

## Link para Collection Postman

---

- ✓ ~~Postman collection~~ - <https://www.getpostman.com/collections/384fd218a1c506e39455>

## Link projeto backend

---

- ✓ ~~Link de acesso ao projeto no github~~ - <https://github.com/ValdirCezar/helpdesk-backend>

## Aula 01 - Introdução

---

- ✓ ~~Aula de Introdução~~

## Aula 02 - Ferramentas (Curso DEV Full Stack)

---

- ✓ ~~Ferramentas Backend~~
  - ✓ 10 de 10
  - ✓ Java JDK 11 LTS - <https://www.oracle.com/java/technologies/javase-downloads.html>
  - ✓ Spring Tools Suit - <https://spring.io/tools>
  - ✓ Xampp - [https://www.apachefriends.org/pt\\_br/download.html](https://www.apachefriends.org/pt_br/download.html)
  - ✓ Git Bash - <https://git-scm.com/downloads>
  - ✓ Postman - <https://www.postman.com/downloads/>
  - ✓ Conta no Heroku - <https://www.heroku.com/>
  - ✓ Conta no Github - <https://github.com/>
  - ✓ 7-Zip - <https://www.7-zip.org/>
  - ✓ Configurar as variáveis de ambiente após instalação do Java JDK
  - ✓ Comando para verificar instalação do JDK "java --version"

- ✓ ~~Ferramentas Frontend~~
  - ✓ 2 de 2
  - ✓ VSCode - <https://code.visualstudio.com/download>
  - ✓ Node.js - <https://nodejs.org/en/>

## Aula 03 - Iniciando a API (Curso DEV Full Stack)

---

- ✓ Startar o STS
- ✓ Adicionar o tema Dark
- ✓ Iniciar um projeto  
✓ 1 de 1
- ✓ Observações sobre a versão do projeto - Seguir a mesma versão do curso

## Aula 04 - Tipos enumerados (Curso DEV Full Stack)

---

- ✓ Configurar o Git
- ✓ Realizar primeiro commit
- ✓ Criar tipos enumerados

## Aula 05 - Classes de domínio (Parte 1) (Curso DEV Full Stack)

---

- ✓ Iniciar as classes de domínio

## Aula 06 - Classes de domínio (Parte 2) (Curso DEV Full Stack)

---

- ✓ Dar continuidade nas classes de domínio após a criação dos tipos Enumerados

## Aula 07 - Carga inicial da base de dados (Curso DEV Full Stack)

---

- ✓ Adicionar configurações no application.properties
- ✓ Informar possíveis falhas no H2 e como resolver
- ✓ Criar os repositories para a persistência no banco
- ✓ Criar instancias para a carga inicial do banco

## Aula 08 - Perfil de teste (Curso DEV Full Stack)

---

- ✓ Criar perfil de teste para um código mais organizado

## Aula 09 - Perfil de desenvolvimento (Curso DEV Full Stack)

---

- ✓ ~~Criar o perfil de dev para armazenar os dados em um banco MySQL~~
  - ✓ 1 de 1
    - ✓ `spring.datasource.password=`  
`spring.jpa.hibernate.ddl-auto=create`

## Aula 10- findById (Tecnico) (Curso DEV Full Stack)

---

- ✓ Implementar endpoint que busca técnico por ID

## Aula 11 - Padrão DTO (Curso DEV Full Stack)

---

- ✓ ~~Explicar o padrão DTO~~
- ✓ Criar a classe TecnicoDTO
- ✓ ~~Alterar o retorno do endpoint que busca um Técnico por ID para TecnicoDTO~~

