

Laboratório de Engenharia de Software

Aplicativo de Controle Financeiro

Nome	RA
Caio Larroza	1680481511006
Giovanni Armane	1680481511016
Leonardo Costa	1680481512015
Matheus Lopes	1680481511044

Relatório de andamento do projeto

Aluno: Caio Larroza

RA: 1680481511006

[illegible]

Relatório de andamento do projeto

Aluno: Giovanni Armane

RA: 1680481511016

[illegible]

Relatório de andamento do projeto

Aluno: Leonardo Costa

RA: 1680481512015

[illegible]

Relatório de andamento do projeto

Aluno: Matheus Lopes

RA: 1680481511044

[illegible]

Sumário

Sumário.....	6
1. Planejamento do Sistema.....	9
1.1. Descrição do projeto.....	9
1.2. Cronograma.....	9
1.3. Recursos.....	10
1.3.1. Recursos humanos.....	10
1.3.2. Recursos físicos.....	10
2. Especificação dos Requisitos do Sistema.....	11
2.1. O produto.....	11
2.1.1 Propósito.....	11
2.1.2 Clientes.....	11
2.1.3 Usuários.....	11
2.1.4 Convenções de nomes e definições.....	11
2.2 Restrições do produto.....	12
2.2.1 Restrição técnica.....	12
2.2.2 Restrição sistêmica.....	12
2.2.3 Cronograma e recursos.....	12
2.3. Escopo do projeto.....	13
2.4. Requisitos funcionais.....	13
2.5. Requisitos não funcionais.....	14
2.6. Protótipos do produto.....	15
2.6.1. Modelo de navegação.....	15
2.6.2. Protótipos de tela.....	16
3. Modelo de Casos de Uso do Sistema.....	18
3.1. Diagrama de Casos de Uso do sistema.....	18
3.2. Documentação dos Atores.....	19
3.2.1. Ator Usuário.....	19
3.3. Documentação dos Casos de Uso.....	20
3.3.1. Caso de uso Inserir Conta Contábil.....	20
3.3.2. Caso de uso Listar Contas Contábeis.....	21
3.3.3. Caso de uso Alterar Conta Contábil.....	21
3.3.4. Caso de uso Remover Conta Contábil.....	22
3.3.5. Caso de uso Inserir Objetivos.....	23
3.3.6. Caso de uso Alterar Idioma.....	23
3.3.7. Caso de uso Consultar Movimentações.....	24
3.3.8. Caso de uso Alterar Movimentação.....	25

3.3.9. Caso de uso Remover Movimentação	25
3.3.10. Caso de uso Inserir Movimentação	26
3.3.11. Caso de uso Agendar Débito.....	27
3.3.12. Caso de uso Inserir Débito.....	27
3.3.13. Caso de uso Agendar Crédito	28
3.3.14. Caso de uso Inserir Crédito	29
3.3.15. Caso de uso transferir entre Contas	29
3.3.16. Caso de uso Fazer Login	30
3.3.17. Caso de uso Fazer Login com Google.....	31
3.3.18. Caso de uso Realizar Cadastro	32
3.3.19. Caso de uso Fazer Login com Facebook	32
3.3.20. Caso de uso Ver Projeção do Objetivo	33
3.3.21. Caso de uso Alterar Objetivo.....	34
3.3.22. Caso de uso Remover Objetivo	34
3.3.23. Caso de uso Arquivar Objetivo.....	35
3.3.24. Caso de uso Listar Objetivos	36
4. Modelo de Análise do Sistema	36
4.1. Especificação das classes de análise do sistema.....	36
4.1.1. Visão Geral dos Pacotes de Análise	37
4.1.1.1. Pacote View	38
4.1.1.1. Pacote ClientControllers.....	38
4.1.1.1. Pacote Controllers.....	38
4.1.1.1. Pacote Models.....	39
4.1.2. Diagramas e especificações das classes.....	39
4.1.2.1. Classe Visão Geral	39
4.1.2.2. Classe Contas.....	40
4.1.2.3. Classe Movimentações	40
4.1.2.4. Classe Login	40
4.1.2.5. Classe Criar Conta	41
4.1.2.6. Classe Histórico	41
4.1.2.7. Classe Configurações	41
4.1.2.8. Classe Objetivos	41
4.1.2.9. Classe ConfiguracoesController	42
4.1.2.10. Classe UsuarioController.....	43
4.1.2.11. Classe MovimentacaoController	44
4.1.2.12. Classe ContaContabilController	46
4.1.2.13. Classe ObjetivoController	47

4.1.2.14. Classe Configuracoes	48
4.1.2.15. Classe Usuario	49
4.1.2.16. Classe Movimentacao.....	49
4.1.2.17. Classe Transferencia	50
4.1.2.18. Classe ContaContabil	50
4.1.2.19. Classe Objetivo.....	50
4.1.2.20. Classe HistoricoObjetivo	51
4.1.2.21. Classe ConfiguracoesController (View).....	51
4.1.2.22. Classe LoginController (View).....	52
4.1.2.23. Classe CadastroController (View)	54
4.1.2.24. Classe HistoricoController	55
4.1.2.25. Classe MovimentaçãoController (View)	55
4.1.2.26. Classe VisaoGeralController (View)	57
4.1.2.27. Classe ContaContabilController (View).....	57
4.1.2.28. Classe ObjetivoController (View)	58
4.2. Realizações de casos de uso	59
4.2.1. Cenário Inserir Conta.....	59
4.2.2. Cenário Inserir Objetivo	60
4.2.3. Cenário Transferência.....	61
4.2.4. Cenário Inserir Movimentação	62
5. Modelos de Projeto de Sistema	63
5.1. Classes de Projeto.....	63
5.2.1. Diagrama de Classes.....	63
5.2.1. Comparativo entre frameworks/bibliotecas.....	63
5.2. Realizações de Casos de Uso – Projeto	64
5.2.1. Diagrama de Sequência - Inserir Conta	65
5.2.2. Diagrama de Sequência – Inserir Objetivo.....	65
5.2.3. Diagrama de Sequência – Inserir Movimentação.....	66
5.2.4. Diagrama de Sequência – Realizar Transferência	66

1. Planejamento do Sistema

1.1. Descrição do projeto

O projeto busca desenvolver um aplicativo de controle financeiro, como tantos outros existentes no mercado, mas com uma proposta diferente. Em conjunto aos recursos que todo aplicativo da categoria possui, este projeto contará com metodologias de projeção financeira que farão previsão de gastos e estimativas de rendimento, auxiliando o usuário a controlar seu fluxo financeiro, pensando no futuro.

Através do aplicativo, o usuário poderá contabilizar seu fluxo financeiro com maior facilidade através de uma interface simples, isso é, responsiva, intuitiva e com poucos controles e entradas de dados.

Qualquer pessoa que tenha no mínimo um fluxo financeiro pode usar o aplicativo, sem que seja necessário o conhecimento de contabilidade, finanças ou ao menos matemática. Para usá-lo, a pessoa só precisa inserir as movimentações financeiras de sua vida pessoal no aplicativo, podendo organizá-las por entrada/saída, conta, categoria e data. Feito isso, o aplicativo responsabiliza-se por analisar tais dados e gerar diversos tipos de relatórios para que o usuário possa entender e planejar com facilidade sua vida financeira.

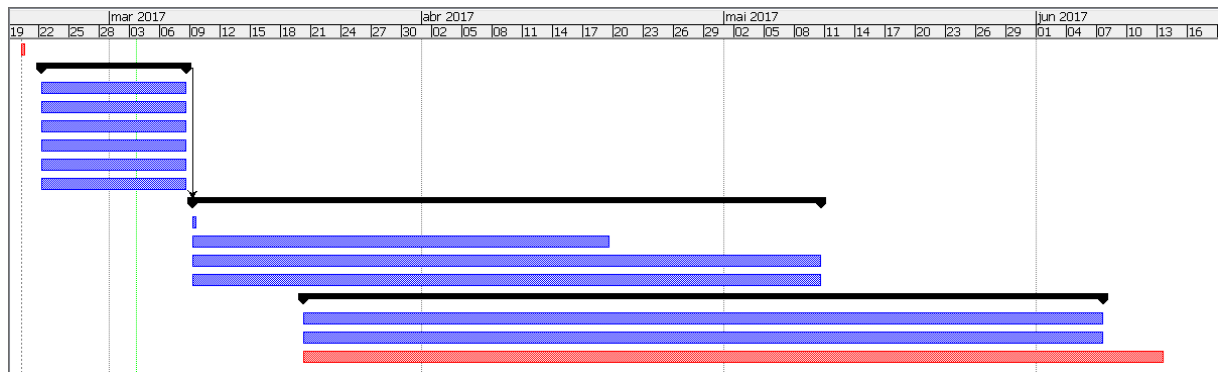
O aplicativo poderá também ser usado para previsão de gastos e estimativas de rendimento, para que o usuário possa, por exemplo, ter uma previsão de quando terá uma meta cumprida ou até mesmo se um gasto agora poderá prejudicar sua situação financeira num futuro próximo, através de metodologias de projeção financeira, permitindo assim ao usuário um controle mais eficaz da sua contabilidade. Tais projeções também serão exibidas ao usuário através de gráficos e listas, e essas projeções poderão ser refinadas com a adição de lançamentos futuros, previsões do usuário, isto é, lançamentos incertos e cadastro de movimentações que se repetem com uma determinada frequência ou em dias específicos.

Além disso, o aplicativo estará disponível em qualquer plataforma que suporte tecnologias web, porém disponível sem o uso de internet apenas para dispositivos móveis.

Contará também com a possibilidade de sincronização dos dados do usuário entre seus dispositivos, exigindo para tal um cadastro que poderá ser feito inserindo-se dados básicos de identificação ou através da conta do Google ou Facebook.

1.2. Cronograma

Nome	Duração	Início
Reunião de planejamento	1 dia	19/02/17 08:00
Especificação de Requisitos	11 dias?	22/02/17 08:00
Especificação do produto	11 dias	22/02/17 08:00
Restrições do produto	11 dias?	22/02/17 08:00
Escopo do projeto	11 dias?	22/02/17 08:00
Requisitos funcionais	11 dias?	22/02/17 08:00
Requisitos não funcionais	11 dias?	22/02/17 08:00
Prototipagem	11 dias?	22/02/17 08:00
Documentação	45 dias?	09/03/17 08:00
Diagrama de caso de uso	1 dia?	09/03/17 08:00
Diagrama de classes de análise	30 dias?	09/03/17 08:00
Diagrama de classes de projeto	45 dias?	09/03/17 08:00
Diagrama entidade-relacionamento	45 dias?	09/03/17 08:00
Implementação	58 dias?	20/03/17 08:00
Front end	58 dias?	20/03/17 08:00
Back end	58 dias?	20/03/17 08:00
Testes	62 dias	20/03/17 08:00



1.3. Recursos

1.3.1. Recursos humanos

- **Caio Larroza:** Será responsável pelo design do sistema. Disponibilidade semanal de 13 horas;
- **Giovanni Armane:** Será responsável pelos diagramas UML e IA do sistema. Disponibilidade semanal de 13 horas;
- **Leonardo Costa:** Responsável pela programação Client-Side e modelagem do banco de dados. Disponibilidade semanal de 13 horas;
- **Matheus Lopes:** Responsável pela programação do Web Service. Disponibilidade semanal de 13 horas.

1.3.2. Recursos físicos

Para o desenvolvimento deste projeto será utilizado:

- Computadores dos laboratórios;
- Notebooks pessoais de cada integrante;
- Visual Studio 2015;
- Visual Studio Code;
- Pacote Office;
- StarUML;
- Trello;
- GitLab;

2. Especificação dos Requisitos do Sistema

2.1. O produto

2.1.1 Propósito

O produto tem como objetivo suprir uma aparente deficiência no mercado de aplicativo de controle de finanças pessoais. Foi notado que as soluções existentes não eram atrativas do ponto de vista estético, mercadológico ou mesmo funcional. O projeto então tem por objetivo trazer soluções modernas e responsivas, que possam ser acessadas independentes de lugar, dispositivo ou até mesmo disponibilidade de conectividade. Que seja entendível por qualquer pessoa, não importando nível técnico para operar a solução ou até mesmo conhecimento profundo de contabilidade.

Com o produto, é esperado atingirmos um mercado que reconhecemos potencial não alcançado.

2.1.2 Clientes

Pessoas ou entidades que tenham o interesse em atrair, educar e/ou ajudar terceiros manter a atividade financeira balanceada. Estes envolvem bancos, agências governamentais ou privadas de investimentos ou profissionais livres no mercado.

2.1.3 Usuários

Consideramos usuários finais todos aqueles que estejam iniciando sua vida financeira ou aqueles que desejam organizar a mesma e necessitam de uma ferramenta prática que os ajude a atingir este ideal. Como o público é vasto, é necessário que o projeto utilize dos meios mais modernos em design inteligente, responsivo, acessível e prático. Não será esperado do usuário final não mais que um dispositivo que tenha suporte à tecnologia web e uma conexão suficientemente estável para que seus dados sejam sincronizados periodicamente.

2.1.4 Convenções de nomes e definições

Termo	Sigla	Descrição
Conta corrente	CC	Conta bancária corrente, possuída por um usuário.
Conta poupança	CP	Conta bancária poupança, possuída por um usuário
Login		Processo de autenticação realizado pelo aplicativo.

2.2 Restrições do produto

2.2.1 Restrição técnica

O projeto deve ser desenvolvido exclusivamente com tecnologias web e mobile, especificamente com o uso do framework de aplicações web progressivas Ionic versão 1 e uma interface de programação de aplicações web com total transferência de estado representacional sobre os frameworks da Microsoft ASP.NET Core e .NET Core.

