Universidade de São Paulo

Instituto de Ciências Matemáticas e de Computação

Departamento de Ciências de Computação

Disciplina de Programação Orientada a Objetos (SSC0103)

Documentação do Trabalho Final de Programação Orientada a Objetos

Portal de acesso de informações de pessoas em ambiente escolar

Prof. Dr. Márcio Delamaro

Grupo 1:
Alcino Salviano Cavalcanti, 11892963

Calvin Suzuki de Camargo, 11232420

Gabriel Takeshi Miyake Batistella, 11232198

Pedro Henrique Raymundi, 11795634

São Carlos - SP
27 de julho de 2021

1. INTRODUÇÃO

1.1 SOBRE O PROJETO

O projeto final criado para a disciplina de Programação Orientada a Objeto consiste na elaboração de um programa que auxiliará a organização de dados de pessoas em ambiente escolar, e para isso, será desenvolvido um portal de acesso hierarquizado em tipos de usuários. Espera-se que o programa facilite a administração da escola no controle do quadro de funcionários e alunos bem como também os auxilie a acessarem suas informações no sistema.

1.2 PROPOSTA

A proposta foi pensada levando em consideração a parte administrativa de uma escola e a quantidade de informação acumulada nesse setor. O programa oferece o acesso de informações como horários, ocorrências, listas de alunos e funcionários e etc. A possibilidade de se registrar e acessar essas informações através de um programa computacional ajudaria em uma melhor gestão desses dados e facilitaria o acesso de todas as pessoas selecionadas a eles.

Além disso, os alunos e seus responsáveis também conseguiriam ter um melhor acompanhamento de sua situação escolar. Ademais, o projeto simplifica o trabalho do diretor e dos pedagogos no colégio em identificar alunos com dificuldades que precisam de um maior acompanhamento pedagógico ou alunos avançados que podem ser indicados para a iniciação científica jr.

Através da motivação apresentada e da proposta de trabalho, espera-se que o desenvolvimento desse projeto possa auxiliar a escola, e todos os seus membros a terem uma administração mais eficiente.

2. OBJETIVOS

O objetivo principal do trabalho foi desenvolver um sistema de acesso de informações de pessoas em uma escola. Esse acesso é hierárquico, i. e., controlado por um sistema de restrição ao acesso de informação com base unicamente no tipo de conta em que o portal é aberto.

Além disso, procuramos aplicar nossos os assuntos abordados na disciplina em programação orientada a objeto e em linguagem *Java*™ para criar:

- *Graphical user interface*;
- Hierarquização de acesso à informação do sistema;
- Sistema de *login*;
- Aplicação de banco de dados;

3. DESCRIÇÃO DO SOFTWARE

3.1 ORGANIZAÇÃO DA HIERARQUIA DE ACESSO

Um dos fundamentos do programa é dar um acesso controlado para as pessoas presentes em um ambiente escolar. Neste, classificamos as pessoas como: Aluno, Professor, Zelador e Diretor. Cada uma dessas pessoas carrega dados importantes sobre sua atuação, mostrada na figura 1, e tem acesso a informações de outros tipos de pessoa, como mostrado na figura 2.

