

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DO RIO
GRANDE DO NORTE
CAMPUS NATAL – CENTRAL
CURSO SUPERIOR DE TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE
SISTEMAS
PROGRAMAÇÃO ORIENTADA A OBJETOS

TIAGO CAIO DE OLIVEIRA LISBOA
RUTE FERNANDES DO NASCIMENTO

RELATORIO DE DESENVOLVIMENTO DA APLICAÇÃO JFINANCAS

NATAL/RN
DEZEMBRO DE 2017

TIAGO CAIO DE OLIVEIRA LISBOA
RUTE FERNANDES DO NASCIMENTO

PROGRAMAÇÃO ORIENTADA A OBJETOS

Relatorio de atividade apresentado ao Prof. Plácido A. Souza Neto, ministrante da disciplina de Programação Orientada a Objetos, do Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Norte, no campus Natal – Central, como atividade avaliativa.

NATAL/RN
DEZEMBRO DE 2017

RESUMO

Esse documento contém o detalhamento do processo de desenvolvimento da aplicação JFinanças. Contendo uma visão geral do processo de desenvolvimento da mesma, a descrição dos seus módulos e as informações necessárias para o entender seu funcionamento.

Palavras Chaves: Java, Programação Orientada a Objetos, Avaliação.

Sumário

1 INTRODUÇÃO.....	4
2 MODELAGEM DA APLICAÇÃO.....	5
3 PROCESSO DE DESENVOLVIMENTO.....	8
4 CONSIDERAÇÕES FINAIS.....	9
4.1 Trabalhos Futuros.....	9

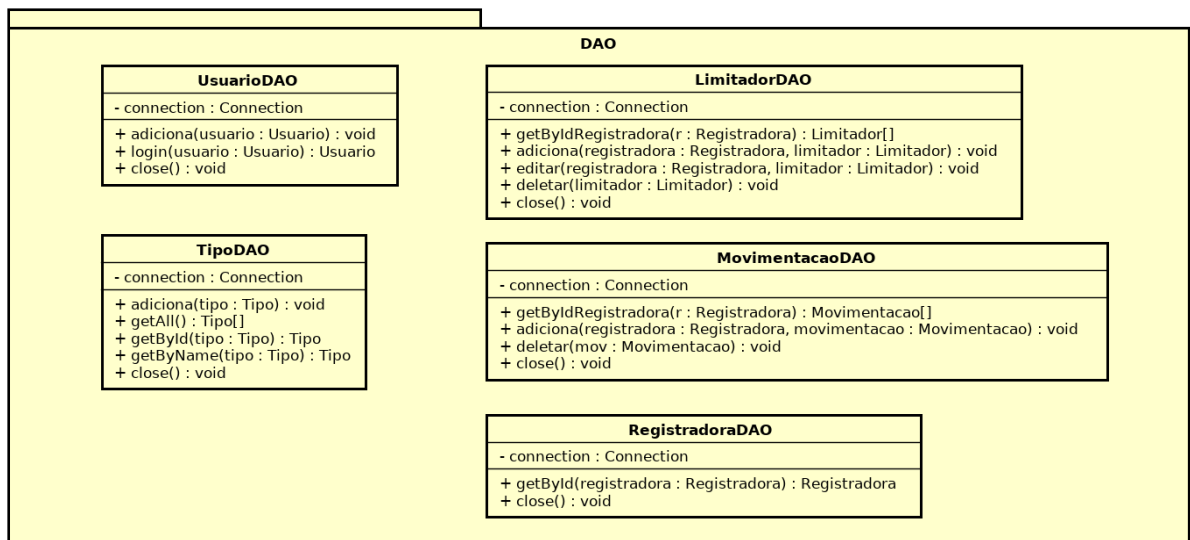
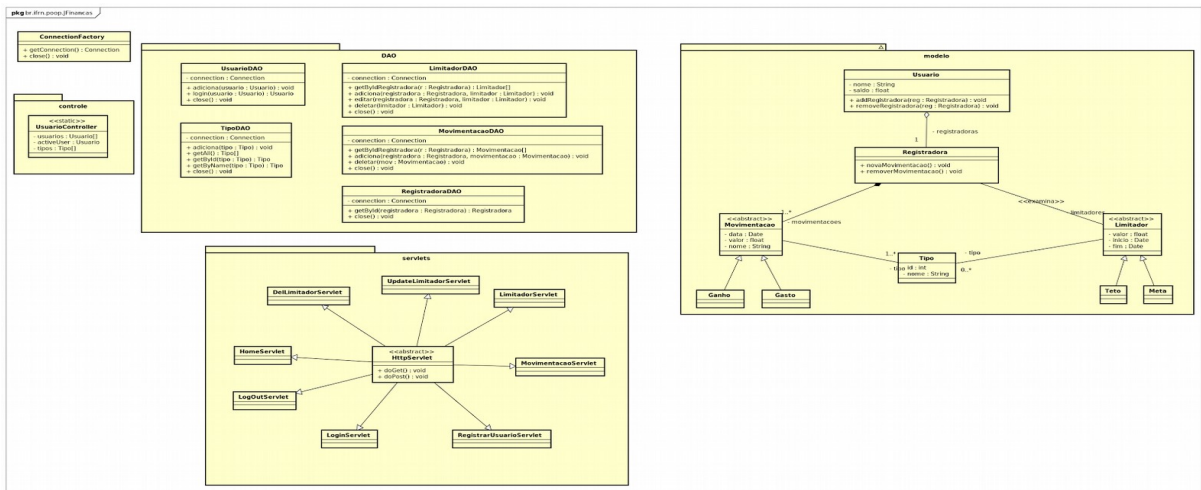
1 INTRODUÇÃO

