



FACULDADE DE TECNOLOGIA DE AMERICANA
Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas

Amanda Carolina Zen
Bryan de Oliveira Germino
Fernanda da Silva dos Santos
Guilherme Augusto Dias Correa de Oliveira

**APLICATIVO PARA GERENCIAMENTO DE CRÉDITOS PARA O USO DO
TRANSPORTE PÚBLICO**

Americana, SP
2018



FACULDADE DE TECNOLOGIA DE AMERICANA
Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas

Amanda Carolina Zen
Bryan de Oliveira Germino
Fernanda da Silva dos Santos
Guilherme Augusto Dias Correa de Oliveira

**APLICATIVO PARA GERENCIAMENTO DE CRÉDITOS PARA O USO DO
TRANSPORTE PÚBLICO**

Relatório técnico desenvolvido em
cumprimento à exigência curricular do Curso
Superior de Tecnologia em Análise e
Desenvolvimento de Sistemas do Prof. Dr.
Kleber de Oliveira Andrade

Americana, SP
2018

FICHA CATALOGRÁFICA – Biblioteca Fatec Americana - CEETEPS
Dados Internacionais de Catalogação-na-fonte

Z53a ZEN, Amanda Carolina

Aplicativo para gerenciamento de créditos para o uso de transporte público. / Amanda Carolina Zen; Bryan de Oliveira Germino; Fernanda da Silva dos Santos; Guilherme Augusto Dias Correa de Oliveira.. – Americana, 2018.

67f.

Monografia (Curso de Tecnologia em Análise e Desenvolvimento de Sistemas) - - Faculdade de Tecnologia de Americana – Centro Estadual de Educação Tecnológica Paula Souza

Orientador: Prof. Dr. Kleber de Oliveira Andrade

1 Desenvolvimento de software 2. Android – aplicativos. I. ANDRADE, Kleber de Oliveira II. GERMINO, Bryan de Oliveira III. SANTOS, Fernanda da Silva IV. OLIVEIRA, Guilherme Augusto Dias Correa de V. Centro Estadual de Educação Tecnológica Paula Souza – Faculdade de Tecnologia de Americana

CDU:

Faculdade de Tecnologia de Americana

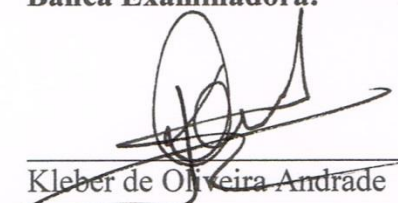
Amanda Carolina Zen
Bryan de Oliveira Germino
Fernanda da Silva dos Santos
Guilherme Augusto Dias Correa de Oliveira

Aplicativo para gerenciamento de créditos para o uso de transporte público

Trabalho de graduação apresentado como exigência parcial para obtenção do título de Tecnólogo em Análise e Desenvolvimento de Sistemas pelo Centro Paula Souza – FATEC Faculdade de Tecnologia de Americana. Área de concentração: Desenvolvimento

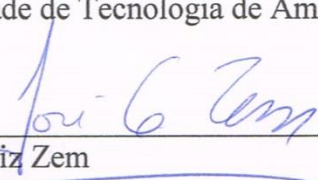
Americana, 28 de Junho de 2018.

Banca Examinadora:


Kleber de Oliveira Andrade

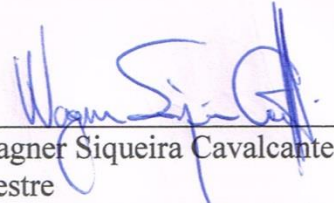
Doutor

Faculdade de Tecnologia de Americana, Fatec, Brasil.


José Luiz Zem

Doutor

Faculdade de Tecnologia de Americana, Fatec, Brasil.


Wagner Siqueira Cavalcante

Mestre

Faculdade de Tecnologia de Americana, Fatec, Brasil.

RESUMO

Este trabalho foi produzido com a finalidade de desenvolver um aplicativo *mobile*, que servirá como cartão de passe para o transporte público. Todas as fases do projeto serão relatadas, passando pela concepção da ideia, os motivos da escolha do *Scrum* como metodologia, os *Sprints* realizados ao longo do projeto, a definição dos requisitos mínimos que agregam mais valor ao projeto, diagramas da UML, criação das tabelas do banco de dados do aplicativo, descrição de toda infraestrutura necessária para o desenvolvimento e implantação do aplicativo, procedimentos e análise dos testes realizados, e por fim uma descrição dos resultados alcançados ao final do projeto juntamente com sugestões de melhorias que agregariam mais valor ao produto, porém devido ao tempo limitado para desenvolvimento ficaram como planos futuros.

Palavras Chave: Aplicativo; desenvolvimento; transporte; *mobile*;

ABSTRACT

This work was produced with the purpose of developing a mobile application, which will serve as a pass card for public transport. All the phases of the project will be reported, passing through the idea design, the reasons for choosing Scrum as methodology, the sprints made throughout the project, the definition of the minimum requirements that add more value to the project, UML diagrams, the creation of the app's database, the description of all the structure need to development and implementation of the app, procedures and analysis of the tests performed, and finally, a description from all the results reached by the end of the project, along with improvements that would add more value to the product, but, due to limited development time they remained as future plans.

Keywords: App; development; transport; mobile;

SUMÁRIO

1	INTRODUÇÃO	10
2	PROJETO DO APLICATIVO	12
2.1	Levantamento de Requisitos	12
2.1.1	Requisitos Funcionais.....	12
2.1.2	Requisitos Não Funcionais	13
2.2	Recursos e Ferramentas	14
2.3	Modelagem	15
2.3.1	Casos De Uso.....	15
2.3.2	Documentação dos Casos de Uso	17
2.3.3	Diagrama de Classe	21
2.3.4	Dicionário de Dados	22
2.3.5	Diagrama de Sequência	25
2.3.6	Diagrama de Atividades.....	31
2.3.7	Diagrama Entidade Relacionamento	35
3	PROTOTIPAGEM	36
3.1	Tela de <i>login</i>	36
3.2	Tela criar conta	37
3.3	Tela recuperar senha.....	38
3.4	Tela <i>home</i>	39

3.5 Tela menu navegação	40
3.6 Tela comprar	41
3.7 Tela histórico de compra	42
3.8 Tela histórico de utilização	43
3.9 Tela configuração	44
3.10 Tela alterar senha	45
3.11 Tela alterar informações do usuário	46
4 METODOLOGIA.....	48
4.1 Utilizando <i>Scrum</i>	51
4.1.1 <i>Sprint</i> um	51
4.1.2 <i>Sprint</i> dois	53
4.1.3 <i>Sprint</i> três	54
4.1.4 <i>Sprint</i> quatro	56
4.1.5 <i>Sprint</i> cinco	58
5 IMPLEMENTAÇÃO	60
5.1 Plano de testes	60
5.2 Resultados dos testes.....	61
6 CONSIDERAÇÕES FINAIS.....	64
REFERÊNCIAS.....	65
APÊNDICE A – Script usado na criação das tabelas do banco de dados.....	66
APÊNDICE B – Simulador catraca.....	68

LISTA DE FIGURAS

Figura 1 – Diagrama de caso de uso.	16
Figura 2 – Diagrama de classe.....	21
Figura 3 - Diagrama de sequência / Cadastro de cliente	26
Figura 4 - Diagrama de sequência / Controle de crédito	27
Figura 5 - Diagrama de sequência / <i>Login</i>	28
Figura 6 - Diagrama de sequência / Menu de opções.....	29
Figura 7 - Diagrama de sequência / Menu de opções (continuação)	30
Figura 8 - Diagrama de Atividades / Alterar cadastro.....	31
Figura 9 - Diagrama de Atividades / Menu	32
Figura 10 - Diagrama de Atividades / Criar conta.....	33
Figura 11 - Diagrama de Atividades / Controle de crédito.....	34
Figura 12 - Diagrama Entidade Relacionamento.....	35
Figura 13 - Tela de <i>login</i>	37
Figura 14 - Tela criar conta	38
Figura 15 - Tela recuperar senha	39
Figura 16 - Tela home	40
Figura 17 - Tela menu navegação.....	41
Figura 18 - Tela comprar.....	42
Figura 19 - Tela histórico de compra	43
Figura 20 - Tela histórico de utilização.....	44
Figura 21 - Tela configuração.....	45
Figura 22 - Tela alterar senha	46

Figura 23 - Tela alterar informações do usuário.....	47
Figura 24 - Gráfico <i>Burndown Sprint</i> um	52
Figura 25 - Gráfico <i>Burndown Sprint</i> dois	54
Figura 26 - Gráfico <i>Burndown Sprint</i> três	55
Figura 27 - Gráfico de <i>Burndown Sprint</i> quatro	57
Figura 28 - Gráfico de <i>Burndown Sprint</i> cinco.....	59
Figura 29 - Tela simulador catraca.....	69

LISTA DE TABELAS

Tabela 1 – Requisitos funcionais do projeto.....	13
Tabela 2 – Requisitos não funcionais do projeto.....	14
Tabela 3 - Caso de uso “Comprar crédito”.....	17
Tabela 4 - Caso de uso “Inserir cliente”.....	18
Tabela 5 - Caso de uso “Realizar <i>login</i> ”.....	18
Tabela 6 - Caso de uso "Visualizar histórico de compras".....	19
Tabela 7 - Caso de uso "Verificar Saldo".....	20
Tabela 8 - Caso de uso "Alterar Cadastro".....	20
Tabela 9 - Métodos das entidades.....	22
Tabela 10 - Descrição das Tabelas.....	23
Tabela 11 - Descrição do atributos.....	23
Tabela 12 - Descrição dos Relacionamentos.....	24
Tabela 13 - Demonstração <i>Sprint</i> um.....	51
Tabela 14 - Retrospectiva <i>Sprint</i> um.....	52
Tabela 15 - Demonstração <i>Sprint</i> dois.....	53
Tabela 16 – Retrospectiva <i>Sprint</i> dois.....	54
Tabela 17 - Demonstração <i>Sprint</i> três.....	55
Tabela 18 - Retrospectiva <i>Sprint</i> três.....	56
Tabela 19 - Demonstração <i>Sprint</i> quatro.....	57
Tabela 20 – Retrospectiva <i>Sprint</i> quatro.....	58
Tabela 21 – Demonstração <i>Sprint</i> cinco.....	58
Tabela 22 - Retrospectiva <i>Sprint</i> cinco.....	59
Tabela 23 - Plano de testes.....	60

1 INTRODUÇÃO

Nas últimas décadas muito se tem falado sobre a importância da tecnologia para a sociedade e para o meio ambiente, como usar a tecnologia para atender as necessidades de uma sociedade, reduzir o uso de matéria prima de forma a preservar o meio ambiente para as próximas gerações.

