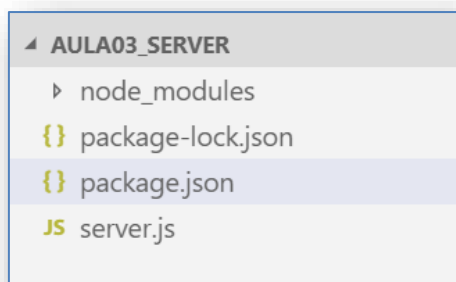


PROJETO SERVER NODE:

Estrutura do projeto depois de finalizado:

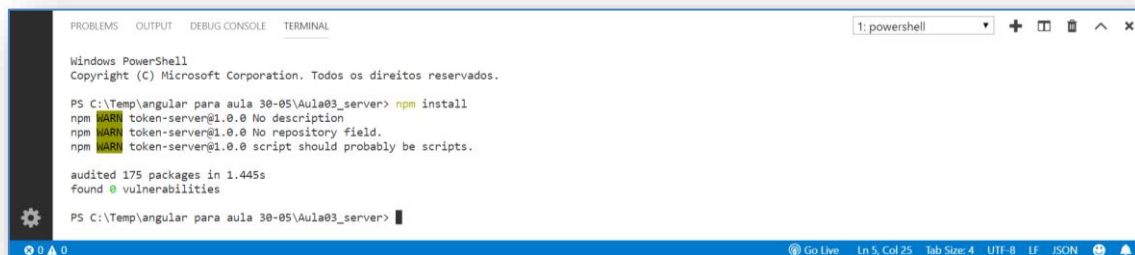


package.json

```
{
  "name": "token-server",
  "version": "1.0.0",
  "description": "",
  "main": "server.js",
  "scripts": {
    "start": "node server.js",
    "test": "echo \\\"Error: no test specified\\\" && exit 1"
  },
  "author": "voces",
  "license": "ISC",
  "dependencies": {
    "body-parser": "^1.18.3",
    "express": "^4.16.4",
    "jsonwebtoken": "^8.4.0"
  }
}
```

Instalar as dependências do package

Digitar no terminal:
"npm install"



```
Windows PowerShell
Copyright (C) Microsoft Corporation. Todos os direitos reservados.

PS C:\Temp\angular para aula 30-05\Aula03_server> npm install
npm WARN token-server@1.0.0 No description
npm WARN token-server@1.0.0 No repository field.
npm WARN token-server@1.0.0 script should probably be scripts.

audited 175 packages in 1.445s
found 0 vulnerabilities

PS C:\Temp\angular para aula 30-05\Aula03_server>
```

server.js

```
const express = require("express");
const body = require("body-parser");
const app = express();
const port = 2233;
const secret = "minhachave";
const jwt = require("jsonwebtoken");
app.use(body.json());

//node server.js
//ou
//nodemon server.js

app.use((req, res, next) => {
  res.header("Access-Control-Allow-Origin", "*");
  res.header("Access-Control-Allow-Methods",
    "POST,GET,DELETE,PUT,OPTIONS");
  res.header("Access-Control-Allow-Headers", "Origin, X-
    Requested-With, Content-Type, Accept, authorization");
  res.header("Cache-Control", "no-cache");
  res.header("Pragma", "no-cache");
  next();
});

app.post('/criptografia', (req, res) => {
```

```
let login = req.body.login;
let senha = req.body.senha;
if ((login == 'lu' && senha == '123') ||
    (login == 'feio' && senha == '123')) {
  const crypto = require("crypto");
  resp2 = crypto.createHash("md5").
    update(senha).
    digest("hex");
  res.send({ senha: resp2 });
} else {
  res.send("error de validacao")
  return;
}
});

app.post("/generate", (req, res) => {
  let login = req.body.login;
  let senha = req.body.senha;
  let resp = null;
  if ((login == 'feio' && senha == '123') ||
      (login == 'lu' && senha == '123')) {
    resp = jwt.sign({ login: login, senha: senha }, secret);
  } else {
    resp = 'falha';
  }
  res.send(resp);
});

app.post("/check", (req, res) => {
  let token = req.body.token; //resgatando do form
  jwt.verify(token, secret, (err, dados) => {
    if (err) {
      res.status(403).send("error");
      return;
    } else {
      res.send(dados);
    }
  });
});
});
```



Angular6 Restful – Noite Quinta e Sexta

Prof. Edson Belém

Quinta-feira, 30 de maio de 2019.