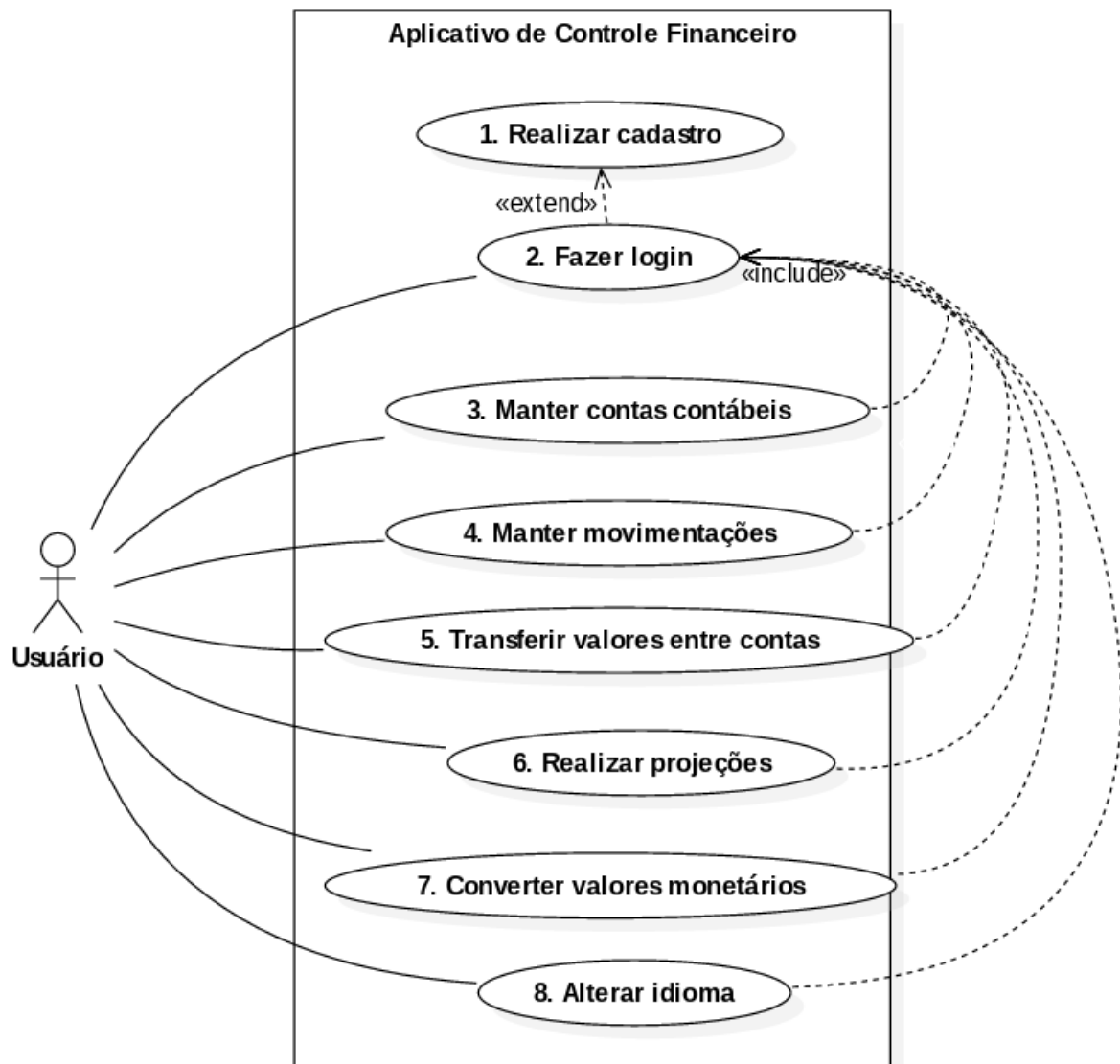
2.2.2 Restrição sistêmica

O projeto estará limitado a uso somente em sistemas operacionais com suporte a HTML5 e conexão à internet ou o uso em sistema operacional Android a partir da versão 4.4.

2.2.3 Cronograma e recursos

Deve-se cumprir o cronograma estabelecido sem possibilidade de adiamento da data final. Não será permitida a contratação de recursos externos.

2.3. Escopo do projeto



2.4. Requisitos funcionais

ID do requisito	Descrição	Caso de uso relacionado
RF01	O sistema deverá manter o cadastro de usuários, com informações de contato	UC18
RF02	O sistema deverá efetuar a autenticação de usuário	UC16, UC17, UC19
RF03	O sistema deverá manter funções CRUD para movimentações financeiras referentes aos usuários cadastrados	UC7, UC8, UC9, UC10, UC11, UC12, UC13, UC14
RF04	O sistema deverá manter funções CRUD para contas contábeis	UC1, UC2, UC3, UC4

	referentes a movimentações financeiras	
RF05	O sistema deverá ser capaz de realizar transferências entre contas contábeis	UC10, UC15
RF06	O sistema deverá, com base nas informações contábeis mantidas, realizar projeções de valores	UC5, UC20, UC21, UC22, UC23, UC24
RF07	O sistema deverá estar disponível em português e inglês.	UC6

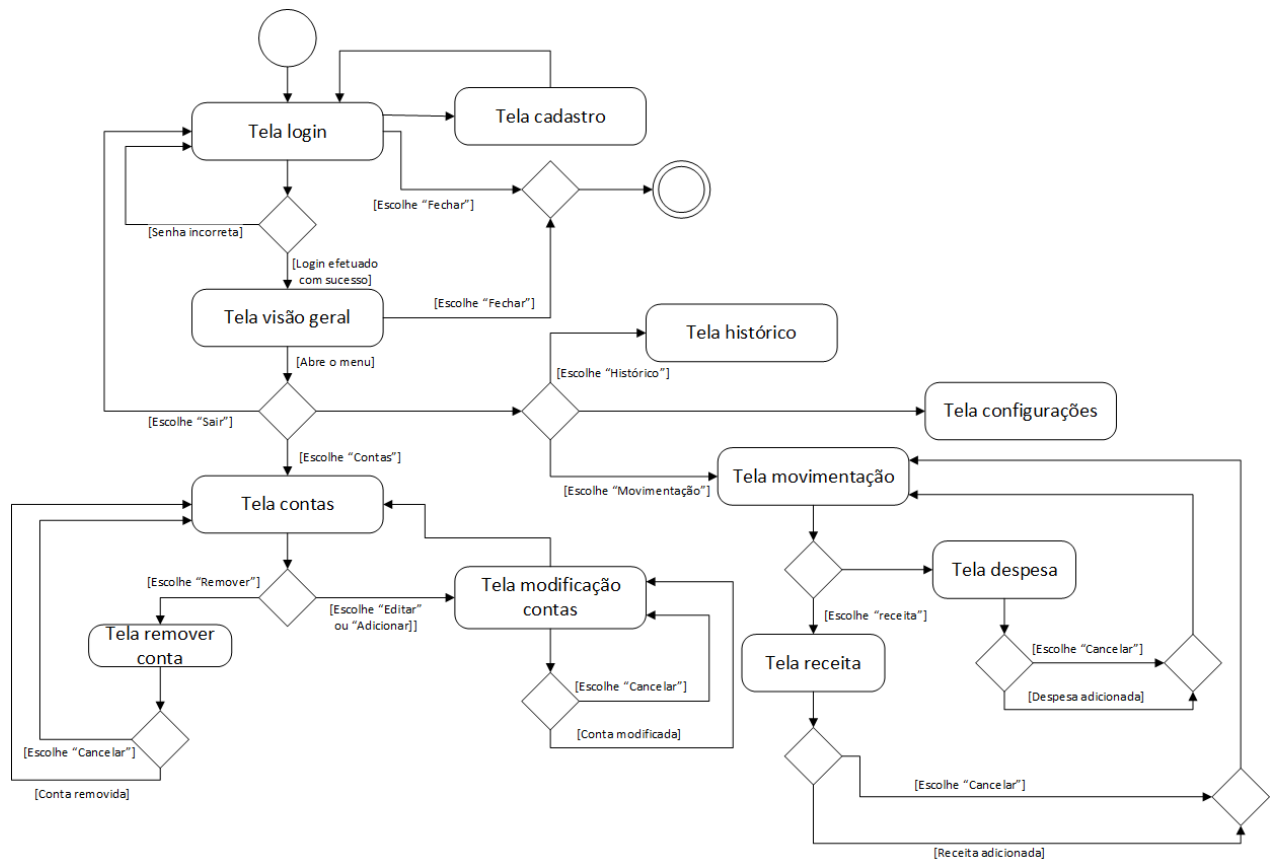
2.5. Requisitos não funcionais

ID do requisito	Descrição
RN01	A interface deve responder a cada ação do usuário em até 350ms
RN02	O servidor deverá ser uma máquina Windows ou Linux
RN03	Somente o usuário da conta pode realizar ações na mesma
RN04	Cada funcionalidade deve ser acessada pelo usuário em no máximo 5 passos
RN05	A interface de operação será baseada em ícones e textos apresentados em uma tela "touchscreen"
RN06	A interface deverá ser baseada em design minimalista
RN07	O sistema deverá ser portátil para outros ambientes sem a necessidade de reprogramação do mesmo
RN08	O sistema permitirá que o usuário realize cadastro utilizando contas de outros serviços (Google, Facebook, etc.)
RN09	O sistema não permitirá que o usuário informe senhas de contas e cartões bancários
RN10	O sistema deverá ser desenvolvido utilizando o padrão MVC
RN11	O sistema deverá ser desenvolvido utilizando as linguagens C#, HTML, CSS e JavaScript, sobre os frameworks Ionic v1, ASP.NET Core e .NET Core
RN12	O sistema deverá armazenar suas informações localmente em formato JSON e no servidor utilizando-se o sistema gerenciador de banco de dados Microsoft SQL Server