## Aula 12 - Exceção para objeto não encontrado (Curso DEV Full Stack)

---

- ✓ ~~Criar um manipulador para a exceção de objeto não encontrado~~

## Aula 13 - FindAll (Tecnico) (Curso DEV Full Stack)

---

- ✓ ~~Implementar o endpoint que fará a listagem de todos os técnicos do banco~~

## Aula 14 - create (Tecnico) (Curso DEV Full Stack)

---

- ✓ ~~Implementar endpoint que cria um novo tecnico~~

## Aula 15 - Exceção para CPF já cadastrado (Curso DEV Full Stack)

---

- ✓ ~~Criar um método que verifique se o CPF já possui cadastro no banco~~

## Aula 16 - Validations (Curso DEV Full Stack)

---

- ✓ Adicionar as anotações de validação dos atributos

## Aula 17 - update (Tecnico) (Curso DEV Full Stack)

---

- ✓ Implementar endpoint para atualização das informações do Tecnico

## Aula 18 - delete (Tecnico) (Curso DEV Full Stack)

---

- ✓ Implementar endpoint para deletar Tecnico por ID

## Aula 19 - CRUD para Cliente (Curso DEV Full Stack)

---

- ✓ O CRUD para Cliente pode ser idêntico ao técnico

## Aula 20 - findByID (Chamado) (Curso DEV Full Stack)

---

- ✓ Busca um chamado por ID

## Aula 21 - findAll (Chamado) (Curso DEV Full Stack)

---

- ✓ Lista todos os chamados do banco

## Aula 22 - create (Chamado) (Curso DEV Full Stack)

---

- ✓ Implementar endpoint para criação de chamado

## Aula 24 - Autenticação com JWT (Curso DEV Full Stack)

---

- ✓ Adicionar as dependências para Spring Security e JWT

- ✓ Criar a classe SecurityConfig
  - ✓ 4 de 4

- ✓ Esta classe herda WebSecurityConfigureAdapter
- ✓ Sobrescrever o método configure(HttpSecurity http)
- ✓ Liberar de forma explicita as configurações de cross-origin
- ✓ Liberar acesso ao h2 pois o mesmo possui uma peculiaridade

- ✓ ~~Adicionar um Bean de BCryptPasswordEncoder em SecurityConfig~~
  - ✓ 1 de 1
  - ✓ Assim teremos um BCrypt em forma de Bean para encodar a senha do usuário
- ✓ ~~Criar classe de usuário UserSS conforme contrato do Spring Security~~
  - ✓ 5 de 5
  - ✓ Essa classe deve implementar o contrato UserDetails
  - ✓ Adicionar métodos do contrato UserDetails
  - ✓ Adicionar atributos id, email, senha e lista de authorities
  - ✓ Criar construtor para UserSS com parametros
  - ✓ Adicionar getter para id
- ✓ ~~Criar classe de serviço UserDetailsServiceImpl conforme contrato do Spring Security~~
  - ✓ 6 de 6
  - ✓ Essa classe deve implementar a interface UserDetailsService
  - ✓ Implementar método loadUserByUsername
  - ✓ Injetar um UsuarioRepository
  - ✓ Instanciar um Usuario chamando o repo.findByEmail
  - ✓ Verificar se o usuário foi encontrado e caso não, lançar uma exceção
  - ✓ Retornar um new UserSS com os parâmetros do usuário buscado
- ✓ ~~Criar uma classe CredenciaisDTO~~
  - ✓ 2 de 2
  - ✓ Incluir atributos email e senha
  - ✓ Construtores e getters e setters
- ✓ ~~Incluir as propriedades de JWT (segredo e tempo de expiração) em application.properties~~
  - ✓ 2 de 2
  - ✓ Jwt.secret=
  - ✓ Jwt.expiration=(Tempo em millis)
- ✓ ~~Criar uma classe JWTUtil (@Component)~~
  - ✓ 1 de 1



- ✓ Criar o método generateToken(String email)

## ✓ Criar um filtro de autenticação ✓ 8 de 8

- ✓ Em security criar a classe JWTAuthenticationFilter
- ✓ Deve estender o UsernamePasswordAuthenticationFilter
- ✓ Implementar método Authentication attemptAuthentication (IDE)
- ✓ Implementar o método successfulAuthentication (IDE)
- ✓ Implementar o método unsuccessfulAuthentication (IDE)
- ✓ Criar construtor para os atributos
- ✓ Registrar filtro de autenticação em SecurityConfig
- ✓ Sobrecarga de configure

