
Universidade Federal Fluminense

Mubank
Documento de Arquitetura de Software

Versão <0.2>

Mubank	Versão: 0.2
Documento de Arquitetura de Software	Data: 15/06/2022
<document identifier>	

Histórico da Revisão

Data	Versão	Descrição	Autor
18/03/2022	0.1	Estado inicial	Grupo
15/06/2022	0.2	Ajustes da arquitetura	Grupo

Mubank	Versão: 0.2
Documento de Arquitetura de Software	Data: 15/06/2022
<document identifier>	

Índice Analítico

1.	Introdução	4
1.1	Finalidade	4
2.	Metas e Restrições da Arquitetura	4
3.	Suposições e Dependências	4
4.	Requisitos Arquiteturalmente Significantes	4
5.	Decisões, Restrições e justificativas	4
6.	Mecanismos Arquiteturais	5
7.	Camadas da Arquitetura	5
8.	Visões da Arquitetura	5
9.	Qualidade	5
10.	Problemas de Projeto e Soluções encontradas	18

Mubank	Versão: 0.2
Documento de Arquitetura de Software	Data: 15/06/2022
<document identifier>	

Documento de Arquitetura de Software

1. Introdução

Finalidade:

- Desenvolver um sistema de um banco digital que descomplica sua vida financeira dos nossos clientes..

Escopo:

- Um cliente pode fazer login no sistema
- Um cliente pode fazer cadastro no sistema
- Um cliente pode editar suas informações cadastrais
- Um cliente pode fazer um saque
- Um cliente pode fazer um depósito
- Um cliente pode ver as movimentações feitas na sua conta
- Um cliente pode fazer uma transferência para outro cliente

1.1 Finalidade

Este documento oferece uma visão geral arquitetural abrangente do sistema, usando diversas visões arquiteturais para representar diferentes aspectos do sistema. O objetivo deste documento é capturar e comunicar as decisões arquiteturais significativas que foram tomadas em relação ao sistema.

2. Metas e Restrições da Arquitetura

O projeto Mubank possui as seguintes metas:

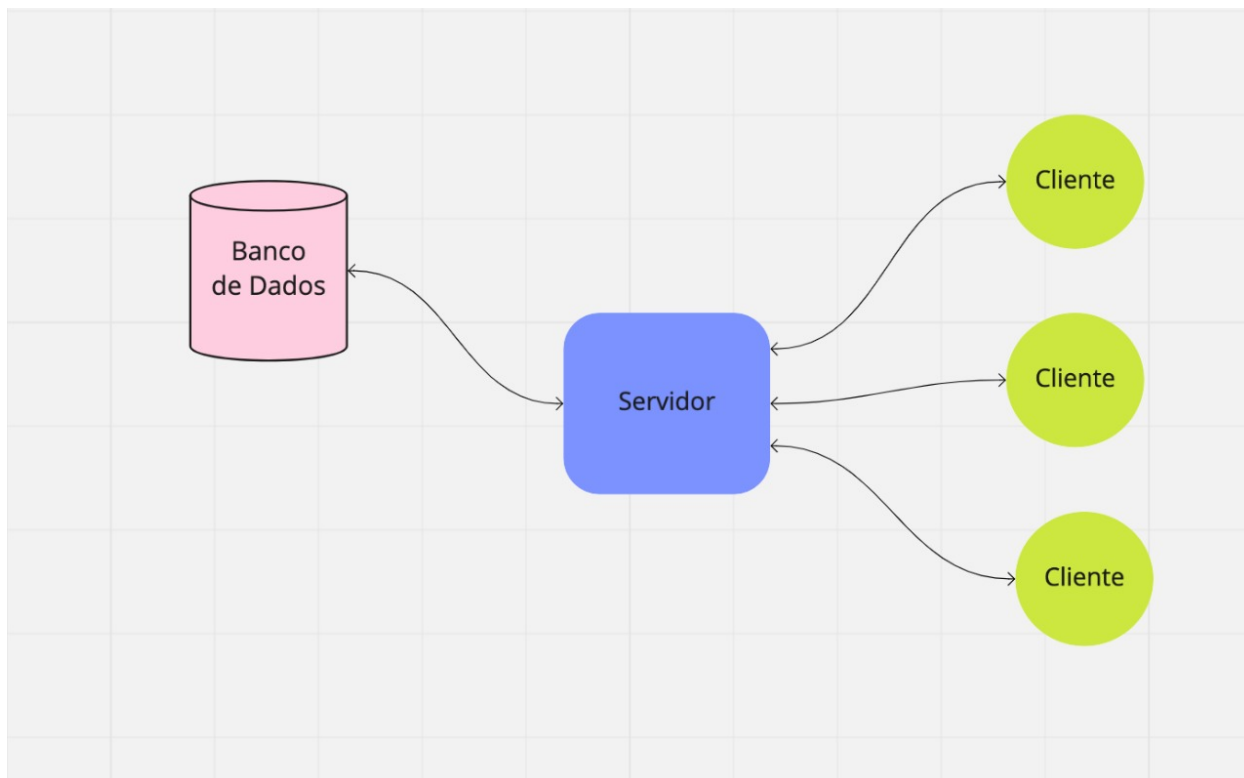
Suporte a diferentes tamanhos e proporções de tela (RNF04)

