

Relatório do Projeto

Gestão de Escola Secundária

Grupo Nº9



Realizado por:

Rita Gomes, Nº49519

Diogo Araújo, Nº49680

Diogo Rodrigues, Nº 49658

Pedro Ferrão, Nº50162

Covilhã, 31 de dezembro de 2023

Conteúdo

1	Introdução	2
2	Descrição geral do sistema	3
2.1	Funcionalidades	3
2.2	Tecnologias utilizadas	3
3	Design	3
3.1	Estrutura de classes	3
3.2	Interface do utilizador	4
4	Funções Implementadas	5
4.1	Menu	5
4.2	Funções dos diferentes menus	5
4.3	Métodos	6
5	Desafios de implementação	7
5.1	Tratamento do input do utilizador	7
5.2	Complexidade da relação estudante-professor-aluno	7
5.3	Complexidade da atribuição de professores	7
6	Conclusão	8

1 Introdução

Realizado na linguagem de programação *Java*, o trabalho tem como objetivo é criar um sistema informático que faça a gestão de uma escola secundária, como gerir horários, atribuir professores e inscrever alunos nos respetivos cursos e turmas. São realizadas também operações base, entre elas criar, atualizar e visualizar dados das entidades.

2 Descrição geral do sistema

2.1 Funcionalidades

O Sistema de Gestão de uma Escola Secundária oferece as seguintes funcionalidades principais:

1. Criação e inscrição de alunos em cursos e turmas.
2. Gestão dos professores, incluindo a atribuição de turmas.
3. Visualização de detalhes de alunos, professores, cursos, turmas e disciplinas.
4. Atribuição e visualização das notas dos alunos.

2.2 Tecnologias utilizadas

O programa é implementado em Java, utilizando princípios de programação orientada a objetos. Inclui uma interface de utilizador simples baseada em consola para interação.

3 Design

3.1 Estrutura de classes

O programa está estruturado em torno das seguintes classes fundamentais:

1. **Student:** Representa um aluno com informações pessoais, identificação (ID), curso e turma em que está inscrito, bem como as suas respetivas notas.
2. **Teacher:** Representa um professor com informações pessoais, identificação (ID), disciplinas lecionadas e respetivo horário.
3. **Course:** Representa um curso com turmas e disciplinas associadas.
4. **SchoolClass:** Representa uma turma dentro de um curso.
5. **Subject:** Representa uma disciplina dentro de um curso.
6. **StudentGrade:** Representa a classificação de um aluno numa disciplina específica.

3.2 Interface do utilizador

A interface do utilizador é baseada na consola, com um sistema orientado por menus. Os utilizadores podem escolher entre várias opções para realizar tarefas como a criação de alunos, a visualização de detalhes, a atribuição de notas e outras funcionalidades.

4 Funções Implementadas

4.1 Menu

Diferentemente do método convencional *case/break* para construir um menu optámos por estruturá-lo da seguinte forma:

1. **Organização do menu:** O menu apresenta um conjunto de menus principais (Ex: "studentMenu", "teacherMenu"...), cada um com as suas próprias opções e submenus.
2. **Opções do Menu:** Cada menu possui opções que correspondem a ações específicas. Essas ações estão associadas a métodos implementados que serão executados quando selecionada a opção (Ex: "Action a1 = () - método a implementar")".
3. **Submenus:** Também foram implementados submenus para agrupar funcionalidades relacionadas.
4. **Construir menu:** Por fim chama-se um método para construir todos os menus e submenus "main.build()".

4.2 Funções dos diferentes menus

1. **Aluno:** Este menu tem como opções e chama os métodos "Criar Aluno e Inserir num Curso", "Ver Detalhes do Aluno", "Editar Detalhes do Aluno", "Editar Detalhes do Aluno", "Remover Aluno", "Mostrar todos os Alunos", "Alterar detalhes dos Alunos".
2. **Professor:** Este menu tem como opções e chama os métodos "Criar Professor e Inserir no Curso", "Ver Detalhes do Professor", "Remover Professor", "Editar Detalhes do Professor", "Ver Todos os Professores".
3. **Curso:** Este menu tem como opções e chama os métodos "Ver Detalhes do Curso".
4. **Turma:** Este menu tem como opções e chama os métodos "Atribuir Professor à Turma", "Ver detalhes da Turma", "Horário", "Ver todas as turmas".

5. **Horário:** Este menu tem como opções e chama os métodos "Criar horário", "Ver Horário".
6. **Disciplina:** Este menu tem como opção e chama o método "Ver Detalhes da Disciplina".
7. **Notas:** Este menu tem como opções e chama os métodos "Atribuir Nota", "Ver Notas".
8. **Estatísticas:** Este menu tem como opções e chama os métodos "Ver o Curso com mais pessoas", "Ver Aluno com Melhor Média", "Ver a Turma com Melhores Notas".

4.3 Métodos

Adicionalmente aos métodos utilizados para os menus citados acima, implementámos outros, como por exemplo:

- Para ler/escrever ficheiros utilizámos o método ***Serializable***, uma vez que o mesmo é utilizado para guardar e atualizar objetos e considerámos que seria uma melhor abordagem, comparada com outros métodos existentes.
- Optámos por utilizar uma ***Interface*** para os *Cursos* e para as *Disciplinas*. Isto permite-nos uma melhor reutilização dos valores em diferentes partes do programa, facilita o processo para adicionar outros cursos ou disciplinas, respetivamente.

5 Desafios de implementação

5.1 Tratamento do input do utilizador

O processamento dos dados do utilizador numa interface de consola orientada por menus apresentava o seu próprio conjunto de desafios. Era crucial garantir que os utilizadores forneciam dados válidos e que o sistema respondia adequadamente a vários cenários. A implementação envolveu mecanismos robustos de validação de entrada e avisos claros ao utilizador. Além disso, foram incorporados mecanismos de tratamento de erros para orientar os utilizadores em caso de seleções inválidas.

5.2 Complexidade da relação estudante-professor-aluno

A intrincada relação entre disciplinas, professores e alunos apresentou diversos desafios na manutenção da integridade dos dados. Garantir que os professores estivessem associados às disciplinas corretas e que os alunos recebessem as notas das disciplinas adequadas exigiu uma análise cuidadosa. Foi dada especial atenção à implementação de métodos como `findSubjectByName` e `findTeacherById` para garantir a exatidão das associações.

5.3 Complexidade da atribuição de professores

Um desafio significativo encontrado durante a implementação foi a atribuição correta dos professores às disciplinas. Garantir que um professor fosse associado adequadamente a uma disciplina exigiu mecanismos meticulosos de validação e verificação de erros. A complexidade surgiu da necessidade de validar a atribuição tanto durante a criação do professor como na associação subsequente à disciplina. Foi implementada uma solução abrangente, verificando a validade das atribuições de professores em cada etapa para evitar discrepâncias.

6 Conclusão

Em conclusão, o Sistema de Gestão de uma Escola Secundária representa um avanço significativo na gestão eficiente dos dados educativos. Embora atualmente baseado em consolas, o sistema proporciona uma base sólida para futuras melhorias, incluindo a potencial implementação de uma interface gráfica de utilizador e a integração com uma base de dados. A conceção modular do programa, aliada à sua robusta funcionalidade, posiciona-o como uma ferramenta valiosa para administradores, professores e outras partes interessadas envolvidas na gestão da informação dos alunos.