## Aula 25 - Autorização (Curso DEV Full Stack)

---

### ✓ Criar a classe JWTAuthorizationFilter ✓ 1 de 1

- ✓ Essa classe estende BasicAuthenticationFilter

### ✓ Construtor com parametros

### ✓ Chamar o metodo do FilterInternal()

### ✓ Adicionar filtro em securityConfig

### ✓ Adicionar atributos para JWTUtil e UserDetailsService

## Aula 26 - Deploy do backend para nuvem (Curso DEV Full Stack)

---

### ✓ Adicionar arquivo system.properties na raiz do projeto

### ✓ Criar repositório no github

- ✓ Adicionar "java.runtime.version=11" em system.properties
- ✓ Associar projeto local ao repositório no github
- ✓ Realizar push

## Aula 27 - Iniciando projeto front (Curso DEV Full Stack)

---

- ✓ Instalar Node.js
- ✓ Instalar Angular CLI na versão 12.0.3
  - ✓ 1 de 1
  - ✓ npm install -g @angular/cli@12.0.3
- ✓ Subir a aplicação
  - ✓ 1 de 1
  - ✓ ng serve
- ✓ Iniciar um projeto usando o Angular CLI
  - ✓ 1 de 1
  - ✓ ng new helpdesk --minimal --strict

## Aula 28 - Estrutura do projeto (Curso DEV Full Stack)

---

- ✓ Desabilitar o strict do typescript
- ✓ Modificar o inlineTemplate para true
- ✓ Criar arquivo html para componente App
- ✓ Adicionar Angular Material ao projeto

## Aula 29 - Adicionar imports para o projeto (Curso DEV Full Stack)

---

- ✓ Imports para Angular Material e Formulários do Angular



## Aula 30 - Criando primeira Rota (Curso DEV Full Stack)

---

- ✓ Criar componente NAV
- ✓ Configurar uma rota para ele

## Aula 31 - HTML e CSS para componente NAV (Curso DEV Full Stack)

---

- ✓ Adicionar o HTML e CSS do componente NAV

## Aula 32 - Componente HOME (Curso DEV Full Stack)

---

- ✓ Criar componente HOME
- ✓ Criar rota para o mesmo
- ✓ Implementar HTML e CSS para o mesmo

## Aula 33 - Componente HEADER (Curso DEV Full Stack)

---

- ✓ Implementar o componente HEADER

## Aula 34 - Componente TECNICO-LIST (Parte 1) (Curso DEV Full Stack)

---

- ✓ Criar componente TECNICO-LIST
- ✓ Criar rota para o mesmo
- ✓ Implementar HTML e CSS para o mesmo

## Aula 35 - Componente LOGIN (Parte 1) (Curso DEV Full Stack)

---

- ✓ Criar componente LOGIN
- ✓ Criar rota para o mesmo



# Concluída

quarta-feira, 21 de julho de 2021

- ✓ ~~HTML e CSS para o mesmo~~

## Aula 36 - Componente LOGIN (Parte 2) (Curso DEV Full Stack)

---

- ✓ ~~Validação de formulários~~
- ✓ ~~Instalar toastr via npm~~
- ✓ ~~Implementar serviço de mensagens~~
  - ✓ 2 de 2
  - ✓ Instalar toastr "npm i ngx-toastr"
  - ✓ Adicionar ToastrModule.forRoot() nos imports

## Aula 37 - Componente LOGIN (Parte 3) (Curso DEV Full Stack)

---

- ✓ ~~Instalar dependência de JwtHelperService~~
- ✓ ~~Criar um serviço para login AuthService~~
  - ✓ 4 de 4
  - ✓ O mesmo deve ter os métodos abaixo
  - ✓ isAuthenticated
  - ✓ authenticate
  - ✓ successfulLogin
- ✓ ~~Criar api.config.ts~~
  - ✓ 1 de 1
  - ✓ Adicionar url base
- ✓ ~~criar storage\_keys.config.ts~~
- ✓ ~~Tentar autenticar usando o service implementado~~

## Aula 38 - Logout (Curso DEV Full Stack)

---

- ✓ Implementar a funcionalidade de logout

## Aula 39 - Intercetor para requisições HTTP (Curso DEV Full Stack)

---

- ✓ Implementar um interceptor que adicione um token no header das requisições http