Suporte com os principais navegadores web da atualidade : Google Chrome, Microsoft Edge e Mozilla Firefox. (RNF02)

Suporte a diferentes sistemas operacionais como Windows, Linux e MacOS, utilizando a JVM. (RNF01)

A arquitetura do sistema deve ser implementada obedecendo o esquema conceitual abaixo:

Mubank	Versão: 0.2
Documento de Arquitetura de Software	Data: 15/06/2022
<document identifier>	



Um banco de dados SQL(PostgreSQL ou MySQL) central, que se comunica diretamente com o servidor. Este servidor, por sua vez, responde as requisições de todos os clientes a uma taxa mínima de 5 requisições por segundo.

O sistema deve estar disponível na nuvem, de modo a atingir o requisito de confiabilidade de estar 99% do tempo disponível.

3. Suposições e Dependências

- O sistema deverá manter os dados das transferências isolados e suas condições de corridas decorrentes do paralelismo tratadas, de modo que, uma transferência, em hipótese alguma, possa gerar uma inconsistência dos valores armazenados no banco de dados
- Por se tratar de um sistema bancário, os valores serão tratados sempre como inteiros, ou seja, os valores em moeda sempre em sua unidade mais básica(por exemplo, os valores em reais serão sempre tratados em centavos), para evitarmos os problemas que são gerados com a manipulação de ponto flutuante.
- O banco de dados do sistema deverá ter uma rotina de backup diária, com objetivo de, no improvável caso de uma falha parar o banco completamente, ser possível retornar os dados a um estado anterior.
- O banco de dados a ser implementado deverá ser obrigatoriamente relacional e dentre uma das opções: PostgreSQL ou MySQL.

Mubank	Versão: 0.2
Documento de Arquitetura de Software	Data: 15/06/2022
<document identifier>	

4. Requisitos Arquiteturalmente Significantes

[Insira uma referência ou link para os requisitos que exploram aspectos relevantes da arquitetura.]

5. Decisões, Restrições e justificativas

[Liste as decisões tomadas em relação às abordagens arquiteturais. Estas decisões servirão como guias para definir as partes arquiteturalmente significantes do sistema. Justifique cada decisão ou restrição para que os desenvolvedores compreendam a importância de construir o sistema dentro desse contexto. Isto pode incluir uma lista de “Faça” ou “Não Faça” para guiar os desenvolvedores no desenvolvimento do sistema]

- Decision or constraint and justification
- Decision or constraint and justification

6. Mecanismos Arquiteturais

[Liste os mecanismos arquiteturais, como mecanismos de persistência, comunicação e tratamento de erros, por exemplo, e descreva ocorrente estado de cada um. Inicialmente, cada mecanismo pode ser somente um nome e uma breve descrição. Eles evoluirão até que o mecanismo se torne um padrão ou uma colaboração de elementos de projeto que possam ser aplicados diretamente em algum aspecto do projeto.]

Mecanismo Arquitetural 1

[Descreva a finalidade, os atributos e funções do mecanismo arquitetural.]

