

**UNIVERSIDADE DO ESTADO DE SANTA CATARINA**

**MARLON HENRY SCHWEIGERT**

**ANÁLISE DE SISTEMAS: RELATÓRIO FINAL**

RELATÓRIO FINAL DESENVOLVIDO PELO ACADÊMICO  
MARLON HENRY SCHWEIGERT PARA A MATÉRIA DE  
ANÁLISE DE SISTEMAS LECIONADO PELO PROFESSOR  
MESTRE LEANDRO ISRAEL PINTO

**JOINVILLE**

**2016**

# INTRODUÇÃO

O trabalho resume-se em desenvolver um sistema bancário simplificado enbasado nos diagramas desenvolvidos durante o semestre letivo para a matéria de Análise e Modelagem de Sistemas.

O sistema escolhido pelo acadêmico é um sistema funcional bancário, podendo realizar consultas no banco de dados, cadastrar clientes e realizar operações básicas no banco.

Ao final, espera-se utilizar todos os diagramas para facilitar o desenvolvimento do devido sistema, e comprovar a eficiência de documentação e planejamento preliminar de projetos de software.

O objetivo do acadêmico é desenvolver uma biblioteca de network idealizada pelo mesmo, auxiliada pelo professor Doutor Omir e professor Doutor Charles em conversas extra classe para utilização em seu projeto particular.

## PLANEJAMENTO

### Levantamento de Requisitos

#### Sistema Bancário

O projeto trata do desenvolvimento de dois softwares para um sistema bancário: o software para o caixa eletrônico que será utilizado pelos clientes do banco e um sistema administrativo que será utilizado por funcionários. O sistema deverá ser seguro, prático e rápido.

#### Requisitos Funcionais

- Acesso controlado por credenciais. Nenhuma função poderá ser feita sem credenciais.

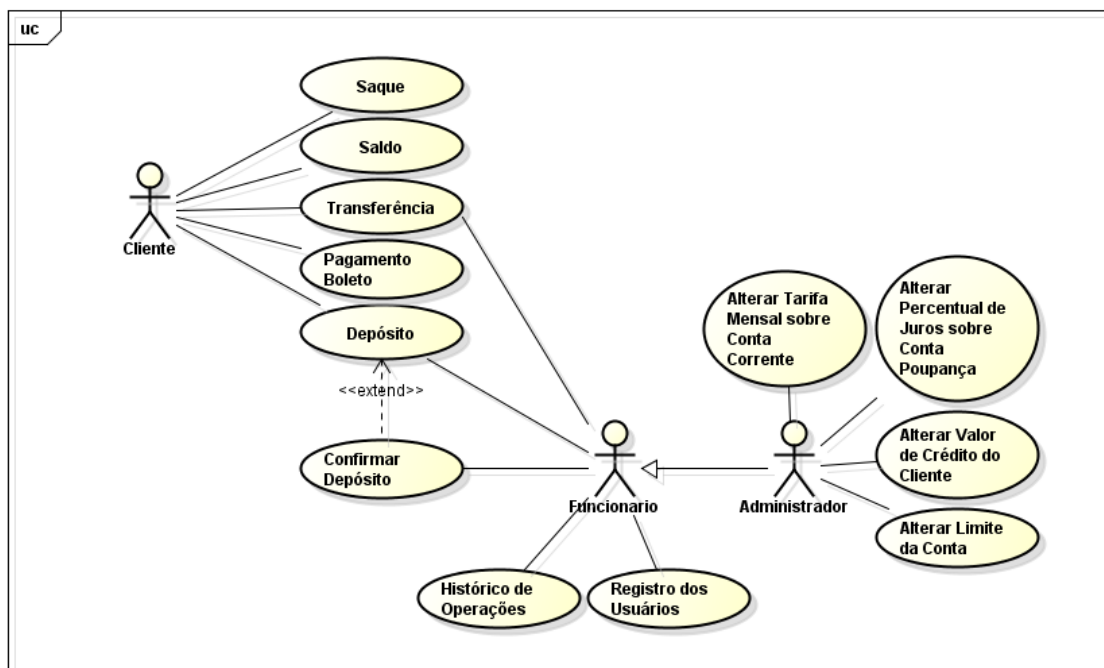
- Consulta de saldo da conta corrente e conta poupança.
- Realiza saque na conta corrente e na conta poupança respeitando o limite diário.
- Realiza um depósito em dinheiro
- Realizar depósitos não automáticos via caixa eletrônico. Será confirmado pelo sistema administrativo.
- Realiza transferência entre contas: O cliente deverá informar de qual conta (corrente ou poupança), qual a quantia que deverá ser transferida e a conta que deverá receber. Além disso, ele poderá escolher a data de transferência. Não poderá ser escolhida uma data retroativa; Se a data atual for escolhida, então a transferência é feita no momento de confirmação. Caso posterior, o sistema criará um log de transferências para efetuar o pagamento depois.
- O sistema de pagamento de boletos deverá pedir o código do boleto e efetuar as operações normalmente.
- O sistema do funcionário deverá ser restrito pelo número de matrícula e senha.
- Existe um funcionário administrador que poderá realizar algumas alterações em especial:
  - Visualizar, criar, alterar e excluir registro dos usuários.
  - Alterar os limites diários da conta corrente ou poupança.
  - Alterar o valor de crédito do cliente.
  - Alterar o percentual de juros sobre uma conta poupança.
  - Alterar a tarifa mensal sobre a conta corrente.