RN13	Um dispositivo deverá possuir apenas 1 usuário conectado por vez
------	--

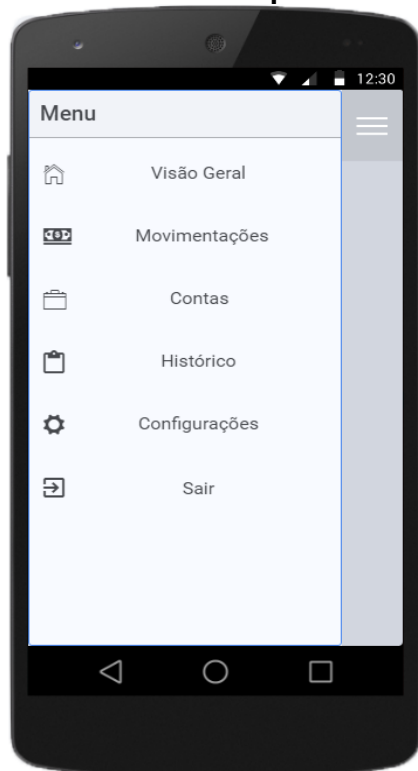
2.6. Protótipos do produto

2.6.1. Modelo de navegação



2.6.2. Protótipos de tela

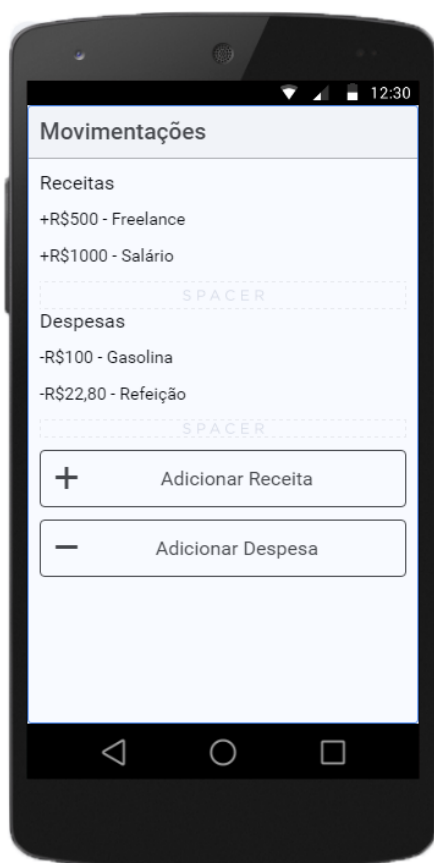
Menu Principal



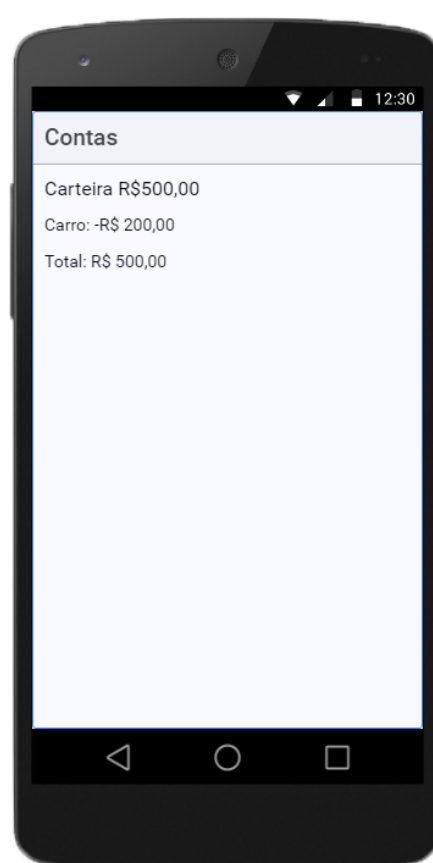
Visão Geral



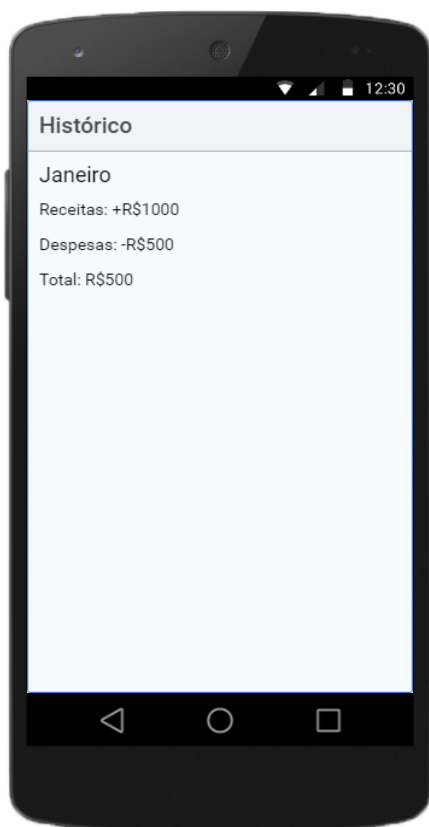
Movimentações



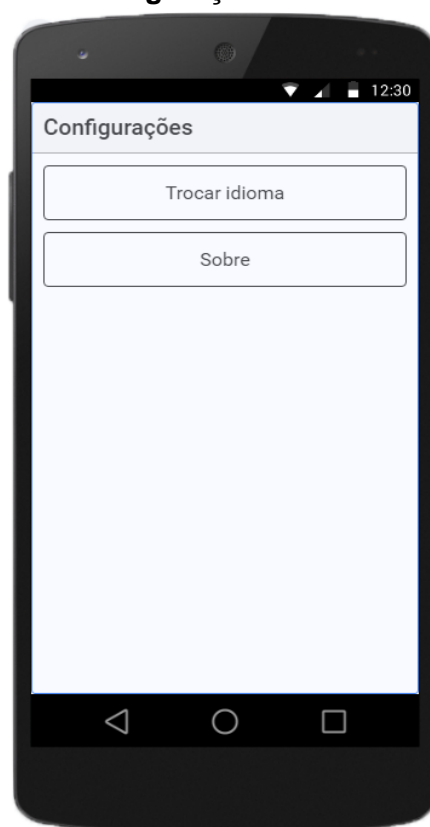
Contas



Histórico



Configurações



Criar Conta

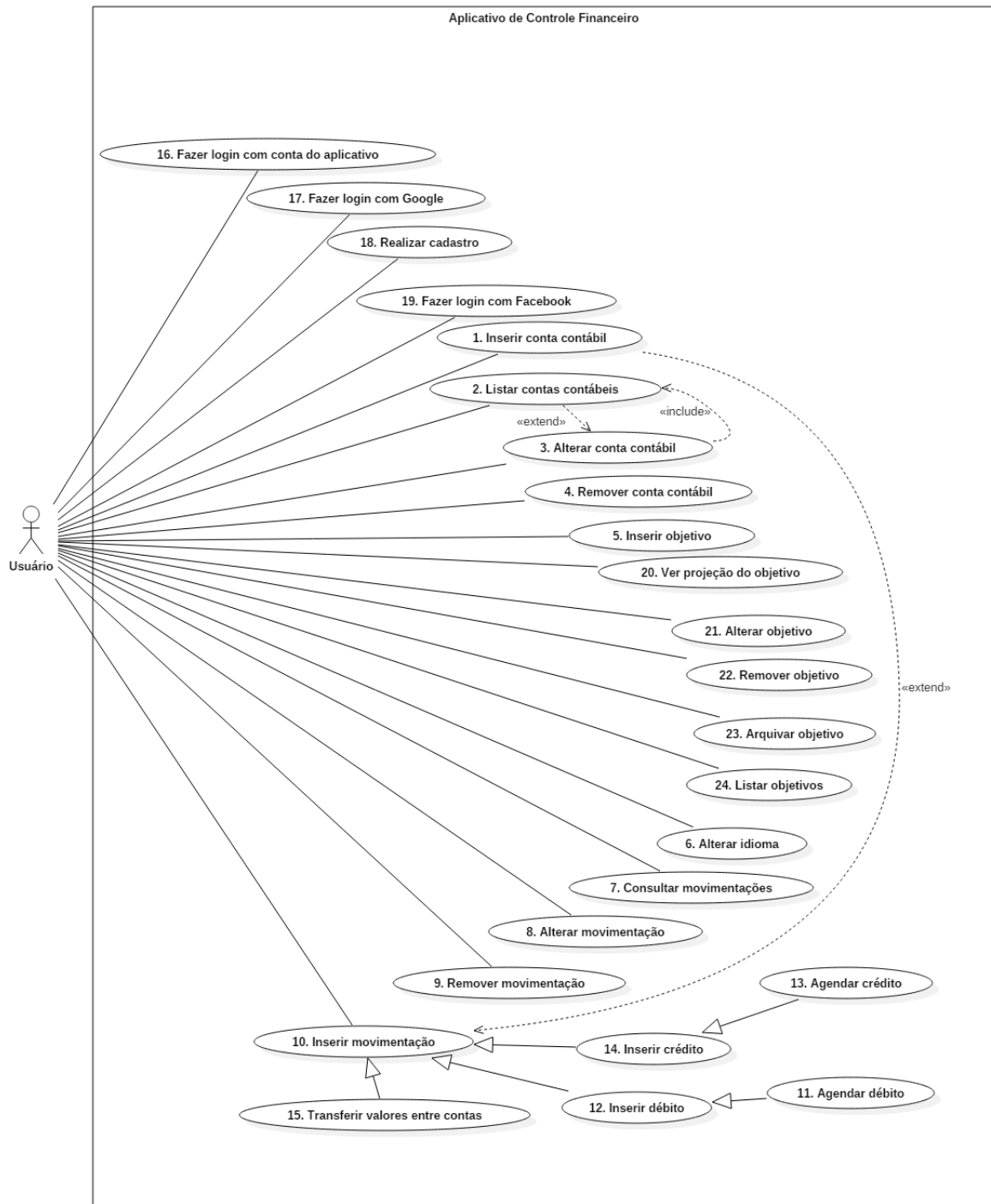


Login



3. Modelo de Casos de Uso do Sistema

3.1. Diagrama de Casos de Uso do sistema



3.2. Documentação dos Atores

3.2.1. Ator Usuário

Ator: Usuário
ID: 001
Resumo: Usuário que irá interagir com o sistema
Responsabilidades: <ol style="list-style-type: none">1. Efetuar inserções de receitas e despesas2. Efetuar atualizações de receitas e despesas3. Efetuar visualizações de receitas e despesas4. Efetuar exclusões de receitas e despesas5. Efetuar inserções de contas contábeis6. Efetuar atualizações de contas contábeis7. Efetuar visualizações de contas contábeis8. Efetuar exclusões de contas contábeis9. Efetuar inserções de objetivos10. Efetuar atualizações de objetivos11. Efetuar visualizações de objetivos12. Efetuar arquivamentos de objetivos13. Efetuar exclusões de objetivos14. Efetuar o cadastro no sistema15. Efetuar o login no sistema
Ambiente Físico: Indiferente
Número e Tipo: 1 ator do tipo humano
Frequência com que usa o sistema: Diariamente

3.3. Documentação dos Casos de Uso

3.3.1. Caso de uso Inserir Conta Contábil

Caso de Uso: Inserir conta contábil
ID: 1
Resumo: O usuário cria uma conta para a realização de movimentações.
Atores primários: Usuário
Atores secundários: Nenhum
Pré-condições: <ol style="list-style-type: none">1. O usuário está logado no sistema.2. O sistema está na tela inicial, aguardando uma ação do usuário.
Fluxo de eventos principais: <ol style="list-style-type: none">1. O usuário acessa a tela de contas através do menu.2. O usuário clica em adicionar.3. O usuário digita o nome da conta. {Verifica nome da conta}4. O usuário seleciona um ícone e cor para a conta.5. O usuário clica em adicionar.6. O sistema cria a nova conta.7. O sistema salva a nova conta no banco de dados.8. O sistema exibe uma confirmação da criação da conta.9. O sistema abre a tela de inserção de movimentações com a nova conta selecionada.
Pós-condições: <ol style="list-style-type: none">1. O usuário está logado no sistema.2. O usuário terá uma nova conta à disposição para uso.3. O sistema estará na tela de inserção de movimentações com a nova conta selecionada, aguardando uma ação do usuário.
Fluxo de eventos alternativo: <ol style="list-style-type: none">1. <i>Nome de conta repetido</i> Em {Verifica nome da conta}, o sistema verifica se há uma conta cadastrada com o nome digitado. Se houver:<ol style="list-style-type: none">1. O sistema solicita que o usuário digite um nome diferente

3.3.2. Caso de uso Listar Contas Contábeis

Caso de Uso: Listar contas contábeis
ID: 2
Resumo: O sistema exibe uma lista com todas as contas contábeis cadastradas.
Atores primários: Usuário
Atores secundários: Nenhum
Pré-condições: <ol style="list-style-type: none">1. O usuário está logado no sistema.2. O sistema está na tela inicial, aguardando uma ação do usuário.
Fluxo de eventos principais: <ol style="list-style-type: none">1. O usuário acessa a tela de contas através do menu.2. O sistema exibe uma lista com o nome de todas as contas contábeis cadastradas.
Pós-condições: <ol style="list-style-type: none">1. O usuário está logado no sistema.2. O usuário verá as contas contábeis cadastradas e poderá editá-las.
Fluxo de eventos alternativo: <p>Nenhum.</p>

3.3.3. Caso de uso Alterar Conta Contábil

Caso de Uso: Alterar conta contábil
ID: 3
Resumo: O usuário altera as propriedades de uma conta contábil.
Atores primários: Usuário
Atores secundários: Nenhum
Pré-condições: <ol style="list-style-type: none">1. O usuário está logado no sistema.2. O sistema está na tela inicial, aguardando uma ação do usuário.
Fluxo de eventos principais: <ol style="list-style-type: none">1. Executar Caso de Uso <u>Listar contas contábeis</u>

<ol style="list-style-type: none"> 2. O usuário clica em uma das contas exibidas. 3. O usuário altera as propriedades desejadas da conta. {Verifica nome da conta} 4. O usuário clica em alterar. 5. O sistema altera a conta. 6. O sistema salva as alterações no banco de dados. 7. O sistema exibe uma confirmação da alteração da conta. 8. O sistema retorna para a lista de contas contábeis.
<p>Pós-condições:</p> <ol style="list-style-type: none"> 1. O usuário está logado no sistema. 2. O usuário estará com a lista de contas aberta.
<p>Fluxo de eventos alternativo:</p> <ol style="list-style-type: none"> 1. <i>Nome de conta repetido</i> Em {Verifica nome da conta}, o sistema verifica se há uma conta cadastrada com o nome digitado. Se houver: <ol style="list-style-type: none"> 1. O sistema solicita que o usuário digite um nome diferente

3.3.4. Caso de uso Remover Conta Contábil

Caso de Uso: Remover conta contábil
ID: 4
Resumo: O usuário apaga uma conta contábil e suas movimentações.
Atores primários: Usuário
Atores secundários: Nenhum
<p>Pré-condições:</p> <ol style="list-style-type: none"> 1. O usuário está logado no sistema. 2. O sistema está na tela inicial, aguardando uma ação do usuário.
<p>Fluxo de eventos principais:</p> <ol style="list-style-type: none"> 1. Executar Caso de Uso <u>Listar contas contábeis</u> 2. O usuário clica em uma das contas exibidas. 3. O usuário clica em excluir conta. 4. O sistema vai exibir uma mensagem pedindo a confirmação do usuário 5. O usuário clica em excluir. 6. O sistema remove a conta e suas movimentações do banco de dados.

<p>7. O sistema exibe uma confirmação da remoção da conta.</p> <p>8. O sistema retorna para a lista de contas contábeis.</p>
<p>Pós-condições:</p> <p>1. O usuário está logado no sistema.</p> <p>2. O usuário estará com a lista de contas aberta.</p>
<p>Fluxo de eventos alternativo: Nenhum.</p>

3.3.5. Caso de uso Inserir Objetivos

Caso de Uso: Inserir Objetivos
ID: 5
Resumo: O usuário irá inserir algum objetivo para sua conta contábil em uma determinada data no futuro
Atores primários: Usuário
Atores secundários: Nenhum
<p>Pré-condições:</p> <p>1. Estar logado no sistema</p> <p>2. Possuir uma conta contábil</p>
<p>Fluxo de eventos principais:</p> <p>1. O usuário irá acessar o menu de opções</p> <p>2. O usuário irá entrar no menu de objetivos</p> <p>3. O usuário irá criar um objetivo novo</p> <p>4. O usuário irá vincular uma conta contábil já existente ao objetivo inserido</p>
<p>Pós-condições:</p> <p>1. Será apresentado uma mensagem de confirmação da ação</p>
Fluxo de eventos alternativo: Nenhum

3.3.6. Caso de uso Alterar Idioma

Caso de Uso: Alterar idioma
ID: 6
Resumo: O usuário altera o idioma da aplicação

Atores primários: Usuário
Atores secundários: Nenhum
Pré-condições: <ol style="list-style-type: none"> 1. O usuário está logado no sistema. 2. O sistema está na tela inicial, aguardando uma ação do usuário.
Fluxo de eventos principais: <ol style="list-style-type: none"> 1. O usuário acessa a tela de configurações através do menu. 2. O usuário clica em alterar idioma. 3. O usuário escolhe um idioma da lista. 4. O usuário confirma a troca de idioma.
Pós-condições: <ol style="list-style-type: none"> 1. O sistema deverá voltar a tela inicial.
Fluxo de eventos alternativo:

3.3.7. Caso de uso Consultar Movimentações

Caso de Uso: Consultar movimentações
ID: 7
Resumo: O usuário consulta todas as suas movimentações.
Atores primários: Usuário
Atores secundários: Nenhum
Pré-condições: <ol style="list-style-type: none"> 1. O usuário está logado no sistema. 2. O sistema está na tela inicial, aguardando uma ação do usuário.
Fluxo de eventos principais: <ol style="list-style-type: none"> 1. O usuário acessa a tela de movimentações através do menu. {Filtrar movimentações} 2. O usuário clica em uma movimentação. 3. O sistema abre uma tela com todos os detalhes da movimentação
Pós-condições:
Fluxo de eventos alternativo: <ol style="list-style-type: none"> 1. <i>Filtro de movimentações</i>

Em {Filtrar movimentações}, o usuário pode aplicar ou não um filtro em suas movimentações (ordem cronológica, maior/menor valor, receitas, despesas, etc.)

3.3.8. Caso de uso Alterar Movimentação

Caso de Uso: Alterar movimentação
ID: 8
Resumo: O usuário altera alguma movimentação realizada previamente.
Atores primários: Usuário
Atores secundários: Nenhum
Pré-condições: <ol style="list-style-type: none">1. O usuário está logado no sistema.2. O sistema está na tela inicial, aguardando uma ação do usuário.
Fluxo de eventos principais: <ol style="list-style-type: none">1. Executar Caso de Uso <u>Consultar movimentações</u>2. O usuário clica no botão para alterar.3. O usuário realiza suas alterações.4. O usuário salva a movimentação alterada.
Pós-condições: <ol style="list-style-type: none">1. O sistema deverá voltar a tela de movimentações.
Fluxo de eventos alternativo:

3.3.9. Caso de uso Remover Movimentação

Caso de Uso: Remover movimentação
ID: 9
Resumo: O usuário remove alguma movimentação realizada anteriormente.
Atores primários: Usuário
Atores secundários: Nenhum
Pré-condições: <ol style="list-style-type: none">1. O usuário está logado no sistema.

2. O sistema está na tela inicial, aguardando uma ação do usuário.
Fluxo de eventos principais: <ol style="list-style-type: none"> 1. Executar Caso de Uso <u>Consultar movimentações</u> 2. O usuário clica no botão para remover. 3. O usuário confirma a remoção.
Pós-condições: <ol style="list-style-type: none"> 1. O sistema deverá voltar a tela de movimentações.
Fluxo de eventos alternativo:

3.3.10. Caso de uso Inserir Movimentação

Caso de Uso: Inserir Movimentação
ID: 10
Resumo: usuário submete ao sistema uma quantia que deseja movimentar em uma de suas contas. O sistema identifica o tipo de movimentação e executa o caso de uso específico. Ao final, o sistema pergunta se o usuário deseja realizar outra movimentação.
Atores primários: usuário
Atores secundários: nenhum
Pré-condições: <ol style="list-style-type: none"> 1. Login efetuado. 2. Ao menos uma conta contábil registrada.
Fluxo de eventos principais: <ol style="list-style-type: none"> 1. Usuário clica na opção de inserir uma movimentação. 2. O sistema abre uma interface com campos para inserção de dados. 3. O usuário informa o tipo de movimentação em um menu. 4. O sistema reconhece o tipo de movimentação e chama o caso de uso específico da movimentação (IDs 12, 14, 15). 5. O sistema pergunta ao usuário se deseja realizar uma nova movimentação. 6. Usuário confirma se deseja ou não uma nova movimentação.
Pós-condições: <ol style="list-style-type: none"> 1. Usuário retorna ao menu principal. 2. Agendado uma sincronização do banco de dados local com o remoto.
Fluxo de eventos alternativo:

3.3.11. Caso de uso Agendar Débito

Caso de Uso: Agendar Débito
ID: 11
Resumo: o sistema recebe do usuário a ação de agendar um débito. O sistema exibe na tela um campo para a data de agendamento e o usuário preenche este campo. O sistema valida o campo e retorna para caso de uso “inserir débito (ID 12).
Atores primários: usuário
Atores secundários: nenhum
Pré-condições: <ol style="list-style-type: none">1. Login efetuado.2. Ao menos uma conta contábil registrada.3. Caso de uso 12 ter sido executado.
Fluxo de eventos principais: <ol style="list-style-type: none">1. O sistema exibe um campo com informações de agendamento.2. O usuário informa os campos de agendamento.3. O sistema valida os campos.
Pós-condições: <ol style="list-style-type: none">1. Retorna ao caso de uso “inserir débito” (ID 12).
Fluxo de eventos alternativo:

3.3.12. Caso de uso Inserir Débito

Caso de Uso: Inserir Débito
ID: 12
Resumo: o usuário informa um montante e se deseja agendar o débito ou realiza-lo prontamente. O sistema valida os campos, chama o caso de uso “agendar débito” (ID 11) se necessário. O sistema armazena os dados no banco de dados local.
Atores primários: usuário
Atores secundários: nenhum
Pré-condições: <ol style="list-style-type: none">1. Login efetuado.