Mecanismo Arquitetural 2

[Descreva a finalidade, os atributos e funções do mecanismo arquitetural.]

7. Camadas da Arquitetura

[Descreva os padrões de arquitetura utilizados e como a arquitetura se manterá consistente e uniforme. Isto pode ser uma simples referência para um conhecido padrão arquitetural, como o padrão de divisão em camadas e uma descrição de como os componentes do sistema podem ser colocados juntos.]

8. Visões da Arquitetura

[Descreva as visões arquiteturais usadas para descrever a arquitetura. Isto ilustra as diferentes perspectivas disponíveis para rever e documentar as decisões arquiteturais.]

Visões Recomendadas:

- **Lógica:** Descreva a estrutura e comportamento de porções arquiteturalmente significantes do sistema. Isto deve incluir a estrutura de pacotes, interfaces críticas, importantes classes e subsistemas e as relações entre estes elementos. Isto também inclui visões físicas e lógicas dos dados persistentes.
- **Operacional:** Descreva os nós físicos do sistema e os processos, threads e componentes que rodam em cada um desses nós. Esta visão não é necessária se o sistema roda num único processo e num único thread.

Mubank	Versão: 0.2
Documento de Arquitetura de Software	Data: 15/06/2022
<document identifier>	

- **Casos de Uso:** Uma lista ou diagrama dos casos de uso que contém requisitos arquiteturalmente relevantes.

Nome do caso de uso	[UC1] Fazer cadastro no sistema
Ator(es)	Usuário
Descrição	O usuário se interessou pelo MuBank e deseja se cadastrar no sistema. Ele deve preencher o formulário com seus dados para fazer o cadastro.
Referências	-
Gatilho	Selecionar a opção “Se cadastrar”
Pré-condições	- O usuário deve concordar com as regras e condutas do banco. - O usuário deve preencher o formulário com seus dados pessoais.
Pós-condições	-
Fluxo principal (cenário típico)	1. O usuário seleciona a opção “Se cadastrar” 2. O usuário preenche o formulário com seus dados pessoais. 3. O usuário concorda com as regras e condutas 4. O usuário seleciona a opção “Concluir cadastro”
Fluxo alternativo (cenário alternativo)	1.1. O usuário seleciona a opção “Login” 1.2. O usuário seleciona a opção “Ainda não tem cadastro? Se cadastrar” 1.3. O usuário preenche o formulário com seus

Mubank	Versão: 0.2
Documento de Arquitetura de Software	Data: 15/06/2022
<document identifier>	

	<p>dados pessoais.</p> <p>1.4. O usuário concorda com as regras e condutas</p> <p>1.5. O usuário seleciona a opção “Concluir cadastro”</p>
--	--

Nome do caso de uso	[UC2] Fazer login no sistema
Ator(es)	Cliente
Descrição	O cliente deseja acessar a plataforma e suas funcionalidades, ele deve fazer login no sistema com suas credenciais.
Referências	-
Gatilho	Selecionar a opção “Login”
Pré-condições	<ul style="list-style-type: none"> - O cliente deve estar cadastrado no sistema - O cliente deve preencher suas credenciais
Pós-condições	-
Fluxo principal (cenário típico)	<ol style="list-style-type: none"> 1. O cliente seleciona a opção de login 2. O cliente preenche suas credenciais 3. O cliente seleciona a opção "Efetuar Login"
Fluxo alternativo (cenário alternativo)	

Mubank	Versão: 0.2
Documento de Arquitetura de Software	Data: 15/06/2022
<document identifier>	

Nome do caso de uso	[UC3] Editar informações cadastrais
Ator(es)	Cliente
Descrição	O cliente muda alguma informação pessoal e deseja fazer a alteração no sistema.
Referências	-
Gatilho	Escolher a opção “Editar conta”
Pré-condições	<ul style="list-style-type: none"> - O cliente deve estar cadastrado e autenticado - O cliente deve ter mudado alguma informação pessoal
Pós-condições	<ul style="list-style-type: none"> - O cliente deve confirmar as alterações
Fluxo principal (cenário típico)	<ol style="list-style-type: none"> 1. O cliente faz login na sua conta 2. O cliente altera os dados que necessita 3. O cliente confirma os dados alterados
Fluxo alternativo (cenário alternativo)	-

