraseña que se llenarán cuando el usuario inicie sesión, con esto ya no será necesario que en la ruta para usar este controlador se tenga que escribir ni la contraseña ni el correo por lo que con esto la contraseña puede tener cualquier símbolo.

Pruebas



Contraseña por la cual se cambió:
!"#\$%&/()=,.-

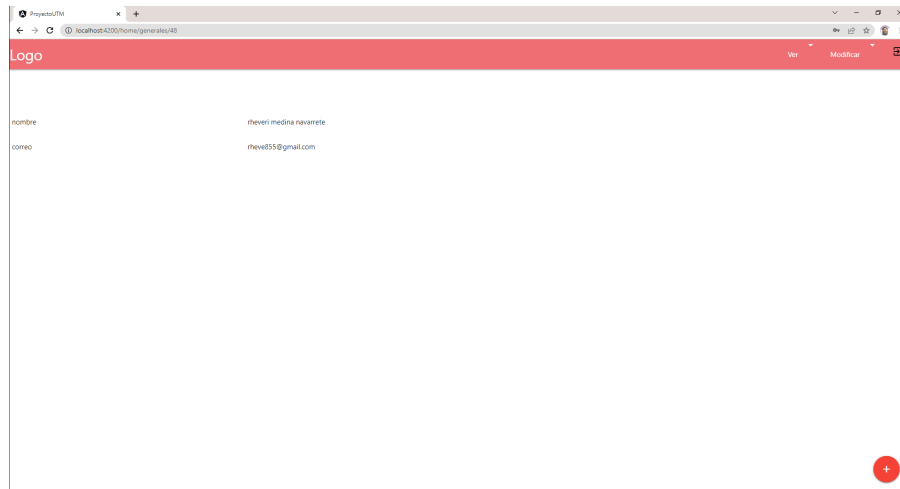


Iniciando seccion con el correo:

rheve855@gmail.com

Y la contraseña:

!"#\$%&/()=,.-



Código

HTML del componente “recuperar”.

```
<div class="bodyLogin">
  <div class="container">
    <div class="row">
      <div class="col s12 m12 l8 offset-l2 loginDiv">
        <div class="row loginContainer">
          <div class="col s12 m6 l6">
            <div class="col s11 offset-s1">
              
            </div>
          </div>
          <div class="col s12 m5 l5 offset-l1 offset-m1">
            <div class="loginTitulo">
              <h5>Cambiar Contraseña</h5>
            </div>
            <div class="row">
              <div class="col s12 customInput">
                <i class="material-icons prefix">vpn_key</i>
                <input [(ngModel)]="contrasena.password"
class="browser-default" type="password" name="password" id="password" placeholder="Nueva
Contraseña" autocomplete="off">
              </div>
              <div class="col s12 customInput">
                <i class="material-icons prefix">vpn_key</i>
                <input [(ngModel)]="contrasena.passwordSecond"
class="browser-default" type="password" name="password" id="password2"
placeholder="Comprobar Contraseña" autocomplete="off">
              </div>
            </div>
          </div>
        </div>
      </div>
    </div>
  </div>
</div>
```

```

        <div class="row">
            <button (click)="cambiar()" class="btn col 112 s12 m12 btnLogin
">
                ENVIAR
            </button>
        </div>
    </div>

</div>

</div>
</div>
</div>
</div>

```

ts del componente “recuperar”.

```

import { Component, OnInit } from '@angular/core';
import { ActivatedRoute } from '@angular/router';
import { CorreoService } from 'src/app/services/correo.service';
import { Router } from '@angular/router';
import Swal from 'sweetalert2';
import { Contraseña } from '../models/contrasena.model';
import { Usuario } from '../models/usuario.model';

@Component({
  selector: 'app-recuperar',
  templateUrl: './recuperar.component.html',
  styleUrls: ['./recuperar.component.css']
})
export class RecuperarComponent implements OnInit {
  token : any;
  contraseña : Contraseña;
  correo : any;
  datosActualizar : Usuario;
  constructor(private route:ActivatedRoute, private correoService:CorreoService,private
router: Router) {
    this.token='';
    this.contraseña = new Contraseña();
    this.datosActualizar = new Usuario();
  }

  ngOnInit(): void {
    this.route.paramMap.subscribe(params =>
    {
      this.token = params.get('token');
      let dato={
        'token':this.token
      }
      this.correoService.decodificarMail(dato).subscribe((resCorreo: any) =>
      {
        console.log('Init',resCorreo);
        this.correo =resCorreo.correoProfesor;

```

```

        },err => console.error(err));
    });
}
cambiar(){
    if (this.contrasena.password == this.contrasena.passwordSecond) {
        this.datosActualizar.correo = this.correo;
        this.datosActualizar.password = this.contrasena.password;
        console.log(this.datosActualizar);
        this.correoService.cambiarContrasenya(this.datosActualizar).subscribe((resCorreo:
any) =>
        {

            },err => console.error(err));
            this.router.navigateByUrl('');
        }
    }
    else{
        Swal.fire({
            position: "center",
            icon: "error",
            title: `Datos incorrectos`,
        });
    }
}
}
}

```

En el servicio “correo” se agregaron.

```

    decodificarMail(token: any) {
        return
this.http.post(`${environment.API_URI_CORREOS}/decodificarMail/`,
token);
    }
    cambiarContrasenya(datos: any) {
        return
this.http.post(`${environment.API_URI}/Profesores/cambiarContrasenya`,
datos);
    }
}

```

El primero es para decodificar el correo y el segundo para cambiar la contraseña.

Cambios que se hicieron en las rutas del profesor, el existe se cambió el “get” por “post” y se modificó la ruta. El segundo es para la ruta para el controlador que va a cambiar la contraseña del Profesor.

```

        this.router.post('/existe', profesoresController.existe);
        this.router.post('/cambiarContrasenya',
profesoresController.cambiarContrasenya);

```

En el controlador de “existe” solo se cambió la forma de asignar de “correoProfesor” y del “password”.

```
let correoProfesor = req.body.correo;
let password = req.body.password;
```

Y por último el controlador de “cambiarContraseña”.

```
public async cambiarContraseña(req: Request, res: Response): Promise<void> {
  let password = req.body.password as any;
  var salt = bcrypt.genSaltSync(10);
  const correoProfesor = req.body.correo as any;
  console.log(req.params);
  bcrypt.hash(password, salt).then(function (nuevoPassword) {
    req.body.password = nuevoPassword;
    console.log(nuevoPassword, ' ', correoProfesor);
    const resp = pool.query("UPDATE profesores set profesores.password = ? WHERE
correoProfesor = ?", [nuevoPassword, correoProfesor]);
    res.json(resp);
  })
}
```