- Visualizar, criar e excluir registro dos clientes.
- Visualizar um histórico de operações, podendo filtrar os resultados por cliente e/ou dia de ação e/ou tipo de operação.
- Realizar a conferência de depósitos: O funcionário deverá depositar a quantia real que existe no envelope.
- Realizar pagamento de boletos e transferência entre contas: Similar ao cliente, mas sem limites de transferência.

### **Requisitos não-Funcionais**

- A Aplicação deverá ser multi plataforma (Linux / Windows)
- O Caixa eletrônico requer interface gráfica simplificada.
- Restrições
- O valor inicial da tarifa mensal da conta corrente é de R\$2,45. O juros na poupança deverá ser aplicado diariamente, com valor de 0,6114% ao mês.

## Diagrama de Casos de Uso



Nome do caso de uso	Pagamento de boleto
Caso de uso geral	Realizar o pagamento de boletos bancários através do caixa eletrônico
Atores	Cliente
Resumo	Este caso de uso descreve as etapas percorridas pelo sistema para efetuaro pagamento de boletos bancários
Pré-condições	1. Entrar no sistema como usuário 2. Possuir saldo suficiente para efetuar o pagamento.
Pós-condições	
Fluxo Principal	

<p>1. O sistema solicita do usuário o código do boleto bancário.</p> <p>2. O usuário envia para o sistema o código do boleto bancário.</p> <p>3. O sistema valida o código do boleto.</p> <p>4. O sistema verifica se a conta do cliente possui saldo suficiente para o pagamento do boleto.</p> <p>5. O sistema debita da conta do cliente o valor cobrado pelo boleto.</p> <p>6. O sistema envia uma mensagem de confirmação do pagamento.</p>
Fluxo Secundário
<p>3 a) Código inválido</p> <p>O sistema envia uma mensagem de erro ao usuário informando que o código informado não é válido.</p> <p>4 a) Saldo insuficiente</p> <p>O sistema envia uma mensagem de erro ao usuário informando que a conta não possui saldo suficiente para efetuar o pagamento do boleto.</p>

Nome do caso de uso	Alterar Limite da Conta
Caso de uso geral	Alterar o valor do limite da conta bancária
Atores	Administrador
Resumo	Este caso de uso descreve as etapas percorridas para realizar a alteração do limite da conta bancária de um usuário

