



SOUL CODE

Projeto Imersão
Squad 06

SQUAD 06

Carolina Queiroz
Débora Campos
Jociane Carvalho
Pedro Lopes
Priscila Alves

OBJETIVOS

- **Trabalhar em grupo e de maneira organizada**
- **Não pegar todas as tarefas, e sim, uma por vez e entregar bem feito.**
- **Trabalhar corretamente com o Github**

Carolina

Front-end

Confirmação de logout do sistema

Criar um novo campo no formulário de cadastro de funcionário para a seleção do cargo

- Confirmação de logout do sistema

▼ confirmar-logout

confirmar-logout.component.css

<> confirmar-logout.component.html

TS confirmar-logout.component.spec.ts

TS confirmar-logout.component.ts

<> navbar.component.html

```
<button matTooltip="Logout" mat-icon-button (click)="logout()">
  <mat-icon>logout</mat-icon>Sair?
</button>
</mat-toolbar-row>
</mat-toolbar>
```

TS navbar.component.ts

```
logout() {
  const dialog= this.dialog.open(ConfirmarLogoutComponent)
  dialog.afterClosed().subscribe((Response) => {
    if(Response) {
      this.authService.signOut()
    }
  })
}
```

logout.component.css




<> confirmar-logout.component.html X

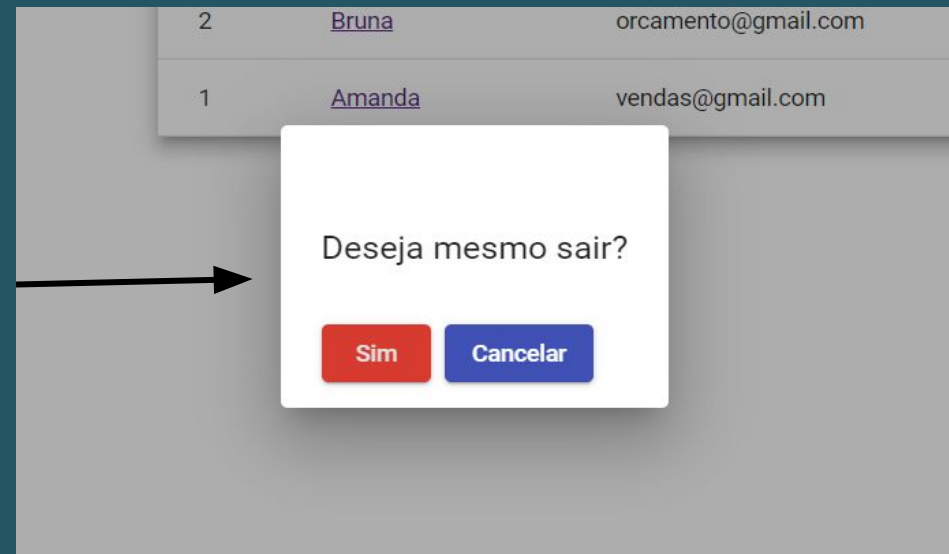
TS confirmar-logout.component.ts

src > app > confirmar-logout > <> confirmar-logout.component.html > div > button

Go to component

```
1 <h2 mat-dialog-title></h2>
2 <div mat-dialog-content>
3   <p>Deseja mesmo sair?</p>
4 </div>
5 <div mat-dialog-actions>
6   <button [mat-dialog-close]="true"
7     mat-raised-button matTooltip="Sim"
8     color="warn">Sim</button>
9   <button [mat-dialog-close]="false"
10     mat-raised-button matTooltip="Cancelar"
11     color="primary">Cancelar</button>
12 </div>
13
```