Projeto Node JWT, Criptografia, Gerar Token, Checar Token, Rotas, Instalação Bootstrap, JQuery, Fontawesome



Aula

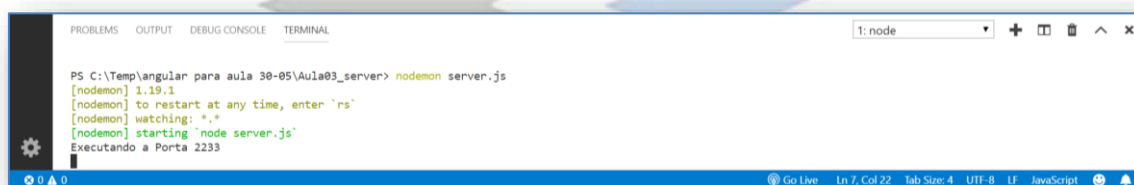
03

```
app.listen(port, () => console.log(`Executando a Porta  
${port}`));
```

Iniciar o servidor.

Digitar no terminal:

“node server.js” ou “nodemon.js”



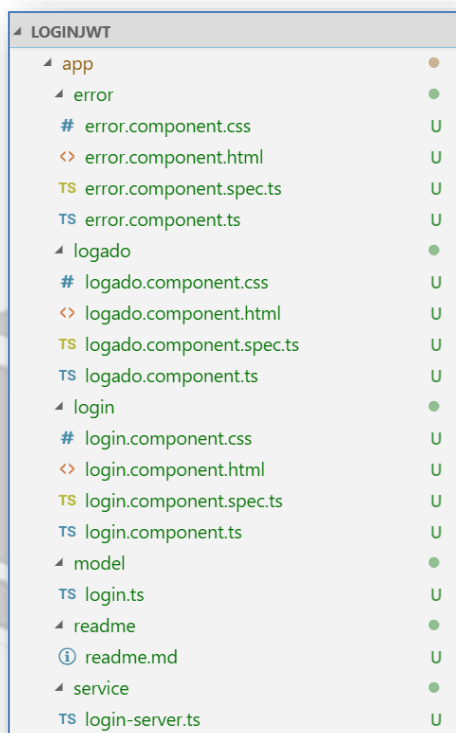
COTI
Informática

Escola de Nerds

www.cotiinformatica.com.br

NOVO PROJETO CLIENT:

Estrutura do projeto:



login.ts

```
export class Login{
  login : string;
  senha : string

  constructor(login ?:string, senha ?:string){
    this.login = login;
    this.senha= senha;
  }
}
```

login-service.ts

```
import { Injectable } from "@angular/core";
import { HttpClient } from "@angular/common/http";
import { Login } from "../model/login";

const url = 'http://localhost:2233'

@Injectable()
export class LoginService {

    constructor(private http: HttpClient) {
    }

    public criptografar(login: Login) {
        return this.http.post<string>(`${url}/criptografia`,
login, { responseType: 'text' as 'json' });
    }

    public gerarToken(login: Login) {
        return this.http.post<string>(`${url}/generate`, login,
        { responseType: 'text' as 'json' }
        );
    }

    public checarToken(vtoken: string) {
        return this.http.post<string>(`${url}/check`, { token:
vtoken },
        { responseType: 'text' as 'json' }
        );
    }
}
```

