PEC – ProEngControl

*Marcos Tullio Silva de Souza & Samuel Grontoski*

O PEC – ProEngControl é uma aplicação web, projetada especificamente para gerenciar de forma eficiente os chamados e as demandas da Engenharia de Testes. A plataforma permite o acompanhamento detalhado de cada chamado, desde a sua abertura até a resolução final. Seu objetivo é otimizar e automatizar os processos, reduzir o tempo de resposta e melhorar a qualidade dos serviços prestados pela Engenharia de Testes.

Domínio: [pec.hi-mix.com.br](http://pec.hi-mix.com.br/)

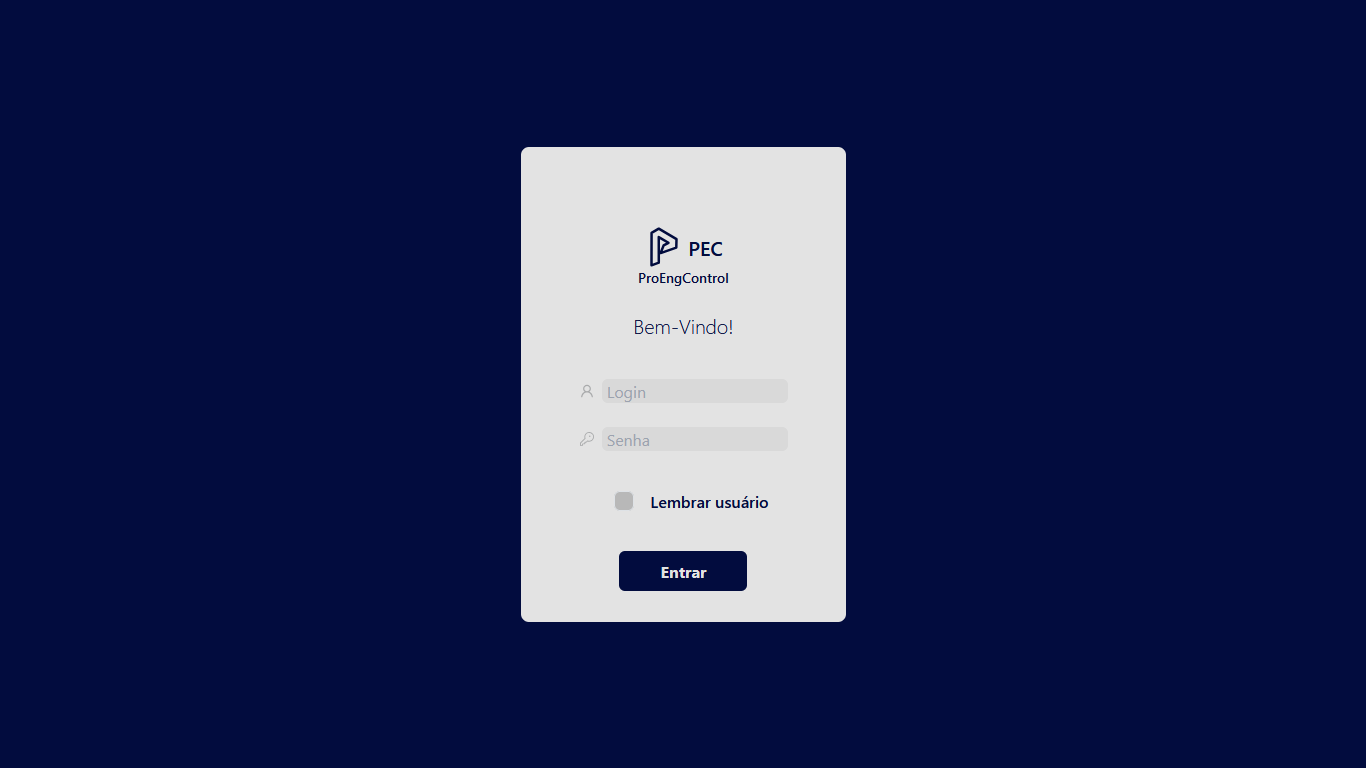
Versão Corrente: Beta

Tecnologias utilizadas

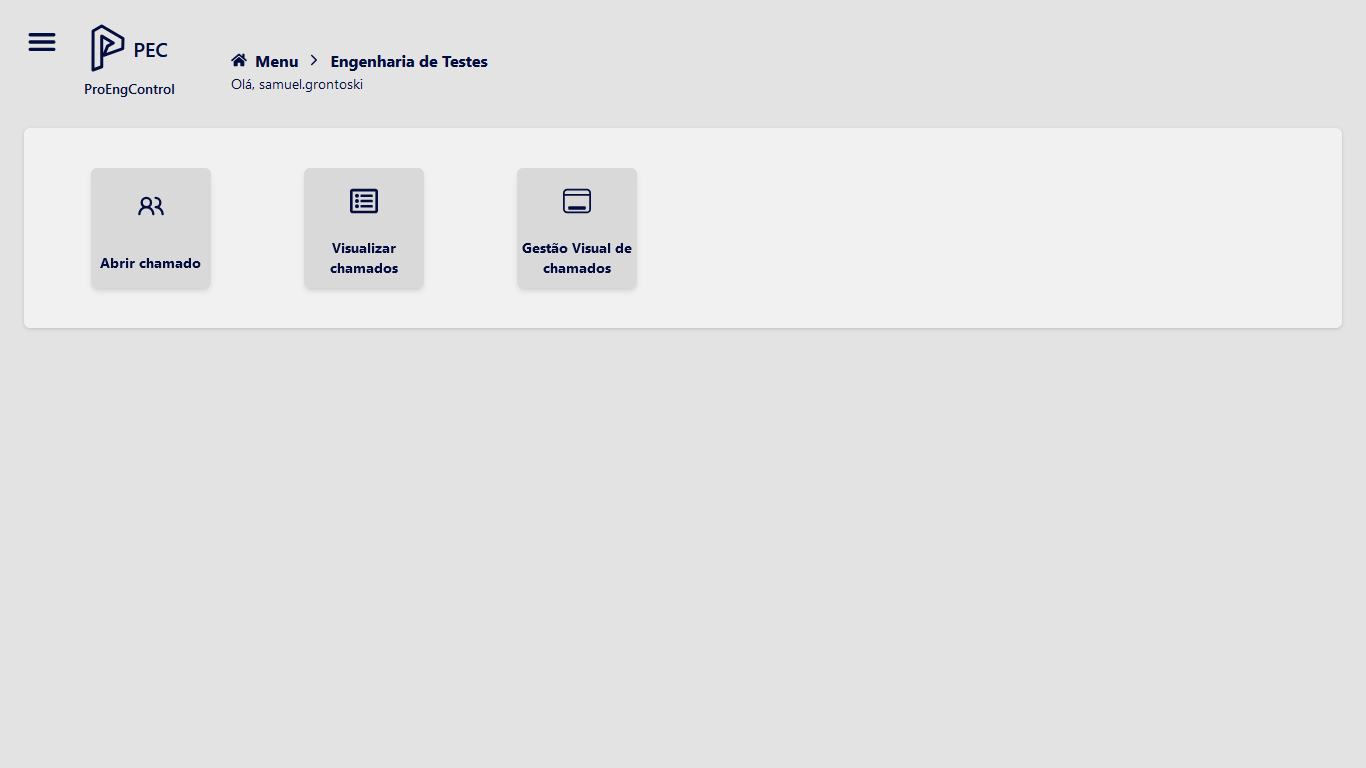
* **Frontend:** TypeScript, React.js, Vite, Tailwind CSS
* **Backend:** JavaScript, Node.js v20.11.1
* **Database:** MySQL
* **Gerenciamento de versão:** GitHub e Docker

