Documentação do Projeto SA2 Final - Daniel, Matheus e Kelvin

**1. Levantamento de Requisitos**

**Requisitos Funcionais:**

- Cadastro de Responsáveis: Permite a gestão completa dos responsáveis pela administração de ambientes e ativos patrimoniais, incluindo operações de cadastro, edição, exclusão e consulta.

- Cadastro de Ambientes: Permite o gerenciamento dos ambientes onde estão localizados os ativos patrimoniais, com funcionalidades similares de CRUD (Create, Read, Update, Delete).

- Cadastro de Ativos Patrimoniais: Permite a gestão dos ativos patrimoniais pertencentes aos ambientes, oferecendo operações completas de CRUD.

**Requisitos Não Funcionais:**

- Segurança: Implementação de autenticação e autorização para proteger os endpoints da API contra acessos não autorizados.

- Desempenho: Garantia de performance adequada mesmo em ambientes com grande volume de dados.

**Compreensão das Necessidades da Unidade Escolar do SENAI-SP:**

- Nosso objetivo foi criar uma solução que resolvesse eficientemente os desafios de gestão de ativos patrimoniais da escola, que antes era centralizado por uma pessoa e todo o controle dos ativos era feito via planilha do Excel.

**2. Design do Banco de Dados**

**Modelo de Dados:**

- Responsável: Representa os responsáveis pela administração dos ambientes.

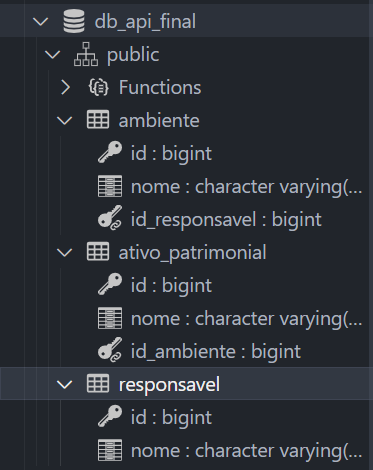
- Ambiente: Representa os locais onde os ativos patrimoniais estão situados.

- Ativo Patrimonial: Representa os bens patrimoniais que são geridos dentro dos ambientes.

**Relacionamentos:**

- Responsável <-> Ambiente: Relacionamento um para n, onde um responsável pode gerenciar vários ambientes.

- Ambiente <-> Ativo Patrimonial: Relacionamento um para n, onde um ambiente pode conter vários ativos patrimoniais.



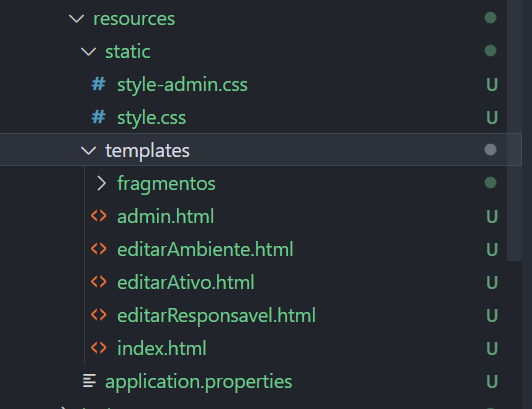
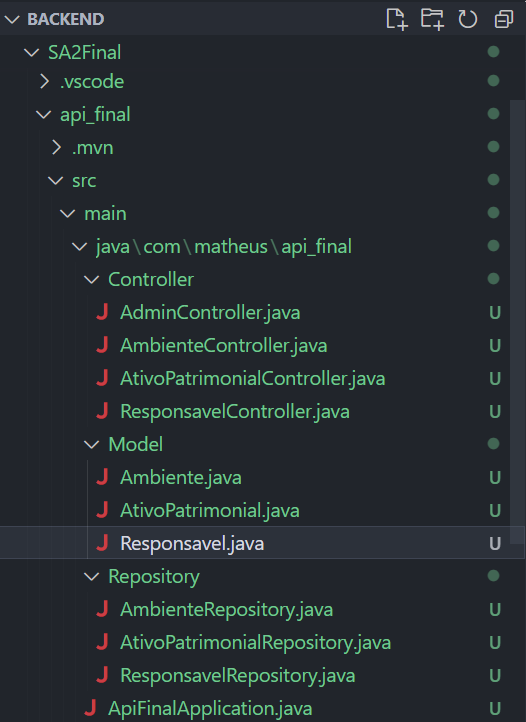
**3. Implementação da API**

**Funcionalidades dos Endpoints:**

- A API foi desenvolvida para oferecer um conjunto completo de operações CRUD para as entidades Responsavel, Ambiente e Ativo Patrimonial. Seguimos as melhores práticas de desenvolvimento RESTful, utilizando métodos HTTP apropriados e códigos de status para feedback adequado.

