

CENTRO UNIVERSITÁRIO UNIRUY WYDEN
CAMPUS IMBUÍ - SALVADOR

ARIANE OLIVEIRA DE SOUZA - RA: 201951445384
BRUNO MENEZES DOS SANTOS - RA 202209053991

APLICAÇÃO GERENCIAMENTO DE ALUNOS EM UMA ACADEMIA - WAYFIT

SALVADOR

2024

ARIANE OLIVEIRA DE SOUZA
BRUNO MENEZES DOS SANTOS

APLICAÇÃO GERENCIAMENTO DE ALUNOS EM UMA ACADEMIA - WAYFIT

Avaliação Final da Disciplina de Programação
Orientada a Objetos em Java.

Salvador
2024

1. Introdução

Este projeto de gerenciamento de alunos foi desenvolvido com o objetivo de facilitar a administração dos alunos matriculados em uma academia. A aplicação permite que os administradores da academia possam gerenciar eficientemente o cadastro, consulta, listagem e remoção de alunos, proporcionando uma solução prática e intuitiva para o controle de informações essenciais dos alunos.

2. Objetivo

O principal objetivo do sistema é centralizar e simplificar as operações administrativas relacionadas aos alunos, garantindo que todas as informações relevantes estejam facilmente acessíveis e possam ser manipuladas de forma segura e eficiente.

3. Funcionalidades Principais

Requisitos Funcionais

Cadastro de Alunos (RF01)

Permite o registro de novos alunos no sistema, armazenando informações como nome, idade e matrícula.

Listagem de Alunos (RF02)

Exibe uma lista completa de todos os alunos cadastrados na academia.

Consulta de Alunos por Matrícula (RF03)

Permite a busca de um aluno específico utilizando a sua matrícula.

Remoção de Alunos (RF04)

Permite a remoção de um aluno do sistema utilizando a sua matrícula.

Requisitos Não Funcionais

Desempenho (RNF01)

O sistema deve realizar operações de adição, listagem, busca e remoção de alunos rapidamente, sem atrasos perceptíveis, garantindo uma experiência de uso eficiente para os administradores.

Usabilidade (RNF02)

A interface do sistema deve ser clara e intuitiva, facilitando a utilização por parte dos administradores, mesmo aqueles com pouca experiência técnica.

Segurança (RNF03)

O sistema deve garantir que operações críticas, como adição e remoção de alunos, sejam realizadas de forma segura, prevenindo duplicações de matrícula e mantendo a integridade dos dados.

Manutenibilidade (RNF04)

O sistema deve ser de fácil manutenção e evolução, permitindo a adição de novas funcionalidades ou ajustes de forma simples e sem grandes impactos na operação existente.

4. Modelo Conceitual e Lógico: Diagrama Entidade - Relacionamento

No projeto de gerenciamento de alunos para uma academia, as principais entidades envolvidas são as classes 'Aluno', 'AlunoController' e 'Repositorio'. Abaixo está uma descrição detalhada de cada uma dessas classes:

Classe Aluno

Representa um aluno da academia, com atributos básicos necessários para identificar e gerenciar os alunos.

Atributos

nome (String): O nome do aluno.

idade (int): A idade do aluno.

matricula (String): O número de matrícula do aluno, usado como identificador único.

AlunoController

Atua como o controlador principal, gerenciando a lógica de negócios associada ao gerenciamento dos alunos. Facilita a interação entre a interface do usuário e a camada de persistência.

Atributos

- 'repositorio' (Repositorio): Instância da classe 'Repositorio' que gerencia a persistência dos dados.

Métodos

adicionarAluno(Aluno aluno): Adiciona um novo aluno ao repositório.

listarAlunos(): Retorna a lista de todos os alunos cadastrados.

buscarAluno(String matricula): Busca um aluno específico pela matrícula.

removerAluno(String matricula): Remove um aluno específico pela matrícula.

Classe 'Repositorio'

Gerencia a persistência dos dados dos alunos, utilizando uma lista em memória para armazenar os objetos 'Aluno'.