Mubank	Versão: 0.2
Documento de Arquitetura de Software	Data: 15/06/2022
<document identifier>	

Nome do caso de uso	[UC4] Um cliente pode fazer um saque
Ator(es)	Cliente
Descrição	O cliente do MuBank deseja fazer um saque em sua conta
Referências	-
Gatilho	Clicar no botão "Sacar"
Pré-condições	<ul style="list-style-type: none"> - O cliente deve possuir uma conta cadastrada no sistema - O cliente deve estar logado - O cliente deve possuir na conta um valor maior do que deseja sacar
Pós-condições	<ul style="list-style-type: none"> - O valor do saque é retirado da conta do cliente - O saque entra na lista de transações.
Fluxo principal (cenário típico)	<ol style="list-style-type: none"> 1. O cliente clica no botão "Fazer saque" 2. O cliente informa o valor do saque 3. O cliente clica no botão "Sacar"
Fluxo alternativo (cenário alternativo)	

Mubank	Versão: 0.2
Documento de Arquitetura de Software	Data: 15/06/2022
<document identifier>	

Nome do caso de uso	[UC5] Um cliente pode fazer um depósito
Ator(es)	Cliente
Descrição	O cliente deseja depositar dinheiro em sua conta MuBank
Referências	-
Gatilho	Clicar no botão “Depositar”
Pré-condições	- A conta de destino deve existir no sistema
Pós-condições	- O valor do depósito é adicionado à conta de destino - O depósito entra na lista de transações
Fluxo principal (cenário típico)	1. O cliente clica no botão “Fazer depósito” 2. O cliente entra com as informações da conta e o valor do depósito 3. O cliente clica no botão "Depositar"
Fluxo alternativo (cenário alternativo)	

Mubank	Versão: 0.2
Documento de Arquitetura de Software	Data: 15/06/2022
<document identifier>	

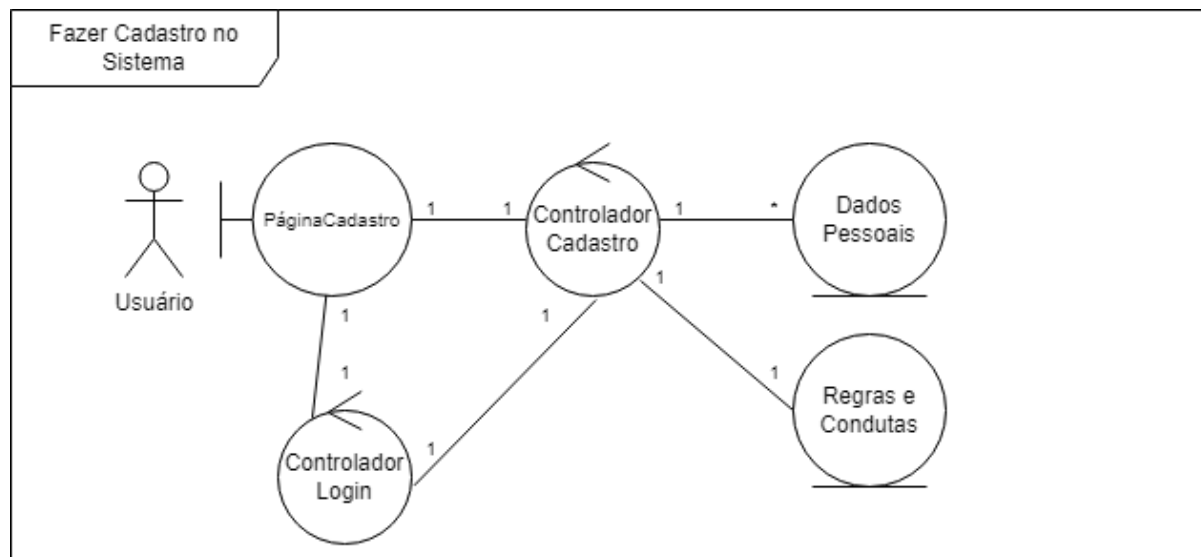
Nome do caso de uso	[UC6] Um cliente pode ver as movimentações feitas na sua conta
Ator(es)	Cliente
Descrição	O cliente deseja ver todas as movimentações (saque, depósito e transferência) que o mesmo fez
Referências	-
Gatilho	Clicar no botão "VER MOVIMENTAÇÕES"
Pré-condições	- Ter alguma movimentação realizada - Estar logado no sistema
Pós-condições	-
Fluxo principal (cenário típico)	1. O cliente, logado, abre o menu de opções 2. O cliente clica em "VER MOVIMENTAÇÕES" 3. É apresentado todas as movimentações bancárias que o usuário fez
Fluxo alternativo (cenário alternativo)	-