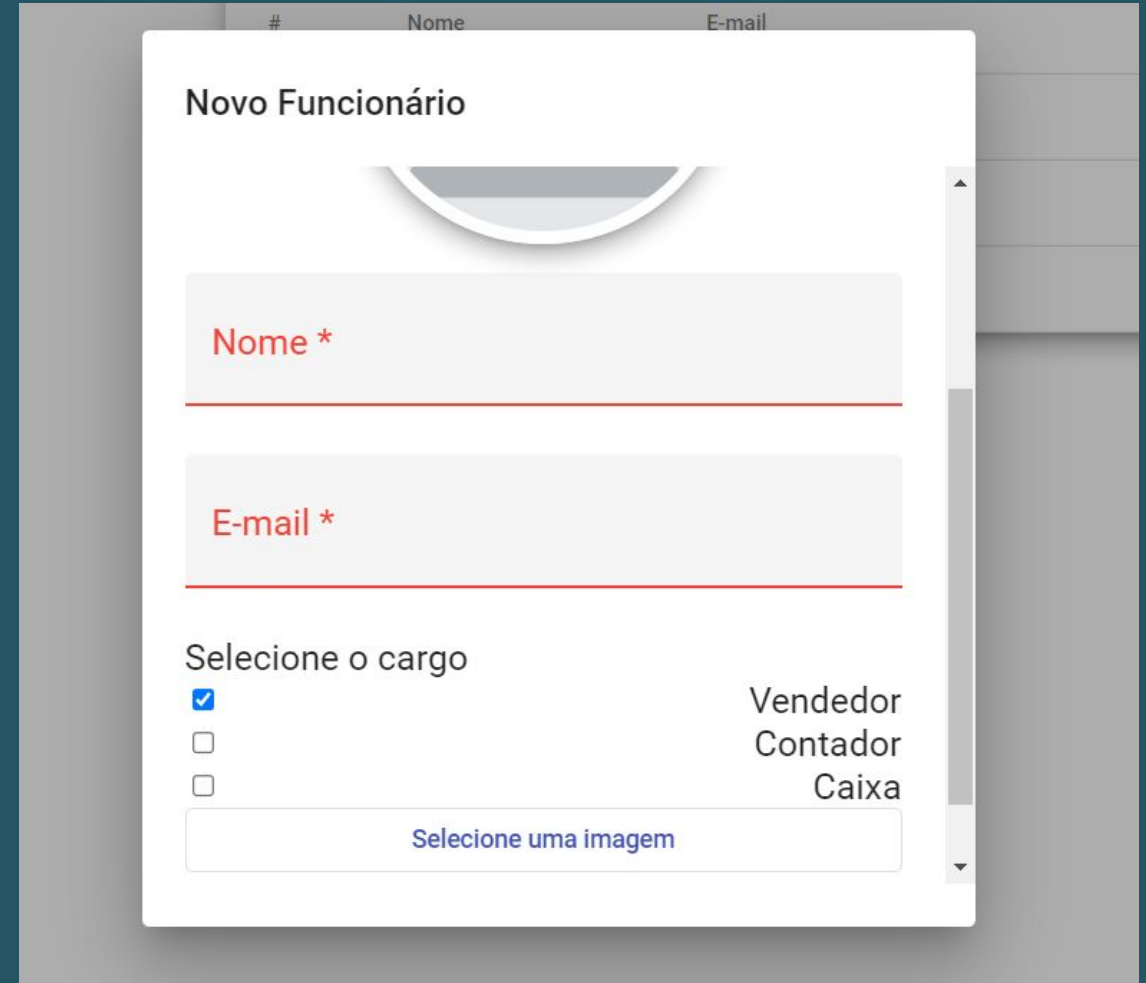
Funcionários [→ Sair?			
#	Nome	E-mail	Excluir
3	Letícia	caixa@gmail.com	
2	Bruna	orcamento@gmail.com	
1	Amanda	vendas@gmail.com	



- Criar um novo campo no formulário de cadastro de funcionário para a seleção do cargo

<> form-funcionario.component.html

```
</mat-form-field>
<label>Selecione o cargo</label>
<div>
  <input type="checkbox" id="scales" name="scales"
    checked>
  <label for="scales">Vendedor</label>
</div>
<div>
  <input type="checkbox" id="scales" name="scales"
    checked>
  <label for="scales">Contador</label>
</div>
<div>
  <input type="checkbox" id="scales" name="scales"
    checked>
  <label for="scales">Caixa</label>
</div>
```



- Endpoint para listar chamados com pagamento lançado

- Endpoint para mostrar a quantidade de chamados agrupados por status


```
60 @GetMapping("/buscarChamadosPeloPagamentoStatusLancado")
61 List<?> buscarChamadosPeloPagamentoStatusLancado(){
62     return chamadoService.buscarChamadosPeloPagamentoStatusLancado();
63 }
64
```

```
65
66 @GetMapping("/quantidadeDeChamadosPorStatus")
67 List<?> quantidadeDeChamadosPorStatus(){
68     return chamadoService.quantidadeDeChamadosPorStatus();
69 }
```

```
24
25 @Query(value = "SELECT chamado.*, pagamento.status\n" +
26     "FROM chamado\n" +
27     "INNER JOIN pagamento ON chamado.id_pagamento = pagamento.id_pagamento\n" +
28     "WHERE pagamento.status = \"LANCADO\"", nativeQuery = true )
29 List<Chamado> findByStatusLancado();
30
31 @Query(value="SELECT COUNT(id_chamado), chamado.status FROM chamado GROUP BY chamado.status", nativeQuery = true)
32 List<?> quantidadeDeChamadosPorStatus();
33
```

```
public List<Chamado> buscarChamadosPeloPagamentoStatusLancado(){
    return chamadoRepository.findByStatusLancado();
}

public List<?> quantidadeDeChamadosPorStatus(){
    return chamadoRepository.quantidadeDeChamadosPorStatus();
}
```