<ol style="list-style-type: none"> 2. Ao menos uma conta contábil registrada. 3. Caso de uso “inserir movimentação” (ID 10) ter sido executado.
Fluxo de eventos principais: <ol style="list-style-type: none"> 1. O sistema exibe campos para inserção de débito. 2. Usuário informa dados de movimentação. 3. O sistema valida os campos referentes a movimentação. 4. O sistema valida se o usuário deseja agendar movimentação. Caso verdadeiro, chamar caso de uso “agendar débito” (ID 11).
Pós-condições: <ol style="list-style-type: none"> 1. O sistema registra débito em banco de dados local. 2. Retorna ao caso de uso “inserir movimentação” (ID 10).
Fluxo de eventos alternativo:

3.3.13. Caso de uso Agendar Crédito

Caso de Uso: Agendar Crédito
ID: 13
Resumo: o sistema recebe do usuário a ação de agendar um crédito. O sistema exibe na tela um campo para a data de agendamento e o usuário preenche este campo. O sistema valida o campo e retorna para caso de uso “inserir crédito” (ID 14).
Atores primários: usuário
Atores secundários: nenhum
Pré-condições: <ol style="list-style-type: none"> 1. Login efetuado. 2. Ao menos uma conta contábil registrada. 3. Caso de uso “inserir crédito” (ID 14) ter sido executado.
Fluxo de eventos principais: <ol style="list-style-type: none"> 1. O sistema exibe um campo com informações de agendamento. 2. O usuário informa os campos de agendamento. 3. O sistema valida os campos.
Pós-condições: <ol style="list-style-type: none"> 1. Retorna ao caso de uso “inserir crédito” (ID 14).
Fluxo de eventos alternativo:

3.3.14. Caso de uso Inserir Crédito

Caso de Uso: inserir crédito
ID: 14
Resumo: o usuário informa um montante e se deseja agendar o crédito ou realiza-lo prontamente. O sistema valida os campos, chama o caso de uso “agendar crédito” (ID 13) se necessário. O sistema armazena os dados no banco de dados local.
Atores primários: usuário
Atores secundários: nenhum
Pré-condições: <ol style="list-style-type: none">1. Login efetuado.2. Ao menos uma conta contábil registrada.3. Caso de uso “inserir movimentação” (ID 10) ter sido executado.
Fluxo de eventos principais: <ol style="list-style-type: none">1. O sistema exibe campos para inserção de crédito.2. Usuário informa dados de movimentação.3. O sistema valida os campos referentes a movimentação.4. O sistema valida se o usuário deseja agendar movimentação. Caso verdadeiro, chamar caso de uso “agendar crédito” (ID 13).
Pós-condições: <ol style="list-style-type: none">1. O sistema registra débito em banco de dados local.2. Retorna ao caso de uso “inserir movimentação” (ID 10).
Fluxo de eventos alternativo:

3.3.15. Caso de uso transferir entre Contas

Caso de Uso: Transferir entre contas
ID: 15
Resumo: o sistema exibe para o usuário suas contas contábeis. O usuário seleciona entre quais contas deseja realizar a transferência. O sistema valida os campos e realiza uma transferência de valores entre as duas contas.
Atores primários: usuário
Atores secundários: nenhum

Pré-condições: <ol style="list-style-type: none"> 1. Login efetuado. 2. Ao menos duas contas contábil registrada. 3. Caso de uso “inserir movimentação” (ID 10) ter sido executado.
Fluxo de eventos principais: <ol style="list-style-type: none"> 1. O sistema exibe as contas contábeis do usuário. 2. O usuário seleciona qual conta terá valores transferidos e qual conta receberá os valores. 3. O sistema valida as informações. 4. O usuário confirma a movimentação.
Pós-condições: <ol style="list-style-type: none"> 1. Transação deve ser registrada no banco de dados local. 2. Retorna ao caso de uso “inserir movimentação” (ID 10).
Fluxo de eventos alternativo:

3.3.16. Caso de uso Fazer Login

Caso de Uso: Fazer Login
ID: 16
Resumo: O usuário insere suas credenciais para entrar no sistema.
Atores primários: Usuário
Atores secundários: Nenhum
Pré-condições: <ol style="list-style-type: none"> 1. O sistema deve estar aberto na tela de login. 2. O login não foi efetuado ainda.
Fluxo de eventos principais: <ol style="list-style-type: none"> 1. O usuário digita seu e-mail. 2. O usuário digita sua senha. 3. O usuário clica em fazer login. {Autenticação de usuário} 4. O sistema abre a tela inicial.
Pós-condições: <ol style="list-style-type: none"> 1. O usuário está logado no sistema.

2. O sistema estará aberto na tela inicial.

Fluxo de eventos alternativo:

1. *E-mail ou senha inválidos*

1. Em {Autenticação de usuário}, o sistema faz a autenticação do usuário juntamente ao servidor. Se o e-mail não se refere a uma conta cadastrada ou a senha está incorreta:

O sistema solicita que o usuário digite um e-mail e senha válidos.

3.3.17. Caso de uso Fazer Login com Google

Caso de Uso: Fazer login com Google
ID: 17
Resumo: o usuário utiliza de sua conta Google para fazer login.
Atores primários: usuário
Atores secundários: nenhum
Pré-condições: <ol style="list-style-type: none">1. O sistema deve estar aberto na tela de login.2. O login não foi efetuado ainda.
Fluxo de eventos principais: <ol style="list-style-type: none">1. O usuário insere seu e-mail cadastrado no Google.2. O usuário insere sua senha de cadastro da conta Google.3. O usuário clica em fazer login.4. O sistema manda as informações para o servidor de autenticação do Google. {Autenticação de usuário}
Pós-condições: <ol style="list-style-type: none">1. O usuário está logado no sistema.2. O sistema estará aberto na tela inicial.
Fluxo de eventos alternativo: <ol style="list-style-type: none">a. <i>E-mail ou senha inválidos</i><ol style="list-style-type: none">1. Em {Autenticação de usuário}, o sistema faz a autenticação de usuário junto dos servidores do Google. Se o e-mail não se refere a uma conta cadastrada ou a senha está incorreta: o sistema solicita que o usuário digite um e-mail e senha válidos.

3.3.18. Caso de uso Realizar Cadastro

Caso de Uso: Realizar cadastro
ID: 18
Resumo: O usuário realiza seu cadastro no sistema.
Atores primários: Usuário
Atores secundários: Nenhum
Pré-condições: <ol style="list-style-type: none">1. O sistema deve estar aberto na tela de login.2. O login não foi efetuado ainda.
Fluxo de eventos principais: <ol style="list-style-type: none">1. O usuário clica em Cadastrar-se2. O usuário insere os dados cadastrais3. O usuário clica em Cadastrar-se. {Validação de dados de cadastro}4. O sistema abre a tela inicial.
Pós-condições: <ol style="list-style-type: none">1. O usuário está logado no sistema.2. O sistema estará aberto na tela inicial.
Fluxo de eventos alternativo: <ol style="list-style-type: none">1. <i>E-mail já cadastrado</i><ol style="list-style-type: none">1. Em {Validação de dados de cadastro}, o sistema envia os dados de cadastro ao servidor. Se o e-mail inserido já está cadastrado: O sistema solicita que o usuário digite um e-mail diferente ou faça login.2. <i>Campos obrigatórios não preenchidos</i><ol style="list-style-type: none">1. Em {Validação de dados de cadastro}, o sistema verifica se todos os campos obrigatórios foram preenchidos antes de enviar os dados ao servidor. Se faltar algum campo: O sistema deixa o campo vazio focado e solicita que o usuário o preencha.

3.3.19. Caso de uso Fazer Login com Facebook

Caso de Uso: Fazer login com Facebook
--

ID: 19
Resumo: o usuário utiliza de sua conta Facebook para fazer login.
Atores primários: usuário
Atores secundários: nenhum
Pré-condições: <ol style="list-style-type: none"> O sistema deve estar aberto na tela de login. O login não foi efetuado ainda.
Fluxo de eventos principais: <ol style="list-style-type: none"> O usuário insere seu e-mail cadastrado no Facebook. O usuário insere sua senha de cadastro da conta Facebook. O usuário clica em fazer login. O sistema manda as informações para o servidor de autenticação do Facebook. {Autenticação de usuário}
Pós-condições: <ol style="list-style-type: none"> O usuário está logado no sistema. O sistema estará aberto na tela inicial.
Fluxo de eventos alternativo: <ol style="list-style-type: none"> <i>E-mail ou senha inválidos</i> <ol style="list-style-type: none"> Em {Autenticação de usuário}, o sistema faz a autenticação de usuário junto dos servidores do Facebook. Se o e-mail não se refere a uma conta cadastrada ou a senha está incorreta: o sistema solicita que o usuário digite um e-mail e senha válidos.

3.3.20. Caso de uso Ver Projeção do Objetivo

Caso de Uso: Ver Projeção do Objetivo
ID: 20
Resumo: O usuário irá ver o acompanhamento e progresso em seu objetivo específico
Atores primários: Usuário
Atores secundários: Nenhum
Pré-condições: <ol style="list-style-type: none"> Estar logado no sistema

2. Possuir uma conta contábil 3. Possuir um objetivo criado e em andamento
Fluxo de eventos principais: <ol style="list-style-type: none"> 1. O usuário irá acessar o menu de opções 2. O usuário irá entrar no menu de objetivos 3. O usuário irá selecionar o objetivo que deseja receber informações detalhadas
Pós-condições: Nenhuma
Fluxo de eventos alternativo: Nenhum

3.3.21. Caso de uso Alterar Objetivo

Caso de Uso: Alterar Objetivo
ID: 21
Resumo: O usuário irá alterar algum objetivo sua conta contábil que esteja em andamento ou planejamento
Atores primários: Usuário
Atores secundários: Nenhum
Pré-condições: <ol style="list-style-type: none"> 1. Estar logado no sistema 2. Possuir uma conta contábil 3. Possuir um objetivo criado
Fluxo de eventos principais: <ol style="list-style-type: none"> 1. O usuário irá acessar o menu de opções 2. O usuário irá entrar no menu de objetivos 3. O usuário irá selecionar o objetivo que deseja realizar alguma alteração 4. O usuário irá realizar a ação
Pós-condições: <ol style="list-style-type: none"> 1. Será apresentado uma mensagem de confirmação da ação
Fluxo de eventos alternativo: Nenhum

3.3.22. Caso de uso Remover Objetivo

Caso de Uso: Remover Objetivo
ID: 22
Resumo: O usuário irá remover algum objetivo de sua conta contábil que já tenha sido cumprido ou abandonado
Atores primários: Usuário
Atores secundários: Nenhum
Pré-condições: <ol style="list-style-type: none"> 1. Estar logado no sistema 2. Possuir uma conta contábil 3. Possuir um objetivo criado
Fluxo de eventos principais: <ol style="list-style-type: none"> 1. O usuário irá acessar o menu de opções 2. O usuário irá entrar no menu de objetivos 3. O usuário irá selecionar o objetivo que deseja remover 4. O usuário irá realizar a ação
Pós-condições: <ol style="list-style-type: none"> 1. Será apresentado uma mensagem de confirmação da ação
Fluxo de eventos alternativo: Nenhum

3.3.23. Caso de uso Arquivar Objetivo

Caso de Uso: Arquivar Objetivo
ID: 23
Resumo: O usuário irá arquivar algum objetivo de sua conta contábil que já tenha sido cumprido ou abandonado e que não deseja remover do sistema
Atores primários: Usuário
Atores secundários: Nenhum
Pré-condições: <ol style="list-style-type: none"> 1. Estar logado no sistema 2. Possuir uma conta contábil 3. Possuir um objetivo criado
Fluxo de eventos principais:

<ol style="list-style-type: none"> 1. O usuário irá acessar o menu de opções 2. O usuário irá entrar no menu de objetivos 3. O usuário irá selecionar o objetivo que deseja arquivar 4. O usuário irá realizar a ação
Pós-condições: <ol style="list-style-type: none"> 1. Será apresentado uma mensagem de confirmação da ação
Fluxo de eventos alternativo: Nenhum

3.3.24. Caso de uso Listar Objetivos

Caso de Uso: Listar Objetivos
ID: 24
Resumo: O usuário irá listar todos os objetivos existentes vinculados à sua conta contábil
Atores primários: Usuário
Atores secundários: Nenhum
Pré-condições: <ol style="list-style-type: none"> 1. Estar logado no sistema 2. Possuir uma conta contábil 3. Possuir no mínimo um objetivo criado
Fluxo de eventos principais: <ol style="list-style-type: none"> 1. O usuário irá acessar o menu de opções 2. O usuário irá entrar no menu de objetivos 3. O usuário irá selecionar a lista de objetivos que deseja visualizar
Pós-condições: Nenhuma
Fluxo de eventos alternativo: Nenhum

4. Modelo de Análise do Sistema

4.1. Especificação das classes de análise do sistema

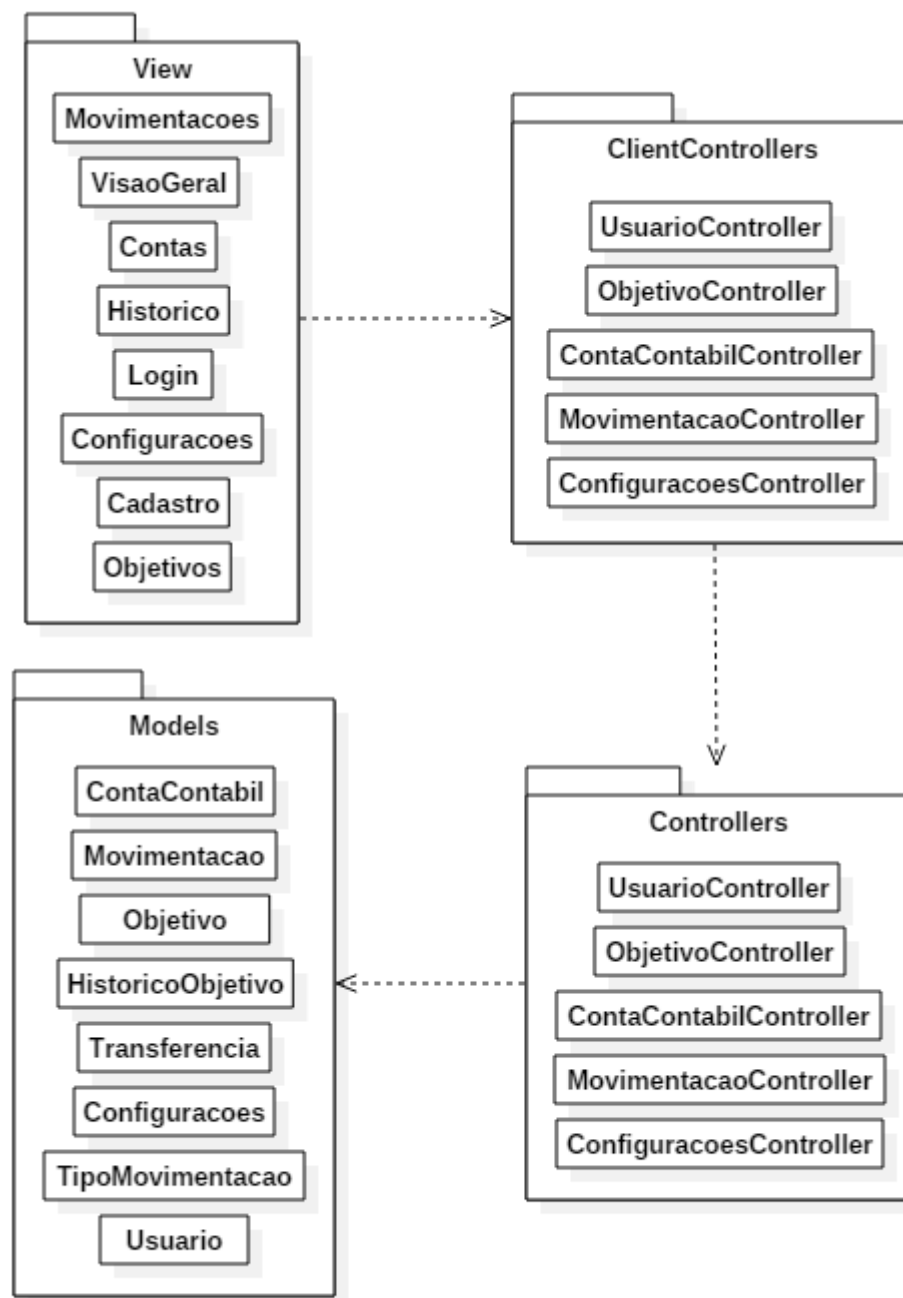
View – O pacote view serve para armazenar as interfaces do sistema, elas serão o meio de comunicação do sistema com o usuário.

