

# Documentação Final: Sistema RegulaSense

## 1. Colaboradores e Supervisão

- Desenvolvimento Frontend & Integração: Paulo Henrique Rodrigues Nogueira
- Desenvolvimento Backend & Arquitetura: Marcos Vinicius Gundel da Silva
- Supervisão Acadêmica: Professor Eduardo Siqueira

O projeto foi guiado por `sprints` de desenvolvimento focadas na integração contínua das funcionalidades de autenticação e dos módulos de trabalho principais.

## 2. Arquitetura Técnica e Ambiente

Área	Tecnologia Principal	Dependências Chave	Função
Backend (Serviço de Usuários & FillSense)	Python / Django	Django REST Framework, SimpleJWT, Psycopg2	Lógica de Negócio, Autenticação, Gerenciamento de Dados e APIs de IA.
Banco de Dados	PostgreSQL	Docker Image	Persistência e armazenamento de dados de usuários e solicitações.
Frontend (Interface Web)	React	<code>react-router-dom</code> , <code>axios</code> , CSS Modular	Interface do Usuário (UI), Roteamento, Gerenciamento de Estado e Consumo de API.
Orquestração	Docker Compose	Dockerfile	Unifica o ambiente de desenvolvimento, garante a comunicação entre serviços ( <code>localhost:3000</code> e <code>:8000</code> ) e automatiza a inicialização.

## 3. Fluxos de Trabalho e Funcionalidades Críticas

### 3.1. Fluxo de Autenticação (Login)

**Status:** Totalmente integrado e funcional.

- **Ponto de Partida:** O usuário acessa a aplicação em `http://localhost:3000` e é levado para a rota `/login`.
- **Ação:** O componente `Login.jsx` envia uma requisição `POST` para `http://localhost:8000/api/auth/login/` via `axios`.
- **Sucesso:** O Backend retorna um **Token JWT** (chave de acesso). O Frontend armazena esse token no `localStorage` e redireciona o usuário para a rota `/home`.
- **Segurança:** O componente `ProtectedRoute.jsx` verifica a existência desse token antes de permitir o acesso a qualquer rota principal.

**3.2. Módulo FillSense (Formulários e Histórico)**

**Objetivo:** Gerenciar a criação e o acompanhamento de solicitações de procedimentos médicos.

Componente	Ação de Integração	Status do Código
Histórico ( <code>/solicitacoes</code> )	Faz uma chamada <code>GET</code> para <code>/api/fillsense/solicitacoes/</code> (com o token JWT no header) e renderiza os resultados em uma tabela dinâmica.	Concluído (Front-end)
Nova Solicitação ( <code>/nova-solicitacao</code> )	Componente com lógica complexa para <code>POST</code> na API principal ( <code>/api/fillsense/solicitacoes/</code> ) e chamadas simuladas de IA.	Concluído (Front-end)
Tabela de Dados	<b>Ajuste de UX:</b> Implementado o ícone de ação dinâmico ( <code>fa-eye</code> ou <code>fa-pen-to-square</code> ) que muda de acordo com o <code>status</code> da solicitação.	Concluído (Front-end)
Formulário	Adicionado o campo <b>"Número do SUS"</b> e ajustado o formato do JSON de envio para se alinhar com as novas regras de dados do Backend.	Concluído (Front-end)

**4. Ajustes de Interface e Usabilidade (Finalização Frontend)**

Diversos refinamentos foram feitos para garantir uma experiência de usuário limpa e profissional:

- **Layout:** Estilos globais (`index.css`) corrigidos para garantir que o formulário de login centralize corretamente e que a tela não fique em branco.
- **Navegação:** Removida a barra de navegação global do `App.js` (que causava o "retângulo azul") e centralizada toda a navegação nos componentes (Home, Solicitacoes, RegulaFlow). Os links foram corrigidos para apontar para as rotas React (ex: `/regulaflow`).
- **Rodapé:** Implementação de um rodapé moderno e informativo (`Footer.jsx`) com a logo do **CIIA**, informações de contato e links de parcerias e redes sociais (com o tamanho da logo ajustado para harmonia).
- **Limpeza da UI de Login:** Os links de "Criar conta" e "Esqueceu a senha?" foram removidos da página de login para simplificar a interface.

## 5. Fluxo de Trabalho Git e Colaboração

- **Fluxo de Trabalho:** O desenvolvimento seguiu o modelo de branch `developer`, garantindo que a branch `main` permanecesse sempre estável.
- **Sincronização:** Foi estabelecida a rotina de `git pull origin main` e `git merge main` para manter a branch `developer` atualizada e evitar grandes conflitos.
- **Resolução de Conflitos:** Foi resolvido com sucesso um conflito de merge no `NovaSolicitacao.jsx`, garantindo a unificação do seu trabalho com as atualizações do seu colega.