- Clientes
 - GET /servicos/clientes
 - GET /servicos/clientes/{idCliente}
 - POST /servicos/clientes
 - PUT /servicos/clientes/{idCliente}
 - DEL /servicos/clientes/{idCliente}
- Usuarios
 - POST /login
 - GET /servicos/usuarios
 - POST /servicos/usuarios
- Chamados
 - GET /servicos/chamados
 - GET /servicos/chamados/{idChamado}
 - GET /servicos/chamadosPeloStatus
 - GET /servicos/buscarChamadosPeloPagamentoS...
 - POST /servicos/chamados/{idCliente}
 - GET /servicos/chamadosPorIntervaloData
 - GET /servicos/quantidadeDeChamadosPorStatus
 - GET /servicos/chamadosPeloFuncionario/{idFunci...
 - POST /servicos/chamados
 - PUT /servicos/chamados/{idChamado}
 - PUT /servicos/chamadosAtribuirFuncionario/{idC...
 - PUT /servicos/chamadosModificarStatus/{idCha...
 - DEL /servicos/chamados/{idChamado}

GET `{{local}}/servicos/buscarChamadosPeloPagamentoStatusLancado` Send

Params Authorization Headers (9) Body ● Pre-request Script Tests Settings Cookies

Body Cookies Headers (14) Test Results Status: 200 OK Time: 87 ms Size: 1023 B Save Response

Pretty Raw Preview Visualize JSON

```
13      "cargo": {
14        "idCargo": 3,
15        "nome": "vendedor",
16        "descricao": "vender",
17        "salario": 1500.0
18      },
19    },
20    "cliente": {
21      "idCliente": 1,
22      "nome": "carol queiroz",
23      "email": "carol@hotmail.com",
24      "enderecoCliente": {
25        "idEndereco": 1,
26        "rua": "mar mediterraneo",
27        "bairro": "centro",
28        "cidade": "madre de Deus",
29        "uf": "ba"
30      }
31    },
32    "pagamento": {
33      "idPagamento": 123,
34      "valor": 1000.0,
35      "formPagamento": "PIX",
36      "status": "LANCADO"
37    }
38  }
39 }
```

PUT /servicos/clientes/{idCliente}

DEL /servicos/clientes/{idCliente}

Usuarios

POST /login

GET /servicos/usuarios

POST /servicos/usuarios

Chamados

GET /servicos/chamados

GET /servicos/chamados/{idChamado}

GET /servicos/chamadosPeloStatus

POST /servicos/chamados/{idCliente}

GET /servicos/chamadosPorIntervaloData

GET /servicos/quantidadeDeChamadosPorStatus

GET /servicos/chamadosPeloFuncionario/{idFunci...

POST /servicos/chamados

PUT /servicos/chamados/{idChamado}

PUT /servicos/chamadosAtribuirFuncionario/{idC...

PUT /servicos/chamadosModificarStatus/{idCha...

DEL /servicos/chamados/{idChamado}

Endereço Cliente

GET /servicos/enderecoCliente

GET /servicos/enderecoCliente/{idEnderecoCil}

POST /servicos/enderecoCliente/{idCliente}

PUT /servicos/enderecoCliente/{idEndereco}

Funcionario

GET /servicos/funcionarios

GET /servicos/funcionarios/{idFuncionario}

GET /servicos/funcionariosDoCargo/{idCargo}

GET /servicos/funcionariosEmail/{email}

POST /servicos/funcionarios/{idCargo}

PUT /servicos/funcionarios/{idFuncionario}

GET {{local}}/servicos/quantidadeDeChamadosPorStatus

Send

Params

Authorization

Headers (7)

Body

Pre-request Script

Tests

Settings

Cookies

Query Params

	KEY	VALUE	DESCRIPTION	...	Bulk Edit
	Key	Value	Description		

Body

Cookies

Headers (14)

Test Results

Status: 200 OK

Time: 20 ms

Size: 467 B

Save Response

Pretty

Raw

Preview

Visualize

JSON

```
1  [
2    [
3      1,
4      "ATRIBUIDO"
5    ],
6    [
7      1,
8      "RECEBIDO"
9    ]
10 ]
```

Débora - Front

Atualização do Arquivo ReadMe

Build

Run `ng build` to build the project. The build artifacts will be stored in the `dist/` directory.