Pré-condições	1. Estar como administrador do sistema interno.
Pós-condições	
Fluxo principal	
1. O administrador solicita a alteração do limite da conta. 2. O sistema verifica o nível de permissão do usuário. 3. O sistema verifica o limite atual da conta. 4. O sistema solicita o novo valor do limite da conta. 5. O administrador insere o novo valor do limite da conta. 6. O sistema lê e altera o valor do limite da conta. 7. O sistema envia para a tela uma mensagem de confirmação de alteração no valor do limite da conta.	
Fluxo Alternativo	
2 a) Permissão insuficiente  O sistema emite uma mensagem informando que o usuário não tem permissão suficiente para realizar a tarefa.  6 a) Caracteres inválidos  O sistema emite uma mensagem de erro informando o uso de caracteres inválidos.  6 b) Campo em branco  O sistema emite uma mensagem de erro informando que o campo não pode	

estar em branco.

Nome do caso de uso	Transferência
Caso de uso geral	Realizar transferências entre contas em um caixa eletrônico
Atores	Cliente
Resumo	Este caso de uso descreve as etapas percorridas pelo sistema para efetuar uma transferência entre contas
Pré-condições	1. Entrar no sistema como usuário 2. Possuir saldo suficiente para efetuar a transferência
Pós-condições	
Fluxo Principal	
1. O sistema solicita a conta de destino 2. O sistema solicita a quantidade e a conta de Saída (Corrente e Poupança) 3. O sistema solicita pede a confirmação da senha 4. O sistema solicita a data para efetuar esta operação 5. O sistema efetua o agendamento da transferência	
Fluxo Secundário	
1 a) Conta Inexistente  O sistema envia uma mensagem de erro ao usuário informando que a conta	



informada não é válida.

#### 2 a) Saldo insuficiente

O sistema envia uma mensagem de erro ao usuário informando que a conta não possui saldo suficiente para efetuar a transferência.

#### 3 a) Senha Errada

O sistema envia uma mensagem de erro ao usuário informando que a senha está errada. A operação é cancelada por segurança.

Nome do caso de uso	Confirmar Depósito
Caso de uso geral	Realiza a confirmação de Depósito
Atores	Funcionário
Resumo	Este caso de uso descreve as etapas percorridas pelo sistema para efetuar a confirmação de Depósito
Pré-condições	1. Entrar no sistema como funcionário
Pós-condições	
Fluxo Principal	
1. O sistema exibe todos os depósitos não confirmados	
2. O sistema requer a quantia depositada.	

3. O sistema exige a senha de confirmação do funcionário.
Fluxo Secundário
<p>3 a) Senha incorreta</p> <p>O sistema exibe uma mensagem de erro informando que a senha digitada está incorreta e cancela a operação.</p>

Nome do caso de uso	Saque
Caso de uso geral	Retirar dinheiro da conta
Atores	Cliente
Resumo	Este caso de uso descreve os passos para a realização de um saque na conta do cliente do banco.
Pré-condições	1. Estar logado no sistema cliente.
Pós-condições	
Fluxo Principal	
<p>O cliente informa o valor do saque.</p> <p>O sistema verifica se o cliente possui o valor informado na sua conta.</p> <p>O sistema manda uma solicitação para a máquina ejetar o dinheiro.</p> <p>O sistema envia uma mensagem confirmando o saque.</p>	
Fluxo Secundário	