Controllers - O pacote controllers serve para armazenar as classes de controle do sistema (controllers), em que são colocados os métodos e funções que serão executados na API.

Models – O pacote models serve para armazenar as classes de modelo do sistema, essas classes armazenarão os atributos de cada entidade presente no banco de dados.

ClientControllers – O pacote ClientControllers serve para armazenar as classes que conterão os métodos e funções que serão executados no sistema pelo aplicativo.

4.1.1. Visão Geral dos Pacotes de Análise



4.1.1.1. Pacote View

O pacote view encapsula todas as classes referentes a interface com o usuário.

4.1.1.1. Pacote ClientControllers

O pacote ClientControllers encapsula todas as classes referentes a lógica de programação cliente-side, isto é, todas as classes que estão realizando seus processos no dispositivo do usuário final.

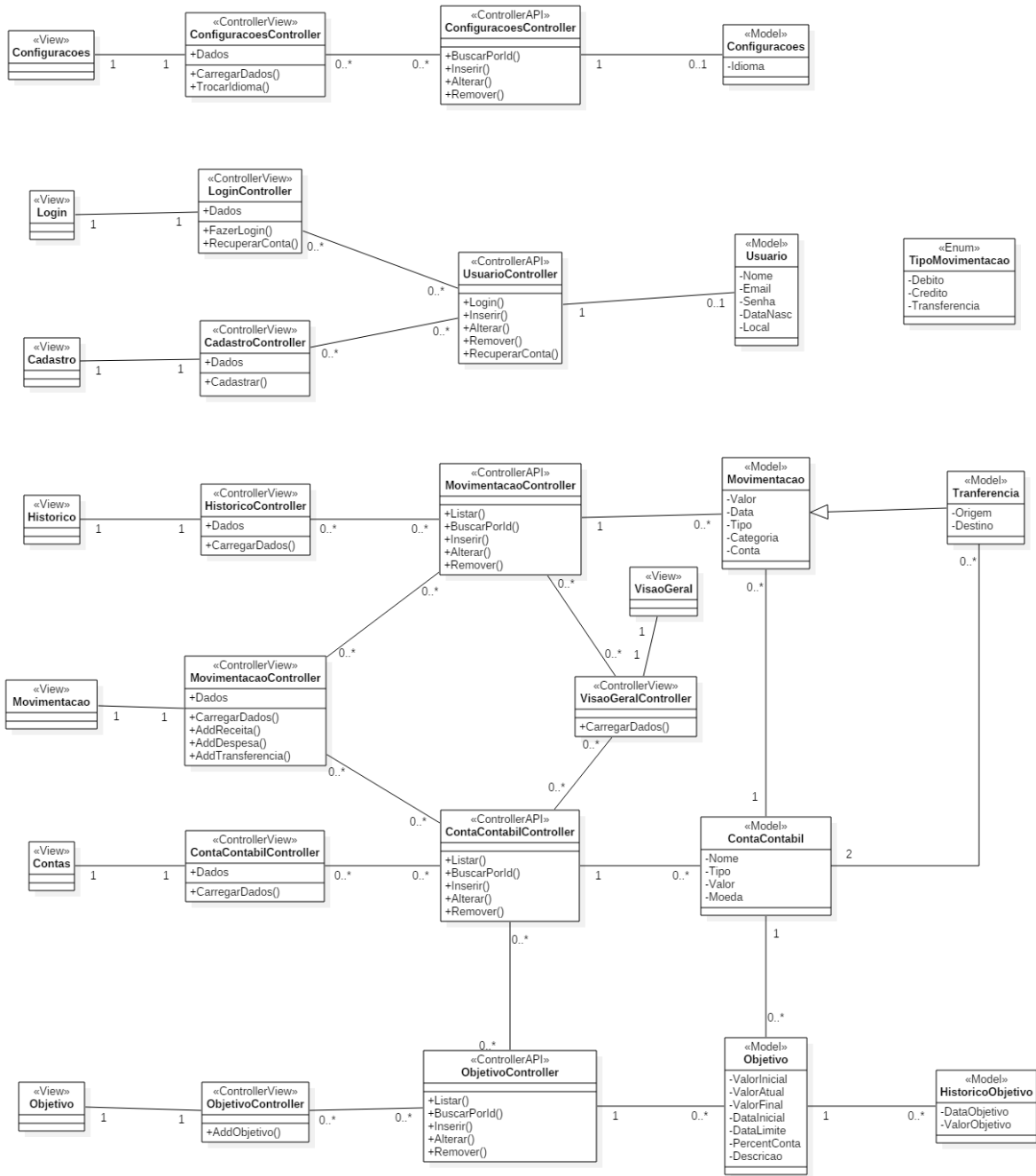
4.1.1.1. Pacote Controllers

O pacote Controllers encapsula todas as classes controladoras referentes ao servidor da aplicação.

4.1.1.1. Pacote Models

O pacote `models` encapsula todas classes referentes aos dados a serem manipulados pela aplicação server-side.

4.1.2. Diagramas e especificações das classes



4.1.2.1. Classe Visão Geral

Descrição: Na classe visão geral será apresentado um resumo básico de todas as últimas atualizações das contas contábeis e todas as contas contábeis existentes.

Superclasse: nenhuma

Subclasse: nenhuma

Atributos: nenhum

Métodos: nenhum

4.1.2.2. Classe Contas

Descrição: Na classe contas será apresentado ao usuário todas as contas contábeis existentes detalhadamente e é onde serão criadas novas.

Superclasse: nenhuma

Subclasse: nenhuma

Atributos: nenhum

Métodos: nenhum

4.1.2.3. Classe Movimentações

Descrição: Na classe movimentações é possível para o usuário consultar todas as movimentações realizadas, sendo elas despesas ou receitas, e também adicionar novas à uma conta existente.

Superclasse: nenhuma

Subclasse: nenhuma

Atributos: nenhum

Métodos: nenhum

4.1.2.4. Classe Login

Descrição: Nesta classe o usuário entrará com seu login e sua senha para acessar sua conta no aplicativo. Caso o usuário não possua uma conta, há a opção de ir para a tela de criação de contas.

Superclasse: nenhuma

Subclasse: nenhuma

Atributos: nenhum

Métodos: nenhum

4.1.2.5. Classe Criar Conta

Descrição: Nesta classe os usuários que não possuem uma conta poderão criar uma para utilizar o aplicativo. É possível criar uma conta vinculada com o Facebook ou com uma conta Google.

Superclasse: nenhuma

Subclasse: nenhuma

Atributos: nenhum

Métodos: nenhum

4.1.2.6. Classe Histórico

Descrição: Na classe histórico será apresentado ao usuário um resumo geral da conta separado por mês, mostrando o total de receitas, despesas e o saldo no final do mês.

Superclasse: nenhuma

Subclasse: nenhuma

Atributos: nenhum

Métodos: nenhum

4.1.2.7. Classe Configurações

Descrição: Nesta classe o usuário poderá realizar configurações básicas em seu aplicativo, como trocar idioma, e também é apresentado informações relacionadas ao aplicativo em si (opção “Sobre”).

Superclasse: nenhuma

Subclasse: nenhuma

Atributos: nenhum

Métodos: nenhum

4.1.2.8. Classe Objetivos

Descrição: Nesta classe o usuário poderá criar metas a serem alcançadas. Nela será possível adicionar uma quantia objetivo a ser poupada com uma data estipulada pelo usuário, as informações serão atualizadas automaticamente à cada movimentação do usuário na conta vinculada.

Superclasse: nenhuma

Subclasse: nenhuma

Atributos: nenhum

Métodos: nenhum

4.1.2.9. Classe ConfiguracoesController

Classe: ConfiguracoesController

Descrição: A classe será o controlador responsável pelas configurações do sistema.

Superclasse: nenhuma

Subclasse: nenhuma

Atributos: nenhum

Operações:

Nome:	BuscarPorId
Descrição:	Busca as configurações do usuário no banco de dados, utilizando seu Id.
Pré-condições:	
1. O usuário deve estar logado no sistema.	
Especificação:	
1. Busca as configurações do usuário no banco de dados utilizando uma Id 2. Retorna essas configurações	
Pós-condições:	
Nenhuma.	

Nome:	Inserir
Descrição:	Insere as configurações do usuário no banco de dados, e as relaciona pelo seu Id.
Pré-condições:	
1. O usuário deve estar logado no sistema.	
Especificação:	
1. Insere as configurações do usuário no banco de dados 2. Retorna um código de estado HTTP	
Pós-condições:	
1. As configurações deverão estar salvas no banco de dados.	

Nome:	Alterar
Descrição:	Altera as configurações do usuário no banco de dados, utilizando seu Id.
Pré-condições:	
1. O usuário deve estar logado no sistema.	
Especificação:	
1. Altera as configurações do usuário no banco de dados utilizando uma Id 2. Retorna um código de estado HTTP	
Pós-condições:	
1. As configurações deverão estar atualizadas no banco de dados.	

Nome:	Remover
Descrição:	Remove as configurações do usuário no banco de dados, utilizando seu Id.
Pré-condições:	
1. O usuário deve estar logado no sistema.	
Especificação:	
1. Remove as configurações do usuário no banco de dados utilizando uma Id	
2. Retorna um código de estado HTTP	
Pós-condições:	
1. As configurações não deverão mais existir no banco de dados.	

4.1.2.10. Classe UsuarioController

Classe: UsuarioController

Descrição: a classe será o controlador responsável pelos usuários do sistema.

Superclasse: nenhuma

Subclasse: nenhuma

Atributos: nenhum

Operações:

Nome:	Login
Descrição:	Valida o usuário do sistema, utilizando seu e-mail e senha.
Pré-condições:	
1. O usuário deve estar cadastrado no sistema.	
Especificação:	
1. Busca as informações do usuário no banco de dados utilizando seu e-mail	
2. Valida essas informações para ver se o usuário é valido	
3. Retorna um código de estado HTTP	
Pós-condições:	
1. O usuário estará logado no sistema.	

Nome:	Inserir
Descrição:	Insere as informações do usuário no banco de dados.
Pré-condições:	
1. As informações devem ser validadas.	
Especificação:	
1. Insere as informações do usuário no banco de dados	
2. Retorna um código de estado HTTP	
Pós-condições:	
1. As informações deverão estar salvas no banco de dados.	

Nome:	Alterar
Descrição:	Altera as informações do usuário no banco de dados.
Pré-condições:	

1. O usuário deve estar logado no sistema.
2. As informações devem ser validadas
Especificação:
1. Altera as informações do usuário no banco de dados utilizando seu e-mail
2. Retorna um código de estado HTTP
Pós-condições:
1. As informações deverão estar atualizadas no banco de dados.

Nome:	Remover
Descrição:	Remove as informações do usuário no banco de dados.
Pré-condições:	
1. O usuário deve estar logado no sistema.	
Especificação:	
1. Remove as informações do usuário no banco de dados utilizando seu e-mail	
2. Retorna um código de estado HTTP	
Pós-condições:	
1. As informações não deverão mais existir no banco de dados.	
2. O usuário será desconectado do sistema.	

Nome:	RecuperarConta
Descrição:	Auxilia o usuário na recuperação da sua conta em caso de perda da sua senha
Pré-condições:	
1. O usuário deve estar logado no sistema.	
Especificação:	
1. Busca o usuário no banco de dados, utilizando um e-mail	
1. Se for valido:	
1. Envia um e-mail de recuperação para o usuário cadastrar uma nova senha	
2. Retorna um código de estado HTTP	
1. Se não for valido:	
1. Retorna um código de estado HTTP	
Pós-condições:	
1. As informações não deverão mais existir no banco de dados.	
2. O usuário será desconectado do sistema.	

4.1.2.11. Classe MovimentacaoController

Classe: MovimentacaoController

Descrição: a classe será o controlador responsável pelas movimentações do usuário no sistema.

Superclasse: nenhuma

Subclasse: nenhuma

Atributos: nenhum

Operações:

Nome:	Listar
--------------	--------

Descrição:	Lista as movimentações do usuário no banco de dados, utilizando seu Id.
Pré-condições:	
1. O usuário deve estar logado no sistema.	
Especificação:	
1. Busca todas as movimentações do usuário no banco de dados, utilizando sua Id	
2. Retorna essas informações	
Pós-condições:	
Nenhuma.	

Nome:	BuscarPorId
Descrição:	Busca as movimentações do usuário no banco de dados, utilizando seu Id.
Pré-condições:	
1. O usuário deve estar logado no sistema.	
Especificação:	
1. Busca uma movimentação específica no banco de dados utilizando uma Id	
2. Retorna essa informação	
Pós-condições:	
Nenhuma.	

Nome:	Inserir
Descrição:	Insere as movimentações do usuário no banco de dados, e as relaciona pelo seu Id.
Pré-condições:	
1. O usuário deve estar logado no sistema.	
2. As informações devem ser validadas.	
Especificação:	
1. Insere a movimentação do usuário no banco de dados	
2. Retorna um código de estado HTTP	
Pós-condições:	
1. As movimentações deverão estar salvas no banco de dados.	

Nome:	Alterar
Descrição:	Altera as movimentações do usuário no banco de dados, utilizando seu Id.
Pré-condições:	
1. O usuário deve estar logado no sistema.	
2. As informações devem ser validadas.	
Especificação:	
1. Altera a movimentação do usuário no banco de dados utilizando uma Id	
2. Retorna um código de estado HTTP	
Pós-condições:	
1. As movimentações deverão estar atualizadas no banco de dados.	

Nome:	Remover
Descrição:	Remove as movimentações do usuário no banco de dados, utilizando seu Id.
Pré-condições:	
1. O usuário deve estar logado no sistema.	

Especificação:
1. Remove a movimentação do usuário no banco de dados utilizando uma Id 2. Retorna um código de estado HTTP
Pós-condições:
1. As movimentações não deverão mais existir no banco de dados.

4.1.2.12. Classe ContaContabilController

Classe: ContaContabilController

Descrição: a classe será o controlador responsável pelas contas contábeis do usuário no sistema.

Superclasse: nenhuma

Subclasse: nenhuma

Atributos: nenhum

Operações:

Nome:	Listar
Descrição:	Lista as contas contábeis do usuário no banco de dados, utilizando seu Id.
Pré-condições:	1. O usuário deve estar logado no sistema.
Especificação:	1. Busca as contas contábeis do usuário no banco de dados utilizando uma Id 2. Retorna essas informações
Pós-condições:	Nenhuma.

Nome:	BuscarPorId
Descrição:	Busca as contas contábeis do usuário no banco de dados, utilizando seu Id.
Pré-condições:	1. O usuário deve estar logado no sistema.
Especificação:	1. Busca uma conta contábil específica, por sua Id 2. Retorna essa informação
Pós-condições:	Nenhuma.