**Funcionalidade do sistema**

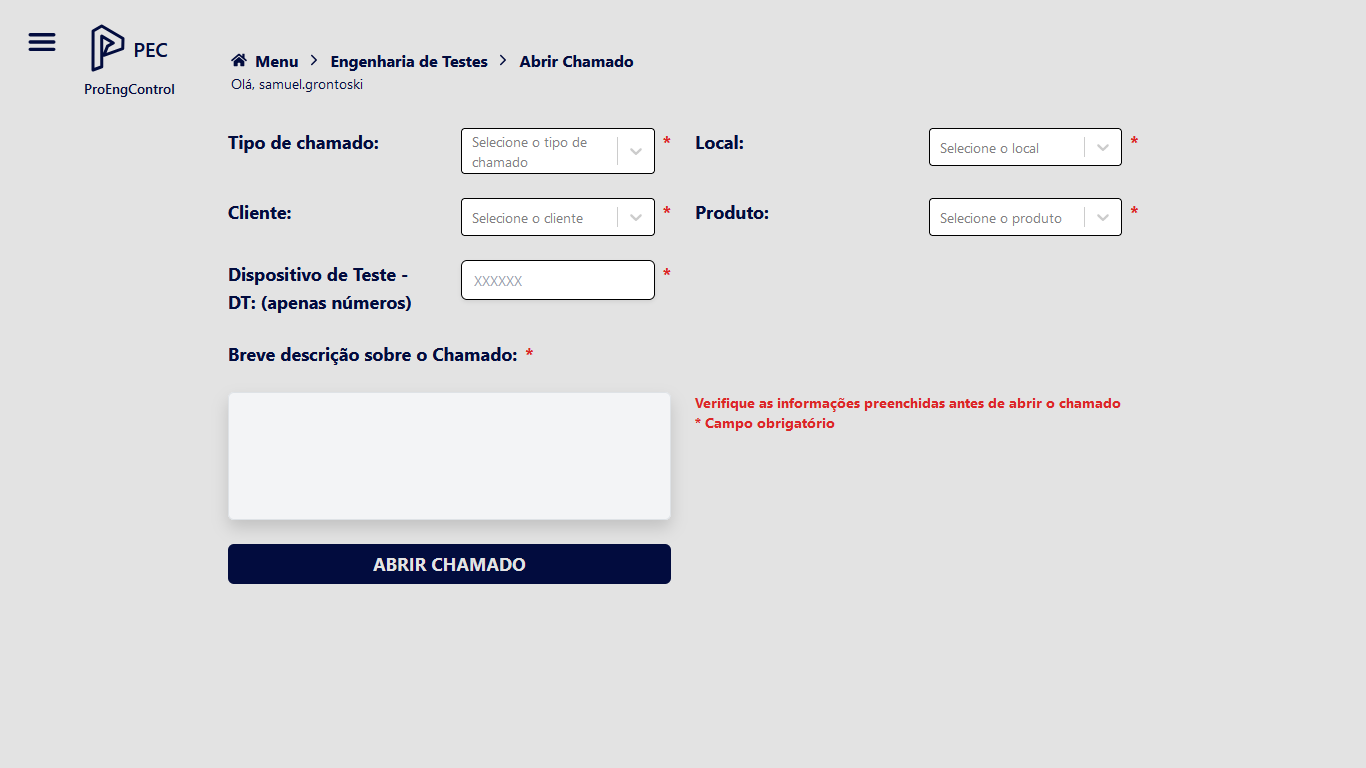
* **Login**

Primeira tela da aplicação, onde o usuário acessa o sistema com mesmo o login e senha utilizados no BetterCall.

* **Menu principal**

Menu onde são exibidas as funcionalidades do sistema, cada botão redireciona a sua função específica.

