



UNIVERSIDADE CATÓLICA DE ANGOLA
FACULDADE DE ENGENHARIA

Curso Engenharia Informática
Disciplina Fundamentos de Programação II
Ano Lectivo 2021/22

Relatório

Sistema de Controlo de Notas e Entrega de Trabalhos

Autor:

ID 1000026185
Nome Gilberto Alexandre Adão de Jesus
Turma AINF

Data 10/07/2022
Docente Engº Pedro Mbote

Resumo

Para consolidar os conhecimentos adquiridos de Programação Orientada a Objetos em Linguagem Java, decidiu-se criar um sistema para gerenciar as entregas de trabalhos dos alunos e a atribuição de notas de modos a que se faça o controlo de matrículas e o rendimento escolar de cada aluno.

1. Introdução

Uma aplicação é um software usado nas mais diversas áreas da vida quotidiana, como forma de automatizar certas resoluções de certos problemas, sendo usado para a resolução de problemas técnicos, sociais, psicológicos entre outras, esses problemas podem ser o tédio e a depressão, inclusive, resolvendo com entretenimento e socialização, jogos e mídias sociais, problemas de aprendizagem desenvolvendo aplicações de acompanhamento psicológico e aprendizagem, mas para esse caso não é nada de diferente do que já disse anteriormente, resolução de problemas, mas nesse caso foi pedido que se desenvolvesse uma aplicação para uma instituição de ensino, de modos que se possa fazer um controlo rigoroso e eficiente das notas que o docente irá atribuir ao aluno e de forma contrária, o aluno entregar os trabalhos para a avaliação contínua.

2. Descrição do Problema

Foi solicitado que se projectasse uma aplicação, para uma realidade onde uma instituição onde cada turma tem uma única disciplina, um único professor e vários alunos, um aluno pode estar em várias turmas, podem existir várias turmas com a mesma disciplina, um professor pode dar aulas em várias turmas, um professor não está associado á uma disciplina em específico, a existência de uma turma depende unicamente da disciplina que nela é lecionada, para isso, foi determinado com os seguintes parâmetros:

1. Classes:

- Aluno: tem o objetivo de armazenar todas informações necessárias de cada aluno.
- Professor: tem o objetivo de armazenar todas informações necessárias de cada professor
- Usuário (Por Questões de atributos Similares entre Aluno e Professor): tem o objetivo de armazenar todas informações que a Classe Aluno e Professor compartilham.
- Disciplina: tem o objetivo de armazenar todas informações necessárias de cada disciplina.
- Turma: consiste em uma relação entre a Classe Professor e a Classe Disciplina.
- Rendimento Escolar: consiste em uma relação existente entre a Classe Aluno e a Classe Turma.

2. Menus:

- Professor: onde o usuário poderá realizar operações inerentes ao professor.
- Aluno: onde o usuário poderá realizar operações inerentes ao aluno.
- Administrador: onde o usuário poderá realizar operações inerentes a um superusuário, controlando tudo.

3. Armazenamento:

Relatório – Sistema de Controlo de Notas e Entrega de Trabalhos

- Ficheiros: o tipo de armazenamento de dados que será usado para guardar os dados.
4. Exceções: serão usadas para encontrar algum erro no sistema e consequentemente mostra-lo.

3. Equipamentos e Ferramentas

Computador usado para o desenvolvimento da aplicação: Computador Portátil **INOVIA** – Especificações:

- **Processador:** Intel Core i3-3120M CPU 2.50GHz, 2 núcleos, 4 processadores lógicos
- **Arquitetura:** 64 bits
- **RAM:** 6 GB
- **GPU:** Intel HD Graphics 4000
- **Sistema Operativo:** Microsoft Windows 10 Pro

Linguagem usada para a implementação da Aplicação: **JAVA** - JDK e JRE - Especificações:

- **Versão:** 1.8.0

Ambiente de Desenvolvimento usado: **Notepad++** – Especificações:

- **Versão:** 8.1

4. Procedimento

Após uma análise do problema enunciado, decidiu-se as implementações do “Set” de classes mencionadas abaixo:

- Usuário
- Aluno
- Professor
- Disciplina
- Turma
- Rendimento Escolar
- Exceções Gerais
- Exceções Rendimento Escolar
- Ficheiro Aluno
- Ficheiro Professor
- Ficheiro Disciplina
- Ficheiro Turma
- Ficheiro Rendimento Escolar
- Iniciar Sessão
- Tipo de Usuário
- Menu Principal
- Executavel

É importante referir que a Sistema foi implementado com conceitos de Programação Orientada a Objectos, para que fosse possível realizar as interações entre todas as classes de forma eficiente e sem qualquer equivocação.