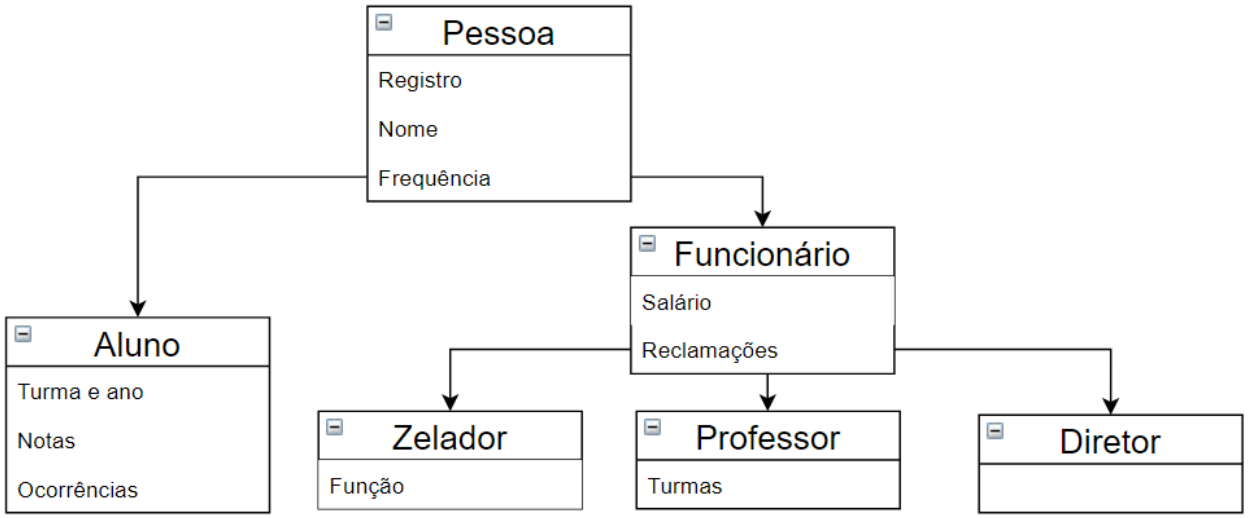

Figura 1: Organização de herança de informações pela orientação à objetos.

Na Figura 1, foi esquematizada a forma das heranças das funções e variáveis entre as classes elaboradas que consistem em carregar os dados de tipos específicos de “Pessoas”, detalhadas na seção 3.2.

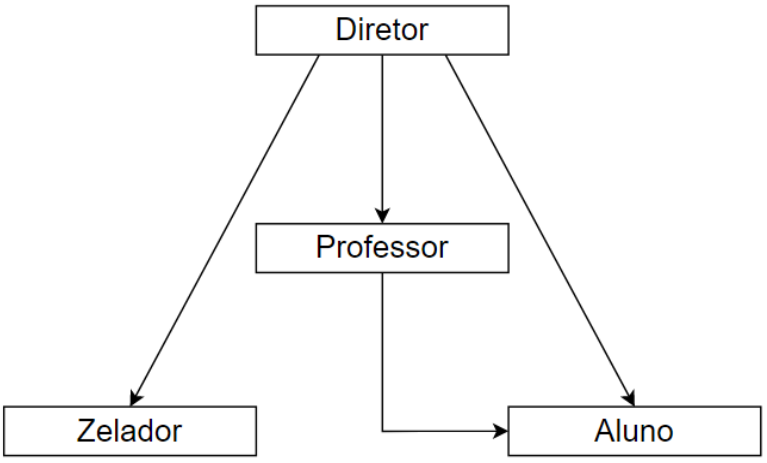

Figura 2: Organização da hierarquia de acesso a informações de pessoas.

A Figura 2 representa a hierarquia de acesso de informações no sistema. A

flecha representa um acesso unilateral de um tipo de conta à outra, por exemplo, o professor acessa os dados de todos os alunos que estão nas turmas pelas quais ele é responsável.

3.2 CLASSES IMPLEMENTADAS

Foram criadas 12 classes principais no trabalho (excluindo *Exceptions*). É possível dividir esse grupo em 3 frentes: a primeira que contém a interface gráfica do projeto; a segunda que salva, insere, remove dados da escola; a terceira que é responsável por cada pessoa e seus dados.