Nome:	Inserir
Descrição:	Insere as contas contábeis do usuário no banco de dados, e as relaciona pelo seu Id.
Pré-condições:	1. O usuário deve estar logado no sistema. 2. As informações devem ser validadas.
Especificação:	1. Insere a conta contábil do usuário no banco de dados 2. Retorna um código de estado HTTP
Pós-condições:	

1. As contas contábeis deverão estar salvas no banco de dados.

Nome:	Alterar
Descrição:	Altera as contas contábeis do usuário no banco de dados, utilizando seu Id.
Pré-condições:	
<ol style="list-style-type: none"> 1. O usuário deve estar logado no sistema. 2. As informações devem ser validadas. 	
Especificação:	
<ol style="list-style-type: none"> 1. Altera a conta contábil do usuário no banco de dados utilizando uma Id 2. Retorna um código de estado HTTP 	
Pós-condições:	
<ol style="list-style-type: none"> 1. As contas contábeis deverão estar atualizadas no banco de dados. 	

Nome:	Remover
Descrição:	Remove as contas contábeis do usuário no banco de dados, utilizando seu Id.
Pré-condições:	
<ol style="list-style-type: none"> 1. O usuário deve estar logado no sistema. 	
Especificação:	
<ol style="list-style-type: none"> 1. Remove a conta contábil do usuário no banco de dados utilizando uma Id 2. Retorna um código de estado HTTP 	
Pós-condições:	
<ol style="list-style-type: none"> 1. As contas contábeis não deverão mais existir no banco de dados. 	

4.1.2.13. Classe ObjetivoController

Classe: ObjetivoController

Descrição: a classe será o controlador responsável pelos objetivos do usuário no sistema.

Superclasse: nenhuma

Subclasse: nenhuma

Atributos: nenhum

Operações:

Nome:	Listar
Descrição:	Lista os objetivos do usuário no banco de dados, utilizando seu Id.
Pré-condições:	
<ol style="list-style-type: none"> 1. O usuário deve estar logado no sistema. 	
Especificação:	
<ol style="list-style-type: none"> 1. Busca os objetivos do usuário no banco de dados utilizando uma Id 2. Retorna essas informações 	
Pós-condições:	
Nenhuma.	

Nome:	BuscarPorId
Descrição:	Busca os objetivos do usuário no banco de dados, utilizando seu Id.
Pré-condições:	

1. O usuário deve estar logado no sistema.
Especificação
1. Busca um objetivo específico do usuário no banco de dados, utilizando uma Id 2. Retorna essa informação
Pós-condições:
Nenhuma.

Nome:	Inserir
Descrição:	Insere os objetivos do usuário no banco de dados, e os relaciona pelo seu Id.
Pré-condições:	
	1. O usuário deve estar logado no sistema. 2. As informações devem ser validadas.
Especificação:	
	1. Insere as informações do usuário no sistema 2. Retorna um código de estado HTTP
Pós-condições:	
	1. Os objetivos deverão estar salvos no banco de dados.

Nome:	Alterar
Descrição:	Altera os objetivos do usuário no banco de dados, utilizando seu Id.
Pré-condições:	
	1. O usuário deve estar logado no sistema. 2. As informações devem ser validadas.
Especificação:	
	1. Altera as informações do usuário no banco de dados utilizando uma Id 2. Retorna um código de estado HTTP
Pós-condições:	
	1. Os objetivos deverão estar atualizados no banco de dados.

Nome:	Remover
Descrição:	Remove os objetivos do usuário no banco de dados, utilizando seu Id.
Pré-condições:	
	1. O usuário deve estar logado no sistema.
Especificação:	
	1. Remove as informações do usuário no banco de dados utilizando uma Id 2. Retorna um código de estado HTTP
Pós-condições:	
	1. Os objetivos não deverão mais existir no banco de dados.

4.1.2.14. Classe Configuracoes

Classe Configuracoes

Descrição: armazena configurações pessoais do usuário do aplicativo.

Superclasse: nenhuma

Subclasse: nenhuma

Atributos:

Nome:	Idioma
Descrição:	Armazena a configuração de idioma desejado.

4.1.2.15. Classe Usuario**Classe Usuario**

Descrição: armazena informações referentes ao usuário da aplicação.

Superclasse: nenhuma

Subclasse: nenhuma

Atributos:

Nome:	Nome
Descrição:	Armazena o nome de usuário.

Nome:	Email
Descrição:	Armazena o email de usuário.

Nome:	Senha
Descrição:	Armazena a senha de usuário.

Nome:	DataNasc
Descrição:	Armazena a data de nascimento de usuário.

Nome:	Local
Descrição:	Armazena o local geográfico do usuário.

4.1.2.16. Classe Movimentacao**Classe Movimentacao**

Descrição: armazena dados de movimentação bancária.

Superclasse: nenhuma

Subclasse: nenhuma

Atributos:

Nome:	Valor
Descrição:	Valor monetário da transferência bancária.

Nome:	Data
Descrição:	Data em que a movimentação bancária ocorreu.

Nome:	Tipo
Descrição:	Caracteriza o tipo de movimentação bancária.

Nome:	Categoria
Descrição:	Categoriza a categoria de movimentação bancária.

Nome:	Conta
Descrição:	Conta de origem da movimentação bancária.

4.1.2.17. Classe Transferencia

Classe Transferencia

Descrição: armazena informações de destino e origem em uma transferência bancária.

Superclasse: nenhuma

Subclasse: nenhuma

Atributos:

Nome:	Origem
Descrição:	Conta de origem em uma transferência bancária.

Nome:	Destino
Descrição:	Conta de destino em uma transferência bancária.

4.1.2.18. Classe ContaContabil

Classe ContaContabil

Descrição: armazena informações sobre uma conta contábil do usuário no aplicativo

Superclasse: nenhuma

Subclasse: nenhuma

Atributos:

Nome:	Nome
Descrição:	Nome de identificação da conta contábil.

Nome:	Tipo
Descrição:	Tipo de conta contábil (conta corrente, poupança, etc).

Nome:	Valor
Descrição:	Valor monetário em crédito/débito na conta contábil.

Nome:	Moeda
Descrição:	Tipo de moeda usada nas transações da conta contábil.

4.1.2.19. Classe Objetivo

Classe Objetivo

Descrição: armazena informações de um objetivo financeiro do usuário.

Superclasse: nenhuma

Subclasse: nenhuma

Atributos:

Nome:	ValorInicial
Descrição:	Valor monetário disponível inicialmente no momento de criação de um objetivo.

Nome:	ValorAtual
Descrição:	Valor atualizado para data atual sobre o objetivo.

Nome:	ValorFinal
Descrição:	Valor desejado para conclusão do objetivo.

Nome:	DataInicial
Descrição:	Data de criação de um objetivo.

Nome:	DataLimite
Descrição:	Data limite (opcional) para conclusão de um objetivo.

Nome:	PercentConta
Descrição:	Porcentagem usada pelo objetivo na conta contábil.

Nome:	Descricao
Descrição:	Descrição do usuário sobre o objetivo.

4.1.2.20. Classe HistoricoObjetivo**Classe HistoricoObjetivo**

Descrição: armazena informações de histórico de um objetivo. Guarda movimentações em cima de um objetivo.

Superclasse: nenhuma

Subclasse: nenhuma

Atributos:

Nome:	DataObjetivo
Descrição:	Data de realização de um histórico de objetivo.

Nome:	ValorObjetivo
Descrição:	Valor de movimentação do elemento histórico de um objetivo.

4.1.2.21. Classe ConfiguracoesController (View)

Descrição: a classe ConfiguracoesController representa o controlador da tela de configurações na aplicação cliente.

Superclasse: nenhuma

Subclasse: nenhuma

Atributos:

Nome:	Dados
Descrição:	Representa o conjunto de dados a serem manipulados pela classe.

Operações:

Nome:	CarregarDados
Descrição:	Faz o carregamento de todos os dados necessários para a realização de outras operações.
Pré-condições:	
Nenhuma.	
Especificação (diagrama de atividade ou pseudocódigo):	
<ol style="list-style-type: none"> 1. Obtém da aplicação servidor os dados atuais. 2. Exibe os dados nos controles de tela. 	
Pós-condições:	
<ol style="list-style-type: none"> 1. Os dados são exibidos na tela. 	

Nome:	TrocarIdioma
Descrição:	Altera o idioma atual da aplicação.
Pré-condições:	
Nenhuma.	
Especificação (diagrama de atividade ou pseudocódigo):	
<ol style="list-style-type: none"> 1. Altera o idioma no objeto Dados. 2. Envia o objeto para a aplicação servidor realizar a persistência. 	
Pós-condições:	
<ol style="list-style-type: none"> 1. Os textos na tela serão exibidos de acordo com o idioma escolhido. 	

4.1.2.22. Classe LoginController (View)

Descrição: a classe LoginController representa o controlador da tela de Login na aplicação cliente.

Superclasse: nenhuma

Subclasse: nenhuma

Atributos:

Nome:	Dados
Descrição:	Representa o conjunto de dados a serem manipulados pela classe.

Operações:

Nome:	FazerLogin
Descrição:	Faz a identificação e autenticação do usuário da aplicação para garantir a posterior autorização do uso de seus dados.
Pré-condições:	

<ol style="list-style-type: none"> 1. O dispositivo em que a aplicação estiver sendo executada deve possuir conexão à internet. 2. Os campos “E-mail” e “Senha” devem estar preenchidos na tela.
Especificação (diagrama de atividade ou pseudocódigo):
<ol style="list-style-type: none"> 3. Envia os os valores de e-mail e senha para a aplicação servidor. 4. Analisa no retorno da aplicação servidor para ver se o usuário está autenticado: <ol style="list-style-type: none"> 1. Se estiver: <ol style="list-style-type: none"> 1. Inicia o download dos dados do usuário em segundo plano. 2. Prossegue para a tela Visão Geral. 2. Se não estiver, alerta o usuário de que o e-mail ou a senha estão incorretos.
Pós-condições:
<ol style="list-style-type: none"> 2. O usuário poderá usar a aplicação com uso de backup e sincronização de seus dados via aplicação servidor, caso seja autenticado com sucesso.

Nome:	LoginFacebook
Descrição:	Faz a identificação e autenticação do usuário da aplicação para garantir a posterior autorização do uso de seus dados, com o uso da API de autenticação do Facebook.
Pré-condições:	<ol style="list-style-type: none"> 3. O dispositivo em que a aplicação estiver sendo executada deve possuir conexão à internet.
Especificação (diagrama de atividade ou pseudocódigo):	<ol style="list-style-type: none"> 1. Invoca-se a API de autenticação do Facebook. 2. Exibe-se ao usuário a interface de Login do Facebook. 3. Obtém-se o retorno da API de login do Facebook. 4. Analisa no retorno para ver se o usuário está autenticado: <ol style="list-style-type: none"> 1. Se estiver: <ol style="list-style-type: none"> 1. Inicia o download dos dados do usuário em segundo plano. 2. Prossegue para a tela Visão Geral. 2. Se não estiver, alerta o usuário de que o e-mail ou a senha estão incorretos.
Pós-condições:	<ol style="list-style-type: none"> 1. O usuário poderá usar a aplicação com uso de backup e sincronização de seus dados via aplicação servidor, caso seja autenticado com sucesso.

Nome:	LoginGoogle
Descrição:	Faz a identificação e autenticação do usuário da aplicação para garantir a posterior autorização do uso de seus dados, com o uso da API de autenticação do Google.
Pré-condições:	<ol style="list-style-type: none"> 4. O dispositivo em que a aplicação estiver sendo executada deve possuir conexão à internet.
Especificação (diagrama de atividade ou pseudocódigo):	<ol style="list-style-type: none"> 5. Invoca-se a API de autenticação do Google. 6. Exibe-se ao usuário a interface de Login do Google. 7. Obtém-se o retorno da API de login do Google. 8. Analisa no retorno para ver se o usuário está autenticado:

<ol style="list-style-type: none"> Se estiver: <ol style="list-style-type: none"> Inicia o download dos dados do usuário em segundo plano. Prossegue para a tela Visão Geral. Se não estiver, alerta o usuário de que o e-mail ou a senha estão incorretos.
Pós-condições:
<ol style="list-style-type: none"> O usuário poderá usar a aplicação com uso de backup e sincronização de seus dados via aplicação servidor, caso seja autenticado com sucesso.

Nome:	RecuperarConta
Descrição:	Auxilia o usuário na recuperação de sua conta caso este tenha esquecido ou perdido sua senha.
Pré-condições:	
<ol style="list-style-type: none"> O dispositivo em que a aplicação estiver sendo executada deve possuir conexão à internet. O campo “E-mail” deve estar preenchido na tela. 	
Especificação (diagrama de atividade ou pseudocódigo):	
<ol style="list-style-type: none"> Envia solicitação de recuperação de conta para a aplicação servidor. Informa o usuário de que um e-mail lhe será enviado com instruções para recuperação de sua conta. 	
Pós-condições:	
<ol style="list-style-type: none"> O usuário receberá um e-mail com instruções para recuperação de conta. 	

4.1.2.23. Classe CadastroController (View)

Descrição: a classe CadastroController representa o controlador da tela de Cadastro de Usuários na aplicação cliente.

Superclasse: nenhuma

Subclasse: nenhuma

Atributos:

Nome:	Dados
Descrição:	Representa o conjunto de dados a serem manipulados pela classe.

Operações:

Nome:	Cadastrar
Descrição:	Faz o cadastro do usuário na aplicação servidor.
Pré-condições:	
<ol style="list-style-type: none"> O dispositivo em que a aplicação estiver sendo executada deve possuir mecanismos de armazenamento local ou conexão à internet. Os campos obrigatórios do formulário de cadastro devem estar devidamente preenchidos. 	
Especificação (diagrama de atividade ou pseudocódigo):	
<ol style="list-style-type: none"> O objeto Dados com os dados de cadastro de usuário é enviado à aplicação servidor. O retorno da da aplicação servidor é analisado para ver se o usuário foi cadastrado: 	

<ol style="list-style-type: none"> 1. Se foi, o usuário é redirecionado para a tela de Visão Geral. 2. Se não, é informado ao usuário qual o erro ocorrido no cadastro.
Pós-condições:
<ol style="list-style-type: none"> 1. O usuário poderá usar a aplicação com uso de backup e sincronização de seus dados via aplicação servidor, caso seja cadastrado com sucesso.

4.1.2.24. Classe HistoricoController

Descrição: a classe HistoricoController representa o controlador da tela de Historico na aplicação cliente.

Superclasse: nenhuma

Subclasse: nenhuma

Atributos:

Nome:	Dados
Descrição:	Representa o conjunto de dados a serem manipulados pela classe.

Operações:

Nome:	CarregarDados
Descrição:	Faz o carregamento de todos os dados necessários para a realização de outras operações.
Pré-condições:	Nenhuma.
Especificação (diagrama de atividade ou pseudocódigo):	<ol style="list-style-type: none"> 1. Obtém da aplicação servidor os dados atuais. 2. Exibe os dados nos controles de tela.
Pós-condições:	<ol style="list-style-type: none"> 1. Os dados são exibidos na tela.

4.1.2.25. Classe MovimentaçãoController (View)

Descrição: a classe MovimentacaoController representa o controlador da tela de Movimentacao na aplicação cliente.

Superclasse: nenhuma

Subclasse: nenhuma

Atributos:

Nome:	Dados
Descrição:	Representa o conjunto de dados a serem manipulados pela classe.