Por meio deste questionamento, foi projetado um sistema móvel, a fim de substituir os passes utilizados em transportes públicos. Este aplicativo proposto realizará funções de compra de passe e utilização por meio de *QR code*.

O aplicativo que será desenvolvido neste trabalho, tem como parceria a Associação das Empresas de Transporte Coletivo Urbano de Campinas (Transurc), responsável pela venda de passes para o transporte público do município, foi com base na forma de venda de tickets utilizada por ela que definiu-se quais informações eram essenciais para o aplicativo, contudo, embora tenha sido utilizado o cenário do transporte público de Campinas para definir as principais funcionalidades do aplicativo, ele pode ser implantado em qualquer cidade, pois a forma de venda e compra de passes não divergem muito de uma cidade para outra.

A Transurc foi fundada em 25 de agosto de 1987, possui 30 anos de atuação. Desde julho de 2003 a Transurc possui uma sede na Rua 11 de Agosto, nº 757, no Centro de Campinas, com capacidade para atender, tanto empresas quanto usuários de vale transporte, passe escolar, idoso e universitário. Hoje, a associação possui 325 pontos de recarga para os cartões eletrônicos, entre postos próprios e estabelecimentos credenciados espalhados por Campinas.

Desde o dia 16 de setembro de 2017 a Transurc implantou a validação por código de barras do tipo *QR code* em algumas linhas de Campinas. Ao todo, foram 1.250 veículos equipados com o leitor para código de barras 2D. O objetivo dessa nova validação é acabar com o pagamento de passagens com dinheiro. Estima-se que 10% dos usuários ainda utilizem essa forma de pagamento, o que faz com que os ônibus sejam frequentemente vítimas de assaltos. Entretanto, o ticket com *QR code* só está sendo disponibilizado em papel, o que oferece um grande impacto ao meio ambiente, além de gerar um desconforto para os usuários, pois a compra dessa passagem só pode ser feita nos pontos de recarga da empresa.

Este trabalho tem como objetivo principal desenvolver um aplicativo Android que permite que qualquer pessoa com um *smartphone* possa se cadastrar e fazer do

seu aparelho um cartão de passe único comum. Para alcançar esse objetivo, foram definidas algumas metas específicas, tais como:

- Estudos sobre QR *code* e suas formas de segurança;
- Entendimento das regras de negócio utilizadas pela empresa Transurc;
- Estudo e aplicação da metodologia *Scrum* para o desenvolvimento do aplicativo;
- Aperfeiçoamento dos conhecimentos sobre Engenharia de Software e desenvolvimento de aplicativos para dispositivos móveis.

No capítulo 2 serão apresentadas informações sobre todas as etapas de desenvolvimento do aplicativo, passando por o levantamento de requisitos, recursos e ferramentas utilizadas até a modelagem dos diagramas.

No capítulo 3 serão apresentadas informações sobre a prototipagem do aplicativo, com imagens de todas as telas do aplicativo, mostrando a versão inicial e em seguida a versão final.

No capítulo 4 será apresentada a metodologia utilizada pela equipe, que é o *Scrum*.

No capítulo 5 serão apresentadas informações sobre o plano de testes e também sobre os resultados desses testes.

E por último, teremos as considerações finais.

2 PROJETO DO APLICATIVO

Este capítulo apresenta todas as etapas de desenvolvimento do aplicativo, iniciando nos diagramas criados com base nos principais requisitos levantados, passando pela modelagem do banco de dados e encerrando na prototipagem do aplicativo.

2.1 Levantamento de Requisitos

A engenharia de requisitos (RE – *Requirements Engineering*) é o processo de descobrir, analisar, documentar e verificar requisitos de um sistema. Um requisito pode ser definido como uma descrição dos serviços fornecidos pelo sistema e as suas restrições operacionais (SOMMERVILLE, 2007). Tradicionalmente, os requisitos são divididos em dois tipos: requisitos funcionais e requisitos não funcionais.

2.1.1 Requisitos Funcionais

Os requisitos funcionais descrevem o que o sistema deve fazer, isto é, definem a funcionalidade desejada do software (SOMMERVILLE, 2007). A Tabela 1 apresenta os requisitos funcionais deste projeto.

Tabela 1 – Requisitos funcionais do projeto.

Identificação	Requisito Funcional
RF001	Um dos requisitos funcionais é o cadastro de usuários, com o qual pode-se realizar o pagamento e o controle de crédito adquiridos. Ao realizar o cadastro o usuário vai usar seu CPF como <i>login</i> e vai ter uma senha própria para acessar o aplicativo. Em caso de perda do cartão o usuário pode recuperar seus dados e os seus créditos.
RF002	<i>Login</i> é um dos passes de segurança, que, além de preservar os dados do usuário em caso de perda, também oferecerá a opção de criar um novo cartão, caso o usuário não tenha. Esse requisito é dependente do RF001.
RF003	O usuário pode inserir créditos para usar o cartão, o pagamento é feito de através de um sistema de <i>e-commerce</i> de terceiros, mas, apesar disso, é importante verificar se a compra está sendo reconhecida e realizada de maneira apropriada, pois é através delas que a aprovação do saldo do cartão irá ocorrer. Uma vez que a compra não seja efetivada não haverá saldo para ser usado.
RF004	Além dos gastos de créditos, será possível ver quais ônibus foram tomados. O sistema de catraca do ônibus já tem esse controle interno, porém a intenção é passar esse histórico para o aplicativo. Esse requisito é dependente do RF003.
RF005	Verificar o total de saldo no cartão é um requisito funcional, pois ele é importante para o usuário ter controle de quanto pode gastar e quanto de crédito ainda lhe resta para a próxima viagem. Esse requisito é dependente do RF002.
RF006	Algumas alterações menores podem ser feitas nos dados dos clientes. Esse requisito está diretamente relacionado com o requisito RF001.
RF007	A leitura dos passes nas catracas dos ônibus será realizada por <i>QR code</i> . O aplicativo que gera um <i>QR code</i> por cartão.
RF008	Com o aplicativo aberto, o usuário passa seu celular na catraca, que lê o <i>QR code</i> e reconhece se esse cartão tem saldo, então a catraca autoriza a entrada do passageiro. Esse requisito é dependente do RF007

Fonte: Elaborado pelo autor (2018)

2.1.2 Requisitos Não Funcionais

“Os requisitos não funcionais são aqueles não diretamente relacionados às funções específicas fornecidas pelo sistema” (SOMMERVILLE, 2007). A Tabela 2 apresenta os requisitos não funcionais deste projeto.

Tabela 2 – Requisitos não funcionais do projeto.

Identificação	Requisito não funcional	Categoria
RNF001	Cada cartão é pessoal e intransferível, portanto é importante ter segurança quanto a isso e também segurança sobre os dados de pagamento e do próprio usuário.	Segurança
RNF002	Como o usuário irá utilizar o celular como um cartão, o programa deve ser ágil em disponibilizar o QR para passar na catraca, por isso um aplicativo leve e rápido será de grande importância para o usuário.	Desempenho
RNF003	Não são todas as pessoas que estão habituadas a usar o celular para pagamentos e/ou cartão, para auxiliar o usuário, o sistema deve ser intuitivo.	Usabilidade

Fonte: Elaborado pelo autor (2018)

2.2 Recursos e Ferramentas

Esta seção contempla as ferramentas de programação e os conceitos necessários para o desenvolvimento do projeto.

A seguir são apresentados os aplicativos utilizados para o desenvolvimento deste trabalho até o momento.

- **Microsoft Word 2013:** Este ambiente de desenvolvimento foi utilizado para construção da escrita do trabalho dentro das normas ABNT;
- **Astah Community:** Ambiente utilizado para criação dos diagramas de caso de uso, classe, atividades e sequência;
- **ERDPlus:** Programa utilizado para a criação do DER para auxiliar no desenvolvimento do banco de dados;
- **MySQL:** Ambiente que utilizado para a criação do banco de dados do aplicativo;
- **Android Studio:** Plataforma de desenvolvimento utilizada para a criação do aplicativo;
- **Atom:** Editor de texto, utilizado para criar e editar o *Web Service*;

- **000webhost:** utilizado para a hospedagem *do Web Service*;
- **In-app Billing API:** Biblioteca que facilita a integração do faturamento no aplicativo ;
- **Volley:** Biblioteca que facilita a interação entre o aplicativo e o *Web Service*;
- **Zxing:** Biblioteca que possibilita a criação e leitura do *QR code*.

2.3 Modelagem

Na fase da modelagem, é feita a documentação do aplicativo, que se tratam de diagramas que facilitam na compreensão do projeto de forma padronizada.

A documentação deste trabalho utilizará a linguagem de modelagem *Unified Modeling Language*¹ (UML) para modelar os diagramas de casos de uso, o de classe, o de sequência e o de atividade.

