
Documentação do Sistema: Gerenciamento Estudantil

Alunas: Júlia Clovandi e Alice Tavares

1. Descrição Geral

O sistema **Gerenciamento Estudantil** foi desenvolvido em Java com a finalidade de gerenciar dados acadêmicos de alunos, professores e cursos. O sistema oferece funcionalidades de cadastro, consulta, edição, exclusão e geração de relatórios, além de permitir a associação entre alunos, professores e cursos.

A interface gráfica foi construída utilizando a biblioteca **Swing** do Java, proporcionando uma navegação intuitiva. Os dados são persistidos em um **banco de dados**, garantindo a integridade e segurança das informações entre as execuções do programa.

2. Modelagem de Dados

2.1. Classes Principais

- **Pessoa (Classe Abstrata)**

A classe `Pessoa` é a superclasse das classes `Estudante` e `Professor`, possuindo os atributos comuns `nome` e `idade`. Além disso, define o método abstrato `exibirDados()`, que será implementado nas subclasses.

- **Estudante (Subclasse de Pessoa)**

A classe `Estudante` herda de `Pessoa` e adiciona o atributo `matricula`. Implementa o método `exibirDados()` para exibir as informações do estudante.

- **Professor (Subclasse de Pessoa)**

A classe `Professor` herda de `Pessoa` e adiciona o atributo `especialidade`. Também implementa o método `exibirDados()` para exibir as informações do professor.

- **Curso**

A classe `Curso` representa um curso acadêmico, com os atributos `nomeCurso`, `cargaHoraria` e `professor` (associando um curso a um único professor). Além disso, permite a manipulação de cursos e suas associações com alunos e professores.

- **GerenciadorDeBancoDeDados**

A classe `GerenciadorDeBancoDeDados` é responsável pela interação com o banco de dados. Ela contém métodos para conectar, realizar consultas, inserções, atualizações e exclusões, além de manter a persistência dos dados em um banco relacional, como MySQL.

- **Relatorios**

A classe `Relatorios` permite gerar relatórios, como o relatório de alunos matriculados nos cursos e o relatório de professores associados aos cursos. Utiliza consultas ao banco de dados para coletar as informações necessárias para gerar os relatórios.

- **JanelaPrincipal (Interface Gráfica)**

A classe `JanelaPrincipal` é a interface gráfica principal do sistema. Utiliza o **Swing** para criar uma interface amigável, permitindo a navegação entre as opções do sistema e interação com os dados armazenados no banco.

2.2. Relacionamento entre as Classes

- A **classe Pessoa** é a superclasse das classes `Estudante` e `Professor`, sendo que ambas compartilham os atributos `nome` e `idade`.
- **Estudante** pode estar matriculado em vários **Cursos**, e a associação é gerida por meio de tabelas de relacionamento no banco de dados.
- **Professor** pode estar associado a vários **Cursos**, sendo também gerido por tabelas de relacionamento no banco de dados.
- **Curso** tem um **Professor** responsável, e pode ter muitos **Estudantes** matriculados. O relacionamento entre os cursos e os professores, assim como entre os cursos e os alunos, é mantido em tabelas de junção.

3. Fluxo do Sistema

1. Tela Inicial:

Ao iniciar o programa, a tela inicial exibe um menu com as opções:

- **Aluno:** Acessa o menu de gerenciamento de alunos.
- **Professor:** Acessa o menu de gerenciamento de professores.
- **Curso:** Acessa o menu de gerenciamento de cursos.
- **Sair:** Encerra a aplicação.

2. Menu de Aluno:

O menu de alunos permite as seguintes ações:

- **Cadastrar Aluno:** Solicita nome, idade e matrícula do aluno e grava as informações na tabela `alunos` do banco de dados.
- **Consultar Aluno:** Permite buscar um aluno por nome ou matrícula e oferece opções de **Editar** (alterar nome e idade) ou **Excluir**.

- **Sair:** Retorna ao menu principal.

3. Menu de Professor:

O menu de professores permite as seguintes ações:

- **Cadastrar Professor:** Solicita nome e especialidade do professor e grava as informações na tabela `professores` do banco de dados.
- **Consultar Professor:** Permite buscar um professor por nome ou especialidade e oferece opções de **Editar** (alterar nome e especialidade) ou **Excluir**.
- **Sair:** Retorna ao menu principal.

4. Menu de Curso:

O menu de cursos permite as seguintes ações:

- **Cadastrar Curso:** Solicita nome e carga horária do curso e grava as informações na tabela `cursos` do banco de dados.
- **Consultar Curso:** Permite buscar um curso por nome e oferece opções de **Editar** (alterar nome e carga horária) ou **Excluir**.
- **Vinculação:** Permite associar alunos e professores aos cursos e grava as vinculações nas tabelas de relacionamento (`matriculas` para alunos e cursos, e `associacoes` para professores e cursos).
- **Sair:** Retorna ao menu principal.

5. Geração de Relatórios:

O sistema oferece relatórios que podem ser gerados:

- **Relatório de Estudantes:** Exibe uma lista de estudantes e os cursos nos quais estão matriculados.
- **Relatório de Professores:** Exibe uma lista de professores e os cursos que estão associados.

4. Funcionalidades

- **Cadastro:** Permite a criação de registros para alunos, professores e cursos, com a persistência desses dados no banco de dados.
- **Consulta:** Permite a busca por alunos, professores e cursos a partir de critérios específicos (nome ou matrícula).
- **Edição:** Permite a atualização dos dados dos alunos, professores e cursos, mantendo a consistência dos dados no banco.
- **Exclusão:** Permite a remoção de registros, garantindo que o banco de dados seja atualizado de forma adequada.
- **Relatórios:** Geração de relatórios sobre os alunos matriculados e os professores associados a cursos, utilizando consultas SQL no banco de dados.

- **Persistência de Dados:** Todos os dados (alunos, professores, cursos e vinculações) são armazenados em um banco de dados relacional, o que garante a integridade e consistência das informações entre as execuções do programa.

5. Interface Gráfica (Swing)

A interface gráfica foi construída utilizando o **Swing**, oferecendo uma navegação simples e intuitiva entre as funcionalidades do sistema. A interface contém os seguintes elementos principais:

- **Menus** para acessar as diferentes opções de cadastro, consulta, edição e geração de relatórios.
- **Formulários** para inserção e edição de dados de alunos, professores e cursos.
- **JTable** ou **JList** para exibição de dados, como lista de alunos, professores e cursos.
- **Botões** para execução das ações, como cadastrar, editar, excluir e gerar relatórios.

6. Conclusão

O sistema de **Gerenciamento Estudantil** permite a organização e controle de dados acadêmicos de forma eficiente e intuitiva. A persistência dos dados é garantida por meio do uso de um banco de dados relacional, e a interface gráfica facilita a navegação, oferecendo ao usuário uma experiência agradável e prática.

7. Instruções de Execução

1. **Configuração do Banco de Dados:**
 - Certifique-se de que o banco de dados está configurado corretamente com as tabelas necessárias (alunos, professores, cursos, matriculas, associacoes).
 - O sistema pode ser configurado para trabalhar com diferentes bancos de dados (MySQL, PostgreSQL, SQLite, etc.), dependendo do ambiente de desenvolvimento.
2. **Compilar o código:** Compile todos os arquivos `.java` utilizando o compilador Java (`javac`).
3. **Executar o programa:** Execute a classe `JanelaPrincipal` para iniciar o sistema.
4. **Interação:** Utilize os menus da interface gráfica para acessar as funcionalidades e interagir com o banco de dados.