Mubank	Versão: 0.2
Documento de Arquitetura de Software	Data: 15/06/2022
<document identifier>	

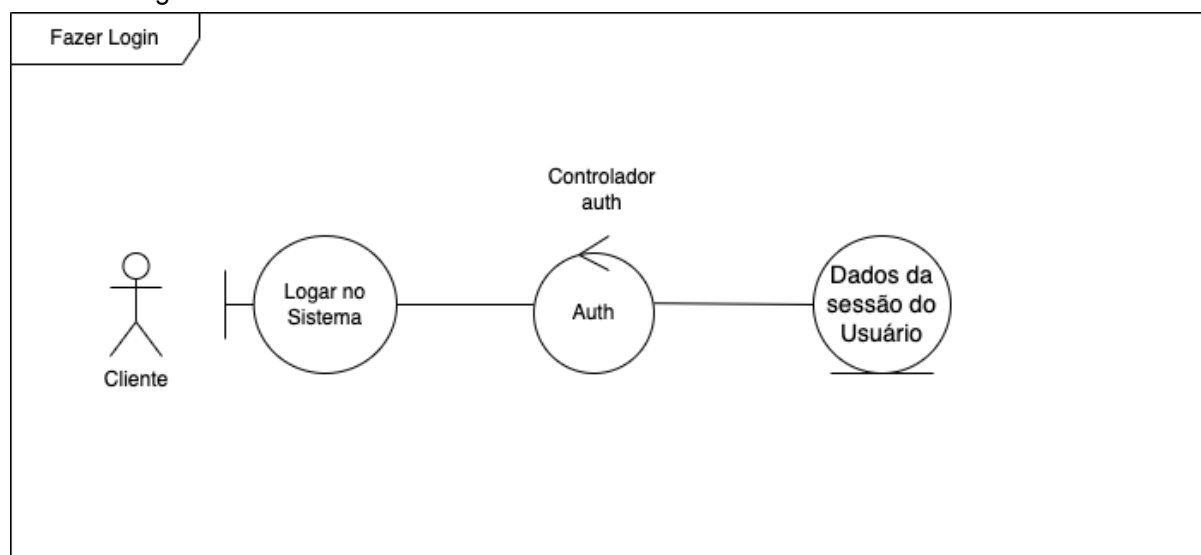
Nome do caso de uso	[UC7] Um cliente pode fazer uma transferência para outro cliente
Ator(es)	Cliente
Descrição	Um cliente deseja enviar dinheiro para outro cliente
Referências	-
Gatilho	Clicar no botão "FAZER UMA TRANSFERÊNCIA"
Pré-condições	<ul style="list-style-type: none"> - Estar logado no sistema - Ter dinheiro suficiente na conta para fazer a transferência
Pós-condições	<ul style="list-style-type: none"> - O valor da transferência é retirado da conta do cliente que enviou o dinheiro - O valor da transferência é adicionado na conta do cliente que recebeu o dinheiro - A transferência entra na lista de transações de ambos clientes.
Fluxo principal (cenário típico)	<ol style="list-style-type: none"> 1. O cliente logado abre o menu de opções 2. O cliente clica no botão "FAZER UMA TRANSFERÊNCIA" 3. Coloca o valor que deseja enviar <ol style="list-style-type: none"> 3.1. Se o valor é maior que o saldo da conta, informa para o cliente que seu saldo não é o suficiente. 3.2. Caso contrário, pergunta os dados do destinatário e o cliente vai para o passo 4. <ol style="list-style-type: none"> 3.2.1 O cliente preenche os dados bancários do destinatário 3.2.2 O cliente clica em "FINALIZAR TRANSAÇÃO"
Fluxo alternativo (cenário alternativo)	-