2.3.1 Casos De Uso

Os diagramas de caso de uso descrevem um cenário de funcionalidades do ponto de vista do usuário, catalogando os requisitos funcionais do sistema. Dentro do diagrama são retratados os atores (representado pelos bonecos), as funcionalidades (representadas pelos balões com a ação escrita por dentro) e as relações (representadas pelas linhas).

Os atores que interagem com o sistema são: o Usuário, API de compras e a Companhia de ônibus. O sistema é um caso de uso explícito e se trata do sistema em si em que os casos de uso acontecem.

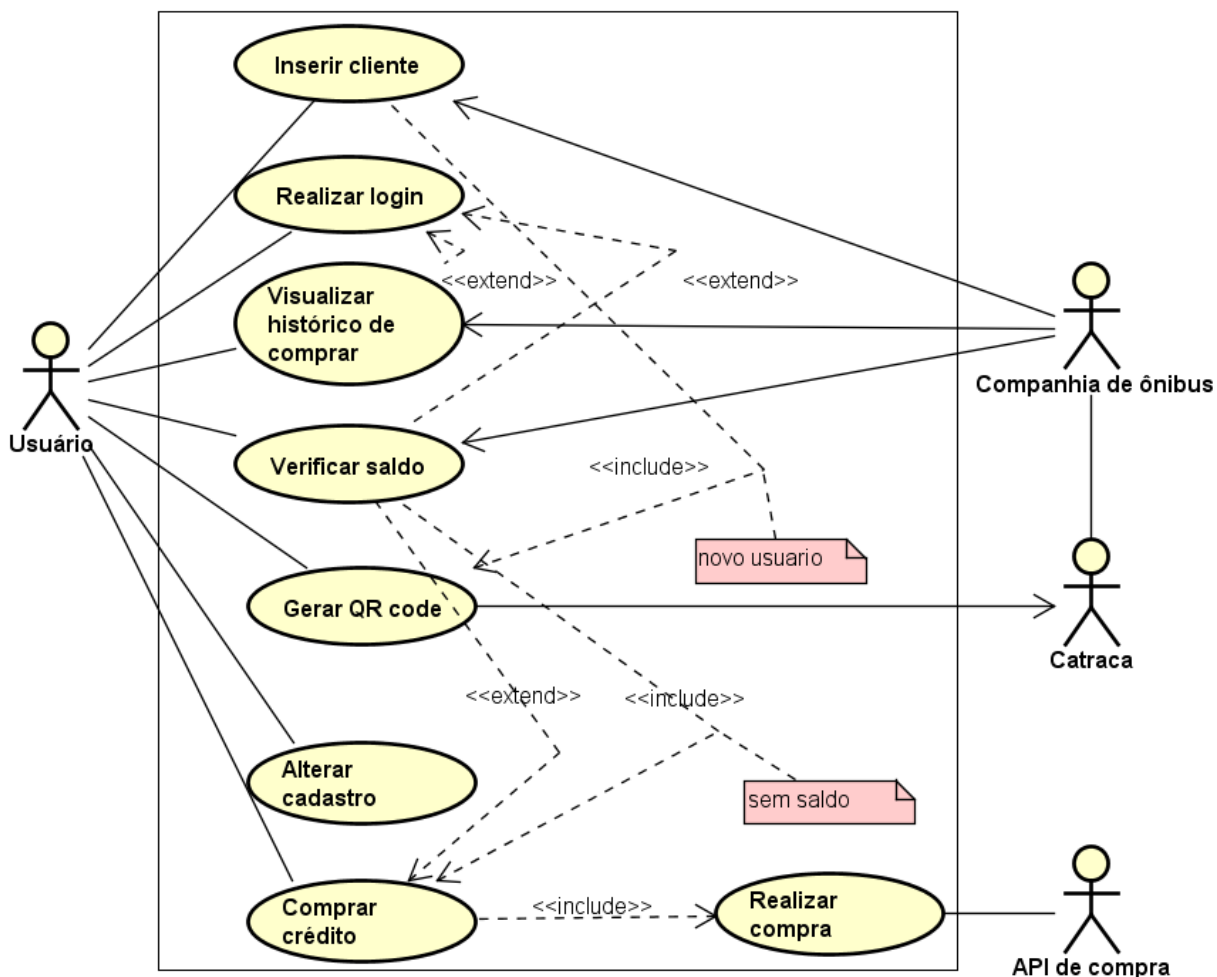
- **Usuário:** É o ator que representa os utilizadores deste aplicativo. Um ator pode, por exemplo, se cadastrar, fazer *login* e comprar créditos;

¹ *Unified Modeling Language* ou Linguagem Unificada de Modelagem (UML) é uma linguagem padrão para modelagem e documentar os sistemas orientados a objetos.

- **API de compra:** Representa o ator da API que permite a interação entre o aplicativo e uma API de e-commerce para efetuar as compras de créditos do aplicativo;
- **Companhia de ônibus:** Representa a empresa de ônibus de Campinas, a responsável pela venda de passes para o transporte público do município.

A Figura 1 apresenta o caso de uso desde a entrada do usuário no sistema, até sua compra de créditos.

Figura 1 – Diagrama de caso de uso.



Fonte: Elaborado pelo autor (2018)

No subcapítulo 2.3.2 será apresentada a documentação dos casos de uso do projeto deste trabalho.

2.3.2 Documentação dos Casos de Uso

Cada funcionalidade dos diagramas de casos de uso será descrita, da Tabela 3 à Tabela 8.

O caso de uso “Comprar crédito” utiliza a API de *e-commerce* para realizar a compra. Após o usuário selecionar o quanto ele deseja comprar e colocar seus dados, a API verifica se são válidos (CPF, nº do cartão e etc.) e realiza a compra. A Tabela 3 detalha o passo a passo da execução deste caso de uso.

Tabela 3 - Caso de uso “Comprar crédito”.

Nome do caso de uso	Comprar crédito.
Atores envolvidos	Usuário, API de compras, Sistema.
Objetivo	Este caso de uso descreve os passos de compra de crédito.
Prioridade de desenvolvimento	Alto.
Ações do ator	Ações do Sistema
1. O usuário clica em “comprar” e digita seus dados.	
	2. O sistema leva para a API de <i>e-commerce</i> , que se encarrega do processo de verificação de dados e aprova a transação.
	3. Ao aprovar a transação, os créditos são inseridos no aplicativo.
	4. Após a tela de “transação aprovada”, é possível visualizar o total adicionado à conta e o aplicativo é direcionado à tela de menu.
Validações	Para que a compra seja autorizada, o usuário deve digitar informações válidas de compra.

Fonte: Elaborado pelo autor (2018).

O caso de uso “Inserir cliente” é o onde o usuário realiza o cadastro para poder acessar o aplicativo. Ao preencher os campos, o aplicativo vai verificar se os

dados são verdadeiros e então realizar o cadastro. A Tabela 4 apresenta o passo a passo da execução deste caso de uso.

Tabela 4 - Caso de uso “Inserir cliente”.

Nome do caso de uso	Inserir cliente.
Atores envolvidos	Usuário, Companhia de ônibus, Sistema.
Objetivo	Este caso de uso descreve os passos para realizar o cadastro do usuário.
Prioridade de desenvolvimento	Alto.
Ações do ator	Ações do Sistema
1. O usuário digita os dados requeridos e clica em “inserir cliente”.	
	2. O sistema envia as informações para o <i>Web Service</i> .
	3. O sistema recebe a resposta e mostra o resultado ao usuário.
Validações	As informações de cadastro devem ser validadas (por exemplo, o campo CPF).

Fonte: Elaborado pelo autor (2018).

O caso de uso “Realizar *login*” é a função de *login* do aplicativo. O usuário deve estar conectado para ter acesso às outras funções. O aplicativo confirma se a senha e o *login* digitados são verdadeiros e então autoriza o acesso. A Tabela 5 apresenta o passo a passo da execução deste caso de uso.

Tabela 5 - Caso de uso “Realizar *login*”.

Nome do caso de uso	Realizar <i>login</i> .
Atores envolvidos	Usuário, Sistema.
Objetivo	Este caso de uso descreve os passos para realizar o <i>login</i> .
Prioridade de desenvolvimento	Alto.

Ações do ator	Ações do Sistema
1. O usuário digita o CPF e senha e clica em logar.	
	2. O sistema envia as informações para o <i>Web Service</i> .
	3. O sistema recebe a resposta e decide se deve ser mostrar a mensagem para o usuário ou estar conectado no sistema.
Validações	As informações devem ser validadas para realizar o <i>Login</i>

Fonte: Elaborado pelo autor (2018).

O caso de uso “Visualizar história de compras” permite que o usuário veja todas as suas compras de créditos realizadas até o momento. A Tabela 6 apresenta o passo a passo da execução deste caso de uso.

Tabela 6 - Caso de uso "Visualizar histórico de compras"

Nome do caso de uso	Visualizar história de compras.
Atores envolvidos	Usuário, Sistema.
Objetivo	Este caso de uso descreve os passos para o usuário visualizar o histórico de compras.
Prioridade de desenvolvimento	Médio.
Ações do ator	Ações do Sistema
1. O usuário requisita a visualização do histórico de compras.	
	2. O sistema envia as informações do cartão pra o <i>Web Service</i> .
	3. Após o retorno dos dados, o sistema mostra as informações ao usuário.

Fonte: Elaborado pelo autor (2018).

O caso de uso “Verificar Saldo” permite que o usuário consiga acessar a quantidade de saldo disponível na conta. O saldo é visível por todo o aplicativo,

sendo exibido na parte superior. A Tabela 7 apresenta o passo a passo da execução deste caso de uso.

Tabela 7 - Caso de uso "Verificar Saldo".