login.component.ts

```
import { Component, OnInit } from '@angular/core';
import { Login } from '../model/login';
import { LoginService } from '../service/login-server';
import { Router } from '@angular/router';

@Component({
  selector: 'app-login',
  templateUrl: './login.component.html',
  styleUrls: ['./login.component.css']
})
export class LoginComponent implements OnInit {

  login: Login;
  mytoken: string = '';
  checagem: string;
  checagemError: string;
  senhaCriptografada: string;

  constructor(private service: LoginService,
    private route: Router) {
    this.login = new Login();
  }

  ngOnInit() {

  }

  criptografia() {
    this.service.criptografar(this.login).subscribe(res => {
      this.senhaCriptografada = res;
    })
  }

  gerar() {
    this.service.gerarToken(this.login).subscribe(res => {
      this.mytoken = res;
      console.log(this.mytoken)
    })
  }

  mostrar() {
```

```
    if (this.mytoken !== 'falha') {
      this.service.checarToken(this.mytoken).subscribe(res => {
        this.checagem = res;
        this.checagemError = '';
      });
    } else {
      this.checagemError = 'Nao foi possivel mostrar o Token';
      this.checagem = '';
    }
  }
}
```

Fazer as instalações do Bootstrap, JQuery e Fontawesome

Digitar no terminal:

"npm i bootstrap jquery"

"npm i @fontawesome/fontawesome-free"

Fazer as alterações no styles.css.

Indicar os endereços de onde estão instalados os arquivos de css e javascripts.

```
"styles": [
  "src/styles.css",
  "node_modules/@fontawesome/fontawesome-free/css/all.css",
  "node_modules/bootstrap/dist/css/bootstrap.css"
],
"scripts": [
  "node_modules/bootstrap/dist/js/bootstrap.js",
  "node_modules/jquery/dist/jquery.js"
]
},
```


login.component.html

```
<h2 class="text-center p-3 bg-secondary mb-5">Designer
Maravilhoso 2</h2>

<div class="container-fluid">
  <div class="card col-md-6 offset-md-3 border-secondary
shadow">
    <div class="card-body">

      <label>Login</label>
      <input type="text" name="login"
[(ngModel)]="login.login" class="form-control mb-2"
placeholder="Digite o E-mail" />

      <label>Senha</label>
      <input type="text" name="senha"
[(ngModel)]="login.senha" class="form-control mb-3"
placeholder="Digite a senha" />

      <div class="row">
        <button (click)="criptografia()" class="btn btn-
secondary">
          Criptografar <i class="fa fa-lock ml-
2"></i></button>
        <button (click)="gerar()" class="btn btn-
secondary">Gerar Token
          <i class="fa fa-cog fa-spin"></i>
        </button>
        <button (click)="mostrar()" class="btn btn-
secondary">Mostrar
          <i class="fa fa-eye ml-2"></i>
        </button>
      </div>

      <!-- ALERTAS -->
      <div class="mt-3">

        <div class="alert alert-secondary role="alert">
          <div *ngIf="senhaCriptografada">
```

```

        {{senhaCriptografada}}
    </div>
</div>

<div class="alert alert-primary" role="alert">
    <div *ngIf="mytoken">
        {{mytoken}}
    </div>
</div>

<div class="alert alert-secondary" role="alert">
    <div *ngIf="checagem">
        {{checagem}}
    </div>
</div>

<div class="alert alert-primary" role="alert">
    <div *ngIf="checagemError">
        {{checagemError}}
    </div>
</div>

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

```

```

Login<br />
<input type="text" name="login" [(ngModel)]="login.login" />
<br />
senha<br />
<input type="text" name="senha" [(ngModel)]="login.senha" />
<br />
<button (click)="criptografia()">Criptografar</button>
<br />
<button (click)="gerar()">Gerar Token</button>
<br />
<button (click)="mostrar()">Mostrar</button>
<br />