Running unit tests

Run `ng test` to execute the unit tests via [Karma](#).

For @angular/core error

Run this sequence: `npm cache clean --force > npm install > ng update > npm update`

Running end-to-end tests

Run `ng e2e` to execute the end-to-end tests via a platform of your choice. To use this command, you need to add a package that implements end-to-end testing capabilities.

Débora - Front

Uso do service Title para
adicionar títulos nas
abas da aplicação
conforme o componente
acessado

```
7 import { Title } from '@angular/platform-browser';
8
9 @Injectable({
10   providedIn: 'root'
11 })
12 export class FuncionarioService {
13
14   private readonly baseUrl: string = 'http://localhost:8080/servicos/funcionarios'
15   atualizarFuncionariosSub$: BehaviorSubject<boolean> = new BehaviorSubject(true)
16
17   constructor(
18     private http: HttpClient,
19     private storage: AngularFireStorage,
20     private authService: AuthService,
21     private title: Title
22   ) { }
23
24   ngOnInit() {
25     this.title.setTitle("Projeto BCW20 Services Frontend Squad 06 - Funcionários");
26   }
27 }
```

Débora - Front

Acessibilidade: Tooltips nos botões da aplicação

```
4 | <h2>Funcionários</h2>
5 | </aside>
6 | <main>
7 | <button mat-mini-fab color="accent"
8 | matTooltip="Adicionar novo Funcionario"
9 | (click)="abrirFormFuncionario()" id="add">
10 |   <mat-icon>add</mat-icon>
11 | </button>
12 | <div>
```

```
9 | import {MAT_TOOLTIP_DEFAULT_OPTIONS, MatTooltipDefaultOptions} from '@angular/material/tooltip';
10 |
11 | @Component({ ...
15 | })
16 |
17 | export class ListarFuncionariosComponent implements OnInit { ...
30 | }
31 |
32 | export class TooltipModifiedDefaultsExample {}
```



Débora - Front

Aprimorar meta tags do arquivo index.html

Uso do service Title para adicionar títulos nas abas

```
4 <meta charset="utf-8">
5 <title>Services Frontend</title> ←
6 <base href="/">
7 → <meta name="description" content="Soul Code Academy -
8   Java FullStack: Projeto BCW20 Services Frontend Squad 06">
9 <link rel="icon" type="image/x-icon" href="favicon.ico">
```

Débora - Back

Criação do arquivo ReadMe para o projeto

Services Backend Project - Squad 06

This project is part of the Java Full Stack bootcamp at SoulCode Academy.

It was generated with Java at IntelliJ, using MySql, Spring and Postman.

Development server

Run `servicosApplication` to start the Java server on IntelliJ.

Run Redis application with the command `redis-server` for cache.

Run Postman application and import the file `BCW20-Servicos_Squad06.postman_collection.json` for the endpoints.

The application will automatically reload if you change any of the source files.

Token

Token expires every 24 hours. To update it, follow the steps below:

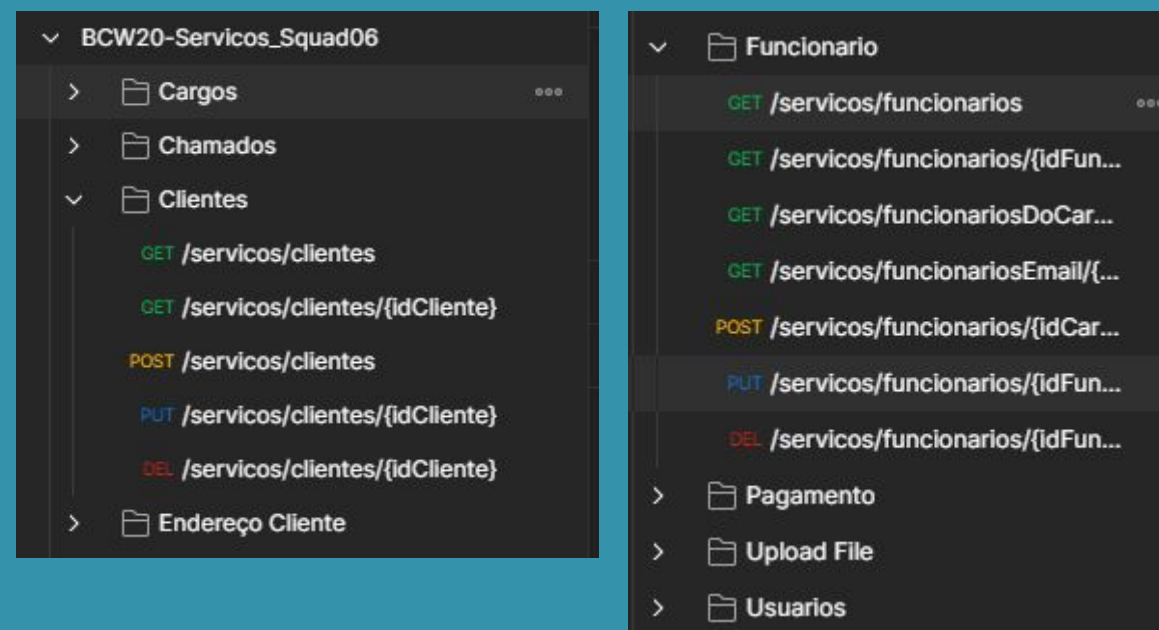
- At Postman , open the folder 'Usuarios' and get the end point 'login' to generate a new token;
- At the main folder called `BCW20-Servicos_Squad06` > `Auth` > Token Type: `Bearer Token` > Copy the generated token