Mubank	Versão: 0.2
Documento de Arquitetura de Software	Data: 15/06/2022
<document identifier>	

4.1 Fazer Cadastro no Sistema

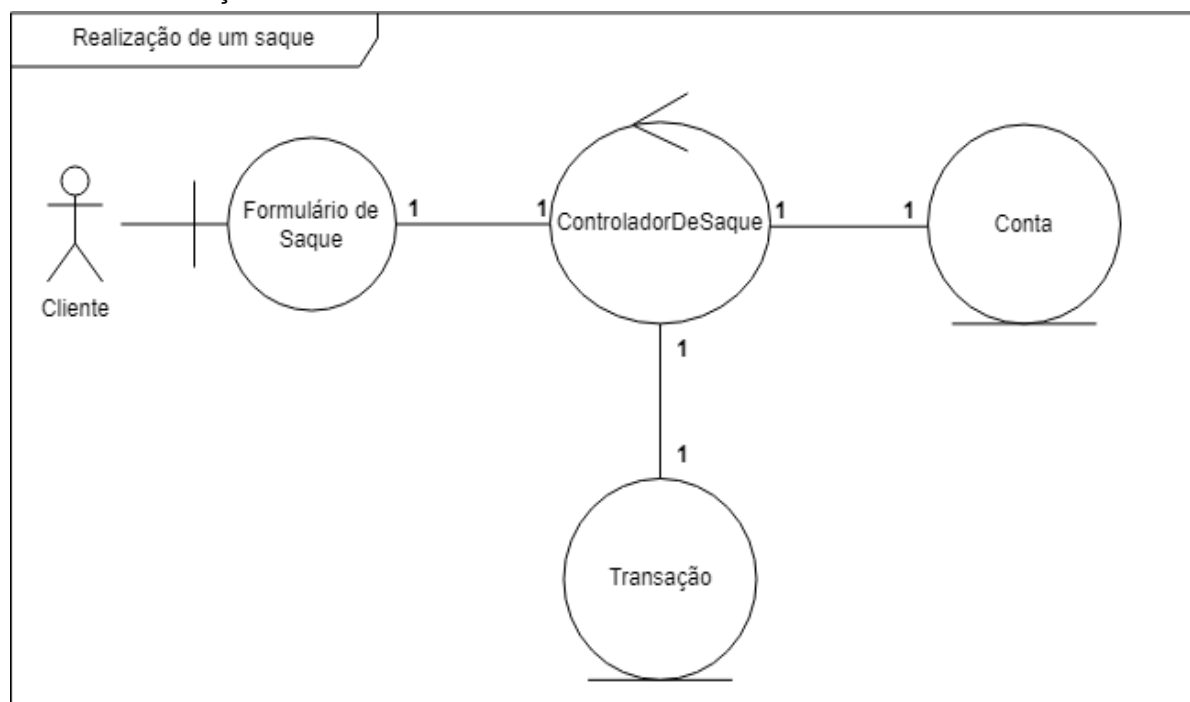


4.2 Fazer Login



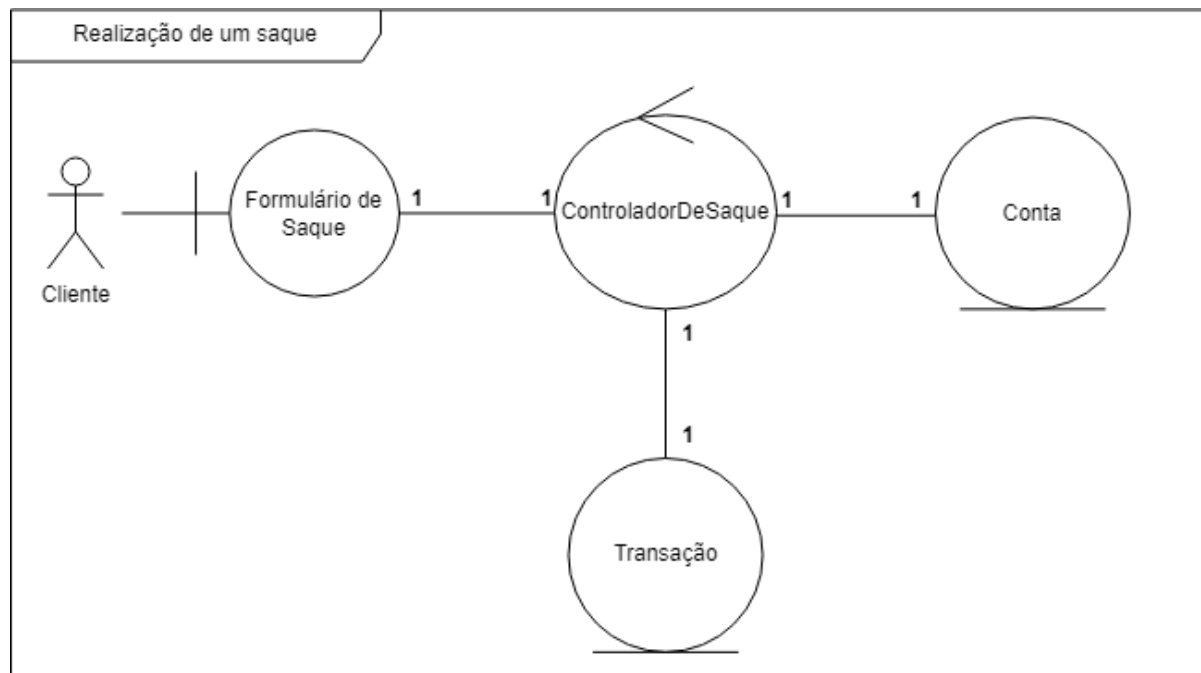
Mubank	Versão: 0.2