```

```
<div *ngIf="senhaCriptografada">
  {{senhaCriptografada}}
</div>
<div *ngIf="mytoken">
  {{mytoken}}
</div>
<div *ngIf="checagem">
  {{checagem}}
</div>
<div *ngIf="checagemError">
  {{checagemError}}
</div>
```

login.component.css

```
.btn{
  margin: auto;
}
```

app.module.ts

```
import { BrowserModule } from '@angular/platform-browser';
import { NgModule } from '@angular/core';
import { AppComponent } from './app.component';
import { LoginComponent } from './login/login.component';
import { FormsModule } from '@angular/forms';
import { HttpClientModule } from '@angular/common/http';
import { LoginService } from './service/login-server';
import { RouterModule } from '@angular/router';
import { LogadoComponent } from './logado/logado.component';
import { ErrorComponent } from './error/error.component';
```

```
@NgModule({
  declarations: [
    AppComponent,
    LoginComponent,
    LogadoComponent,
    ErrorComponent
```

```
],
imports: [
  BrowserModule,
  FormsModule,
  HttpClientModule,
  RouterModule.forRoot([
    { path: '', redirectTo: '/login', pathMatch: 'full' },
    { path: 'login', component: LoginComponent },
    { path: 'logado', component: LogadoComponent },
    { path: 'error', component: ErrorComponent },
    { path: "**", component: ErrorComponent }
  ])
],
providers: [LoginService],
bootstrap: [AppComponent]
})
export class AppModule { }
```

app.component.html

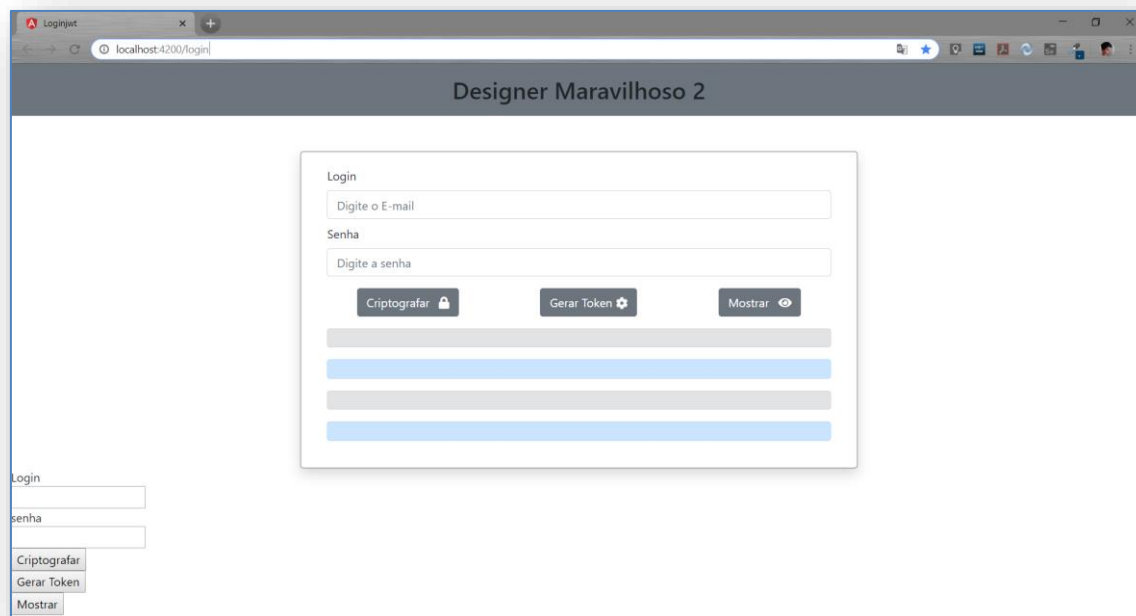
```
<app-login></app-login>
```

Rodar o projeto.

Digitar no terminal:

“ng s -o”