4.1. Classe Usuário

A classe contém os seguintes atributos e métodos:

- **int id:** Um atributo unívoco usado para efeitos de consulta e busca e identifica o aluno ou o professor como sendo único na instituição.
- **String nome:** Contém o nome ou uma informação que identifica o aluno ou o professor.
- **Date dataNascimento:** contém a data de nascimento do professor.
- **String username:** contém o nome de usuário para início de sessão.
- **String senha:** contém a senha para início de sessão.
- Métodos acessores (get) dos atributos.
- Métodos modificadores (set) dos atributos.
- Métodos construtores: implementou-se um construtor padrão e um outro construtor que recebe os 5 atributos.

4.2. Classe Aluno

A classe herdou de Usuario todos os seus comportamentos e atributos de formas á minimizar o código da classe:

- **int período :** contém o período do aluno.
- **Método resumo:** Contém informações resumidas de todos atributos da classe.
- Métodos assessores (get) dos atributos.
- Métodos modificadores (set) dos atributos.
- Métodos construtores: implementou-se um construtor padrão e um outro construtor que recebe 6 atributos, um da classe Aluno e o resto da classe Usuario.

4.3. Classe Professor

A classe herdou de Usuario todos os seus comportamentos e atributos de formas á minimizar o código da classe:

- **String cargo:** contém o cargo que o professor ocupa na instituição.
- Métodos assessores (get) dos atributos.
- Métodos modificadores (set) dos atributos.
- **Método resumo:** Contém informações resumidas de todos atributos da classe incluindo os da superclasse.
- Métodos construtores: implementou-se um construtor padrão e um outro construtor que recebe 6 atributos, um da classe Aluno e o resto da classe Usuario.

4.4. Classe Disciplina

A classe contém os seguintes atributos e métodos:

- **int id:** Um atributo unívoco usado para efeitos de consulta e busca e identifica a disciplina como sendo única na instituição.
- **String nome:** Contém o nome ou uma informação que identifica a disciplina.
- **String ementa:** Contém informações descritivas da cadeira do curso.
- **Métodos acessores (get)** dos atributos.
- **Métodos modificadores (set)** dos atributos.
- **Método resumo:** Contém informações resumidas de todos atributos da classe.
- **Métodos construtores:** implementou-se um construtor padrão e um outro construtor que recebe os 3 atributos.

4.5. Classe Turma

A classe relaciona Disciplina e Professor e contém os seguintes atributos e métodos:

- **int id:** Um atributo unívoco usado para efeitos de consulta e busca e identifica a turma como sendo única na instituição.
- **Disciplina disciplina:** Contém um objeto do tipo Disciplina que não pode ser null, visto que para existência de uma turma é obrigatório uma disciplina.
- **Professor professor:** Contém um objeto do tipo Professor que pode ser null caso não existir nenhum professor lecionando a turma e existir no caso contrário.
- **Int capacidade:** Contém a quantidade permitida de alunos numa certa turma.
- **Método resumo:** Contém informações resumidas de todos atributos da classe.
- **Métodos acessores (get)** dos atributos.
- **Métodos modificadores (set)** dos atributos.
- **Métodos construtores:** implementou-se um construtor padrão e um outro construtor que recebe os 4 atributos.

4.6. Classe Rendimento Escolar

A classe relaciona Turma e Aluno e contém os seguintes atributos e métodos:

- **Turma turma:** Contém um objeto do tipo Turma que não pode ser null, visto que para existência de um rendimento escolar é obrigatório uma turma.
- **Aluno aluno:** Contém um objeto do tipo Aluno que não pode ser null, visto que para existência de um rendimento escolar é obrigatório um aluno.
- **float prova1:** Contém a nota do 1º QG.
- **float prova2:** Contém a nota do 2º QG.
- **Array de String trabalhos(4 posições):** Contém um conjunto de trabalhos que o aluno deve entregar.
- **Array de float notaTrabalhos(4 posições):** Contém um conjunto de notas equivalentes aos trabalhos entregues.