1. **UI:**
    - *AddPessoaUI:* Gera uma janela que permite que uma conta de diretor possa adicionar uma nova pessoa ao sistema (aluno, professor, zelador, diretor), tendo que informar as informações iniciais básicas dessa nova pessoa, e não pode utilizar um registro que já exista. As informações extras são salvas como as *default*;

    - *InfoPessoaUI:* Gera uma janela que permite ao usuário ver as informações de outra pessoa (ou suas próprias), contanto que esse usuário tenha acesso às informações que deseja ver (de acordo com as relações da Figura 2). Além disso, permite a alteração de algumas informações (como as notas de um aluno, ou o número de reclamações de um professor), e até mesmo a remoção de uma dada pessoa, também de acordo com as mesmas relações de acesso. Também fornece avisos/sugestões de como “ajudar a pessoa” baseados no rendimento dela (classificado por um algoritmo), por exemplo pode-se sugerir um acompanhamento pedagógico a alunos com dificuldades ou uma iniciação científica jr. a alunos avançados;

    - *LoginUI:* Gera a janela inicial do programa, permitindo ao usuário inserir seu registro e sua senha. O programa então checa se os dados inseridos batem com o banco de dados e, se sim, inicia a página inicial. No caso de uma pessoa recém-inserida, que ainda não possui senha, a senha escrita no primeiro acesso será definida como a senha oficial dessa pessoa;

    - *PagPrincipalUI:* Gera a janela principal do programa, que mostra todas as pessoas no nosso banco de dados, permitindo: a navegação entre várias páginas, a busca de pessoas específicas por registro ou por nome, a filtragem das pessoas a serem mostradas por tipo (aluno, professor, zelador, diretor) e a ordenação das pessoas por ordem crescente de registro ou por ordem alfabética dos nomes. Além disso, se o usuário clicar em uma pessoa específica, o programa checa se o usuário tem acesso às informações da pessoa “clicada” e, se sim, inicia a janela info pessoa. Similarmente, permite também a rápida

visualização das próprias informações do usuário. No caso do usuário logado ser um diretor, o programa possibilita inserção de uma nova pessoa no banco de dados, iniciando a janela add pessoa;

2. **Gerenciamento dos dados da escola:**
    - *Escola*: É responsável por adicionar e remover logicamente pessoas da escola, também é usada para retornar as informações pertinentes para a UI e checar a tentativa de login de um usuário. Além disso, essa classe realiza as ordenações dos dados para a impressão na interface gráfica. No geral é utilizado polimorfismo para essas operações já que, na maioria dos casos, não é importante os dados específicos de cada Pessoa, mas sim sua classe base.
    - *GerenciadorDados*: Faz a leitura e a escrita dos dados de todas as pessoas na escola dentro do arquivo *.csv, para que assim seja possível salvar os dados para execuções futuras. Cada classe que estende Pessoa é salva e lida de forma um pouco diferente para acomodar corretamente os dados na execução.

3. **Pessoas:**
    - Pessoa: Carrega as informações em comum de todas as pessoas no sistema: Nome, Senha, Número de Registro, Frequência:
        - Aluno: Possui dados de Turma, Número de Ocorrências, Notas;
        - Funcionário: Armazena Salário e Reclamações:
            - Diretor: Cargo com maior privilégio no sistema da escola;
            - Professor: Possui os dados das Turmas ministradas;
            - Zelador: Carrega a função que exerce na escola

4. INTERFACE GRÁFICA

A interface gráfica foi implementada utilizando *Swing*, que nos foi apresentado em sala de aula. Inicialmente, é apresentada uma tela de *login* com campos a serem preenchidos com o número de registro e senha do usuário (Figura 3). Ao pressionar o botão "*Login*", é usado uma base de dados em arquivo **.csv* para a confirmação do *login* e acesso ao sistema. Há, ainda, nessa tela um ícone de ocultar que permite tornar ou não a senha visível ao usuário.

Figura 3: Tela de *login* do aplicativo.