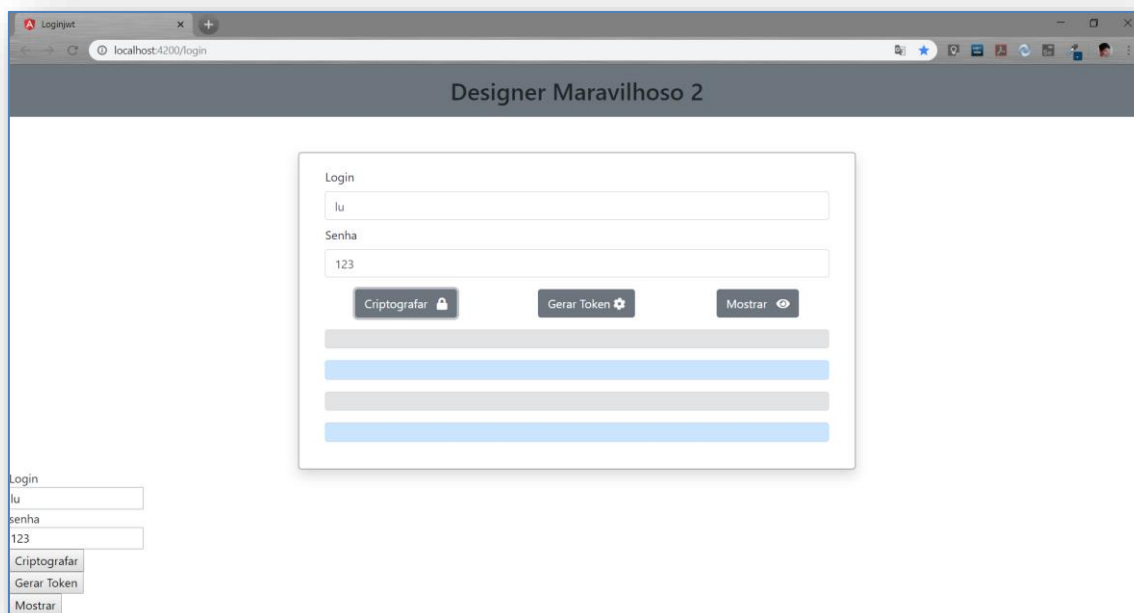
<http://localhost:4200/login>



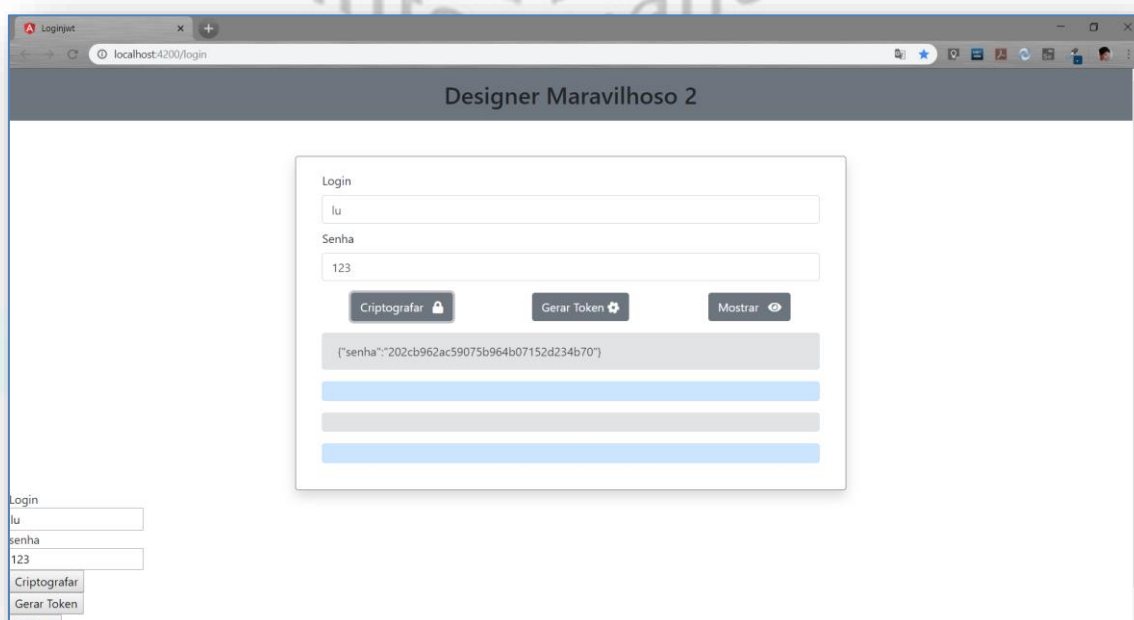
Escola de Nerds

www.cotiinformatica.com.br