- **Array de String** estados: contém os 3 possíveis estados que um aluno pode estar, que são APROVADO, NA FINAL ou REPROVADO;
- **Método notaComExtra:** retorna um float e tem como objetivo calcular as notas das provas incluindo as notas dos trabalhos.
- **Método mediaPonderada:** retorna um float e calcula a média, baseada nos critérios de 1º QG tem peso de 40% e 2º QG tem peso de 60% da média final.
- **Método notaParaFinal:** calcula a nota necessária para poder passar para a final apartir da fórmula:

$$NotaNecessária = \frac{NotaMínima - (NotaProva1 \times 0.4)}{0.6}$$

Observe que NotaMínima é uma constante igual a 10

- **Método resumo:** Contém informações resumidas de todos atributos da classe.
- **Método estado:** retorna uma string que mostra o estado de cada estudante apartir da média ponderada calculada apartir do método mediaPonderada. Apartir do seguinte critério:
 - Média acima de 9.5 = Aprovado.
 - Média entre 7.0 e 9.4 = Na Final.
 - Média abaixo de 7.0 = Reprovado.
- **Métodos acessores** (get) dos atributos.
- **Métodos modificadores** (set) dos atributos: lançam exceções relevantes sobre atribuição de notas de trabalho e provas, entrega de trabalhos.
- **Métodos construtores:** implementou-se um construtor padrão e um outro construtor que recebe os 4 atributos (turma, aluno, prova1,prova2).

4.7. Exceções Implementadas

As exceções foram implementadas baseado nos possíveis erros de lógica que possam ocorrer no sistema, abaixo as possíveis exceções e suas circunstâncias de lançamento:

- **Turma Lotada:** exceção que surge quando se deseja adicionar um aluno a uma turma, mas o número de alunos é equivalente a capacidade da turma.
- **Usuario Inexistente:** exceção que surge quando se deseja iniciar sessão, mas não existe qualquer usuário no sistema com credenciais iguais às digitadas.
- **Id Inexistente:** exceção para buscas por id, surge quando não foi encontrado nenhum id ao procurado na lista.
- **Admin Definido:** exceção que surge quando se deseja cadastrar um novo administrador.
- **Elemento Nulo:** exceção que surge quando aparece um elemento igual á null.
- **Já Existente:** exceção chamada quando na adição de um vínculo entre um elemento e outro, já exista um vínculo anterior entre eles. Exemplo: um aluno tentando se matricular numa turma que já está matriculado.
- **Posição Inválida:** exceção que aparece quando a posição é negativa ou maior ou igual que o tamanho de um certo array.

- **Nota Inválida:** exceção surge quando se insere ou atribui uma nota de valor negativo ou maior que 20.
- **Trabalho Vazio:** exceção que surge quando um trabalho é entregue vazio ou nulo.
- **Nota Sem Trabalho:** exceção para casos de atribuição de notas para um aluno que não entregou trabalho.
- **Trabalho Já Entregue:** exceção para casos de tentar entregar mais de uma vez um mesmo trabalho.
- **Usuário Já Existente:** exceção usada para evitar redundância de dados e possivelmente usuários com usernames e senhas iguais, de modos á fazer um usuário único durante o cadastro.

4.8. Ficheiros - Armazenamento

Por razões de melhor manipulação de ficheiros foram usadas as classes, PrintWriter para escrita e Scanner para leitura.

4.8.1. Ficheiros Aluno, Professor e Disciplina

- **Método inicializar:** cria um novo ficheiro caso ele não exista.
- **Método gerarId:** retorna um inteiro que é o id gerado para um novo registro.
- **Método adicionar:** adiciona um novo registro recebido como parâmetro ao ficheiro.
- **Método remover:** remove um registro do ficheiro, apartir do id recebido como parâmetro.
- **Método editar:** edita um registro do ficheiro recebido como parâmetro.
- **Método Listar:** retorna uma string que é a lista dos resumos dos registros do ficheiro.
- **Método Total:** retorna um inteiro que é a contagem dos registros existentes no ficheiro.
- **Método procurar ou Método de busca:** retorna um objeto da classe correspondente, encontrado no ficheiro apartir do id passado como parâmetro.