**Boas Práticas de Codificação:**

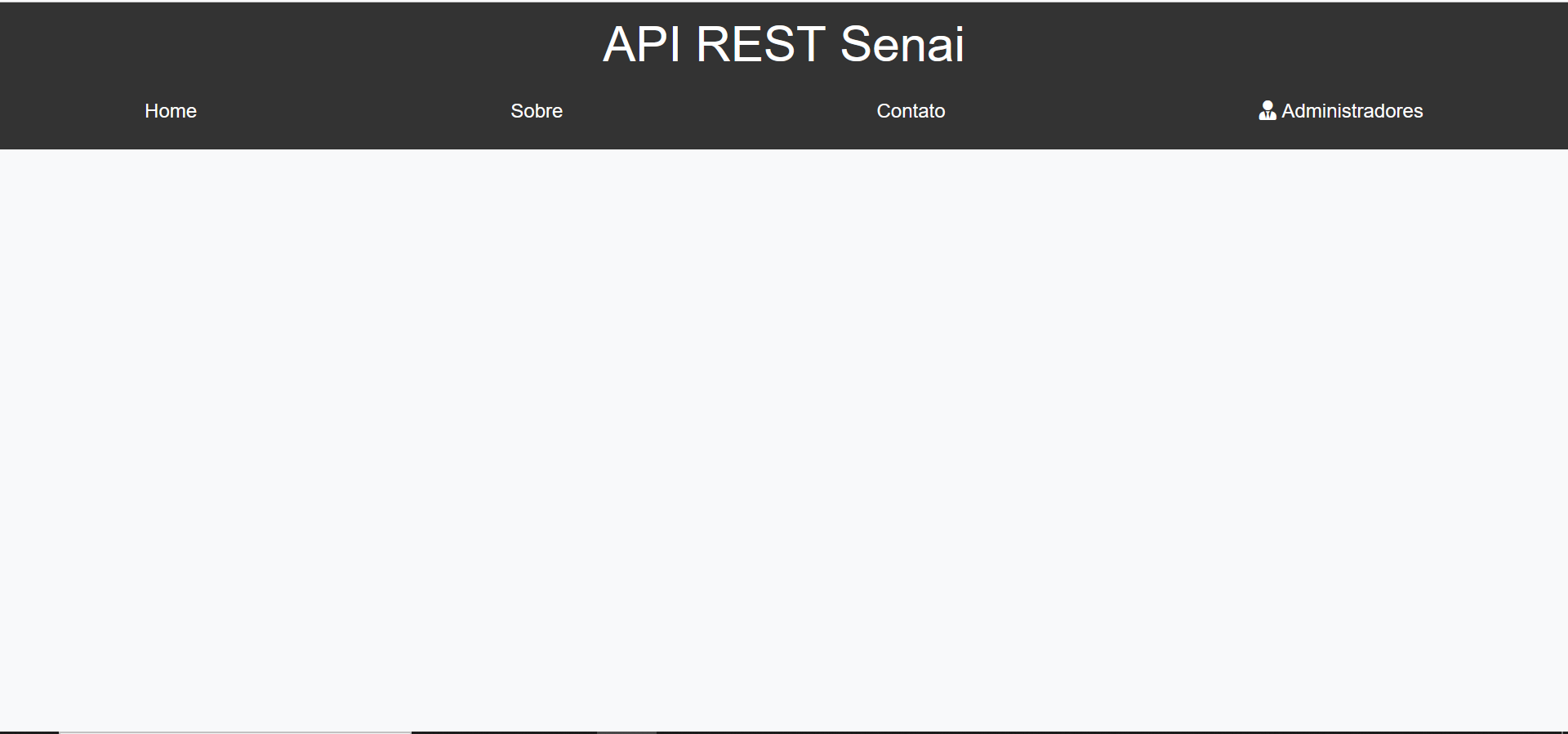
- Adotamos a arquitetura MVC (Model-View-Controller) para uma separação clara de responsabilidades e utilizamos injeção de dependência para gerenciar as dependências entre os componentes da aplicação de forma eficiente.