Na tela seguinte, tem-se a Página Principal do programa, visto na Figura 4. No topo da página, encontram-se botões que permitem acessar facilmente as informações do usuário logado (aluno, professor, zelador ou diretor) e fazer a seleção de filtros que alteram a listagem das pessoas exibidas. Há ainda uma barra de pesquisas que possibilita encontrar uma pessoa através de seu nome ou número de registro. A Página Principal exibe as pessoas cadastradas, e suas informações básicas, e possui uma barra de rolagem para auxiliar na exibição. É possível navegar entre as diferentes páginas através dos botões de navegação presentes ao final de cada página.

Figura 4: *Layout* da tela principal do sistema.

Ainda na Página Principal, é possível selecionar alguma pessoa para visualizar ou alterar alguma informação, dependendo da autorização da conta logada. Ao selecionar uma pessoa (e possuindo a autorização para visualizá-la), é aberta uma janela com as informações desta (Figura 5). Essas páginas com informações pessoais variam de acordo com o tipo de pessoa. Para alunos, é possível visualizar e/ou modificar notas, média, frequência e ver a necessidade de um acompanhamento pedagógico; Para os demais funcionários, é possível visualizar e/ou modificar informações como salário, frequência, reclamações e as informações específicas de cada um (função de um zelador, turmas de um professor, etc). É possível, ainda, salvar ou não as modificações e remover a pessoa do sistema por essa janela. Além disso, um algoritmo baseado na frequência e no número de reclamações/ocorrências da pessoa (e também nas notas no caso do aluno) classifica o rendimento dela e, se necessário, fornece avisos/sugestões de como “ajudar a pessoa”, por exemplo é sugerido um acompanhamento pedagógico a alunos com dificuldades ou uma iniciação científica jr. a alunos avançados, um exemplo pode ser visto na Figura 6.

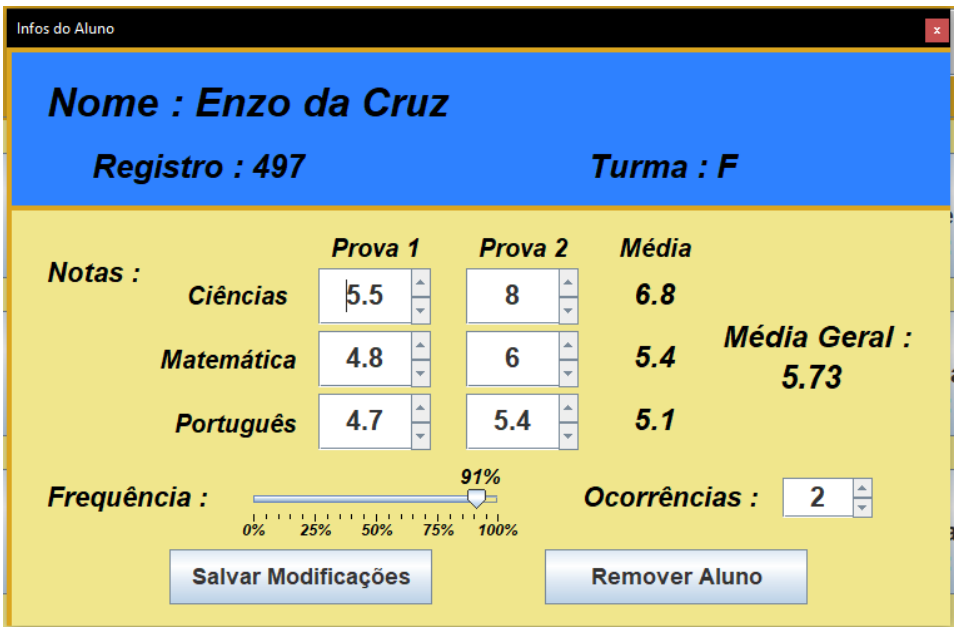

Figura 5: Painel de controle das informações do aluno.