Débora - Back

Remoção de comentários didáticos

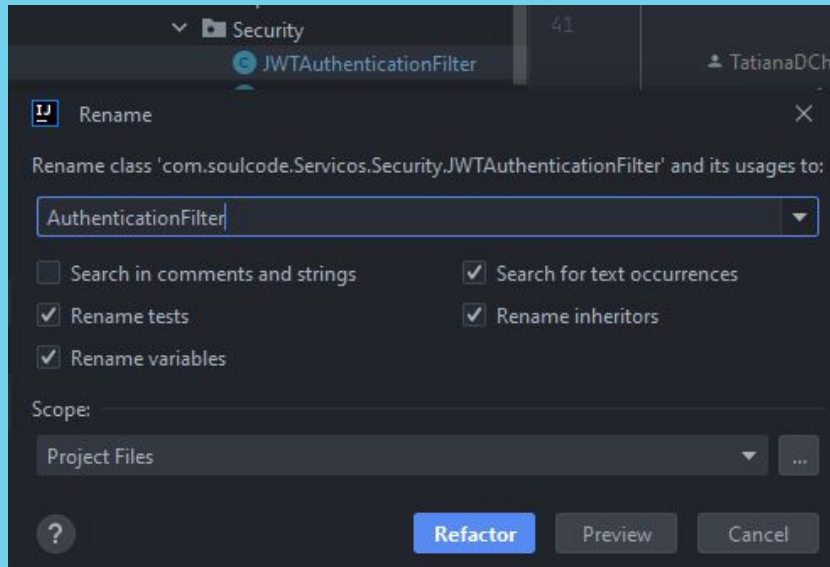
Utilizei esta atividade para lembrar os comandos do Github, entender a sequência necessária para fazer o Pull Request corretamente.

Documentar API completa no postman



Débora - Back

Refatorar Security



Utilizado a Refatoração do IntelliJ

Jociane - Back

Aplicar cache com Redis no ChamadoService

```
32 1 usage  Anecarvalho +1
33 @Cacheable(value = "chamadosCache")
34 public List<Chamado> mostrarTodosChamados(){
35     return chamadoRepository.findAll();
36 }
37
38 4 usages  TatianaDChiacchio +1
39 @Cacheable(value = "chamadosCache", key = "#idchamado")
40 public Chamado mostrarUmChamado(Integer idChamado) {
41     Optional<Chamado> chamado = chamadoRepository.findById(idChamado);
42     return chamado.orElseThrow();
43 }
44
45 1 usage  TatianaDChiacchio +1
46 @Cacheable(value = "chamadosCache", key = "#idCliente")
47 public List<Chamado> buscarChamadosPeloCliente(Integer idCliente){
48     Optional<Cliente> cliente = clienteRepository.findById(idCliente);
49     return chamadoRepository.findByCliente(cliente);
50 }
51
52 1 usage  TatianaDChiacchio +1
53 @Cacheable(value = "chamadosCache", key = "#Funcionario.idFuncionario")
54 public List<Chamado> buscarChamadosPeloFuncionario(Integer idFuncionario){
55     Optional<Funcionario> funcionario = funcionarioRepository.findById(idFuncionario);
56     return chamadoRepository.findByFuncionario(funcionario);
57 }
```

```
54 }
55 1 usage  TatianaDChiacchio +1
56 @Cacheable(value = "chamadosCache", key = "#status")
57 public List<Chamado> buscarChamadosPeloStatus(String status) { return chamadoRepository.findByStatus(status);
58 }
59
60 1 usage  TatianaDChiacchio +1
61 @Cacheable(value = "chamadosCache", key = "(#data1, #data2)")
62 public List<Chamado> buscarPorIntervaloData(Date data1, Date data2){
63     return chamadoRepository.findByIntervaloData(data1,data2);
64 }
65
66 1 usage  Carol
67 public List<Chamado> buscarChamadosPeloPagamentoStatusLancado(){
68     return chamadoRepository.findByStatusLancado();
69 }
70
71 1 usage  TatianaDChiacchio +1
72 @CachePut(value = "chamadosCache", key = "#chamado.idChamado")
73 @Transactional
74 public Chamado cadastrarChamado(Chamado chamado, Integer idCliente){
75     chamado.setStatus(StatusChamado.RECEBIDO);
76     chamado.setFuncionario(null);
77     Optional<Cliente> cliente = clienteRepository.findById(idCliente);
78     chamado.setCliente(cliente.get());
79     return chamadoRepository.save(chamado);
80 }
81
82 1 usage  TatianaDChiacchio +1
83 @CacheEvict(value = "chamadosCache", key = "#idChamado", allEntries = true)
84 public void excluirChamado(Integer idChamado) { chamadoRepository.deleteById(idChamado); }
```