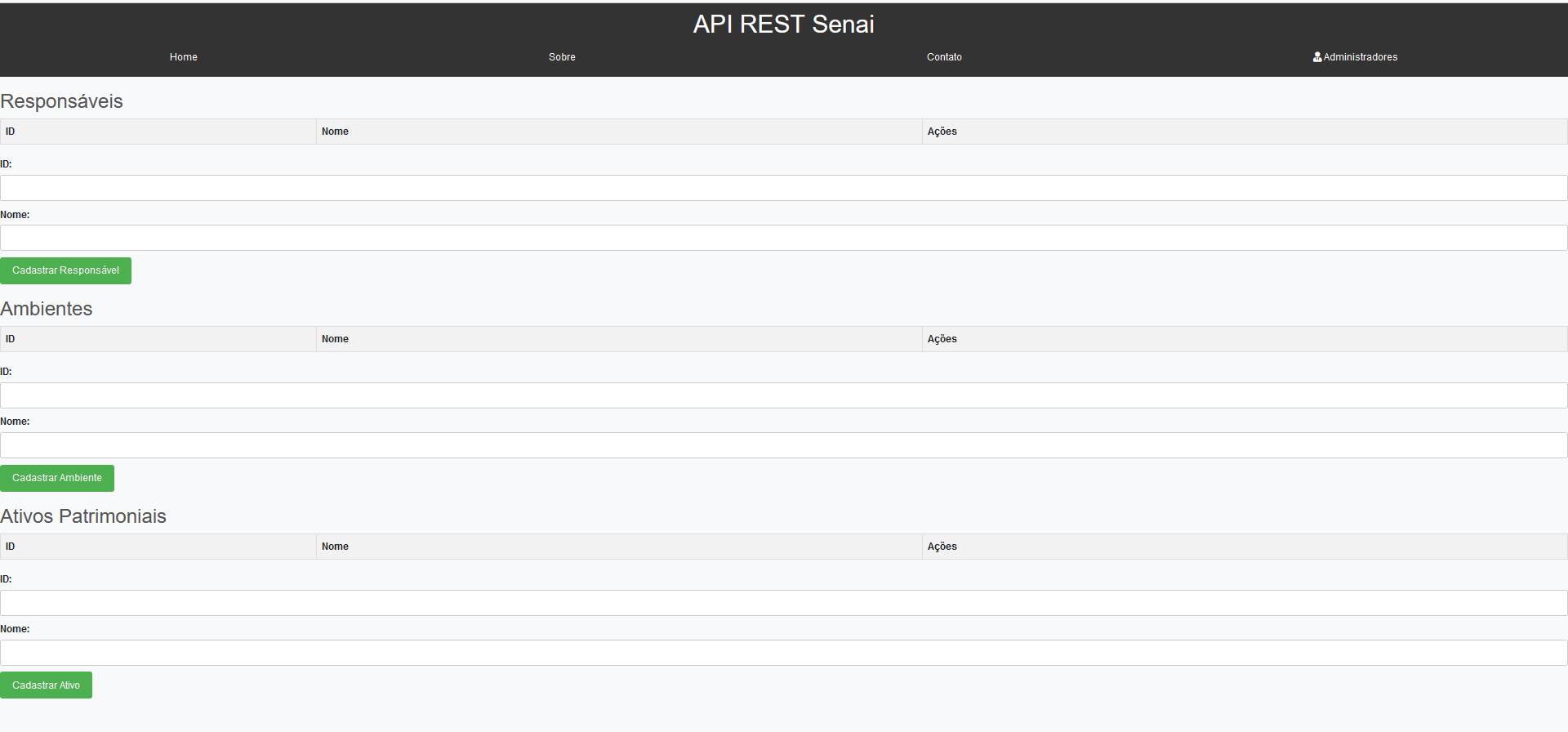


**4. Desenvolvimento da Interface de Usuário**

**Interface de Usuário:**

- Criamos uma interface intuitiva que facilita o cadastro, edição e exclusão de responsáveis, ambientes e ativos patrimoniais. A interface interage diretamente com a API, permitindo uma experiência completa de gerenciamento de dados.





**5. Testes**

**Tipos de Testes Realizados:**

- Realizamos testes unitários para validar componentes individuais do código e testes de integração para garantir a interação adequada entre os componentes da aplicação e a base de dados. Nosso objetivo foi cobrir uma ampla gama de cenários para assegurar a robustez do sistema.

**6. Deploy e Disponibilidade**

**Configuração do Ambiente de Produção:**

- Configuramos o ambiente de produção utilizando PostgreSQL como banco de dados, garantindo uma configuração adequada para suportar a aplicação em produção. Mantemos a disponibilidade contínua do sistema após o deploy, minimizando interrupções significativas.

**7. Atendimento aos Requisitos do Cliente**

**Alinhamento com Expectativas do Cliente:**

- Nosso sistema atendeu aos requisitos específicos de gestão de patrimônio da unidade escolar do SENAI-SP. Durante o desenvolvimento, adaptamos o sistema para refletir a evolução contínua dos requisitos e incorporar melhorias conforme necessário.

**8. Usabilidade e Experiência do Usuário**

**Usabilidade da Interface:**

- A interface foi projetada para ser amigável, com navegação intuitiva e responsividade para suportar diferentes dispositivos. Nosso foco foi garantir uma experiência de usuário agradável e eficiente.

**9. Comunicação e Colaboração em Grupo**

**Colaboração Efetiva:**

- Durante o desenvolvimento, mantivemos uma comunicação clara e colaboração eficaz entre os membros da equipe. Documentamos o progresso do projeto e as decisões tomadas para garantir transparência e alinhamento.

**10. Inovação e Criatividade**

**Soluções Criativas Implementadas:**

- Implementamos funcionalidades além dos requisitos mínimos, como a utilização de Thymeleaf para a criação de templates HTML dinâmicos. Nosso foco foi aplicar pensamento crítico para encontrar soluções originais e resolver desafios técnicos de forma eficaz.

- Esta documentação reflete nosso compromisso em desenvolver uma solução robusta e eficiente para atender às necessidades específicas da unidade escolar do SENAI-SP, garantindo qualidade, segurança e usabilidade em todas as etapas do projeto.