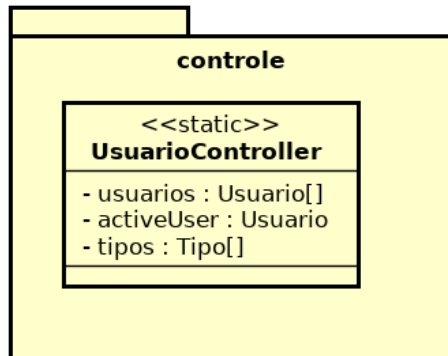
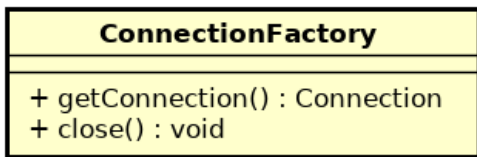
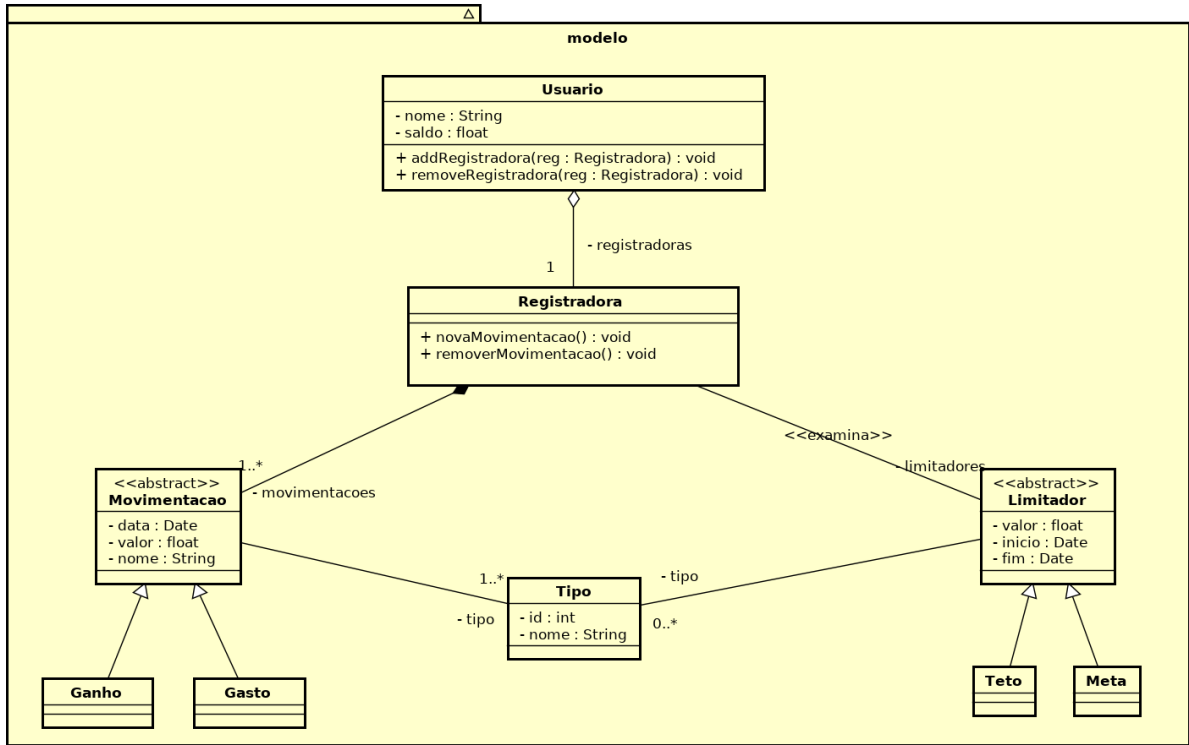
No processo de aprendizagem de programação, é imprescindível a aplicação de atividades práticas, a fim de avaliar os conhecimentos teóricos adquiridos pelos alunos durante as aulas. Partindo dessa premissa, foi proposto que os alunos da turma 2017.1M do curso superior em TADS (Tecnologia em Análise e Desenvolvimento de Sistemas) no IFRN CNAT (Instituto Federal de Educação, Ciências e Tecnologia Campus Natal-Central) cursantes da disciplina de POO (Programação Orientada a Objetos) desenvolvessem um sistema utilizando todos os conceitos aprendidos na disciplina supracitada.