Operações:

Nome:	CarregarDados
Descrição:	Faz o carregamento de todos os dados necessários para a realização de outras operações.
Pré-condições:	
Nenhuma.	
Especificação (diagrama de atividade ou pseudocódigo):	
<ol style="list-style-type: none"> 1. Obtém da aplicação servidor os dados atuais. 2. Exibe os dados nos controles de tela. 	
Pós-condições:	
<ol style="list-style-type: none"> 1. Os dados são exibidos na tela. 	

Nome:	AddReceita
Descrição:	Adiciona uma receita nas movimentações de uma das contas contábeis do usuário.
Pré-condições:	
Nenhuma.	
Especificação (diagrama de atividade ou pseudocódigo):	
<ol style="list-style-type: none"> 1. Exibe o formulário de movimentação. 2. Aguarda o usuário preencher os dados e clicar em OK. 3. Se o formulário estiver com os campos obrigatórios devidamente preenchidos: <ol style="list-style-type: none"> 1. Adiciona a movimentação de receita à conta contábil. 2. Altera o total da conta de acordo com a receita. 3. Envia os novos dados para a aplicação servidor, para realizar a persistência destes. 	
Pós-condições:	
<ol style="list-style-type: none"> 1. O saldo da conta será alterado e a esta será adicionada a movimentação do tipo receita. 	

Nome:	AddDespesa
Descrição:	Adiciona uma despesa nas movimentações de uma das contas contábeis do usuário.
Pré-condições:	
Nenhuma.	
Especificação (diagrama de atividade ou pseudocódigo):	
<ol style="list-style-type: none"> 1. Exibe o formulário de movimentação. 2. Aguarda o usuário preencher os dados e clicar em OK. 3. Se o formulário estiver com os campos obrigatórios devidamente preenchidos: <ol style="list-style-type: none"> 1. Adiciona a movimentação de despesa à conta contábil. 2. Altera o total da conta de acordo com a despesa. 3. Envia os novos dados para a aplicação servidor, para realizar a persistência destes. 	
Pós-condições:	
<ol style="list-style-type: none"> 1. O saldo da conta será alterado e a esta será adicionada a movimentação do tipo despesa. 	

Nome:	AddTransferencia
--------------	------------------

Descrição:	Realiza uma transferencia de valores entre duas contas contábeis do usuário.
Pré-condições:	
1. O usuário precisa ter mais de uma conta contábil.	
Especificação (diagrama de atividade ou pseudocódigo):	
4. Exibe o formulário de transferência. 5. Aguarda o usuário preencher os dados e clicar em OK. 6. Se o formulário estiver com os campos obrigatórios devidamente preenchidos: <ol style="list-style-type: none"> 1. Adiciona a movimentação de transferência em ambas as contas contábeis. 2. Altera o total das contas de acordo com a transferência. 3. Envia os novos dados para a aplicação servidor, para realizar a persistência destes. 	
Pós-condições:	
1. O saldo das contas será alterado e a estas será adicionada a movimentação do tipo transferência.	

4.1.2.26. Classe VisaoGeralController (View)

Descrição: a classe Controller representa o controlador da tela de X na aplicação cliente.

Superclasse: nenhuma

Subclasse: nenhuma

Atributos:

Nome:	Dados
Descrição:	Representa o conjunto de dados a serem manipulados pela classe.

Operações:

Nome:	CarregarDados
Descrição:	Faz o carregamento de todos os dados necessários para a realização de outras operações.
Pré-condições:.	
Nenhuma.	
Especificação (diagrama de atividade ou pseudocódigo):	
<ol style="list-style-type: none"> 1. Obtém da aplicação servidor os dados atuais. 2. Exibe os dados nos controles de tela. 	
Pós-condições:	
1. Os dados são exibidos na tela.	

4.1.2.27. Classe ContaContabilController (View)

Descrição: a classe ContaContabilController representa o controlador da tela de Contas Contábeis na aplicação cliente.

Superclasse: nenhuma

Subclasse: nenhuma

Atributos:

Nome:	Dados
Descrição:	Representa o conjunto de dados a serem manipulados pela classe.

Operações:

Nome:	CarregarDados
Descrição:	Faz o carregamento de todos os dados necessários para a realização de outras operações.
Pré-condições:	
Nenhuma.	
Especificação (diagrama de atividade ou pseudocódigo):	
1. Obtém da aplicação servidor os dados atuais. 2. Exibe os dados nos controles de tela.	
Pós-condições:	
1. Os dados são exibidos na tela.	

4.1.2.28. Classe ObjetoController (View)

Descrição: a classe ObjetoController representa o controlador da tela de Objeto na aplicação cliente.

Superclasse: nenhuma

Subclasse: nenhuma

Atributos

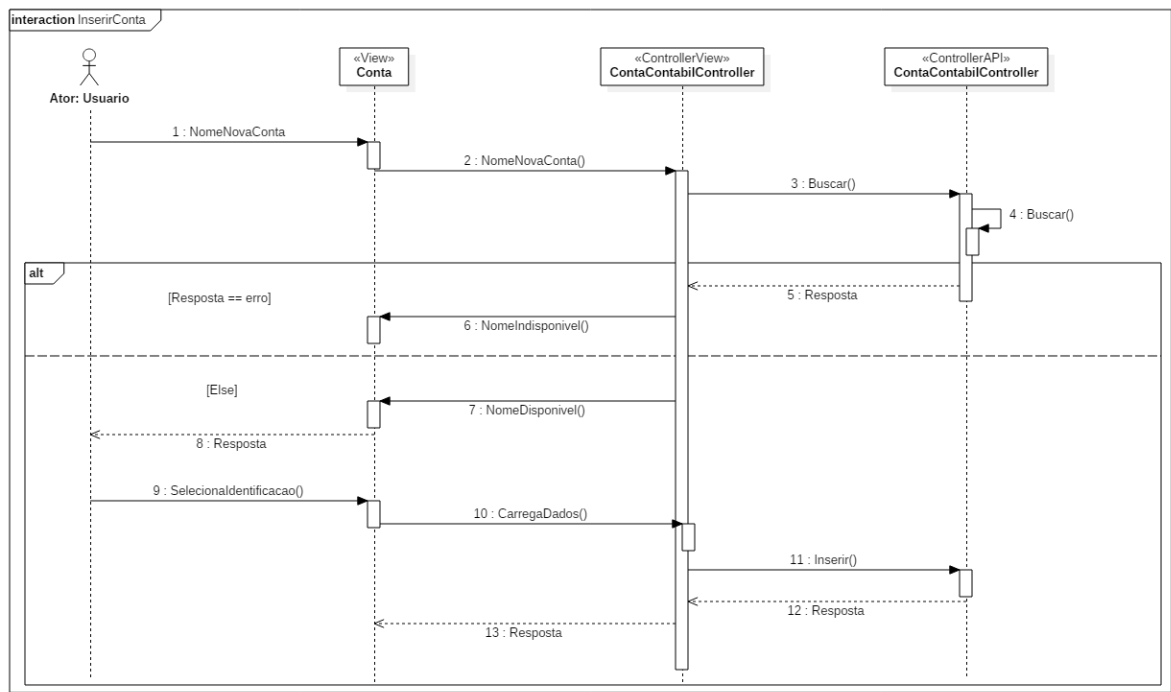
Nome:	Dados
Descrição:	Representa o conjunto de dados a serem manipulados pela classe.

Operações

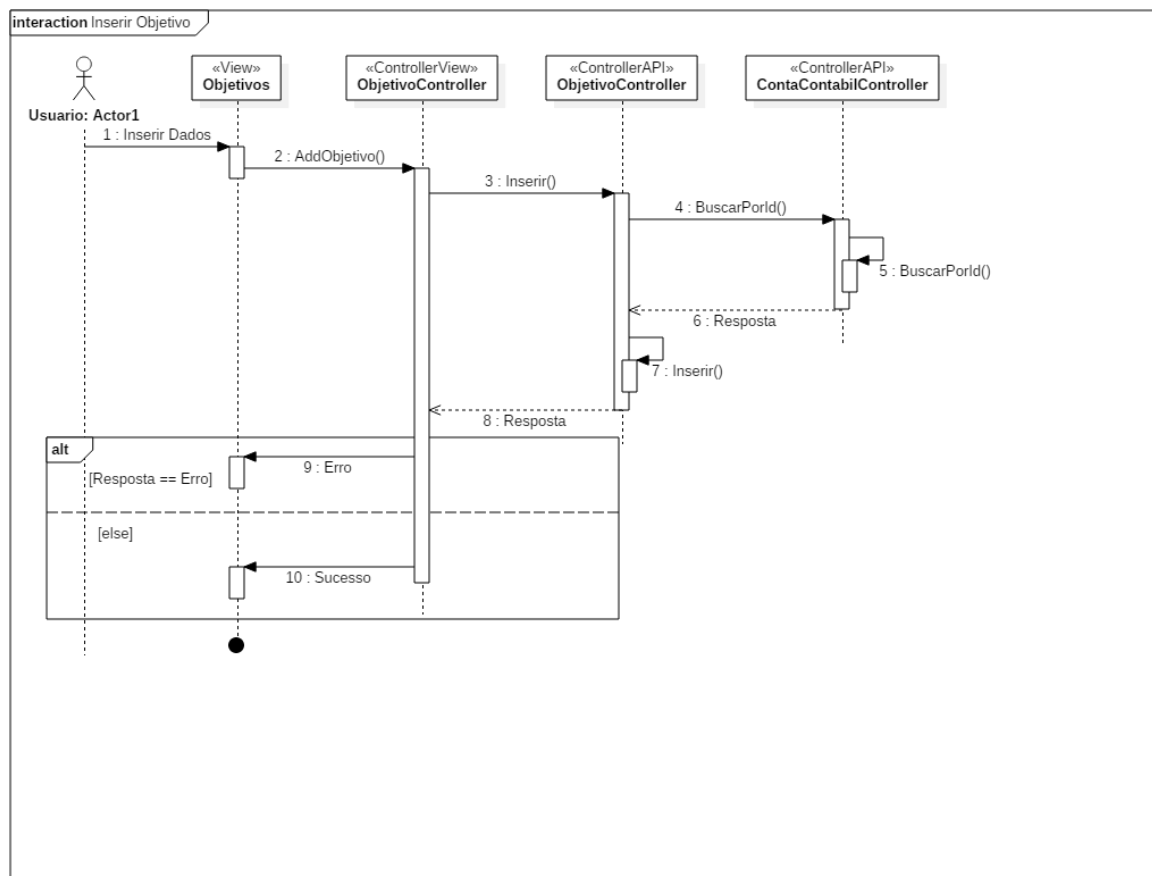
Nome:	CarregarDados
Descrição:	Faz o carregamento de todos os dados necessários para a realização de outras operações.
Pré-condições:	
Nenhuma.	
Especificação (diagrama de atividade ou pseudocódigo):	
1. Obtém da aplicação servidor os dados atuais. 2. Exibe os dados nos controles de tela.	
Pós-condições:	
1. Os dados são exibidos na tela.	