Nome do caso de uso	Verificar Saldo.
Atores envolvidos	Usuário, Sistema.
Objetivo	Este caso de uso descreve como é possível ver o saldo do usuário.
Prioridade de desenvolvimento	Alto.
Ações do ator	Ações do Sistema
1. O usuário requisita a atualização do saldo	
	2. O sistema envia as informações do cartão para o <i>Web Service</i> .
	3. Após o retorno dos dados, o saldo é demonstrado.

Fonte: Elaborado pelo autor (2018).

O caso de uso “Alterar cadastro” possibilita que o usuário altere informações do seu cadastro. Para salvar as alterações, sua senha é requisitada. O aplicativo valida as informações novamente, parecido com o caso de uso “Realizar cadastro”, mas o “Alterar cadastro” tem uma tela própria. A Tabela 8 apresenta o passo a passo da execução deste caso de uso.

Tabela 8 - Caso de uso "Alterar Cadastro".

Nome do caso de uso	Alterar cadastro.
Atores envolvidos	Usuário, Sistema.
Objetivo	Este caso de uso descreve os passos para alterar o cadastro do usuário.
Prioridade de desenvolvimento	Baixo.
Ações do ator	Ações do Sistema

1. O usuário digita as informações requisitadas e clica em Alterar Cadastro .	
	2. O sistema envia as informações para o <i>Web Service</i> .
	3. O sistema recebe a resposta e mostra o resultado ao usuário.
Validações	As informações de cadastro devem ser validadas e senha digitada também.

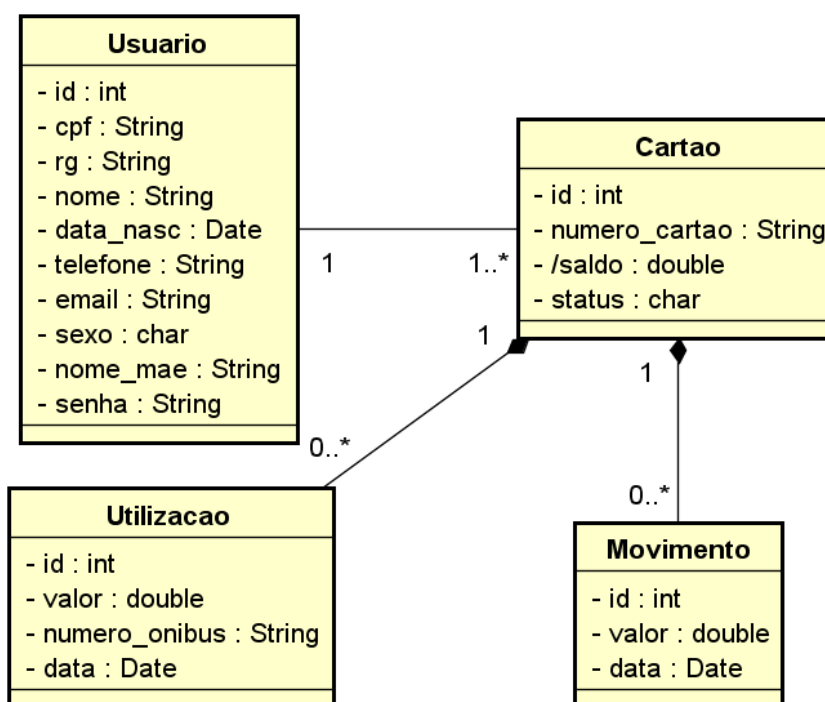
Fonte: Elaborado pelo autor (2018).

2.3.3 Diagrama de Classe

Diagrama de classe é um dos mais importantes. Contém as classes com seus atributos e métodos que irão compor o sistema, além de mostrar como elas se associam (BOOCH;RUMBAUGH;JACOBSON,2005).

Na Figura 2, é possível visualizar o diagrama de classe, com suas entidades e seus atributos.

Figura 2 – Diagrama de classe.



Fonte: Elaborado pelo autor (2018)

Na Tabela 9 é possível observar os métodos de cada entidade.

Tabela 9 - Métodos das entidades

Tabela	Método	Parâmetros	Descrição
Usuario	InsertUsuario	cpf, rg, nome, data, telefone, email, sexo, mae, senha	Inserir os dados do usuário.
	SelectUsuario	cpf, senha	Seleciona o usuário pelo CPF.
	AlterUsuarioSenha	id, senha	Altera a senha do usuário.
	AlterUsuario	id, nome, data, telefone, email, sexo, mae	Altera os dados do usuário.
	SelectUsuarioById	id	Seleciona o usuário pelo ID.
Cartao	InsertCartao	id	Inserir o cartão baseado no id do usuário.
	SelectCartao	id	Seleciona o cartão pelo id do mesmo.
	SelectCartaoByNumero	numero_cartao	Seleciona o cartão pelo número dele.
	AlterCartaoStatus	numero_cartao, status	Altera o status do cartão.
	AlterCartaoNumero	numero_cartao	Altera o número do cartão.
Movimento	InsertSaldo	numero_cartao, data, valor	Inserir saldo na data especificada.
	InsertSaldoNow	numero_cartao, valor	Inserir o saldo na data de agora.
	SelectSaldo	numero_cartao	Seleciona o saldo pelo número do cartão.
Utilizacao	InsertUtilizacao	numero_cartao, valor	Inserir a utilização (que é quando o cartão é passado na catraca).
	SelectUtilizacao	numero_cartao	Seleciona a utilização.

Fonte: Elaborado pelo autor (2018)

2.3.4 Dicionário de Dados

Dicionário de dados é um documento onde se encontram as informações sobre as tabelas, seu conteúdo, bem como os relacionamentos entre os seus

elementos. Na Tabela 10 é possível ver as descrições das tabelas do banco de dados.

Tabela 10 - Descrição das Tabelas

Tabela	Relacionamento	Nome do relacionamento	Descrição
Usuario	Cartao	Tem	Tabela para cadastro do usuário que possui o cartão de passe.
Cartao	Movimento	Possui	Tabela onde são guardadas as informações do cartão.
	Utilização	Realiza	
	Usuario	Tem	
Utilizacao	Cartao	Realiza	Tabela onde são guardadas as informações sobre a utilização, mais precisamente quando o cartão é passado na catraca.
Movimento	Cartao	Possui	Tabela onde as informações sobre o saldo do cartão são armazenadas.

Fonte: Elaborado pelo autor (2018)

Na Tabela 11 é possível ver as descrições dos atributos do banco de dados.

Tabela 11 - Descrição do atributos

Tabela	Nome da Coluna	Tipo de dados	Tamanho	Restrições	Descrição
Usuario	id	int	-	Primary Key, auto_increment	Número de identificação do usuário, gerado automaticamente.
	cpf	varchar	15	Not null, Unique	Número do CPF do usuário.
	rg	varchar	15	Not null, Unique	Número do RG do usuário.
	nome	varchar	80	Not null	Nome do usuário.
	data_nasc	date	-	Not null	Data de nascimento do usuário.
	telefone	varchar	11	Not null	Número de telefone do usuário.
	email	varchar	50	Not null	E-mail do usuário.
	sexo	char	1	Not null	Sexo do usuário (Feminino ou masculino).

	nome_mae	varchar	80	Not null	Nome da mãe do usuário.
	senha	varchar	32	Not null	Senha do usuário para usar no aplicativo.
Cartao	id	int	-	Primary Key, auto_increment	Número de identificação do cartão, gerado automaticamente.
	numero_cartao	varchar	10	Not null, Unique	Número do cartão.
	saldo	decimal	10;2	Not null	Valor do saldo do cartão.
	status	char	1	Not null	Status do cartão, ele pode ser desbloqueado, bloqueado e inativo.
	id_usu	int	-	Foreing key, Not null	Id do usuário importado da tabela Usuario (campo id).
Utilizacao	id	int	-	Primary Key, auto_increment	Número de identificação da utilização, gerado automaticamente.
	valor	decimal	10;2	Not null	Preço da passagem.
	data	datetime	-	Not null	A data da utilização.
	id_car	int	-	Foreing key, Not null	Id do cartão importado da tabela Cartao (campo id).
Movimento	id	int	-	Primary Key, auto_increment	Número de identificação do saldo, gerado automaticamente.
	valor	decimal	10;2	Not null	Quanto de saldo foi inserido na data.
	data	datetime	-	Not null	Quando o saldo foi inserido no cartão.
	id_car	int	-	Foreing key, Not null	Id do cartão importado da tabela Cartao(campo id).

Fonte: Elaborado pelo autor (2018)

Na Tabela 12 é possível ver as descrições dos relacionamentos do banco de dados.

Tabela 12 - Descrição dos Relacionamentos

Relacionamento	Chave estrangeira	Chave primária	Descrição
Tem	Cartao	Usuario	Relacionamento que descreve qual usuário tem o cartão.

Realiza	Utilizacao	Cartao	Relacionamento que descreve qual cartão realizou a operação na catraca do ônibus.
Possui	Movimento	Cartao	Relacionamento que descreve qual cartão possui o saldo.

