

Projeto feedback.edu - Aplicação Completa (Back-end + Protótipo Front-end)

1. Visão Geral do Projeto

Feedback.edu é uma plataforma robusta projetada para aprimorar a comunicação em ambientes acadêmicos, permitindo que alunos forneçam feedbacks anônimos sobre disciplinas e professores. Este repositório contém o projeto completo, que abrange desde a concepção e análise de requisitos até a implementação de uma solução funcional.

A entrega consiste em duas partes principais:

- **API Back-end:** Desenvolvida em **Java** com o framework **Spring Boot**, seguindo uma arquitetura em camadas bem definida para garantir manutenibilidade, testabilidade e escalabilidade, conforme detalhado no Documento de arquitetura de software.
- **Protótipo Front-end:** Uma interface de usuário interativa desenvolvida com **HTML, CSS e JavaScript**, que simula o fluxo completo da aplicação como uma Single-Page Application (SPA), representando fielmente os protótipos e requisitos definidos no Documento de requisitos.

2. Arquitetura e Design

O sistema foi projetado seguindo o padrão **Arquitetura em Camadas**, garantindo uma clara separação de responsabilidades:

1. **Apresentação (Front-end):** Interface com o usuário, desenvolvida como uma SPA e servida diretamente pelo back-end.
2. **Serviço/Aplicação (Controllers):** Expõe os endpoints da API REST, recebendo requisições HTTP e orquestrando as operações.
3. **Domínio (Services):** Contém toda a lógica de negócio, regras de validação e o núcleo do sistema.
4. **Persistência (Repositories):** Abstrai o acesso ao banco de dados utilizando os padrões de projeto **Repository** e **DAO (Data Access Object)**, implementados com Spring Data JPA

3. Estrutura do Projeto

O código-fonte está organizado de forma a refletir a arquitetura do sistema:

```
api/
├── src/
│   ├── main/
│   │   ├── java/
│   │   │   └── com/feedbackedu/
│   │   │       ├── controller/ # Camada de Aplicação (API REST)
│   │   │       ├── domain/    # Entidades do negócio
│   │   │       ├── dto/       # Data Transfer Objects
│   │   │       ├── repository/ # Camada de Persistência
│   │   │       ├── service/    # Camada de Domínio (Lógica de Negócio)
│   │   │       └── Application.java
│   │   └── resources/
│   │       ├── static/        # Camada de Apresentação (HTML, CSS, JS)
│   │       └── application.properties
│   └── test/
```

4. Como Executar o Projeto Completo

Este guia passo a passo permitirá que qualquer pessoa execute a aplicação completa (back-end e front-end) em um ambiente local a partir do código-fonte.

Pré-requisitos

- **Git:** Para clonar o repositório.
- **Java JDK 17** ou superior instalado [cite: README.md].
- **XAMPP:** Para um ambiente de desenvolvimento rápido com Apache e MySQL. Outro servidor MySQL também pode ser utilizado [cite: README.md].

Passo 1: Obter o Código-Fonte

Clone o repositório do GitHub para a sua máquina local (ou descompacte o arquivo .zip em uma pasta).

git clone https://github.com/FelipeSilva96/Engenharia_de_software_2.git

cd Engenharia_de_software_2/api

Passo 2: Configurar o Banco de Dados com XAMPP

A aplicação precisa de um banco de dados MySQL para funcionar.

1. **Inicie o XAMPP:** Abra o painel de controle do XAMPP e inicie os módulos **Apache** e **MySQL**.
2. **Acesse o phpMyAdmin:** No painel do XAMPP, na linha do MySQL, clique no botão **Admin**.
3. **Crie o Banco de Dados:**
 - Na interface do phpMyAdmin, vá para a aba "Bases de Dados".
 - No campo "Criar base de dados", digite o nome `feedbackedu_db` e clique em "Criar"

Importante: Crie apenas o banco de dados. **Não crie nenhuma tabela.** A aplicação, através do Hibernate, criará toda a estrutura de tabelas automaticamente na primeira vez que for executada, graças à configuração no arquivo `application.properties` [cite: README.md, `application.properties`].

Passo 3: Configurar as Credenciais (se necessário)

Por padrão, o XAMPP utiliza o usuário `root` sem senha. O projeto já está configurado para esta situação [cite: `application.properties`]. Se a sua configuração do MySQL for diferente, edite o seguinte arquivo:

- **Arquivo:** `api/src/main/resources/application.properties`

Altere as linhas:

```
spring.datasource.username=seu_usuario_mysql  
spring.datasource.password=sua_senha_mysql
```

Passo 4: Executar a Aplicação

Com o banco de dados pronto, inicie o servidor da aplicação.

1. **Abra um Terminal** (Prompt de Comando, PowerShell, Git Bash, etc.).

Navegue até a pasta `api`, que é a raiz do projeto Spring Boot.

Exemplo de caminho

```
cd caminho/para/Engenharia_de_software_2/api
```

2. **Execute o comando de inicialização** usando o Maven Wrapper, que já está incluído no projeto

Para Windows:

```
.\mvnw.cmd spring-boot:run
```

Para Linux/Mac:

```
./mvnw spring-boot:run
```