4.8.2. Ficheiro Turma

- Contém todos os métodos existentes no 4.8.1.
- **Método turmaLotada:** retorna um boolean indicando se a turma está lotada ou não, uma certa turma com certo id passado como parâmetro.
- **Método listarTurmasProfs:** retorna uma string contendo ou a lista de resumos das turmas lecionadas ou a lista de resumos de turmas disponíveis do professor passado como parâmetro.
- **Método listarTurmasNãoMatriculadas:** retorna uma string contendo a lista de turmas em que um aluno não está matriculado, um aluno com id passado como parâmetro.

4.8.3. Ficheiro Rendimento Escolar

Contém todos os métodos existentes no 4.8.1 excepto o gerarId.

- **Métodos listar Alunos de uma turma e listar turmas de um aluno:** retorna uma string contendo uma listagem.
- **Métodos total alunos de uma turma e total turmas de um aluno:** retorna um inteiro contendo a contagem de registros.
- **Método listarAlunosPorcentual:** retorna uma string contendo a listagem de alunos com as médias, o estado académico e a estatística dos alunos de uma certa turma passada como parâmetro.

4.9. Classe Iniciar Sessão

A classe realiza as operações de Início de Sessão e Cadastro de um Usuário e funciona como um menu, e estes são os métodos desta classe:

- **Método Executar:** inicia o programa.
- **Método inializarFicheiros:** inicializa os ficheiros, caso eles não existam ou forem apagados.
- **Método escolher:** escolher uma opção do menu.
- **Método animação:** criar uma animação de inialização de sistema e desligamento do sistema.
- **Método inicioSessao:** menu da tela de login, onde são realizadas o início de sessão e os cadastros.
- **Método selecionarTipo:** selecionar o tipo de usuário que se deseja logar ou cadastrar, seja ele Professor, Aluno ou Admin.
- **Método lerData:** leitura de data em um certo formato.
- **Método cadastrarUsuario:** cadastrar um novo usuário selecionando o seu tipo.
- **Método inserirCredenciais:** iniciar sessão de um usuário.

4.10. Classe Menu Principal

Tal como a classe Iniciar Sessão, esta classe consiste em juntar todos os menus em um único e usar todos os métodos existentes em cada ficheiro e os tratamentos de métodos, basta visualizar o código da classe **MenuPrincipal**, pretende-se abstrair de explicações, pois a quantidade de códigos usada não permitirá explicar de forma totalmente o código da classe. Este é o resumo da classe.

O resultados poderão ser verificados logo á seguir, no próximo capítulo do documento.

5. Resultados e Análise dos Resultados

```
Iniciando o Sistema...
+
++
+++
++++
+++++
//Inicio de Sessao\
1 - Iniciar Sessao
2 - Cadastrar Usuario

0 - Encerrar o Programa
Insira:
_
```

Este é o visual do Início de Sessão do programa, antes de ir para qualquer sítio, basta tu cadastrares.

```
Administrador: C:\Windows\System32\cmd.exe - java Executavel
//Inicio de Sessao\
1 - Iniciar Sessao
2 - Cadastrar Usuario

0 - Encerrar o Programa
Insira:
2
//Tipo de Usuario\
1 - Professor
2 - Aluno
3 - Administrador

0 - Voltar
Insira:
1
Nome: Pedro Mbote
Insira a Data no Formato(dd-MM-yyyy)
23-09-1989
Nome de Usuario: panzo
Senha: 1234
Cargo: Vice-Decano
Seja Bem-Vindo Sr(a) Pedro Mbote
//Menu Professor\
1 - Listar Turmas Lecionadas
2 - Listar Medias com Estatisticas
3 - Listar Turmas Disponiveis

0 - Encerrar Sessao
Insira:
```

Já registrado, te será dado uma mensagem de boas vindas e mostrará o menu do professor ,no caso.