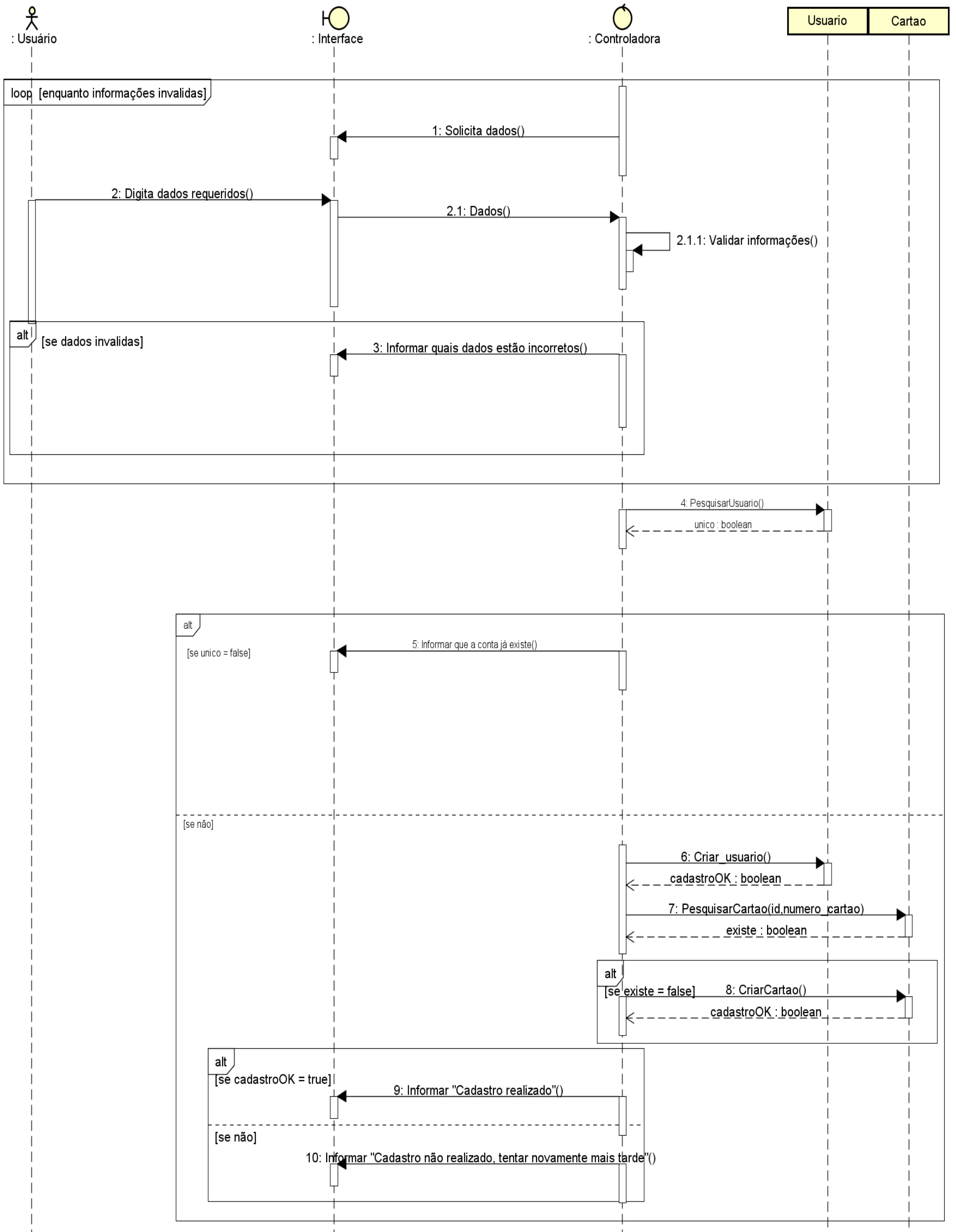
Fonte: Elaborado pelo autor (2018)

2.3.5 Diagrama de Sequência

O Diagrama de sequência mostra a sequência de eventos que irá acontecer dentro de um processo e em que ordem isso ocorrerá (BOOCH; RUMBAUGH; JACOBSON, 2005).

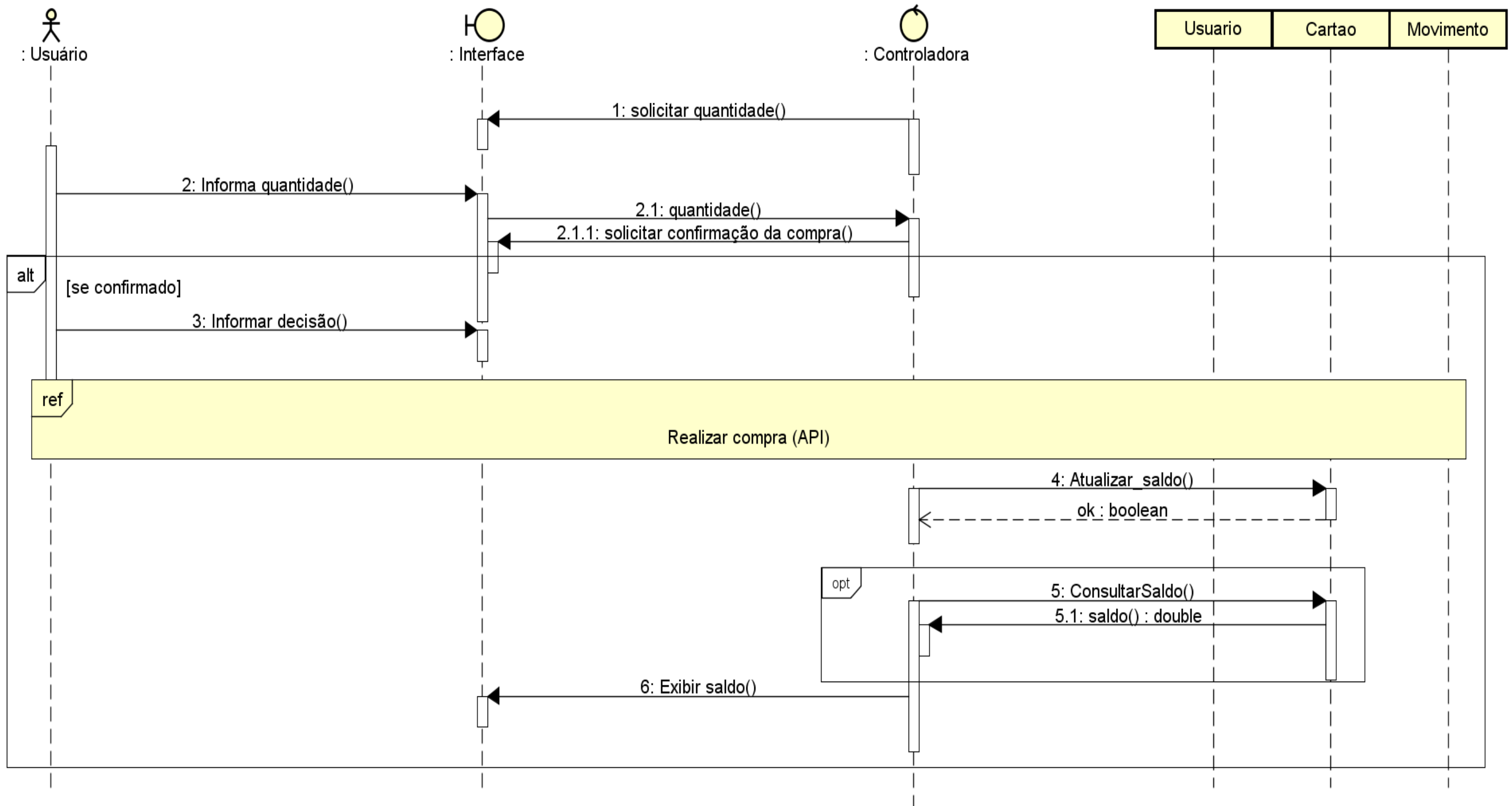
O diagrama foi dividido em várias partes para melhor visualização.

Figura 3 - Diagrama de sequência / Cadastro de cliente



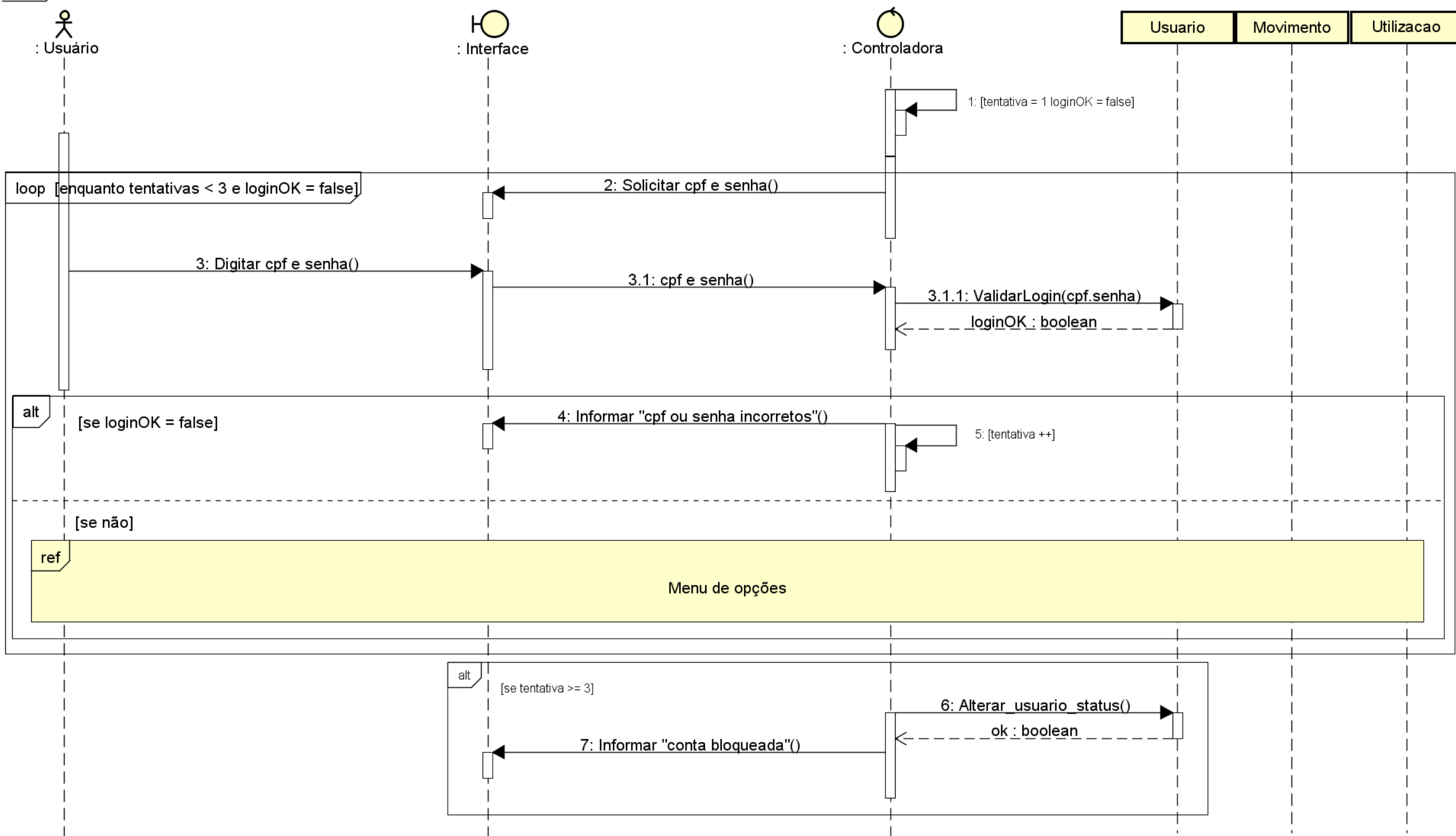
Fonte: Elaborado pelo autor (2018)

