

Programação Orientada a Objetos com .Net

Engenharia de Software.Net

Edson José da Rocha Junior

Sumário

Escopo do Projeto 2

Modelo Conceitual..... 3

Abordagem Domain-Driven Desing (DDD)..... 4

Arquitetura do Projeto 7

 Modelo 7

 Repositório 7

 Serviço 7

 Teste 7

Padrão de Projeto..... 7

Cobertura de Código..... 8

Escopo do Projeto

A consultoria XPTO foi contratada para o desenvolvimento de um sistema simples de controle bancário.

O sistema deve permitir o cadastramento de um novo cliente com seus dados pessoais básicos e gerar, após a aceitação do banco, um número de conta corrente. Dependendo do perfil do cliente a conta corrente pode ser Normal ou Especial, e o que difere uma da outra é que a Conta Especial possui um limite de crédito pré-definido para o cliente. O Cliente pode realizar transações bancárias como depósito, saque e transferência entre contas. O sistema deve ser capaz de gerar um extrato de movimentação financeira da conta do cliente dos últimos 30 dias.

A conta do cliente pode ser fechada a qualquer momento desde que o cliente não possua nenhum crédito ou débito em sua conta. A conta do cliente também pode ser bloqueada por ordem judicial, não permitindo ao cliente mais nenhuma transação bancária durante o período de bloqueio e somente será desbloqueada quando houver uma liberação judicial.

A empresa XPTO tem 6 semanas para entregar uma prova de conceito do sistema acima descrito atendendo aos seguintes termos contratuais de qualidade:

- Desenvolver o sistema utilizando o paradigma Orientado a Objeto;
- Desenvolver o sistema utilizando as melhores práticas de desenvolvimento de sistemas e engenharia de software;
- Possuir uma cobertura de código testado unitariamente maior que 95%;
- Desenvolver modelo de domínio que atenda a complexidade do negócio proposto;
- Possuir armazenamento de objetos em tempo de execução para teste;
- Possuir uma documentação explicativa da arquitetura inicial do projeto;
- Possuir uma documentação sinalizando os padrões utilizados e justificando sua utilização;

Ao final destas 6 semanas uma ferramenta analisará o código no Github e verificará se os requisitos de qualidade estão sendo atendidos.

Modelo Conceitual

O Modelo conceitual é a forma natural dos fatos e estão mais próximas da realidade do ambiente do cliente. Nessa fase são levantados os dados que darão suporte à construção de todo o modelo.

Partindo desse conceito, abaixo foi desenvolvido o segundo modelo conceitual para o projeto proposto.