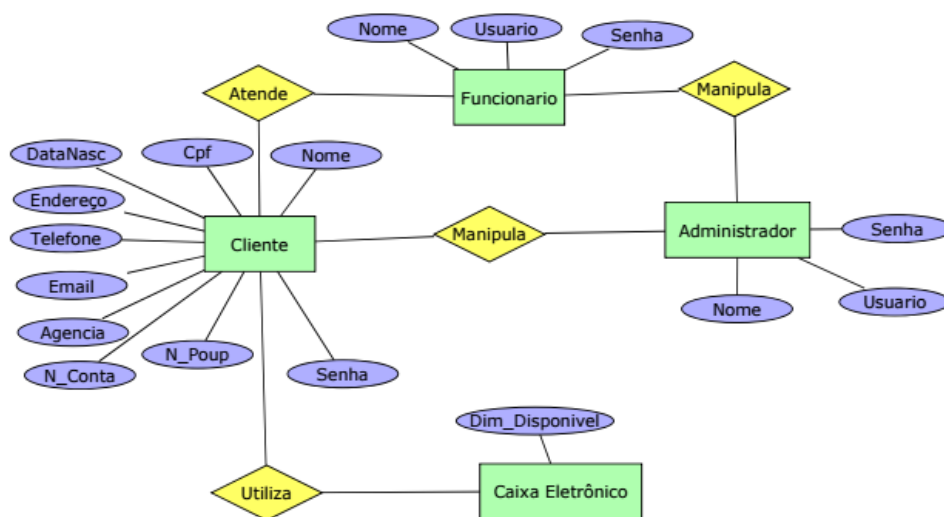
## 2 a) Saldo insuficiente

O sistema emite uma mensagem informando que o cliente não possui saldo suficiente para realizar o saque.

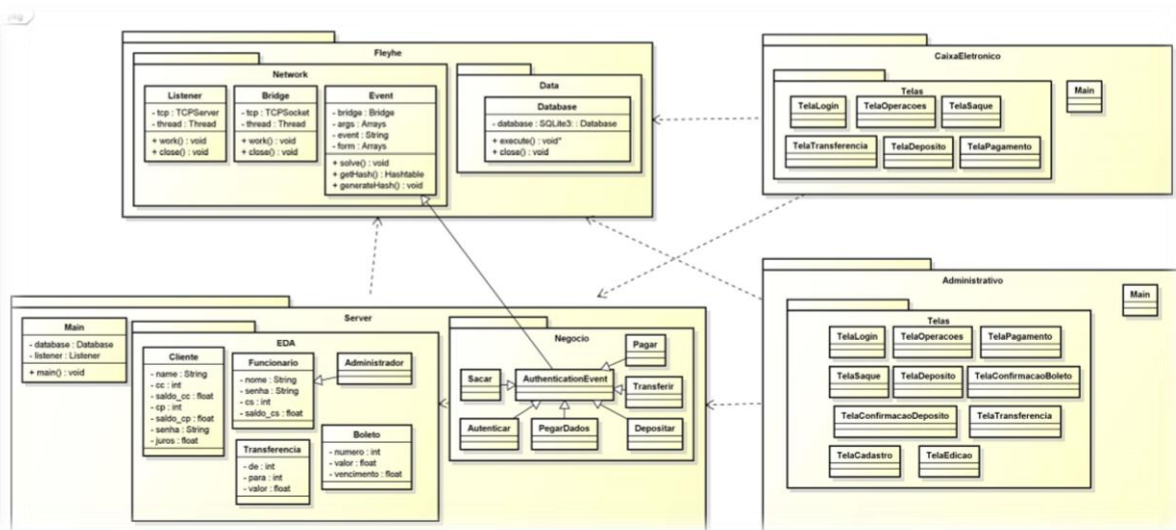
## 3 a) Notas insuficientes

O sistema emite uma mensagem informando que o caixa eletrônico não possui notas suficientes para completar o valor solicitado.