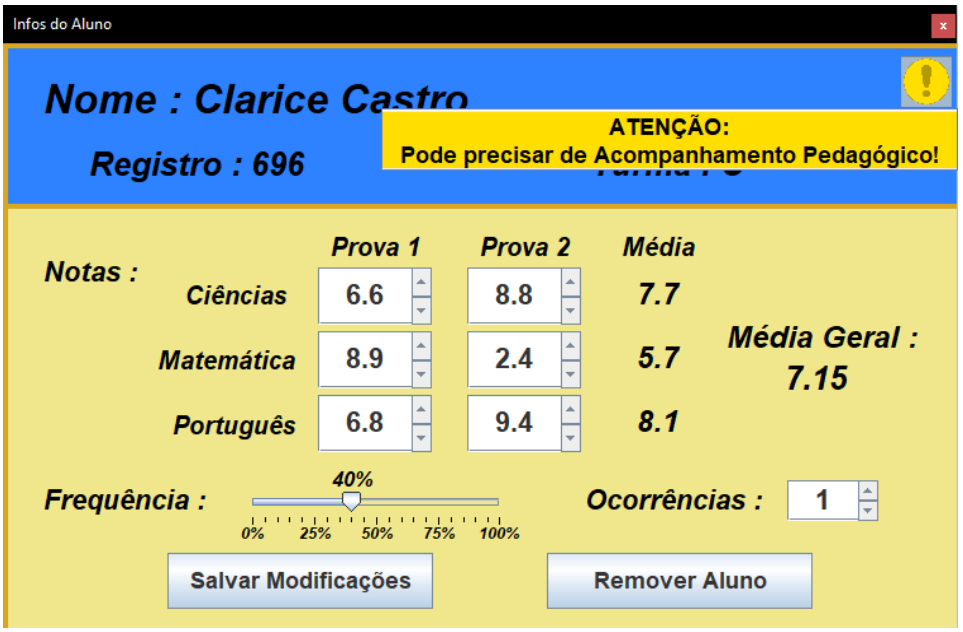

Figura 6: Exemplo de aluno com notificação de mal desempenho

Quando um diretor estiver logado no sistema, aparecerá ao final da listagem de pessoas da Página Principal um botão que permite adicionar uma nova pessoa ao sistema. Ao selecioná-lo, é aberta uma janela para a adição da pessoa. Nela encontram-se quatro *Radio Buttons* que permitem a seleção do tipo de pessoa: aluno, professor, zelador e diretor. Para cada pessoa selecionada, aparecerão apenas os campos de informações referentes a ela. Após o preenchimento dos campos com as informações pedidas, basta pressionar o botão “Adicionar” e um *pop-up* aparecerá informando o resultado da operação.