Jociane - Back

Endpoint para listar funcionários sem chamado

```
1 usage Anecarvalhoss
35 public List<Funcionario> mostrarFuncionariosSemChamado(){
36
37     return funcionarioRepository.findFuncSemChamado();
38 }
39
40
2 usages TatianaDChiacchio +1
```

```
Anecarvalhoss
26 @GetMapping("/funcionarios/semChamado")
27 public List<Funcionario> mostrarFuncionariosSemChamado(){
28     List<Funcionario> funcionarios = funcionarioService.mostrarFuncionariosSemChamado();
29     return funcionarios;
30 }
31
32
```

BCW20-Servicos_Squad06 / Funcionario / /servicos/funcionarios

GET ▼ {{local}}/servicos/funcionarios/semChamado

Params Authorization Headers (7) Body Pre-request Script Tests Settings

Query Params

KEY	VALUE
-----	-------

Body Cookies Headers (14) Test Results

Pretty Raw Preview Visualize JSON ▼ 🔗

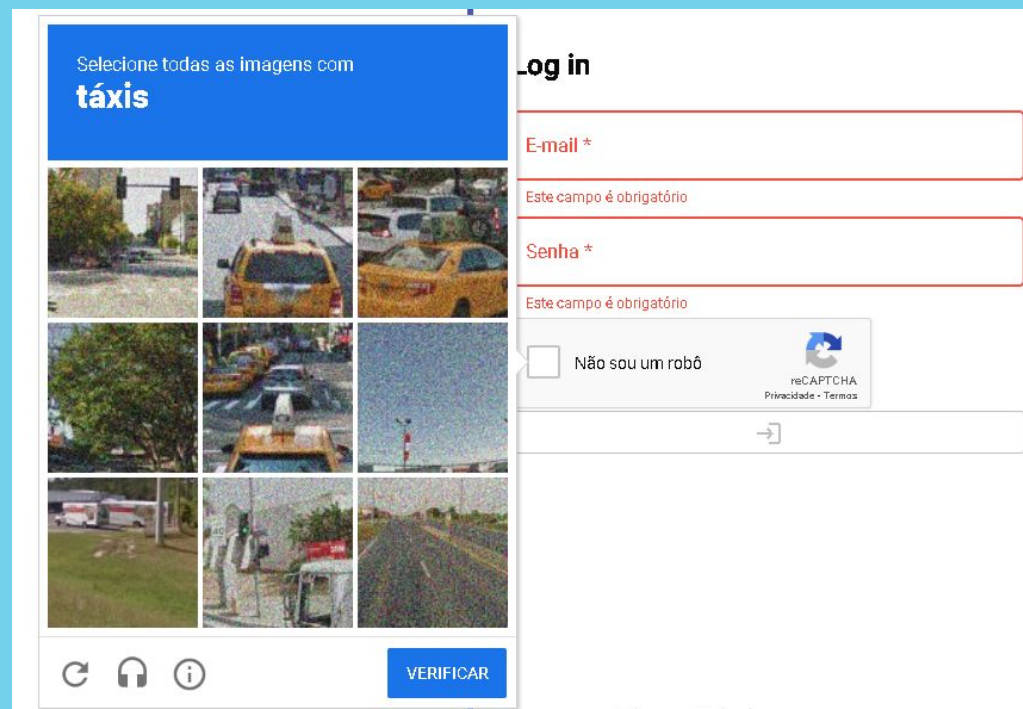
```
1 {
2   {
3     "idFuncionario": 3,
4     "nome": "Natielle",
5     "email": "nat@gmail.com",
6     "foto": null,
7     "cargo": null
8   }
9 }
```


Pedro - Front

Adição de reCAPTCHA na tela de login

O que é o Google reCAPTCHA ?