Figura 4 - Diagrama de sequência / Controle de crédito



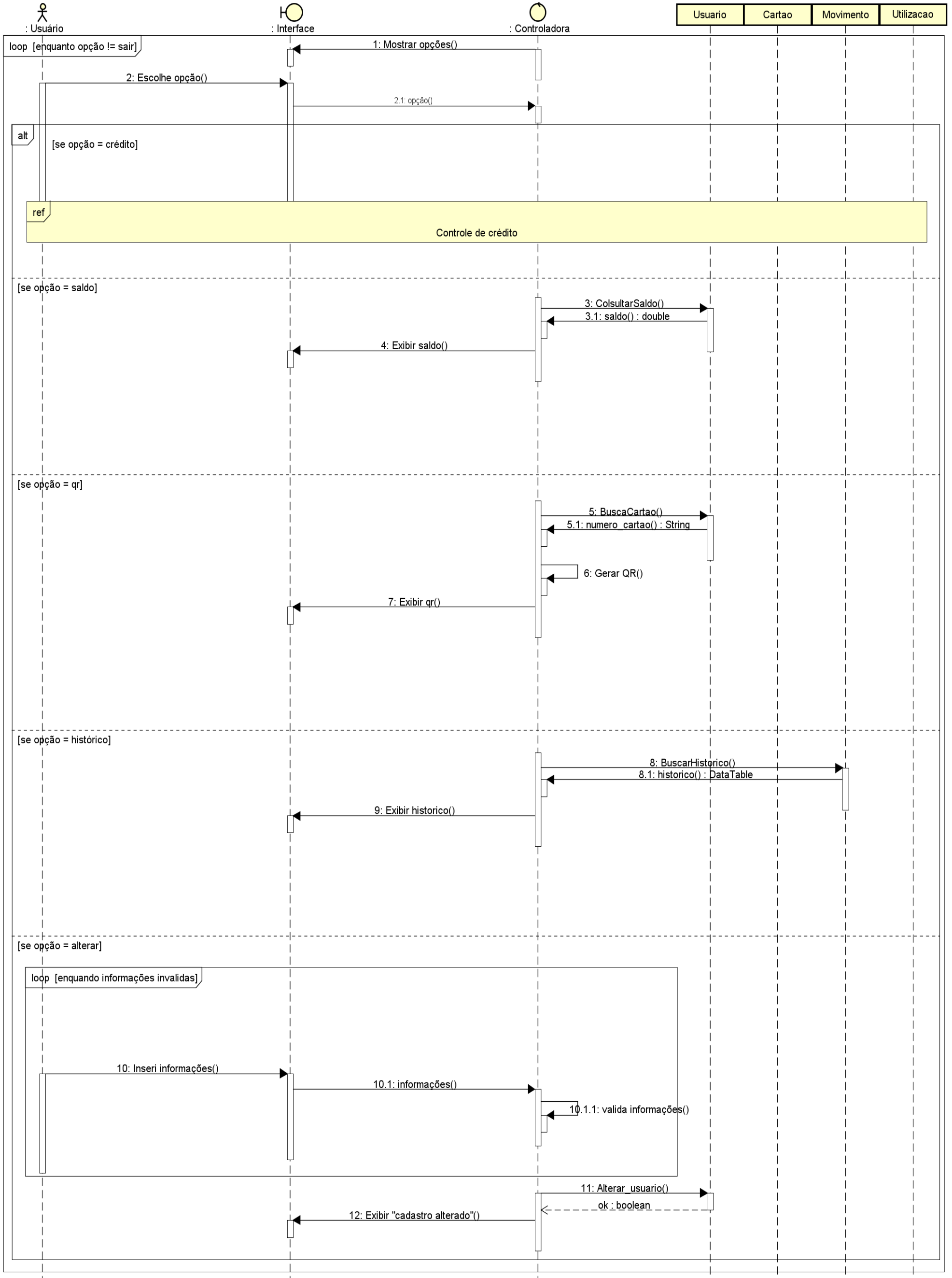
Fonte: Elaborado pelo autor (2018)

Figura 5 - Diagrama de sequência / Login



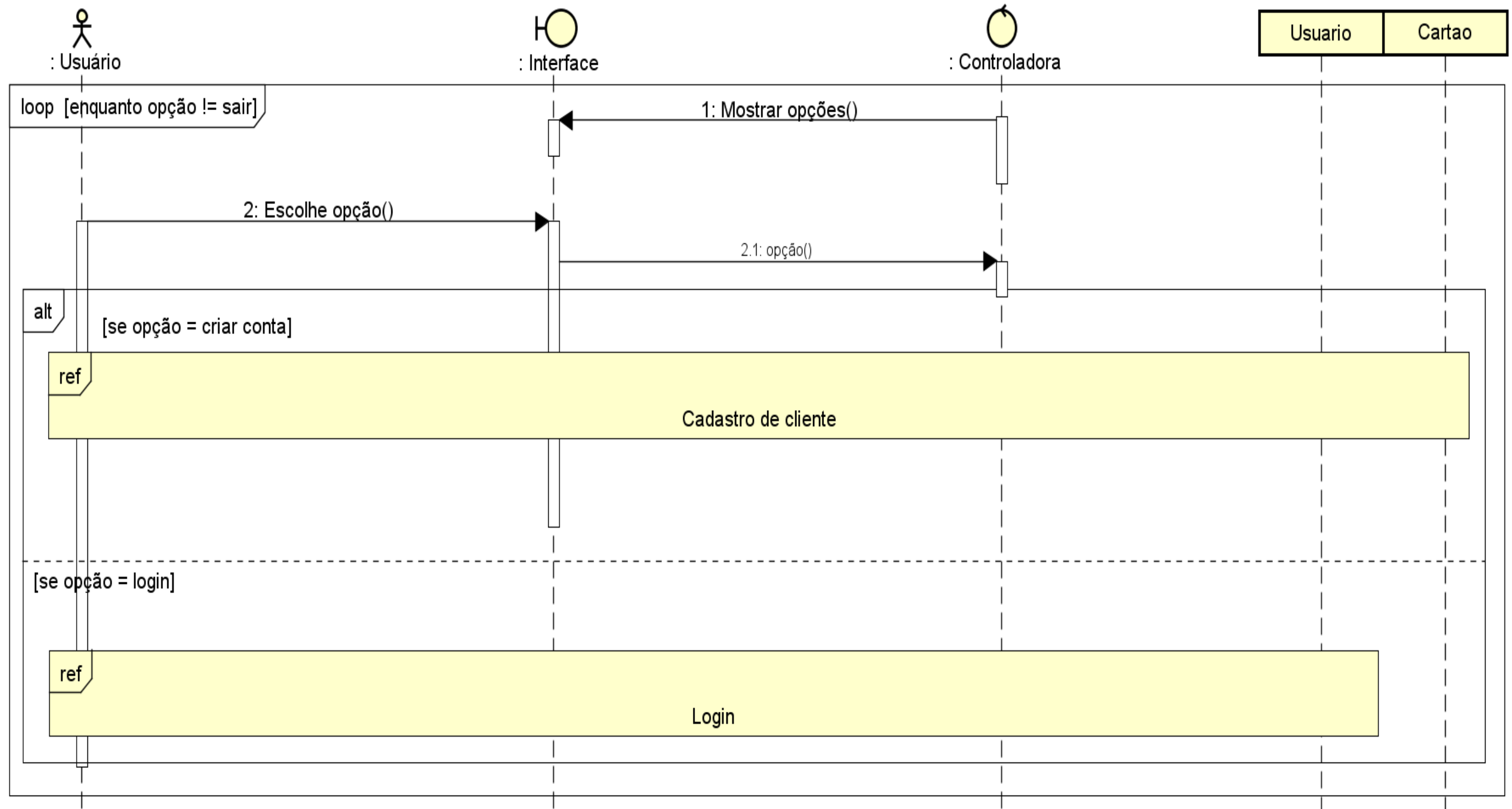
Fonte: Elaborado pelo autor (2018)

Figura 6 - Diagrama de sequência / Menu de opções



Fonte: Elaborado pelo autor (2018)

Figura 7 - Diagrama de sequência / Menu de opções (continuação)



