

## Documento de Decisão Arquitetural e de Ferramentas

### Introdução

A implementação proposta não levou em conta a necessidade de persistência e desenvolvimento de interface com o usuário.

O foco principal foi prover alta coesão e baixo acoplamento. Preocupação em suprir a cobertura dos casos de testes levantados no documento passado como referência.

### Problema

O problema principal é prover uma solução para os cálculos dos diferentes tipos de taxa, efetivar a transferência e buscar as informações de transferências agendadas.

### Arquitetura

Para a solução principal de cálculos das taxas foi utilizado o *Strategy Pattern* que define algoritmos e encapsula cada um deles e os torna intercambiáveis.

A camada de serviço possui o método de agendamento que utiliza a solução *strategy* para o cálculo da taxa e agendamento da transferência.

O pacote DAO possui a futura interface e implementação com o banco de dados requisitando buscas e persistências. Esta camada foi mockada nos testes.

O pacote Model contém a entidade envolvida.

O pacote de testes provê a cobertura completa do pacote *strategy* e do pacote *service*. São testados unitariamente todos os casos propostos e utiliza-se mocks para o DAO.

O pacote Enum possui o enum *TipoOperacao* que retorna a instância correta do *strategy* que executará o cálculo da taxa.

Todos os pacotes possuem suas interfaces e estão preparados para a injeção de dependência.

### Versões e Ferramentas

Utilizado as seguintes ferramentas e frameworks:

Eclipse Mars - versão Mars.1 Release (4.5.1)

JRE 1.8.0\_66

JUnit 4.11

Mockito 2.0.2

Maven 3.3.9