4.2. Realizações de casos de uso

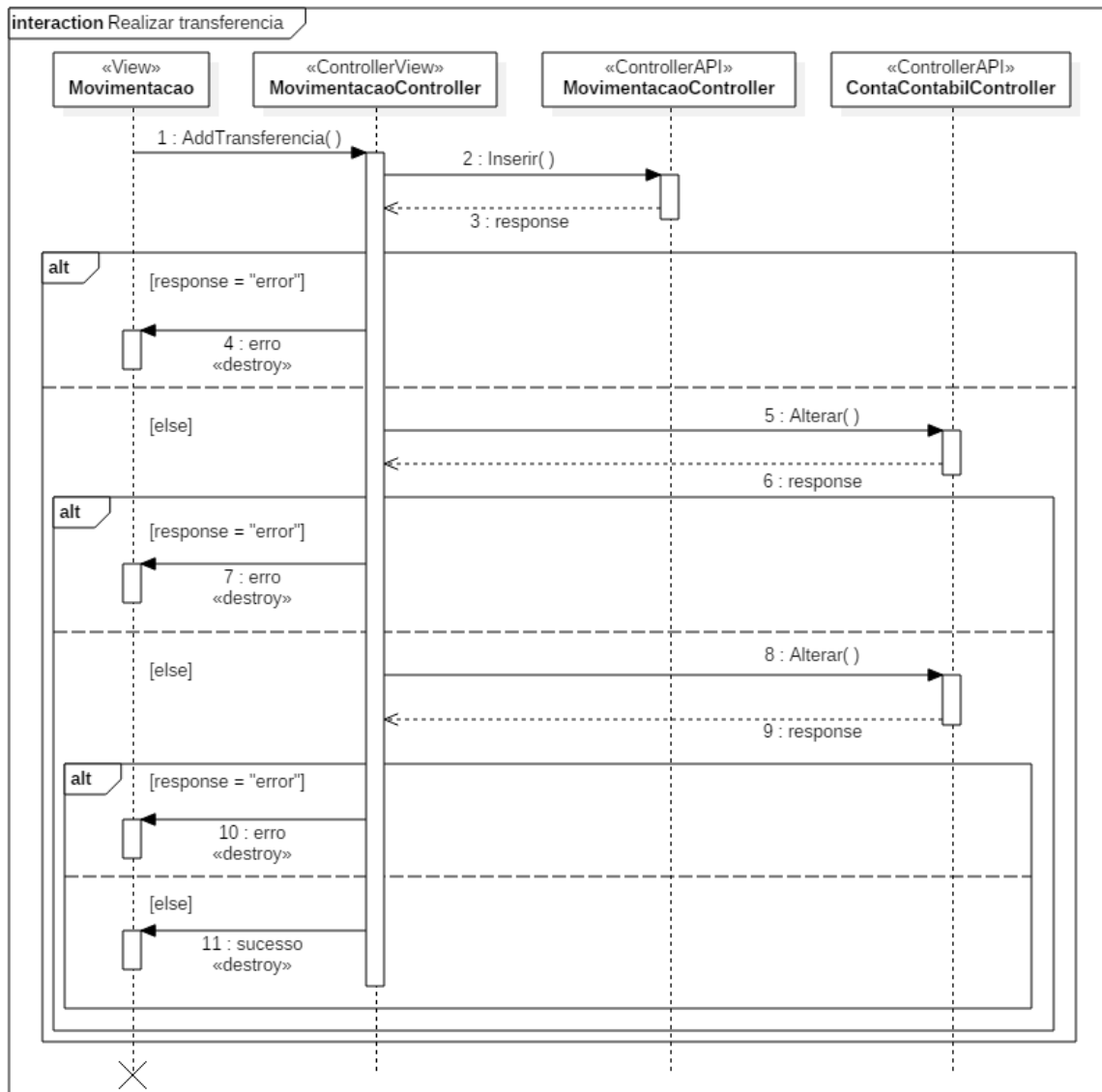
4.2.1. Cenário Inserir Conta



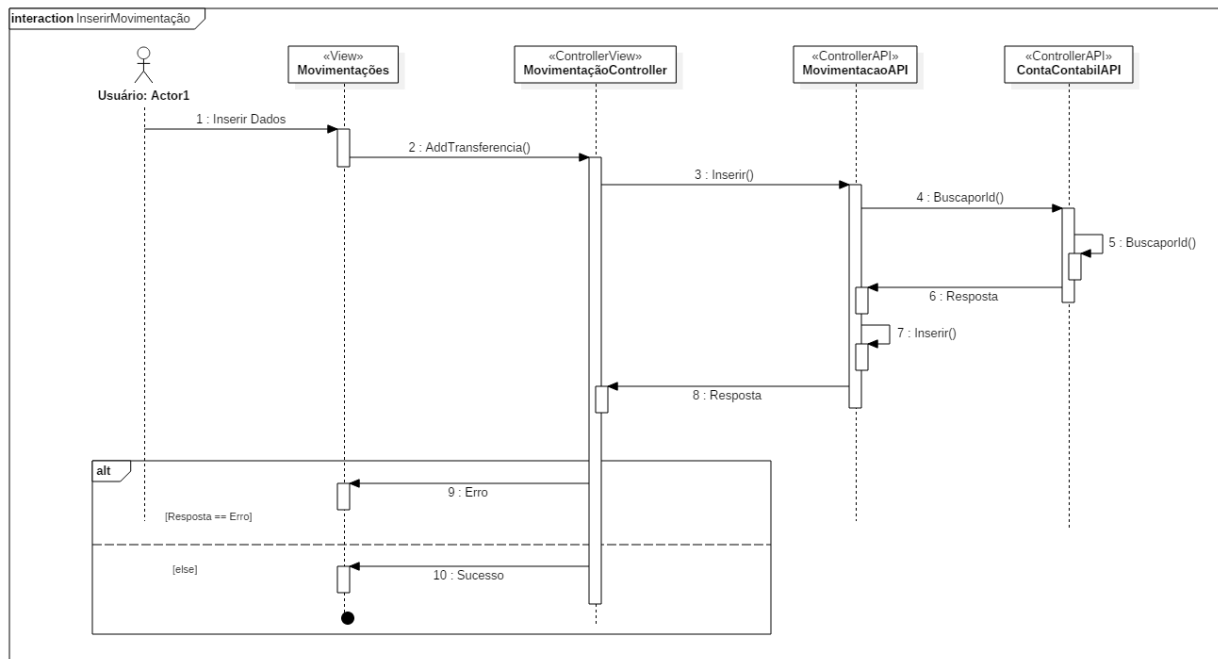
4.2.2. Cenário Inserir Objetivo



4.2.3. Cenário Transferência



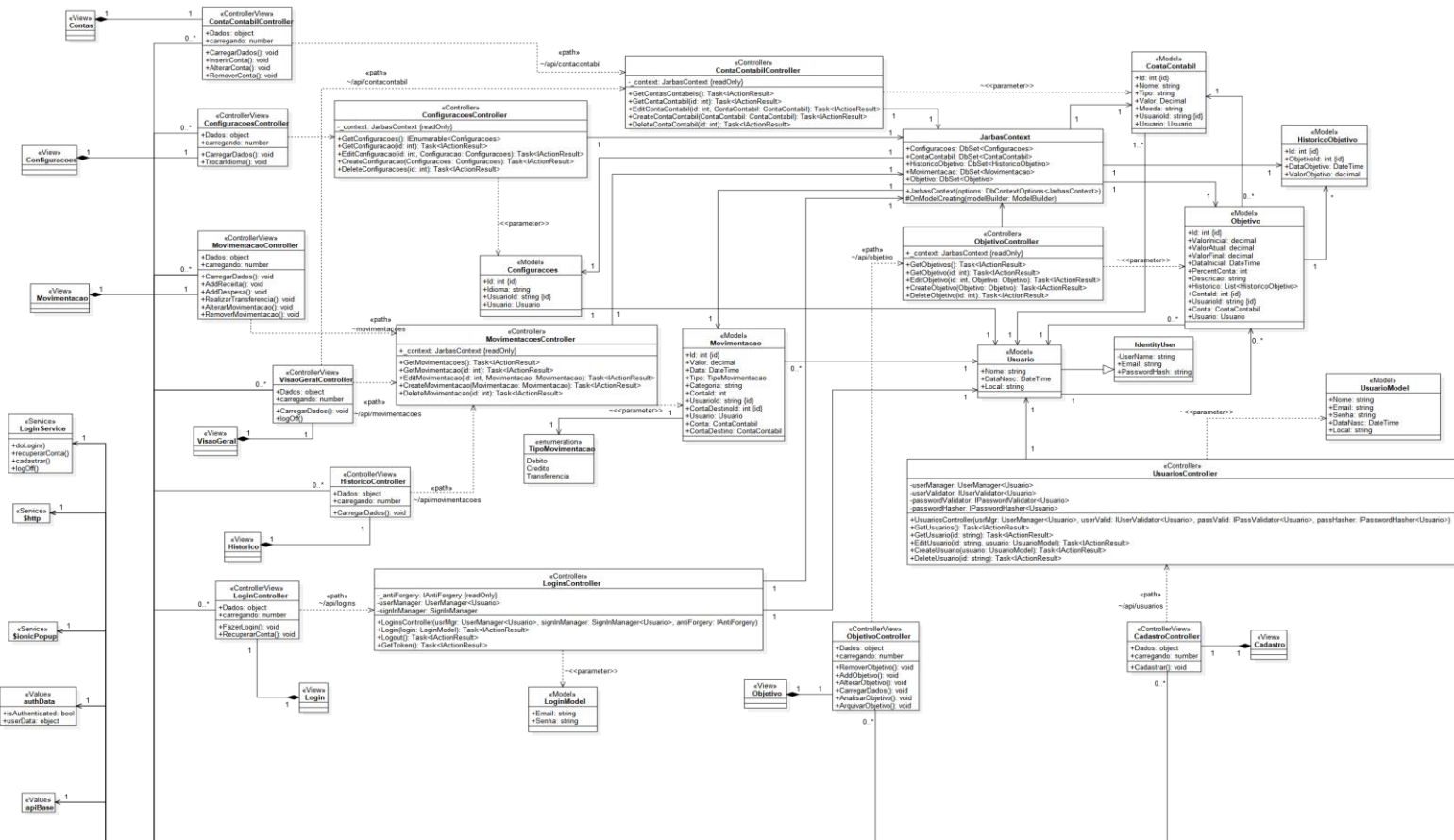
4.2.4. Cenário Inserir Movimentação



5. Modelos de Projeto de Sistema

5.1. Classes de Projeto

5.2.1. Diagrama de Classes



5.2.1. Comparativo entre frameworks/bibliotecas

AngularJS:

AngularJS é, segundo sua documentação disponível online, um framework estrutural open-source para aplicações web dinâmicas, desenvolvido em JavaScript e mantido pela Google, que auxilia na execução de single-page applications. Foi construído sobre o padrão model-view-view-model. Escolhemos AngularJS ao invés de por exemplo, ReactJS, por ser um framework mais maduro, com uma grande comunidade que contribui para sua melhora, e por ter compatibilidade com qualquer browser ou plataforma.

Link: <https://angularjs.org/>

Ionic:

Ionic é, segundo sua documentação disponível online, um kit de desenvolvimento de software (SDK) para a criação de aplicações híbridas e progressivas para dispositivos móveis. Desenvolvido com base no AngularJS e Apache Cordova, provê ferramentas

e serviços para o desenvolvimento de aplicações com o uso de tecnologias Web, como CSS, HTML5 e JavaScript. Escolhemos o Ionic ao invés de seus concorrentes, como Phonegap ou Xamarim, por ele ser gratuito, baseado em AngularJS e oferecer um grande leque de possibilidades no que diz a aparência.

Link: <https://ionicframework.com/>

ASP.NET Core:

ASP.NET Core é, segundo sua documentação disponível online, uma nova framework de código aberto e multi plataforma para construir aplicações modernas, baseadas na nuvem, como aplicativos web, aplicativos de internet das coisas, e API's. Foi desenvolvido para ser um framework otimizado para aplicativos que rodam na nuvem. Consiste em componentes modulares, que mantêm as aplicações flexíveis enquanto se construí uma solução. Escolhemos ASP.NET Core, ao invés de concorrentes como por exemplo, NodeJS, por ser uma framework gratuita, estar entre as frameworks mais rápidas do mercado e contar com ferramentas como o Entity Framework Core, que será citado abaixo.

Link: <https://www.microsoft.com/net/download/core>

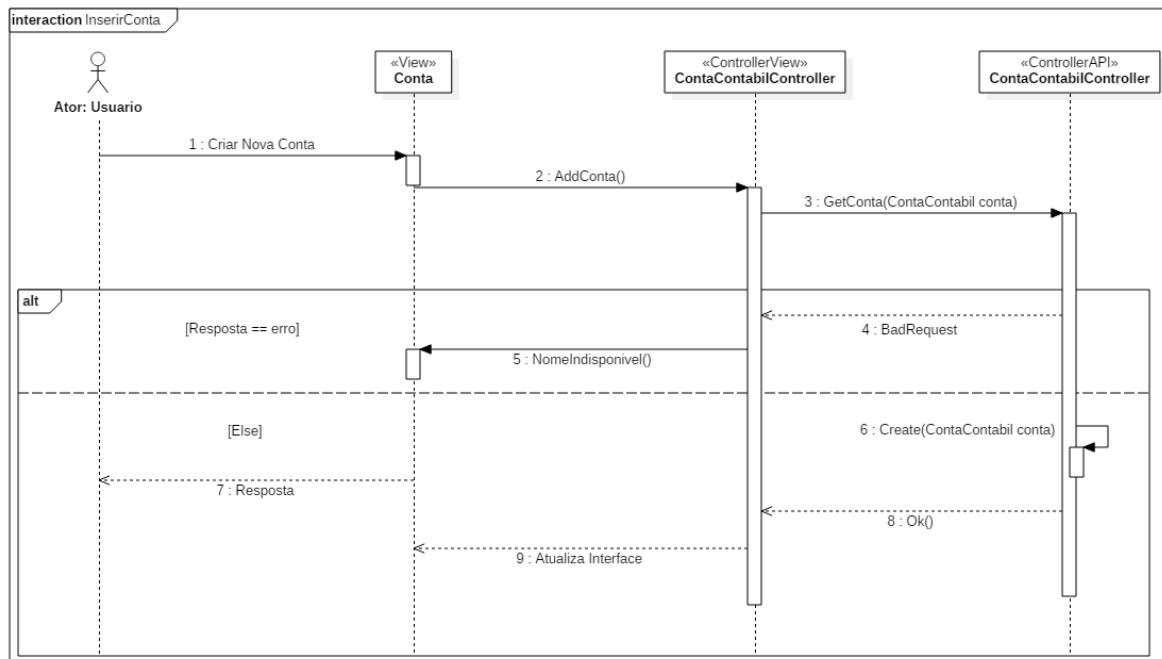
Entity Framework Core:

Entity Framework Core é uma versão mais leve e multi plataforma da Entity Framework. EF Core é um mapeador de objeto-relacional (O/RM) que permite que desenvolvedores .Net Core trabalhem com o banco de dados usando objetos .Net. Isso elimina a necessidade de escrever código para acesso de dados. Pode ser adquirido via Nuget. Escolhemos essa framework ao invés de utilizar o acesso direto ao banco de dados por, permitir um desenvolvimento extremamente rápido, ao ser necessário criar apenas as classes que seriam persistentes, já que a ferramenta, utilizando essas classes, cria toda a estrutura do banco de dados, e também permite a utilização dos dados de forma igualmente fácil.

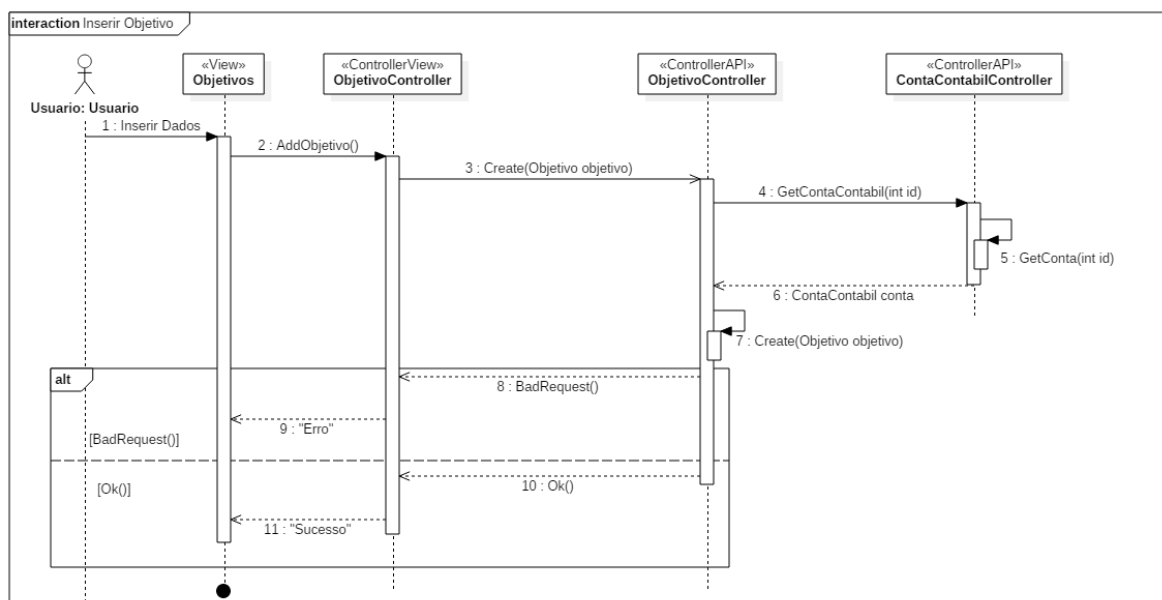
Link: <https://docs.microsoft.com/en-us/ef/core/>

5.2. Realizações de Casos de Uso – Projeto

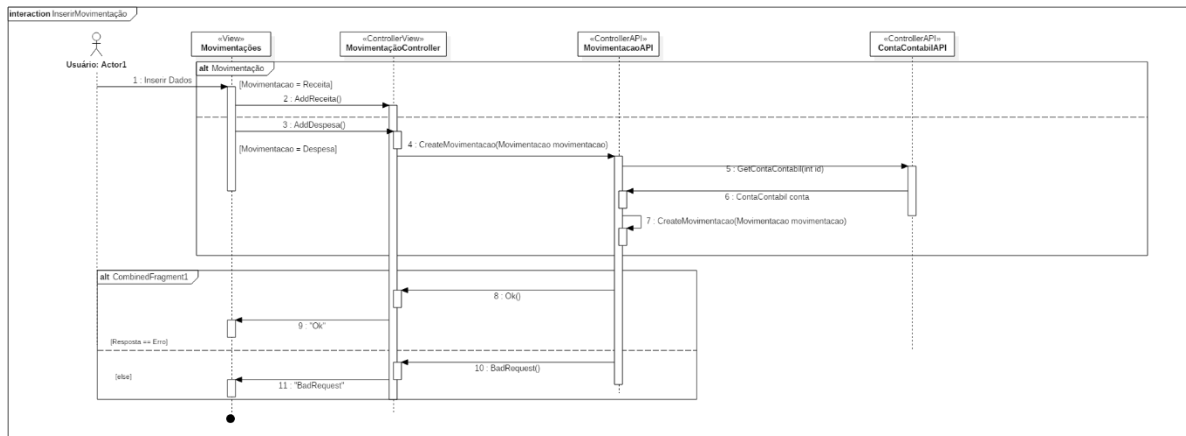
5.2.1. Diagrama de Sequência - Inserir Conta



5.2.2. Diagrama de Sequência – Inserir Objetivo



5.2.3. Diagrama de Sequência – Inserir Movimentação



5.2.4. Diagrama de Sequência – Realizar Transferência