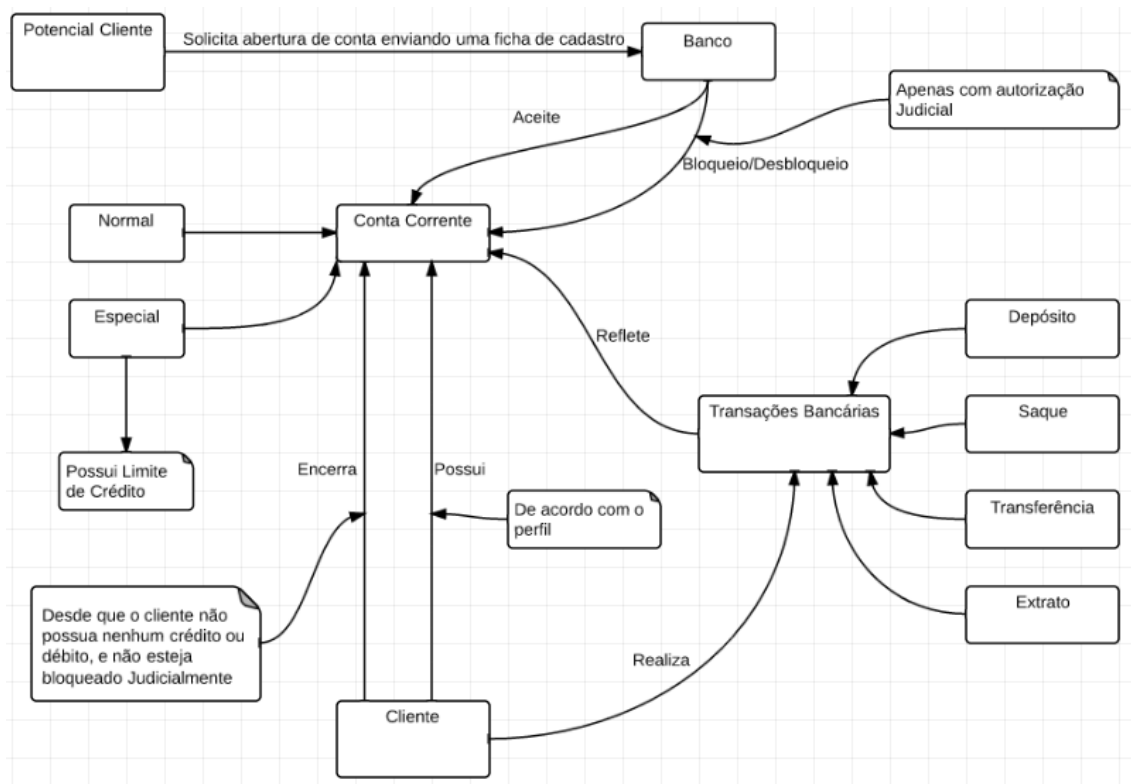


Figura 1 – Modelo Conceitual

Abordagem Domain-Driven Design (DDD)

A abordagem utilizada para o projeto bancário foi o DDD, que é uma abordagem de desenvolvimento de software que reúne um conjunto de conceitos, princípios e técnicas cujo foco está no domínio e na lógica do domínio com o objetivo de criar um modelo do domínio.

Decidi criar um modelo usando Model Driven Design (MDD), inicialmente, isolando o modelo de domínio das demais partes que compõem o sistema. Essa separação dividiu a minha aplicação em quatro partes:

- Interface de Usuário – Responsável pela exibição de informações do sistema ao usuário;
- Aplicação – Responsável por conectar a interface de usuário às camadas inferiores;
- Domínio – Representa os conceitos, regras e lógicas de negócio, onde o foco de DDD está nessa camada;
- Infraestrutura – Responsável por fornecer recursos técnicos que darão suporte às camadas superiores, por exemplo, persistência de dados, conexões com bancos de dados entre outros;

Depois de dividir o sistema em camadas, me concentrei somente na camada de domínio que é o foco do trabalho proposto em sala de aula. Utilizei padrões propostos em DDD chamados blocos de construção, para fazer a modelagem, sendo eles: entidades, objetos de valores, agregados, fábricas, serviços, repositórios, módulos.