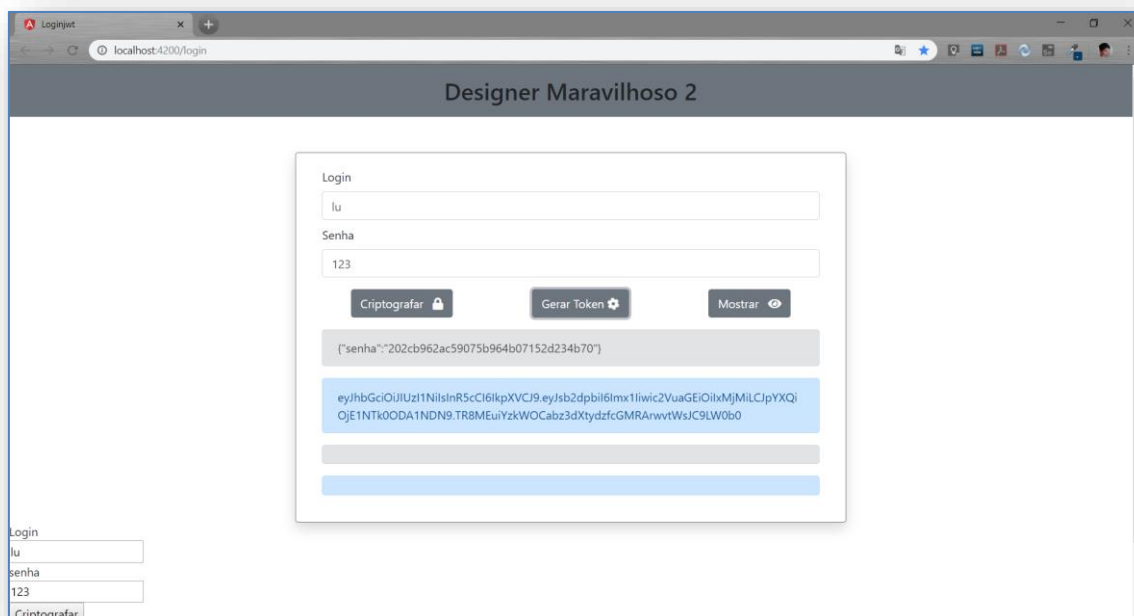
Digitando o login e senha que existem



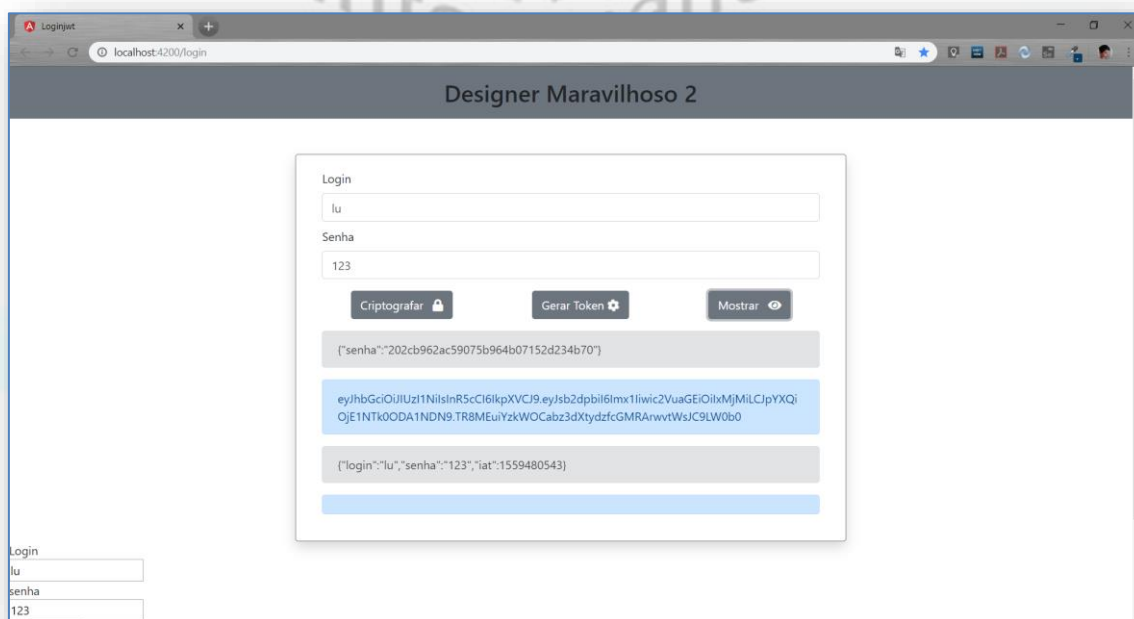
Clicando em criptografar