Atributos

alunos (ArrayList<Aluno>): Lista de alunos cadastrados.

Métodos

adicionar(Aluno aluno): Adiciona um aluno à lista.

listar(): Retorna a lista de todos os alunos.

buscar(String matricula): Busca um aluno específico pela matrícula.

remover(String matricula): Remove um aluno específico pela matrícula.

5. Tabelas do Banco de Dados

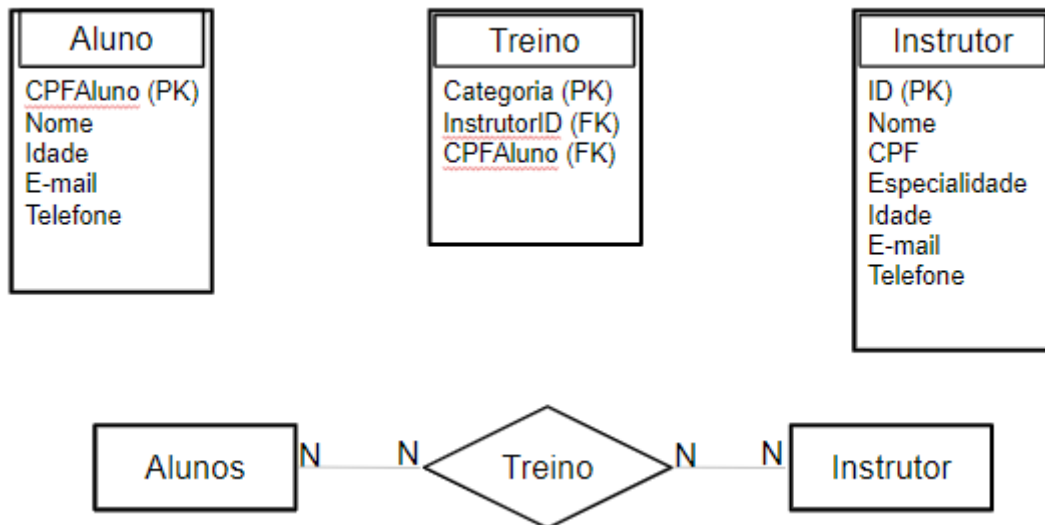


Figura 1

```
CREATE TABLE Aluno (
  CPFAluno INT PRIMARY KEY,
  Nome VARCHAR(255),
  Telefone VARCHAR(20),
  Email VARCHAR(255)
  Idade INT;
)
```

```
CREATE TABLE Instrutor (
  IDInstrutor SERIAL PRIMARY KEY,
  Nome VARCHAR(255),
  Especialidade VARCHAR(255),
  Telefone VARCHAR(20),
  Email VARCHAR(255)
  CPF VARCHAR(20),
  Idade INT;
)
```

```
CREATE TABLE Treino (
  Categoria VARCHAR(255) PRIMARY KEY,
  InstrutorID INT,
  CPFAluno INT,
  FOREIGN KEY (InstrutorID) REFERENCES Instrutor(ID),
  FOREIGN KEY (CPFAluno) REFERENCES Aluno(ID);
)
```