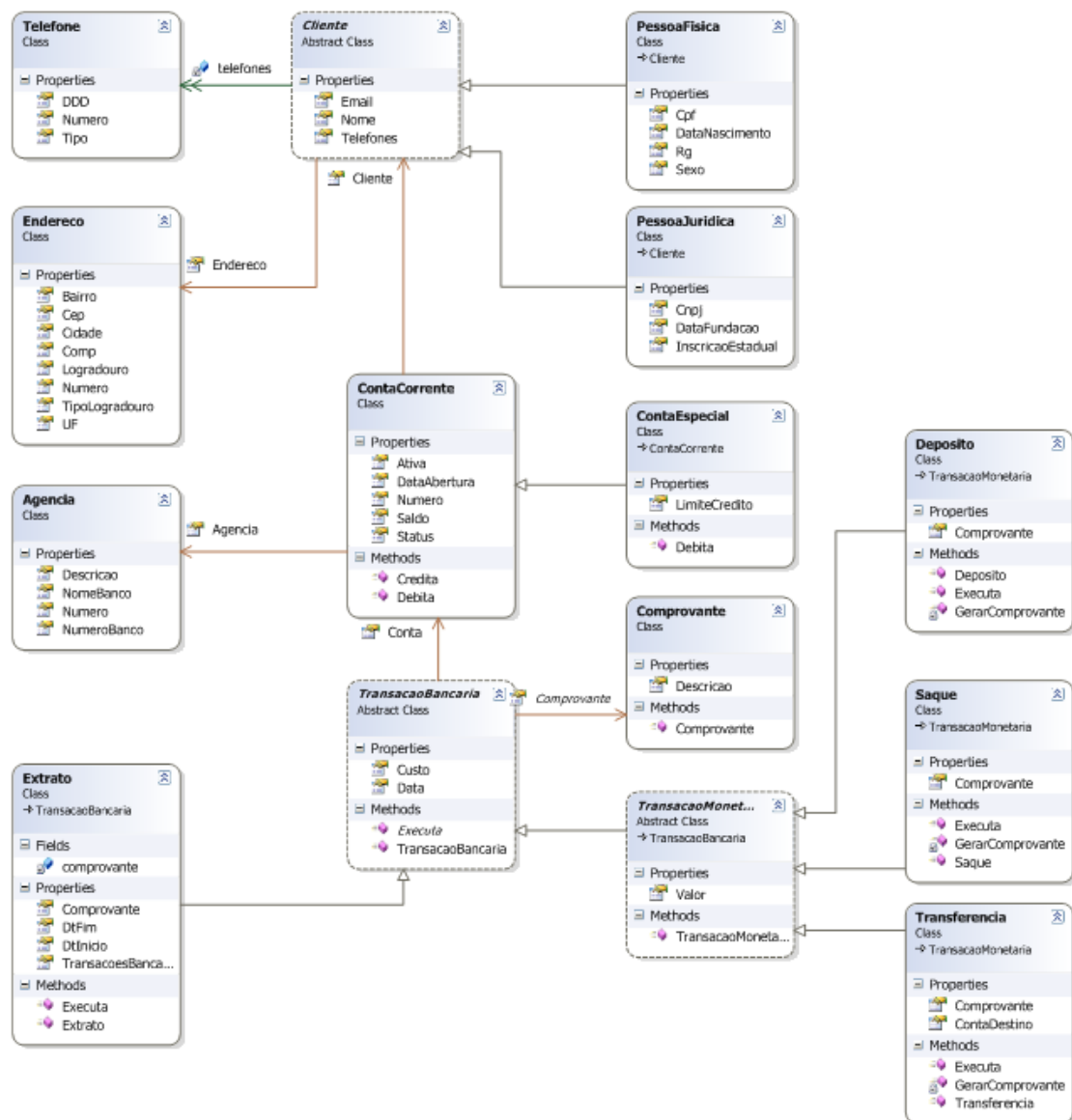


Figura 2 – Diagrama de Classe – Visual Studio 2010

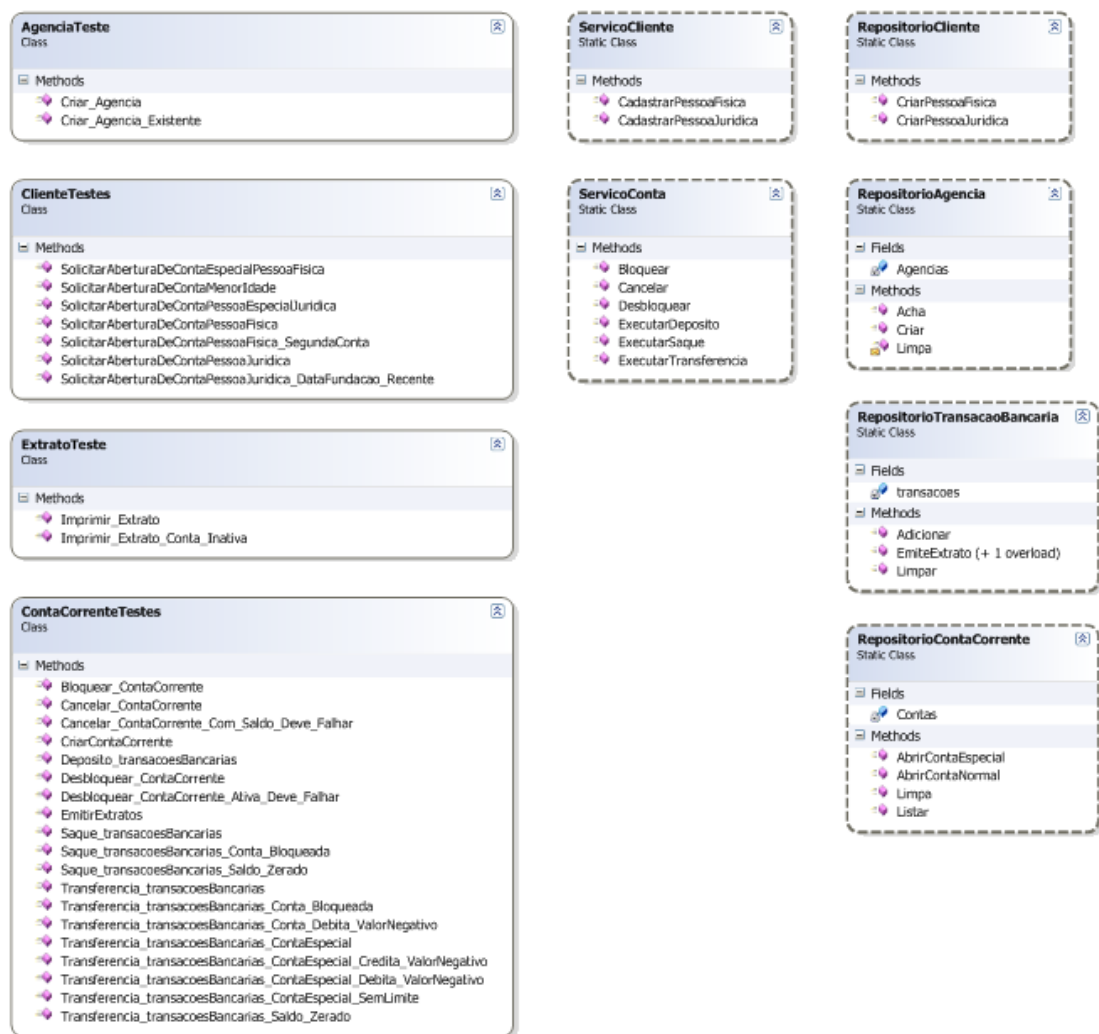


Figura 3 – Diagrama com os testes, serviços e repositórios – Visual Studio 2010.

Arquitetura do Projeto

O projeto está dividido com a seguinte arquitetura:

Modelo

Classes de objetos, elementos do domínio que possuem ciclo de vida dentro da aplicação.

Repositório

Classes responsáveis por administrar o ciclo de vida dos outros objetos. Centralizam operações de criação, alteração e remoção de objetos.

Serviço

Classes que contêm lógica de negócio, mas que não pertence a nenhuma entidade ou objetos de valor.

Teste