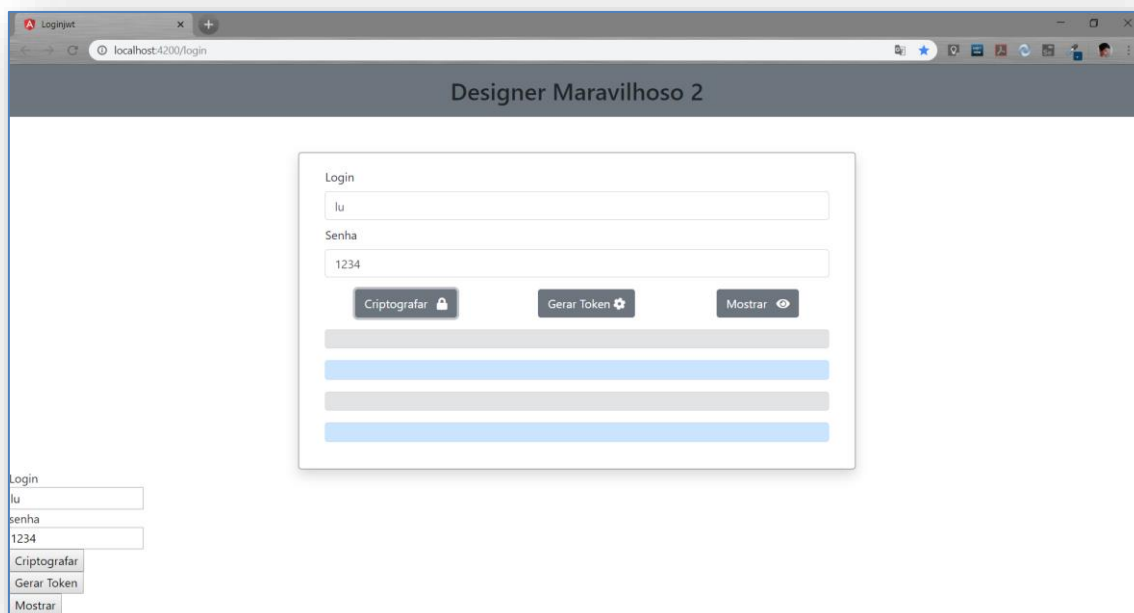
Clicando em gerar token



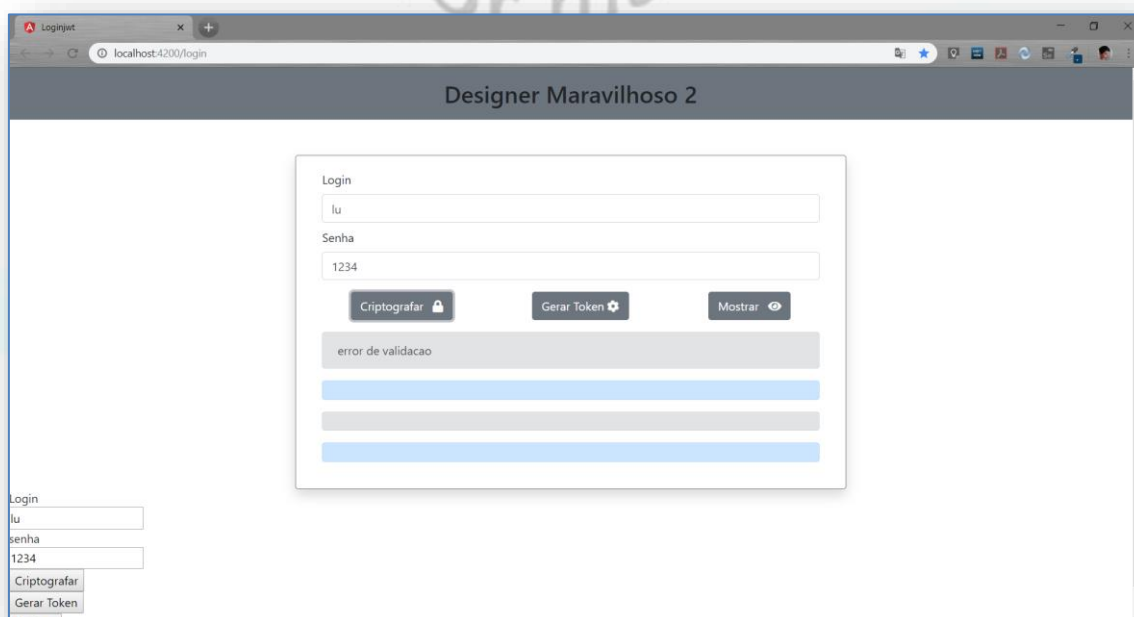
Clicar em Mostrar