O reCAPTCHA usa um mecanismo avançado de análise de risco e desafios adaptáveis para impedir que softwares maliciosos se envolvam em atividades abusivas em seu site. Enquanto isso, usuários legítimos poderão fazer login, fazer compras, visualizar páginas ou criar contas e usuários falsos serão bloqueados.



The image shows a web interface for a login page. On the left, there is a reCAPTCHA challenge titled "Selecione todas as imagens com táxis" (Select all images with taxis). It consists of a 3x3 grid of images. The top row shows a street scene, a yellow taxi, and a yellow taxi. The middle row shows a street scene, a yellow taxi, and a street scene. The bottom row shows a street scene, a yellow taxi, and a street scene. Below the grid are icons for a refresh button, a help button, and an information button, followed by a blue "VERIFICAR" button.

On the right, there is a "Log in" form. It has two input fields: "E-mail *" and "Senha *". Below each field is a red error message: "Este campo é obrigatório". Below the "Senha *" field is a checkbox labeled "Não sou um robô" and a reCAPTCHA logo. Below the checkbox is a blue button with a right arrow icon. At the bottom of the form is a blue button labeled "Log in".

Pedro - Front

Adição de reCAPTCHA na tela de login

Criação do token de autenticação

```
recaptcha: {  
  siteKey:  
    '6Lf42xwhAAAAAJLRTlm-Zk1L6mqH6_MsveaLN8k6'  
}
```

instalação da biblioteca

```
npm install ng-recaptcha
```

Importação dos módulos

```
import { RECAPTCHA_SETTINGS, RecaptchaFormsModule,  
RecaptchaModule, RecaptchaSettings } from  
'ng-recaptcha';
```

Fator de validação

```
loginForm: FormGroup = this.fb.group({  
  login: ['', [ Validators.required ]],  
  password: ['', [ Validators.required ]],  
  recaptcha: ['', [Validators.required]]  
})
```

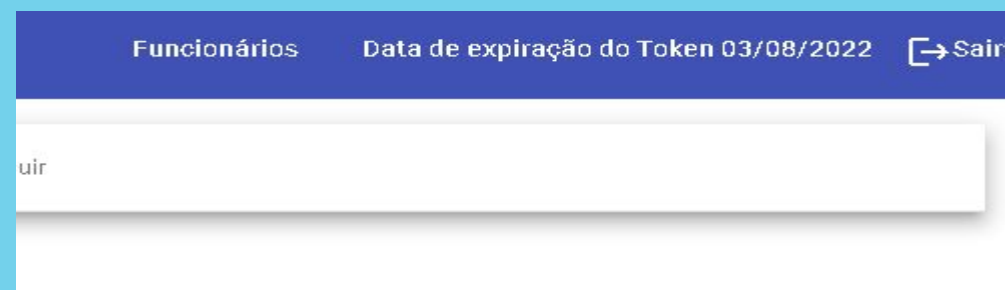
criação da tag html

```
<re-captcha formControlName = "recaptcha"></  
re-captcha>
```

Pedro - Front

Data de expiração da sessão de usuário a partir do

```
usuario() {  
  /* console.log(this.authService.usuario()); */  
  
  this.dataExpiration = this.authService.dataToken();  
  
  this.dataExpiration.getDate();  
  
  const dataAtual = new Date().getDate();  
  
  const sub = this.dataExpiration.getDate() - dataAtual;  
  
  this.timer = new Date(sub);  
}  
  
dataExpiration!: Date  
  
timer!: Date
```



Priscila - Back

Endpoint para listar quais funcionários tem foto nula.

prisfranca

```
public List<Object> findByFuncionarioSemFoto(){return funcionarioRepository.findByFuncionarioSemFoto();}
```

FuncionarioService

2 usages prisfranca

```
@Query(value = "SELECT * FROM funcionario WHERE foto is null",nativeQuery = true )  
List<Object> findByFuncionarioSemFoto();
```

prisfranca

```
@GetMapping("/funcionarioSemFoto")  
public List<Object> findByFuncionarioSemFoto(){  
  
    return this.funcionarioRepository.findByFuncionarioSemFoto();  
}
```

FuncionarioController

Priscila - Back

Endpoint para listar quais funcionários tem foto nula.

The screenshot shows a REST client interface with a GET request to `{{local}}/servicos/funcionarioSemFoto ...`. The response is a JSON array of two objects, each representing an employee with a null photo.