Classes de teste de unidade e asserções no código que validam, após a chamada dos comandos, as alterações de estado esperadas.

Padrão de Projeto

Para a realização do projeto foi utilizado o padrão singleton que tem como objetivo garantir que existirá apenas uma instância de um objeto. Utilizei esse padrão em algumas partes do projeto utilizando classes estáticas, porque com o uso demasiado do singleton leva a solução ter forte dependência entre os objetos e testabilidade fraca.

Cobertura de Código

Para informar o percentual de cobertura de código do projeto bancário, utilizei a ferramenta NCover. O Projeto possui uma cobertura de código testado de 99%, conforme figura 4.

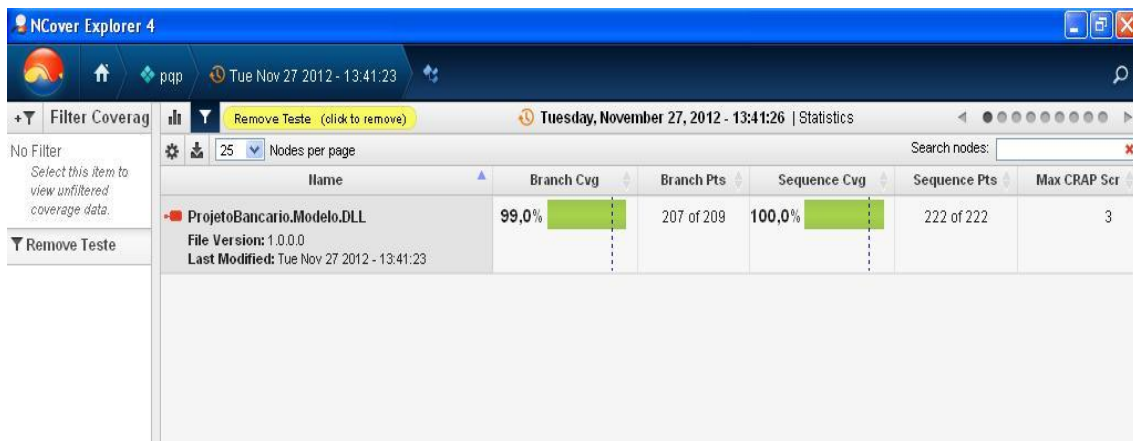


Figura 4 – Cobertura de código - NCover