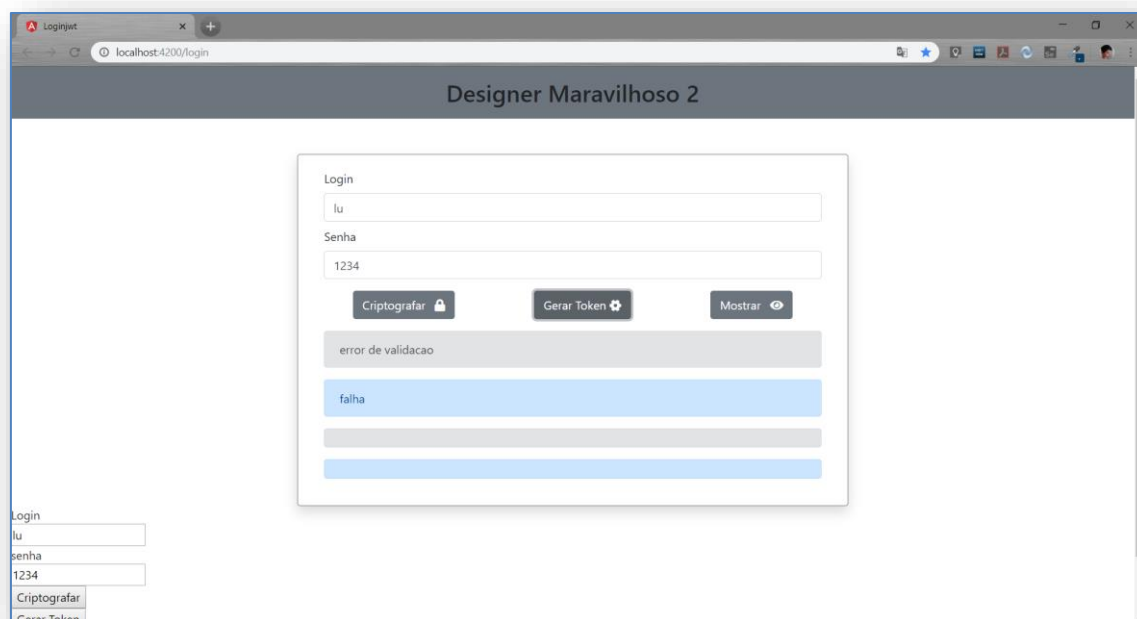
Digitando um login e senha errados



clikando em criptografar



Clicando em gerar token



Clicando em mostrar