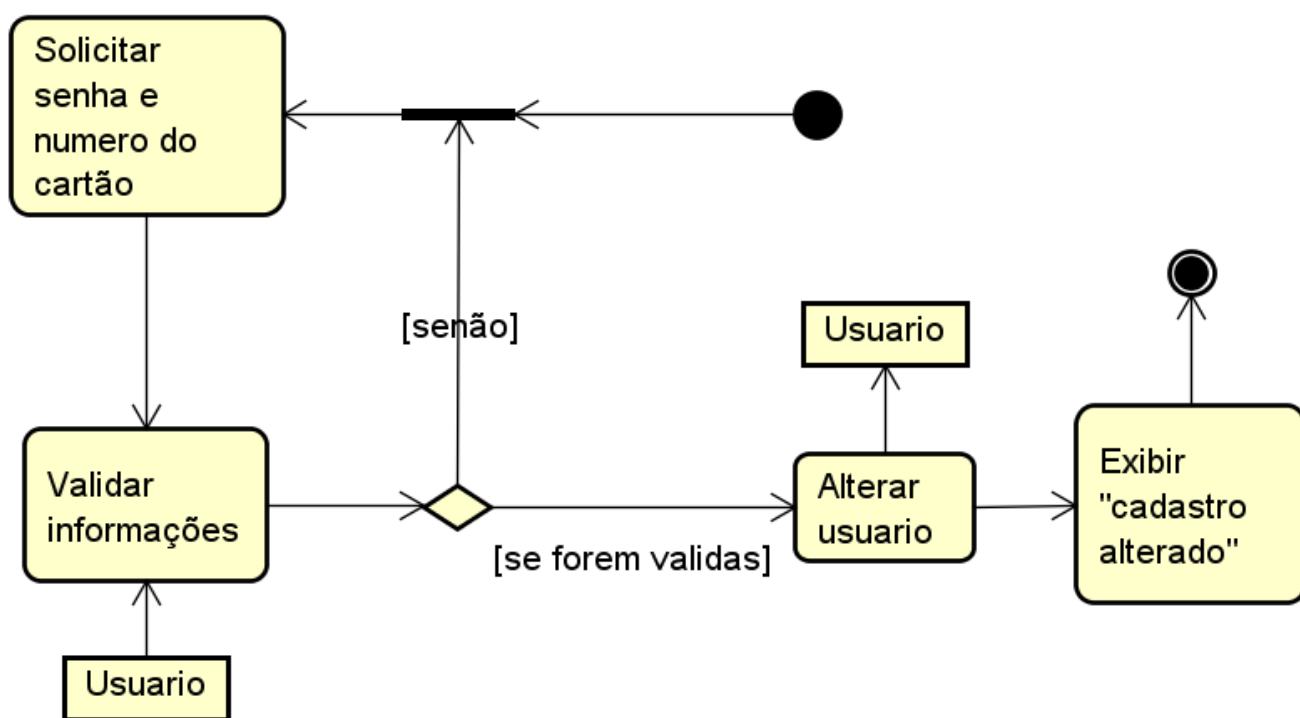
Fonte: Elaborado pelo autor (2018)

2.3.6 Diagrama de Atividades

O diagrama de Atividades é semelhante ao que se demonstra em um fluxograma, porém separa um processo do sistema em diversas atividades que serão executadas em uma determinada sequência para a execução do processo como um todo (BOOCH; RUMBAUGH; JACOBSON, 2005).

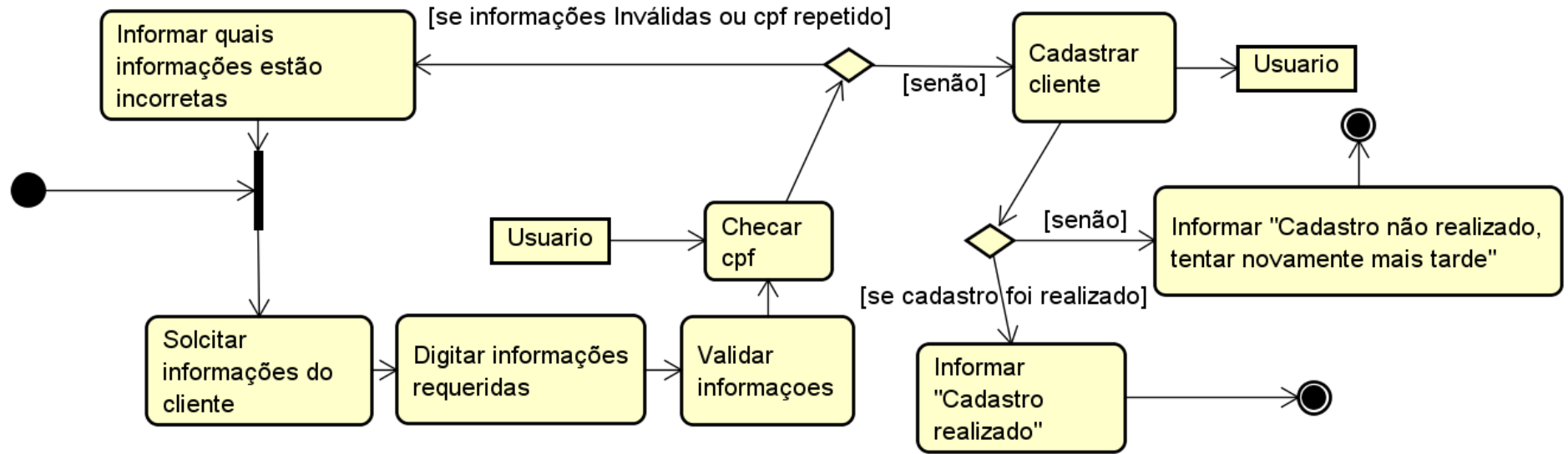
O diagrama foi dividido em várias partes para melhor visualização.

Figura 8 - Diagrama de Atividades / Alterar cadastro



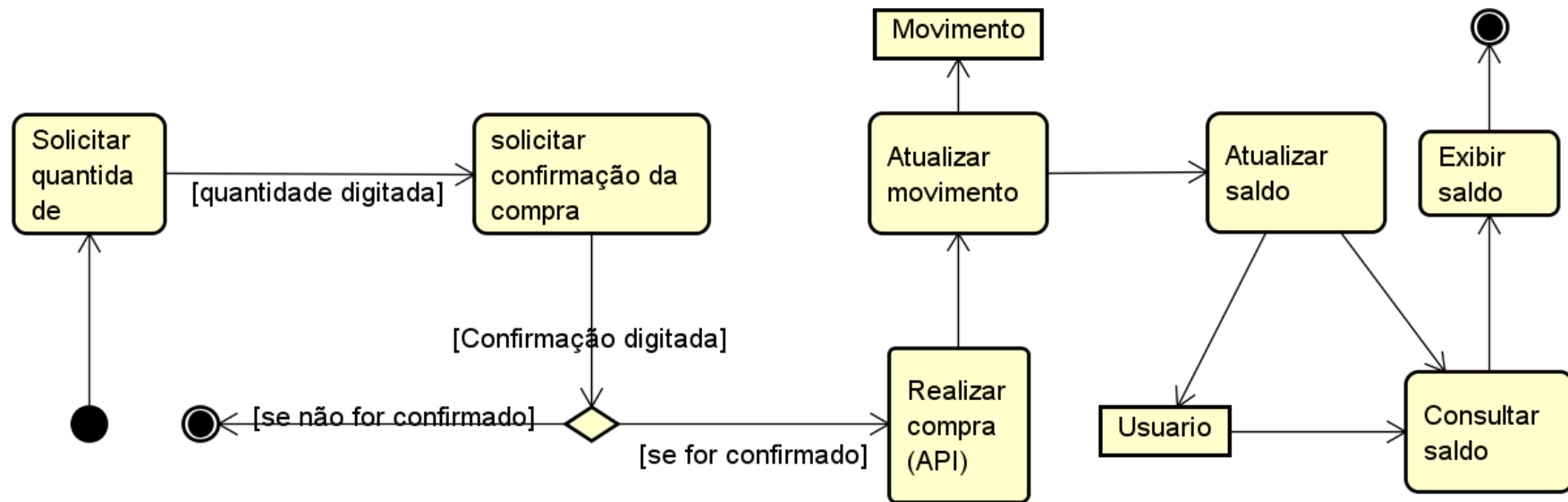
Fonte: Elaborado pelo autor (2018)

Figura 10 - Diagrama de Atividades / Criar conta



Fonte: Elaborado pelo autor (2018)