Relatório – Sistema de Controlo de Notas e Entrega de Trabalhos

```
Administrador: C:\Windows\System32\cmd.exe - java Executavel
Saíndo...
//Início de Sessão\
1 - Iniciar Sessão
2 - Cadastrar Usuário

0 - Encerrar o Programa
Insira:
2
//Tipo de Usuário\
1 - Professor
2 - Aluno
3 - Administrador

0 - Voltar
Insira:
2
Nome: Mario Pedro
Insira a Data no Formato(dd-MM-yyyy)
30-05-2002
Nome de Usuário: mpedro
Senha: mpedro
Período(Inteiro): 6
Seja Bem-Vindo Sr(a) Mario Pedro
//Menu Aluno\
1 - Listar Turmas Matriculadas
2 - Listar Turmas Disponíveis

0 - Encerrar Sessão
Insira:
```

Já para o menu do Aluno tem apenas duas opções, que são as turmas disponíveis e as turmas matriculadas.

```
Administrador: C:\Windows\System32\cmd.exe - java Executavel
Insira:
0
Saíndo...
//Início de Sessão\
1 - Iniciar Sessão
2 - Cadastrar Usuário

0 - Encerrar o Programa
Insira:
1
//Tipo de Usuário\
1 - Professor
2 - Aluno
3 - Administrador

0 - Voltar
Insira:
3
Nome de Usuário: admin
Senha: 1234
Seja Bem-Vindo Sr(a) admin
//Menu Admin\
1 - Disciplinas
2 - Turmas
3 - Alunos
4 - Professores

0 - Encerrar Sessão
Insira:
```

No modo Administrador, tu poderás ver e escolher um leque maior de opções, sendo operações simples e padrão.

Relatório – Sistema de Controlo de Notas e Entrega de Trabalhos

```
Administrador: C:\Windows\System32\cmd.exe - java Executavel
Insira:
1
//Tipo de Usuario\
1 - Professor
2 - Aluno
3 - Administrador

0 - Voltar
Insira:
3
Nome de Usuario: admin
Senha: 1234
Seja Bem-Vindo Sr(a) admin
//Menu Admin\
1 - Disciplinas
2 - Turmas
3 - Alunos
4 - Professores

0 - Encerrar Sessao
Insira:
4
//Menu Admin-Professores\
1 - Remover
2 - Consultar
3 - Listar

0 - Voltar
Insira:
```

Como pode ver, muito mais opções.

```
Administrador: C:\Windows\System32\cmd.exe - java Executavel
Disciplina: [ id: 2022, Nome: Fisica, Ementa: Conceitos Elementares Fundamentais ]
Professor: [ id: 2000, Nome: Menas Boy, Cargo: Decano, BirthDate: 25-07-2000, Nome de Usuario: menas, Senha: boy ]

//Selecionar Elemento\
Insira o Id

0 - Sair
Insira:
2300
Alunos Matriculados
[ id: 1995, Nome: Maluma Mboco, Período: 4.º, BirthDate: 31-01-1995, Nome de Usuario: moco, Senha: moco ]
Média: 11.0
Estado: Aprovado
[ id: 2001, Nome: Alípio Nunes, Período: 2.º, BirthDate: 30-05-2001, Nome de Usuario: nunes, Senha: 1234 ]
Média: 9.0
Estado: Final {Precisa tirar: 9.0}
[ id: 1999, Nome: Pedro Mbote, Período: 5.º, BirthDate: 20-10-1999, Nome de Usuario: pmbote, Senha: 1234 ]
Média: 10.0
Estado: Aprovado
Alunos Aprovados: 2(66.66667%)
Alunos Na Final: 1(33.33333%)
Alunos Reprovados: 0(0.0%)
//Menu Professor\
1 - Listar Turmas Lecionadas
2 - Listar Médias com Estatísticas
3 - Listar Turmas Disponíveis

0 - Encerrar Sessão
Insira:
```

Na opção listar médias com estatísticas, tu escolhes a turma e visualizas os alunos que estão nesta turma e poderás ver as médias e os estados de cada aluno e inclusive poderás ver a sondagem de alunos existentes lá.

As outras operações estarão disponíveis assim que entrar no sistema.

6. Considerações Finais

Este programa serviu de simulação para a realização de forma eficaz e usando as boas práticas de programação, do Sistema de Controlo de Notas e Entrega de Trabalhos, se baseando nos parâmetros dados pelo enunciado, executando todas e quaisquer operações referentes ao monitoramento e gerenciamento das notas e dos trabalhos.

7. Referências Bibliográficas

- Fonseca P. – Guia para a redação de Relatórios – 11ª Edição – 2012
- Deitel P., Deitel H. – JAVA - Como Programar – 10ª Edição – Pearson Education do Brasil, 2017

8. Anexos