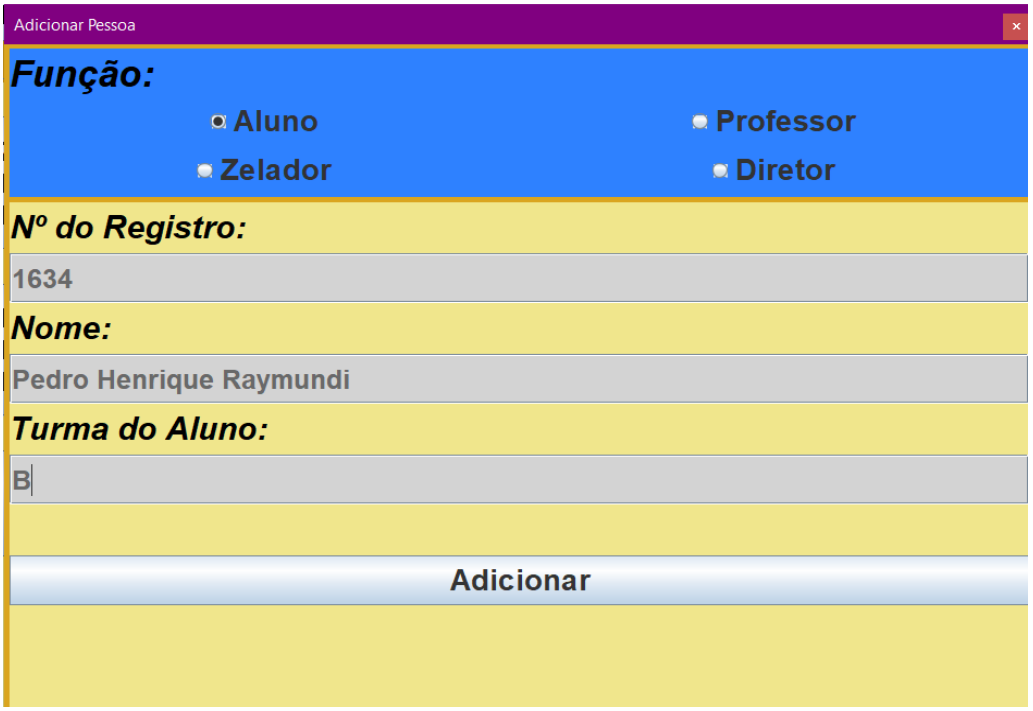

Figura 7: *Layout* da tela de adicionar novo aluno.

Por fim, ao encerrar o sistema, fechando a Página Principal, a base de dados *.csv é atualizada com as modificações realizadas durante a execução do programa.

5. CLASSES JAVA UTILIZADA

- *java.io…* foi utilizado no GerenciadorDados para a escrita e leitura dos dados nos arquivos csv, além disso esse pacote possui exceções úteis para suas funções;
- *java.util.Comparator* foi utilizada para facilitar o processo ordenação do *ArrayList* de Pessoas;
- *java.util.ArrayList* foi usada pela Escola para organizar todas as Pessoas que são inseridas lá. Isso possibilita a adição e remoção de novas pessoas (entre outras coisas) sem que saibamos quantas vão ser necessárias;
- *java.util.Dictionary* e java.util.Hashtable foram utilizadas para organizar *labels* no *slider* contido na classe *InfoPessoaUI*;
- *javax.swing…* foi utilizada nas classes da UI para a montagem da interface gráfica das diferentes páginas do programa. A partir dela foram desenvolvidos os painéis, campos de texto, botões e demais itens da interface;
- *java.awt…* também foi utilizada nas classes da UI do programa. Por meio dela foi implementado as ações referentes aos botões e demais itens interativos da interface. Ela também foi utilizada para a organização de fontes, imagens, cores e demais itens visuais da interface gráfica.

6. CONCLUSÃO E CONTRIBUIÇÃO POR ALUNO

O resultado final do trabalho correspondeu às expectativas do grupo e utilizou, se não todos, a maioria dos conceitos aprendidos em aula, como hierarquia, sobrecarga, arrays e strings, polimorfismo etc. Além disso acreditamos que o projeto final organiza de maneira simples diversas necessidades que alguém pode sentir no ambiente escolar, por exemplo, para os alunos isso significa o acesso de suas notas, para os diretores é interessante poder acompanhar o andamento geral dos funcionários. O grupo teve sim alguns problemas na implementação do código, mas tentamos resolver tudo com discussões semanais para que cada um pudesse se ajudar.

No fim, sabemos que existem algumas coisas que podem ser melhoradas, das quais não temos o conhecimento teórico, nem as ferramentas necessárias para a aplicação (por exemplo a hospedagem do serviço em servidor que assegura o envio de informações à usuários autenticados), outras melhorias demandam uma grande elevação da complexidade do código e, para os objetivos de uma atividade da disciplina POO, não valiam muito a pena (um exemplo seria algum método de senhas com hash criptográfica).

Os autores do código estão definidos no topo de cada arquivo .java. Todo o trabalho foi dividido igualmente, dessa forma a contribuição de cada aluno foi:

Alcino Salviano Cavalcanti - 25%
Calvin Suzuki de Camargo - 25%
Gabriel Takeshi Miyake Batistella - 25%
Pedro Henrique Raymundi - 25%

6. REFERÊNCIAS

Aulas do professor e monitores