

UNIVERSIDADE COMUNITÁRIA DA REGIÃO DE CHAPECÓ

Estudante: Lucas Guerra Rodrigues.

PROJETO DOCTORCALL – AGENDA DE CONSULTAS

1. DESENVOLVIMENTO E CONTRIBUIÇÕES

- **Página de Login (login.html):** Criada para permitir que o usuário acesse o sistema através de autenticação simples por sessão.
- **Página de informações gerais (sobre-nos.html):** desenvolvida para exibir informações gerais sobre a equipe fictícia de atendimento e agendamento;
- **Página de contato (contato.html):** desenvolvida para permitir que os usuários enviem mensagens, dúvidas ou solicitações ao sistema.
- **clienteController.js:** Controller que gerencia ações relacionadas aos clientes (cadastro, listagem, exclusão, etc.).
- **consultaController.js:** Controla o fluxo de consultas, como criação e listagem de agendamentos.
- **profissionalController.js:** Controller que lida com os dados e operações dos médicos do sistema.
- **clienteService.js:** Service que contém as lógicas de validação e manipulação dos dados dos clientes.
- **consultaService.js:** Service que gerencia a lógica de marcação, alteração e cancelamento de consultas.
- **profissionalService.js:** Service que lida com as operações sobre os profissionais, como associação de consultas e listagem.
- **requireAuth.js:** Middleware que verifica se o usuário está autenticado na sessão antes de permitir o acesso a rotas privadas.
- **simpleLogger.js:** Middleware que cria logs simples de requisições, registrando data, rota e método HTTP para monitorar acessos.

- **routes (app.js):** Arquivo que organiza e define as rotas públicas e privadas da aplicação, associadas ao seu respectivo controller.

2. LINK DO REPOSITÓRIO GIT

O repositório público do projeto GitHub está disponível neste link: <https://github.com/LuckyGRodrigues/Agenda-de-Consultas>.

3. DESAFIOS E APRENDIZADOS

Manter o código organizado entre as camadas, evitando que a lógica de negócio ficasse misturada com as rotas.

Aplicação prática de *middlewares*, já que foi a primeira experiência utilizando essa estrutura para controle de autenticação e registro de acessos, além do tratamento de erros com a ferramenta de versionamento de código Github, devido a conflitos nos primeiros commits.

Foram utilizadas IAs como apoio para revisar a estrutura de pastas e exemplos de middlewares, garantindo boas práticas de separação de responsabilidades. Todo o código foi escrito e adaptado manualmente após compreender os exemplos e adequar à proposta do projeto

Foram aprimorados os conhecimentos de aplicação prática da arquitetura **Controller >> Service >> Repository**, entender melhor o fluxo entre as camadas e a importância de aplicações de middlewares e o uso de sessões e autenticação simples, além de reforçar o versionamento no GitHub e a colaboração em equipe no mesmo repositório.

Se houvesse mais tempo para o desenvolvimento do projeto, seriam implementados:

- Um sistema de login mais completo, com perfis diferentes (paciente, médico e administrador);
- Mensagens de erro e feedbacks visuais mais detalhados para o usuário;
- Dinâmica para armazenar as consultas agendadas pelo próprio usuário.