## Aula 40 - Finalizando componente TECNICO-LIST (Curso DEV Full Stack)

---

- ✓ Finalizar a listagem dos técnicos

## Aula 41 - Componente TECNICO-CREATE (Parte 1) (Curso DEV Full Stack)

---

- ✓ Implementar HTML e CSS para o mesmo

- ✓ Criar componente

- ✓ Criar uma rota para o mesmo

## Aula 42 - Componente TECNICO-CREATE (Parte 2) (Curso DEV Full Stack)

---

- ✓ Implementar validação de formulários para o componente TECNICO-CREATE

- ✓ Adicionar o ngx-mask para adicionar uma mascara para o CPF

## Aula 43 - Componente TECNICO-CREATE (Parte 3) (Curso DEV Full Stack)

---

- ✓ Implementar serviço para criar um novo técnico

## Aula 44 - Botões de ação em TECNICO-LIST (Curso DEV Full Stack)

---

- ✓ Implementar botões de ação em TECNICO-LIST

## Aula 45 - Componente TECNICO-UPDATE (Curso DEV Full Stack)

---

- ✓ Criar componente TECNICO-UPDATE

- ✓ Criar rota para o mesmo
- ✓ Implementar o HTML e CSS do mesmo
- ✓ Implementar serviço para o mesmo

## Aula 46 - Bugfix para alteração de senha (Curso DEV Full Stack)

---

- ✓ Corrigir bug na alteração de senha

## Aula 47 - Componente TECNICO-DELETE (Curso DEV Full Stack)

---

- ✓ Implementar o componente TECNICO-DELETE

## Aula 48 - CRUD completo para cliente (Curso DEV Full Stack)

---

- ✓ Implementar CRUD completo para Cliente com base no template de Técnico

## Aula 49 - Componente CHAMADO-LIST (Parte 1) (Curso DEV Full Stack)

---

- ✓ Criar interface Chamado
- ✓ Gerar componente CHAMADO-LIST
- ✓ Adicionar link para o mesmo no menu lateral
- ✓ Iniciar HTML e CSS para o mesmo

## Aula 50 - Implementar serviço para Chamado (Curso DEV Full Stack)

---

- ✓ Criar um serviço para Chamado
- ✓ Criar método que lista os chamados



- ✓ Criar métodos para ordenar chamados por status

## Aula 51 - Componente CHAMADO-CREATE (Parte 1) (Curso DEV Full Stack)

---

- ✓ Criar componente CHAMADO-CREATE

- ✓ Implementar HTML e CSS para o mesmo

## Aula 52 - Validação de formulário CHAMADO-CREATE (Curso DEV Full Stack)

---

- ✓ Implementar validação de formulário para componente CHAMADO-CREATE

## Aula 53 - Serviço para CHAMADO-CREATE (Curso DEV Full Stack)

---

- ✓ Implementar método que liste os técnicos

- ✓ Implementar método que liste os clientes

- ✓ Implementar método que faça uma requisição para criar novo chamado

## Aula 54 - Componente CHAMADO-UPDATE (Curso DEV Full Stack)

---

- ✓ Implementar HTML e CSS para o mesmo

## Aula 55 - Serviço para atualizar chamado (Curso DEV Full Stack)

---

- ✓ Implementar serviço que atualize as informações de um chamado

## Aula 56 - Ajustes no HTML de CHAMADO-UPDATE (Curso DEV Full Stack)

---

- ✓ Realizar ajustes no HTML do CHAMADO-UPDATE

## Aula 57 - Componente CHAMADO-READ (Curso DEV Full Stack)

---

- ✓ Implementar um componente com template padrão simples para visualizar o chamado

- ✓ ~~O mesmo não deve ser editável, apenas para visualizar~~

## Aula 58 - Ajustes e melhorias de CSS (Curso DEV Full Stack)

---

- ✓ ~~Melhorias no CSS da página de LOGIN - (Adicionar ícones nos inputs)~~
- ✓ ~~Mudar aparência dos inputs para outline~~
- ✓ ~~Adicionar link do projeto no Github no ícone code~~

## Aula 59 - Deploy do front para nuvem (Curso DEV Full Stack)

---

- ✓ ~~Criar conta na vercel~~
- ✓ ~~Associar seu GitHub com a plataforma~~
- ✓ ~~Realizar deploy~~