Documento de Arquitetura de Software	Data: 15/06/2022
<document identifier>	

4.3 Editar Informações Cadastrais

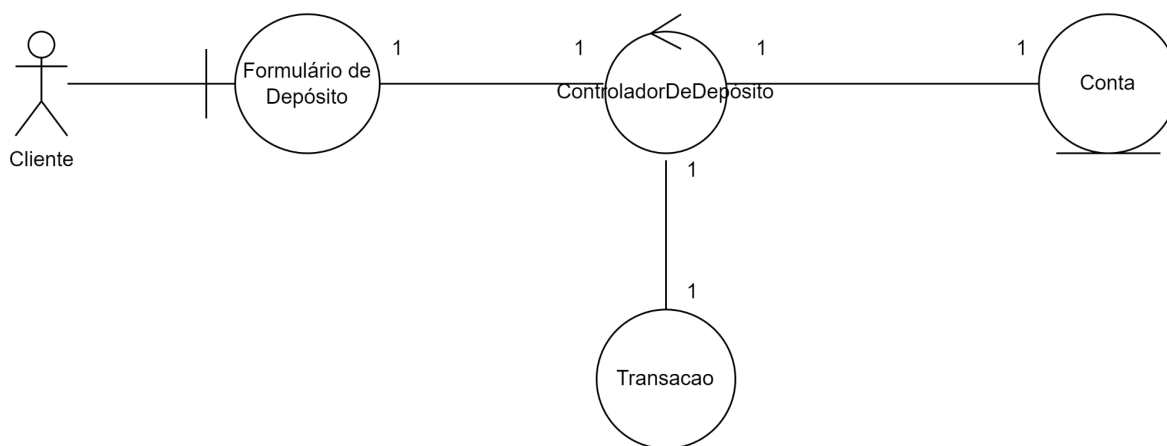


4.4 Realizar um Saque

Mubank	Versão: 0.2
Documento de Arquitetura de Software	Data: 15/06/2022
<document identifier>	