Figura 2

6. Aplicação Protótipo

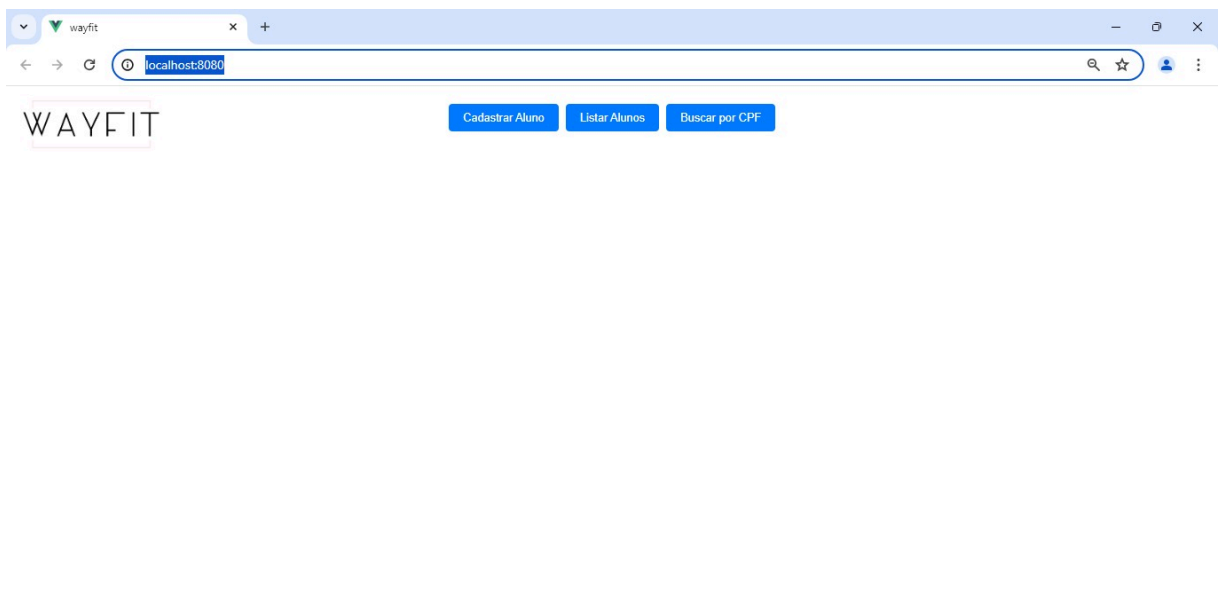


Figura 3 - Página Inicial

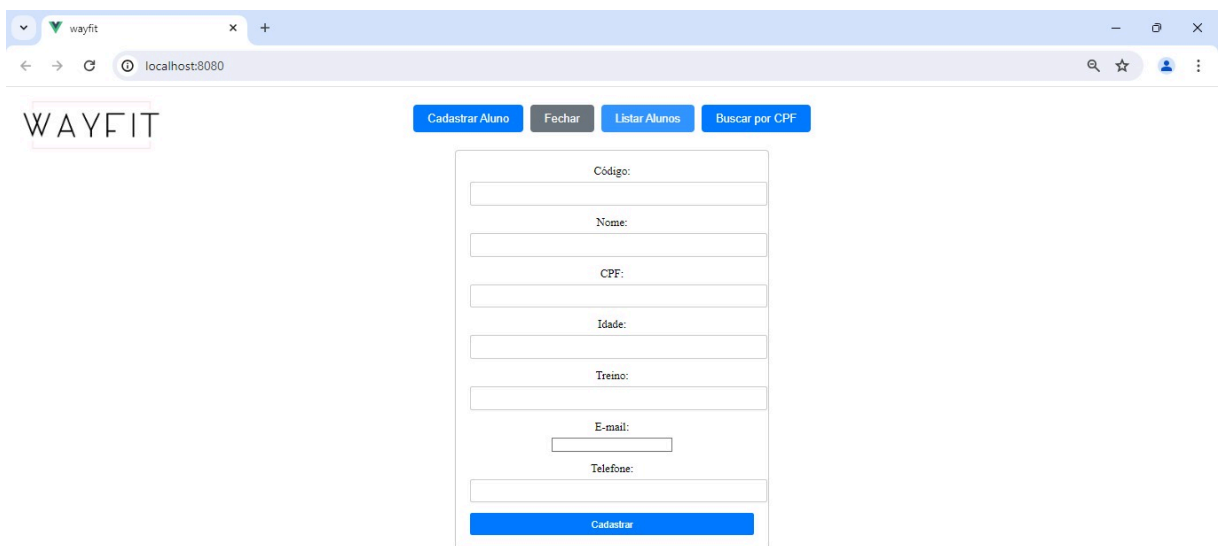


Figura 4 - Cadastro do Aluno



Figura 5 - Consulta ou exclusão por CPF