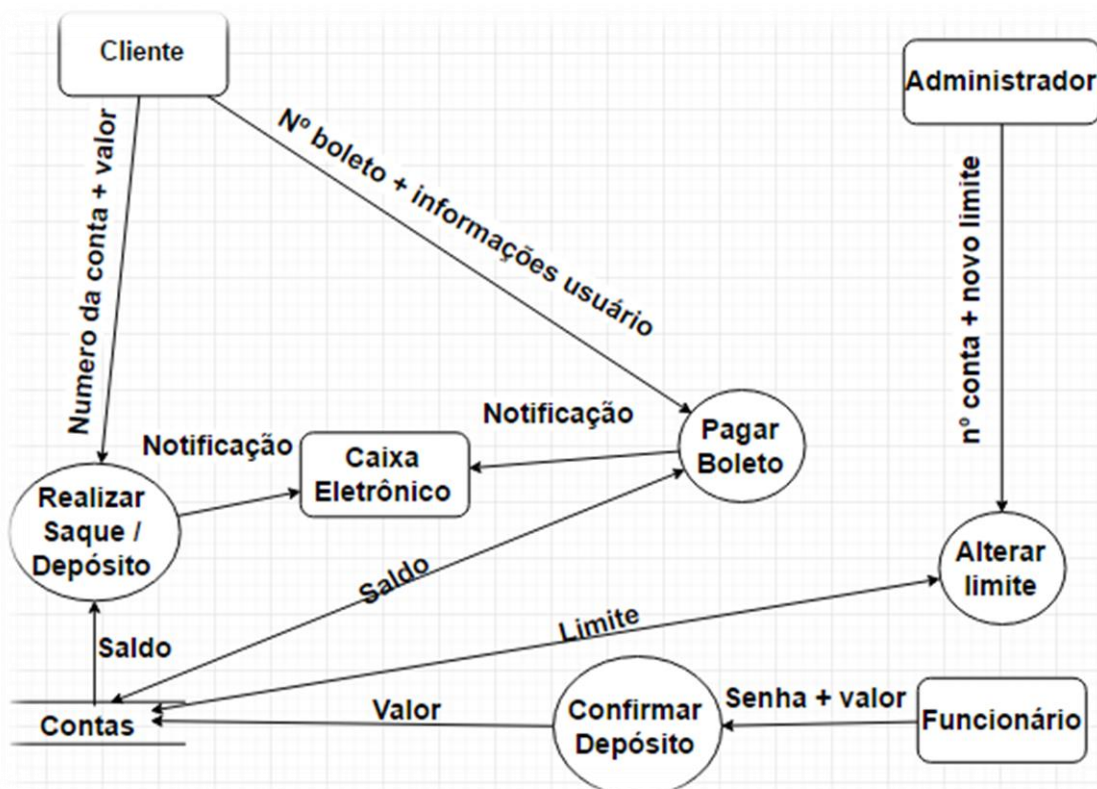
## Diagrama Entidade Relacionamento



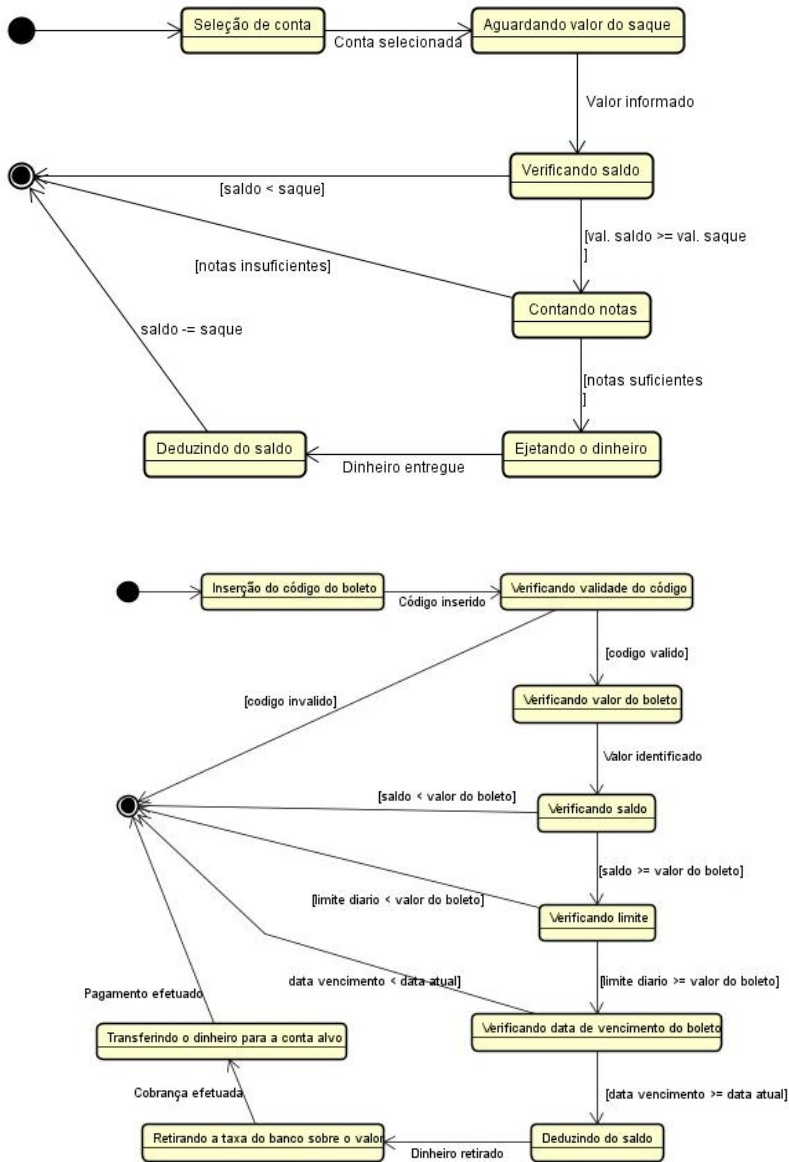
## Diagrama de Classes Completo



## Fluxo de Dados



## Diagrama de Estados



## Código Fonte

Utilizando o diagrama de Classes Completo, iniciei o desenvolvimento pela biblioteca de Network, batizada de Fleyhe. A biblioteca de network foi idealizada

junto ao Professor Doutor Charles e Professor Doutor Omir, ambos do Departamento de Ciência da Computação.

O algoritmo se baseia em comunicar-se via TCP/IP para realizar chamadas de eventos. Estes eventos são todos subclasses de algum evento mestre, no caso, Event. Ao criar algum listener, ele procura todos os subfilhos, salva em uma hash pelo seu nome e então passa os argumentos e o cliente que chamou este evento para efetuar uma interface rápida de comunicação.

A biblioteca de networking foi publicada para a comunidade, tendo até o dia 13 de junho de 2016, 230 downloads (<https://rubygems.org/gems/Fleyhe>). Também foi publicada uma documentação detalhada pela comunidade (disponível em <http://www.rubydoc.info/gems/Fleyhe/1.0.1>).

A mesma biblioteca já implementa o sistema de Banco de Dados. O mesmo precisa do sqlite3 instalado na máquina devidamente para o seu funcionamento.

O Código fonte da biblioteca pode ser encontrado em <https://github.com/fleyhe/Fleyhe>.

O servidor foi implementado utilizando o sistema de network com eventos. Basicamente, ele executa consultas SQL no banco de dados, e retorna o resultado para o cliente, descrito pelo diagrama de eventos.

O cliente manipula a interface gráfica, tendo uma classe em especial para manipulação das chamadas de eventos chamada Request, dentro da biblioteca Network::Fleyhe.

O lado do cliente foi programado na Linguagem Csharp pela facilidade de desenvolvimento de formulários que convém de encontro com o conhecimento da linguagem pelo Acadêmico.

## **Conclusão**

Como relato final, conclue-se que os diagramas ajudam a poupar tempo e replanejamento do programador durante o tempo de desenvolvimento do projeto. O mesmo foi confirmado durante o próprio desenvolvimento do trabalho final.

Como objetivo final do acadêmico, a biblioteca foi publicada de forma OpenSource para a comunidade, possuindo diversos outros acadêmicos que já estão utilizando nas matérias de Sistemas Distribuidos e Rede de Computadores, além de outros diversos programadores da comunidade que estão dando continuidade com o projeto em sua documentação e complementação.