* **Abrir chamado**



Ao clicar no botão de ‘Abrir chamado’, o usuário será redirecionado para janela acima onde deverá informar o tipo de chamado, local, cliente, produto, DT da jiga e inserir uma descrição sobre o problema.

Ao clicar em ‘Abrir Chamado’, o chamado é criado e os campos de input são resetados.

* **Visualizar chamados**

Ao clicar no botão de ‘Visualizar chamado’, o usuário será redirecionado para a janela onde é possível visualizar os chamados em atendimento, aguardando atendimento e atendidos. Ao clicar em um dos chamados um card é aberto na tela mostrando os detalhes do chamado específico:



* **Gestão Visual de chamados**

Esta tela fica exibida na televisão na Engenharia de Testes. Mostrando os chamados que estão sendo atendidos, chamados atrasados, e indicadores diários, semanais e mensais. Está tela possuí acesso restrito apenas a administradores.

Gerenciamento de versão

**Pré-requisitos:** Ter o Git e Node.js instalados e configurados na máquina.

Atualmente é utilizado o GitHub para versionamento do código, no seguinte repositório:

<https://github.com/marcostulliosouza/proengcontrol-pec/>

Porém, por se tratar de um repositório privado, é necessário pedir permissão para acessar o mesmo.

Clonando o repositório

Abra o terminal e navegue até o diretório onde deseja clonar o repositório:

*cd /caminho/do/diretorio/de/destino*

Execute o comando git clone com a URL do repositório:

git clone https://github.com/marcostulliosouza/proengcontrol-pec/

O Git irá clonar o repositório para o seu diretório especificado. Para verificar se o clone foi bem-sucedido, navegue até o diretório clonado:

*cd proengcontrol-pec*

Após clonar o repositório, é necessário instalar os pacotes das bibliotecas utilizadas no projeto.

Na pasta principal do projeto, execute o comando:

*npm install*

Navegue até a pasta do back-end e execute o comando novamente:  
*cd back-end*

*npm install*

Repita o processo para o front-end:

*cd front-end*

*npm install*

Rodando a aplicação localmente

No terminal, acesse a pasta do back-end e execute o comando:

*cd back-end*

npm start

Navegue até a pasta do front-end e execute o comando:

*cd front-end*

npm run dev

Agora a aplicação estará rodando localmente em sua máquina.

Processo de Deploy

**Pré-requisitos:** Certifique-se de que a aplicação esteja funcionando corretamente na máquina local e de ter o Docker instalado. Verifique também se os Dockerfiles estão configurados adequadamente.

Criando imagens da aplicação

1. Na pasta raiz do projeto \proengcontrol-pec\, execute o comando:

*docker-compose build*

2. Em seguida, execute o comando:

*docker-compose* *up –d*

Estes comandos irão criar as imagens para o front-end e back-end da aplicação, e iniciar containers com essas imagens.

Enviando imagens para o Docker Hub

Para enviar (push) as imagens para o Docker Hub, primeiro atribua tags únicas a cada imagem:

Para o front-end:

*docker* *tag <ID\_da\_imagem\_frontend> nome\_usuário/nome\_repositório:frontend*

*docker push nome\_usuário/nome\_repositório:frontend*

Para o back-end:

*docker push nome\_usuário/nome\_repositório:backend*

*docker tag <ID\_da\_imagem\_backend> nome\_usuário/nome\_repositório:backend*

Puxando imagens do Docker Hub

Para clonar (pull) as imagens do Docker Hub, execute os seguintes comandos:

*docker pull nome\_usuário/nome\_repositório:frontend*

*docker pull nome\_usuário/nome\_repositório:backend*

Seguindo esses processos, você pode executar as imagens diretamente no Docker Desktop, sem se preocupar com problemas de compatibilidade ou versões de pacotes usados no desenvolvimento da aplicação.

A versão beta do PEC se encontra no seguinte repositório:

<https://hub.docker.com/r/samuelgrontoski01/proengcontrol-pec/tags>