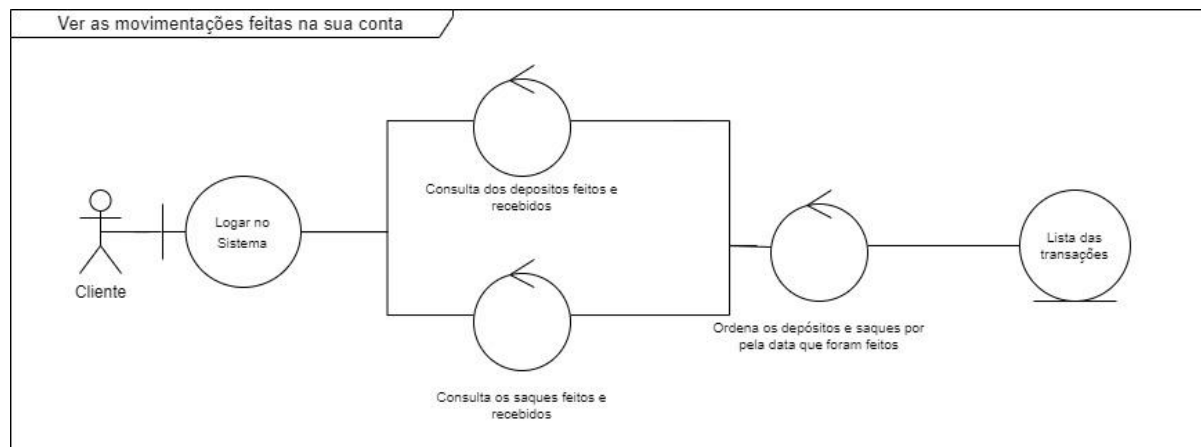


4.5 Fazer depósito

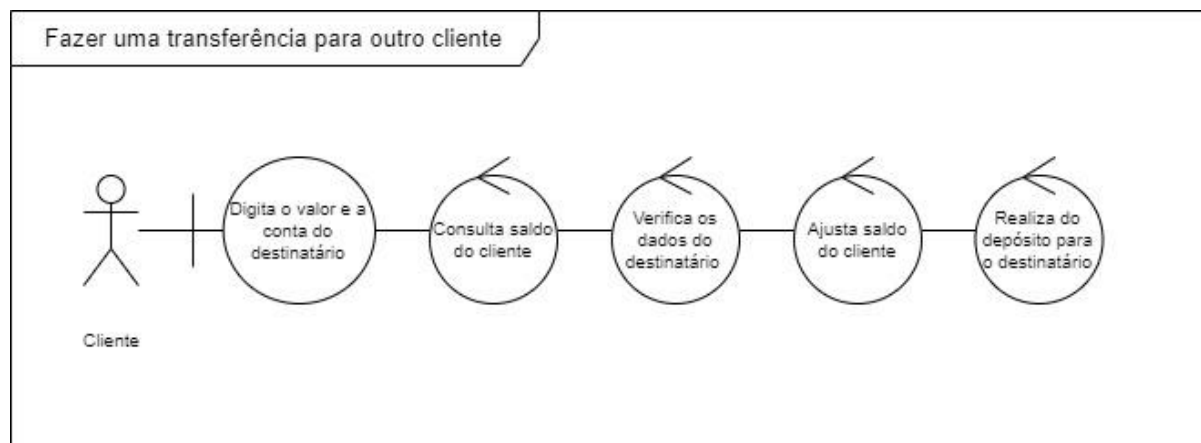


Mubank	Versão: 0.2
Documento de Arquitetura de Software	Data: 15/06/2022
<document identifier>	

4.6. Extrato



4.7 Transferência



9. Qualidade

[Uma descrição de como a arquitetura do software contribui para todos os recursos (exceto a funcionalidade) do sistema: extensibilidade, confiabilidade, portabilidade e assim por diante. Se essas características possuírem significado especial, como implicações de segurança, garantia ou privacidade, elas deverão ser delineadas claramente.]

Mubank	Versão: 0.2
Documento de Arquitetura de Software	Data: 15/06/2022
<document identifier>	

10. Problemas de Projeto e Soluções encontradas

Seguindo os padrões GRASP, no contexto de Alta Coesão, analisamos que na modelagem da classe Transação seria interessante dividir os tipos de Transação em subclasses diferentes, sendo essas Saque, Transferência e Depósito. O objetivo foi para que a coesão se mantivesse alta, fosse atribuído responsabilidades diferentes às classes e não houvesse sobrecarga da classe Transação