

Trabalho Prático 1

Rede Social de Pesquisadores

Édipo Fernandes Vieira de Oliveira - 2011054324

Diego Henrique de Castro Aniceto - 2011054286

Departamento de Ciência da Computação – Universidade Federal de Minas Gerais

April 14, 2015

Resumo:

Este relatório descreve a implementação da solução proposta para o problema de manipulação de e armazenamento de dados de uma rede social de pesquisadores. Para que fosse possível essa implementação foi utilizado a linguagem Java de programação além de teorias de Orientação a Objeto e Modularização.

O resultado obtido foi satisfatório, tanto em relação a solução do problema, quanto aos conceitos envolvidos.

1. Introdução

Este trabalho tem como objetivo, introduzir os principais conceitos da Programação Orientada ao Objeto e modularização de código. Para que estes fossem exercitados foi proposto a solução de um problemas de manipulação e armazenamento de dados referentes a uma rede social de pesquisadores onde eles podem se relacionar através de artigos publicados, podem publicar seus artigos desenhados, entre outras coisas

A modelagem do problema gira em torno de três entidades principais, que são os Pesquisadores, os Veículos de Comunicação e os Artigos, e a partir delas e dos arquivos de entradas disponibilizados, e a partir disso realizar os calculos solicitados pela especificação do trabalho, que são: o calculo de popularidade de cada Pesquisador, o fator de impacto de cada Veiculo de Comunicação, e a pontuação de cada Artigo.

- A seção 2 discute detalhes de implementação.
- A seção 3 traz os testes realizados para verificar a solução do trabalho, bem como a saída gerada
- A seção 4 apresenta uma breve conclusão sobre o trabalho.
- E por fim a seção 5 traz as referências bibliográficas.

2. Implementação

2.1. Modularização e Encapsulamento

Como dito na secção anterior, a implementação da solução gira em torno das entidades Pesquisador, Veículos de Comunicação e Artigo, estas foram mapeadas através da implementação de classes Java, onde seus atributos e métodos identificam-nas. Porém, apenas elas não são o suficiente para a resolução do problema, outras classes dependentes delas são necessárias para que a solução obtivesse êxito.

Das classes-entidades citadas acima, uma merece atenção especial, a entidade Artigo, pois ela se relaciona diretamente com as classes *Veículos de Comunicação* e *Pesquisador*, além disso é preciso identificar as citações feitas nos artigos em questão. Afim de modularizar o código e facilitar seu entendimento as classes foram estruturadas da seguinte forma:

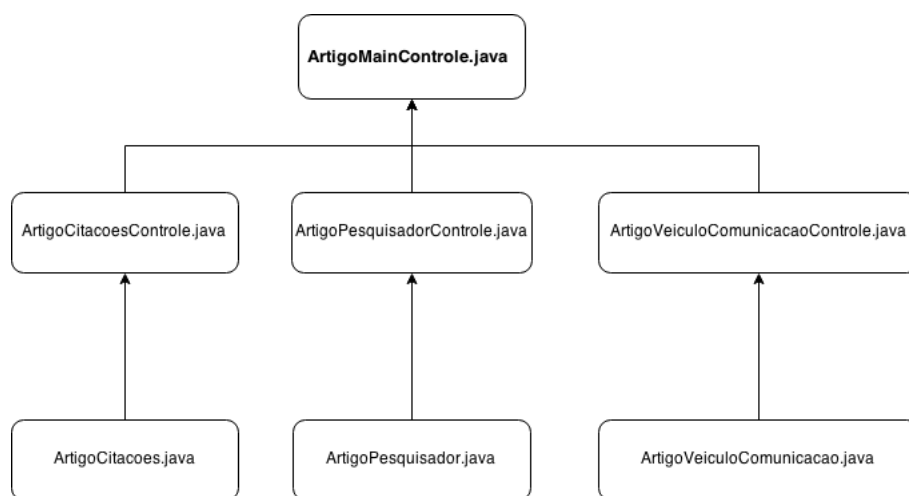


Figure 1: Estrutura de Dependência de Artigo

Como pode ser visto no diagrama acima, a classe **ArtigoMainControle.java** faz o encapsulamento das demais classes, fornecendo o conjunto de todas as operações necessárias envolvendo um Artigo, não sendo necessário assim nenhuma alteração nas classes bases.

A modularização das entidades Pesquisador e Veiculo de Comunicação foram feitas de forma similar a classe Artigo, porém de forma mais simples pois essas classes não tem dependências de demais classes, o que facilita o reuso e a manutenção destas.

2.2. Conceitos OO

Para a modularização e o encapsulamento das classes fossem possíveis, foi necessária a implementação dos conceitos de *Orientação a Objeto*.

O conceito utilizado para realizar o relacionamento entre as classes foi o de **Composição** onde temos uma classe que funciona como o *todo* e classes que funcionam como uma parte deste todo, por exemplo, a classe **PesquisadorControle** (responsável pela implementação dos métodos referentes a entidade Pesquisador) é o todo e a classe **Pesquisador** é parte da classe **PesquisadorControle**. Foi utilizado também o conceito de Objeto, que foi utilizado para que o acesso as classes dependentes fosse possível.

2.3. Calculos e Execução

O principal objetivo do trabalho era fornecer uma lista com a popularidade de cada pesquisador, o fator de impacto de cada veículo de comunicação e a pontuação de cada artigo. Para resolver esses problemas, foi criada uma classe específica que processa todos esses dados e gera os dados solicitados.

Esta classe é composta por objetos de todas as classes **Controle** que fornecem as operações de suas respectivas entidades, tornando possível o processamento dos dados e o fornecimento das respostas.

Para que a solução fosse executada, foi desenvolvida uma classe principal, **MainClass**, que contém um construtor e um objeto da classe **Resultado** que realiza os cálculos necessários para solucionar o problema proposto.

3. Testes

A solução proposta resolve o problema proposto de forma eficaz e eficiente, gerando a saída esperada, com os dados de Popularidade do pesquisador, fator de impacto dos veículos de comunicação e a pontuação dos artigos de forma ordenada. Segue abaixo a saída gerada pelo programa

popularidade_pesquisador.txt	7;1.4286
	8;0.0000
1;731.0000	9;0.0000
2;35.5000	10;0.0000
3;64.0000	11;1.3000
4;341.5000	12;0.0000
5;509.5000	13;0.0000
	14;0.0000
	15;0.0000
fatorImpacto_veiculo.txt	16;0.0000
	17;0.0000
1;1.4286	18;0.0000
2;1.0000	19;0.0000
3;1.3000	20;2.8571
	21;0.0000
pontuacao_artigo.txt	22;0.0000
	23;1.3000
1;0.0000	24;1.3000
2;0.0000	25;0.0000
3;0.0000	26;0.0000
4;0.0000	
5;0.0000	
6;0.0000	

4. Conclusão

Observando o que descrito acima no relatório, conclui-se que o êxito no desenvolvimento da solução do trabalho foi obtido, tanto em relação ao resultado prático do algoritmo quanto aos conhecimento relacionado a modularização de código e orientação a objeto foram absorvidos.

A principal dificuldade encontrada no decorrer do desenvolvimento da solução foi o paradigma de programação trazido pela orientação a objeto, muito diferente da programação estruturada, na qual estávamos familiarizados. Outra dificuldade foi encontrar a melhor forma de modularizar o código utilizando as técnicas de OO.

Foi um trabalho muito prazeroso de fazer, pois fomos expostos a um novo paradigma de programação nunca antes utilizado, fazendo com que nosso leque de conhecimento aumentasse com algo extremamente relevante para nossa formação profissional.

5. Referências bibliográficas