Query Params

KEY	VALUE	DESCRIPTION	...	Bulk Edit
Key	Value	Description		

Body Cookies Headers (14) Test Results Status: 200 OK Time: 126 ms Size: 537 B Save Response

Pretty Raw Preview Visualize JSON

```
[
  3,
  "lucas@gmail.com",
  null,
  "Lucas Lopes",
  2,
  ""
],
[
  4,
  "antonioalves@gmail.com",
  null,
  "Antonio Alves",
  2,
```

Priscila - Back

Listar QUANTOS funcionários estão em um determinado Cargo.

```
prisfranca  
public List<Object> FuncionariosQtdPeloCargo() {return funcionarioRepository.findByFuncionariosQtdPeloCargo();}
```

FuncionarioService

```
prisfranca  
@GetMapping("/funcionariosQtdPeloCargo/{idCargo}")  
public List<Object> FuncionariosQtdPeloCargo(@PathVariable Integer idCargo) {  
    return this.funcionarioRepository.findByFuncionariosQtdPeloCargo();  
}
```

FuncionarioController

```
2 usages prisfranca  
@Query(value = "SELECT COUNT(funcionario.id_cargo) as QUANTIDADE, cargo.nome\n" +  
    "FROM funcionario \n" +  
    "LEFT JOIN cargo\n" +  
    "\tON funcionario.id_cargo = cargo.id_cargo\n" +  
    "GROUP BY funcionario.id_cargo", nativeQuery = true)  
List<Object> findByFuncionariosQtdPeloCargo();
```

FuncionarioRepository

Priscila - Back

Listar QUANTOS funcionários estão em um determinado Cargo.

The screenshot shows a REST client interface with the following details:

- URL:** `{{local}}/servicos/funcionariosQtdPeloCargo/1`
- Method:** GET
- Response Status:** 200 OK, Time: 75 ms, Size: 465 B
- Response Body (JSON):**

```
1 [
2   [
3     1,
4     "Supervisor"
5   ],
6   [
7     2,
8     "Caixa"
9   ]
10 ]
```

Priscila - Back

Melhorar tratamento de exceções dos Services

4 usages TatianaDChiacchio +1 *

```
@Cacheable(value = "chamadosCache", key = "idchamado")
public Chamado mostrarUmChamado(Integer idChamado) {
    Optional<Chamado> chamado = chamadoRepository.findById(idChamado);
    return chamado.orElseThrow(() -> new EntityNotFoundException("Chamado não encontrado " + idChamado));
}
```



MostrarChamadoService

Priscila - Back

Melhorar tratamento de exceções dos Services

BCW20-Servicos_Squad06 Copy / Chamados / /servicos/chamados/{idChamado}

GET `{{local}}/servicos/chamados/1` Send

Params Authorization Headers (7) Body Pre-request Script Tests Settings Cookies

Query Params

KEY	VALUE	DESCRIPTION	...	Bulk Edit
Key	Value	Description		

Body Cookies Headers (14) Test Results Status: 200 OK Time: 99 ms Size: 1.02 KB Save Response

Pretty Raw Preview Visualize JSON

```
1  {
2    "idChamado": 1,
3    "titulo": "limpeza",
4    "descricao": "limpeza de pc",
5    "dataEntrada": "2022-07-12T03:00:00.000+00:00",
6    "status": "ATRIBUIDO",
7    "funcionario": {
8      "idFuncionario": 1,
9      "nome": "",
10     "email": "mel@gmail.com",
11     "foto": "C:/Users/prisc/Downloads/fotoFunc/nome2",
12     "cargo": {
13       "idCargo": 1,
14       "nome": "Supervisor",
```

BCW20-Servicos_Squad06 Copy / Chamados / /servicos/chamados/{idChamado}

GET `{{local}}/servicos/chamados/2` Send

Params Authorization Headers (7) Body Pre-request Script Tests Settings Cookies

Query Params

KEY	VALUE	DESCRIPTION	...	Bulk Edit
Key	Value	Description		

Body Cookies Headers (14) Test Results Status: 404 Not Found Time: 41 ms Size: 603 B Save Response

Pretty Raw Preview Visualize JSON

```
1  {
2    "timestamp": "2022-08-01T17:44:40.833062700Z",
3    "status": 404,
4    "error": "Registro não encontrado",
5    "message": "Chamado nao encontrado 2",
6    "path": "/servicos/chamados/2"
7  }
```

AGRADECIMENTOS

Professor Renato, Tatiana
(saudades!) e Almir pelas
orientações

Aos colegas de bootcamp por
compartilhar idéias e ajudar na
resolução de algumas tarefas