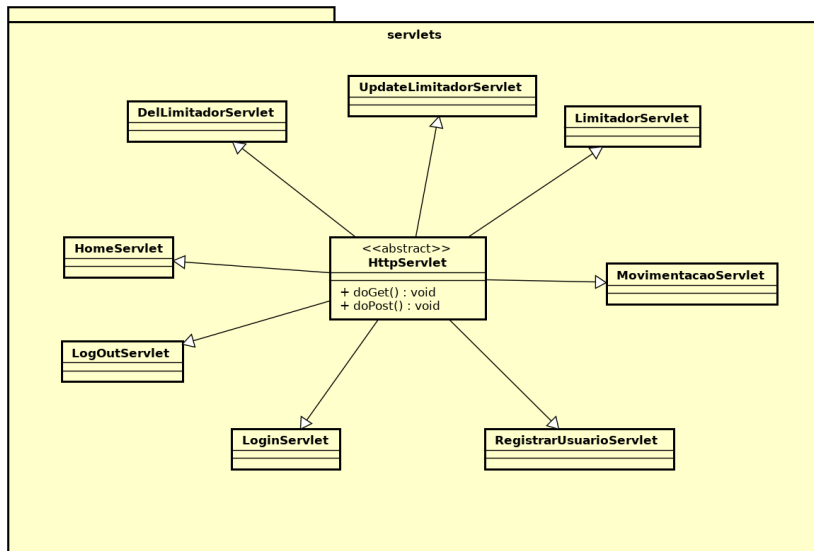
Foi nesse contexto que surgiu a ideia de desenvolver uma aplicação para controle de finanças domésticas, o JFinanças. Seu nome vem de Java (o J) e o que a aplicação deve gerenciar (Finanças). Ele se trata de um sistema de um SPT (Sistema de Processamento de Transações) com características de um SAD (Sistema de Apoio a Decisão).

2 MODELAGEM DA APLICAÇÃO

Uma vez que foi escrita em Java, a aplicação segue um modelo estritamente orientado a objetos e é dividida em pacotes. Ela segue brevemente os *Design Pattern's* MVC (*Model, View, Control*), *Object Factory* e DAO (*Data Access Objects*). Desse modo, seus pacotes são: modelo, onde está o caracterizado o domínio da aplicação; controle, onde originalmente estaria as regras de negócio, mas apenas contém a classe estática *UsuarioController* que serve como a seção de usuário; DAO, onde estão as classes de acesso ao banco de dados; e servlets, onde estão os *HttpServlets*, classes do java que controlam requisições HTTP (*Hypertext Transfer Protocol*). As *Views* (A visão do sistema) estão localizadas na pasta *WebContent*, e são páginas JSP (*JavaServer Pages*).







- WebContent
 - css
 - fontawesome-free-5.0.1
 - js
 - META-INF
 - WEB-INF
 - Extrato.jsp
 - index.jsp
 - Limitadores.jsp
 - login.jsp
 - NovaMovimentacao.jsp
 - novoLimitador.jsp
 - registrar.jsp
 - updateLimitador.jsp
 - Usuario.jsp

3 PROCESSO DE DESENVOLVIMENTO

Esse projeto foi desenvolvido em conjunto. Os membros do grupo se reuniam para poderem implementar a aplicação juntos, assim um corrigia as falhas do outro e se auxiliavam mutuamente, balanceando assim a diferença de experiências e garantindo que ambos participassem de todos os processos de desenvolvimento.

4 CONSIDERAÇÕES FINAIS

Todo o processo de desenvolvimento foi bastante produtivo. Nele foi possível aprimorar os conhecimentos sobre a plataforma Java e os conceitos de Programação Orientada a Objetos de ambos os membros do grupo.

4.1 Trabalhos Futuros

Apesar de finalizado, existem diversas melhorias que poderiam ser aplicadas a esse projeto. Uma delas é melhorar a modelagem do sistema, onde a classe Registradora poderia ser representada por uma relação direta entre o usuário e as suas movimentações e seus limitadores.

Também é possível melhorar a utilização de rotas, necessitando assim de menos HttpServlets. E utilizar a sessão do próprio HttpServlet, ao invés de uma classe estática.