

Projeto: **Unimpact**

Documentação – Código

Tecnologias Utilizadas:

Backend:

- Java: Linguagem de programação orientada a objetos utilizada para a implementação da lógica de negócios do projeto.
- Spring Boot: Framework baseado em Java, amplamente utilizado para criar aplicações web robustas e escaláveis com configuração mínima. No projeto, ele gerencia a camada de backend, incluindo segurança, persistência de dados e gerenciamento de dependências.

Frontend:

- React: Biblioteca JavaScript para construção de interfaces de usuário. No projeto, o React é utilizado para criar a camada de frontend, interagindo com o backend via API REST.

Banco de Dados:

- PostgreSQL: Sistema de gerenciamento de banco de dados relacional (SGBD). Ele é utilizado para armazenar as informações de usuários, propostas e outras entidades relacionadas ao sistema.
- Flyway: Ferramenta de migração de banco de dados, utilizada para versionar e aplicar alterações nas tabelas de forma incremental e segura.
- pgAdmin4: Interface gráfica para administração e gerenciamento do banco de dados PostgreSQL. Utilizado para visualizar e manipular dados diretamente.

Containerização:

- Docker: Ferramenta que cria contêineres para isolar e padronizar o ambiente de desenvolvimento e produção. No projeto, Docker é utilizado para configurar o ambiente de execução da aplicação.
- **Docker Desktop**: Versão do Docker que facilita a execução de contêineres em máquinas locais, permitindo o desenvolvimento local e integração com ferramentas como o Docker Compose.

Estrutura de pastas:

/database-models:

- Nessa pasta estão presentes o modelo conceitual e o modelo lógico do banco de dados, que é a base do projeto. Eles estão salvos no formato *.brM3* que é o formato do brModelo, onde foram criados os modelos.

/demo:

Essa pasta é onde está presente todo backend do projeto, a partir de uma inicialização do Springboot. Dentro da pasta demo estão as seguintes pastas:

- /demo/src/main/java:
 - /controller: Contém as classes responsáveis por lidar com as requisições HTTP e enviar respostas adequadas.
 - /enum1: Contém enumeradores, como o CargoEnum.java, que define constantes específicas usadas no projeto.
 - /models: Contém as classes de entidade, que representam os dados persistidos no banco de dados. Exemplo: Usuario.java, Proposta.java, Endereco.java são exemplos de classes de modelo que correspondem a tabelas no banco de dados.
 - /repositories: Armazena as interfaces que estendem o Spring Data JPA, facilitando a comunicação com o banco de dados.
 - /service:
 - /BusinessObject: Contém as classes que implementam a lógica de negócios. Cada classe aqui implementa as regras específicas de suas entidades, como PropostaBusinessObject.java para lógica de negócios relacionada às propostas.
 - /DataProvider: Implementa as consultas ao banco de dados, recuperando dados e os preparando para uso. Por exemplo, PropostaDataProvider.java faz a busca de propostas no banco de dados.
 - /Facade (a ser implementada): Centralizará as chamadas para as classes BusinessObject e DataProvider, encapsulando a complexidade e fornecendo uma interface simples para as outras partes da aplicação.
- /demo/src/main/resources:
 - /db: Contém arquivos de configuração, como application.properties, e templates para e-mails ou páginas HTML.
 - /migration: Scripts SQL gerenciados pelo Flyway para criar e alterar as tabelas do banco de dados.

/documents:

Essa pasta é onde a documentação do projeto está presente. Diagrama de Classes, documentação sobre código, etc.