Figura 11 - Diagrama de Atividades / Controle de crédito

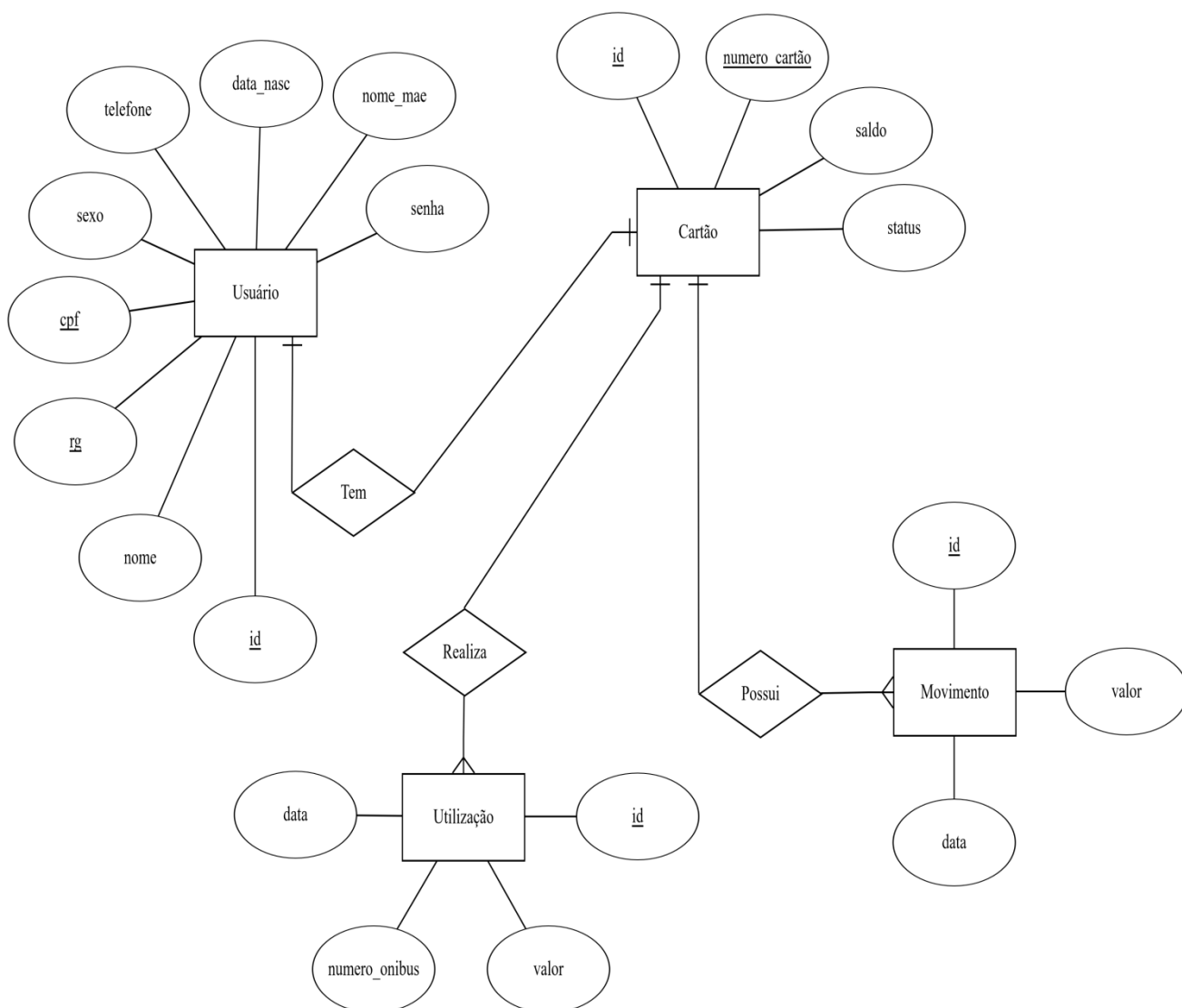


Fonte: Elaborado pelo autor (2018)

2.3.7 Diagrama Entidade Relacionamento

O diagrama entidade relacionamento é uma representação das tabelas do banco de dados e como elas se relacionam (BOOCH; RUMBAUGH; JACOBSON, 2005).

Figura 12 - Diagrama Entidade Relacionamento



Fonte: Elaborado pelo autor (2018)

3 PROTOTIPAGEM

Mais do que apenas um modelo do produto final, a prototipagem tem a função de identificar requisitos detalhados que são necessários para que os recursos gerais dos aplicativos sejam eficazes, além de validar se a iteração do aplicativo com o usuário é intuitiva e fornecer um *feedback* que é usado para refinar os requisitos. (PRESSMAN, 2005).

3.1 Tela de *login*

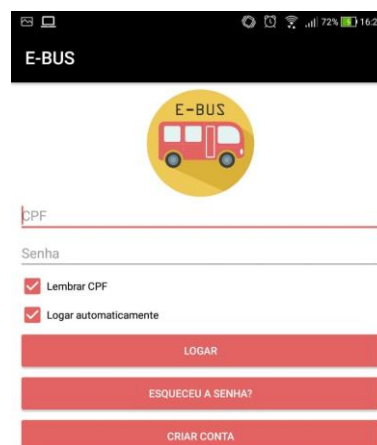
A figura 10 apresenta o protótipo da tela “*Login*” na imagem (a), e, na imagem (b), a versão final da tela no aplicativo composta por:

- **Campo CPF:** Para que o usuário digite seu CPF;
- **Campo senha:** Para que o usuário digite sua senha;
- **Lembrar CPF:** Para que o aplicativo traga o campo CPF preenchido, dessa forma, que o usuário digite apenas sua senha;
- **checkbox logar automaticamente:** Quando selecionada, essa opção evita que o usuário seja obrigado a digitar seu CPF e senha sempre que for utilizar o aplicativo;
- **Botão esqueceu a senha? :** Redireciona o usuário para a tela onde poderá recuperar sua senha, caso a tenha esquecido;
- **Botão entrar:** Realiza a ação de Entrar na Conta do usuário;
- **Botão criar conta:** Redireciona o usuário para a tela onde realizará o seu cadastro para utilização do aplicativo.

Figura 13 - Tela de *login*



(a)



(b)

Fonte: Elaborado pelo autor (2018)

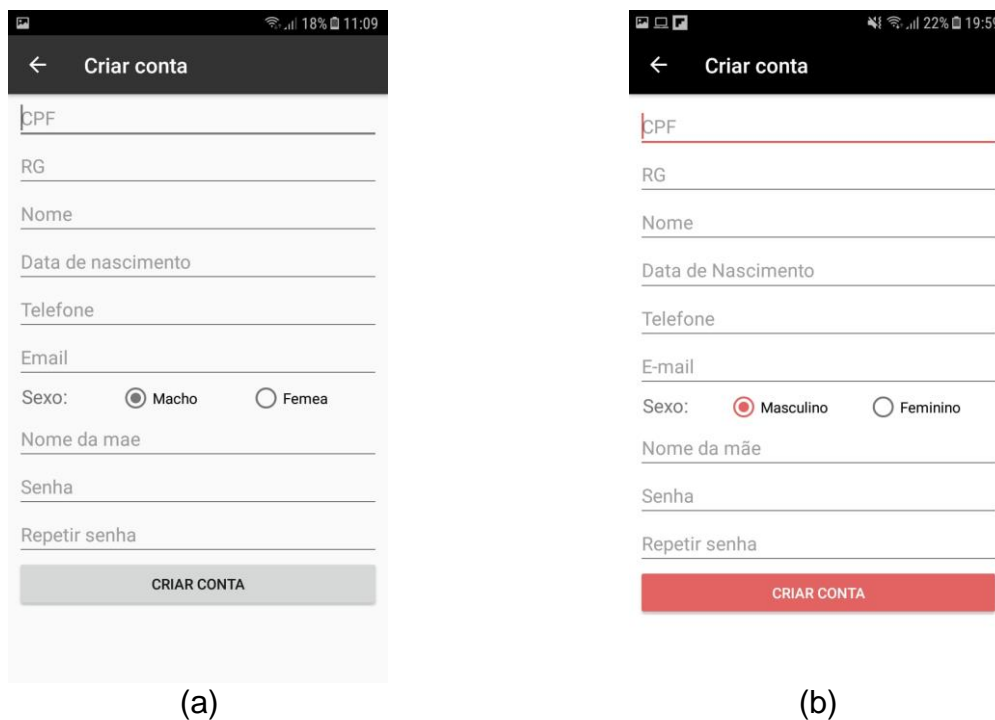
3.2 Tela criar conta

A figura 11 apresenta o protótipo da tela “Criar conta” na imagem (a), e na imagem (b), a versão final da tela no aplicativo composta por:

- **Campo CPF:** Para que o usuário digite seu CPF;
- **Campo RG:** Para que o usuário digite seu RG;
- **Campo Nome:** Para que o usuário digite seu nome;
- **Campo data de nascimento:** Para que o usuário digite seu dia, mês e ano de nascimento;
- **Campo telefone:** Para que o usuário digite seu número de telefone preferido para contato;
- **Campo e-mail:** Para que o usuário digite seu endereço de e-mail;
- **Radio button sexo:** Para que o usuário escolha a opção Feminino ou Masculino;
- **Campo nome da mãe:** Para que o usuário informe o nome completo de sua mãe;

- **Campo senha:** Para que o usuário digite sua senha;
- **Campo repetir senha:** Para que o usuário digite novamente a senha inserida no campo senha;
- **Botão Concluir:** Executa a ação de realizar o cadastro do usuário.

Figura 14 - Tela criar conta



Fonte: Elaborado pelo autor (2018)

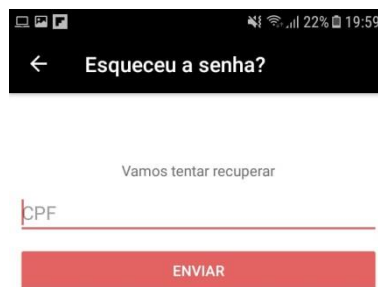
3.3 Tela recuperar senha

A figura 12 apresenta o protótipo da tela “Recuperar senha” na imagem (a), e na imagem (b), a versão final da tela no aplicativo composta por:

- **Campo CPF:** Para que o usuário digite seu CPF;
- **Botão Enviar:** Ao digitar o CPF o sistema deverá verificar se o CPF está cadastrado, se sim, ele irá retomar um e-mail contendo senha do usuário, senão ele irá informar que o CPF não foi encontrado.

Figura 15 - Tela recuperar senha

(a)



(b)

Fonte: Elaborado pelo autor (2018)

3.4 Tela *home*

A figura 13 apresenta o protótipo da tela “*Home*” na imagem (a), e na imagem (b), a versão final da tela no aplicativo composta por:

- **Ícone Menu:** Abre o menu de navegação;
- **Botão atualizar:** Atualiza o saldo do usuário;
- **Campo Saldo:** Mostrar o valor do saldo atualizado do usuário;
- **QR code:** Mostra o QR code do usuário.

Figura 16 - Tela home



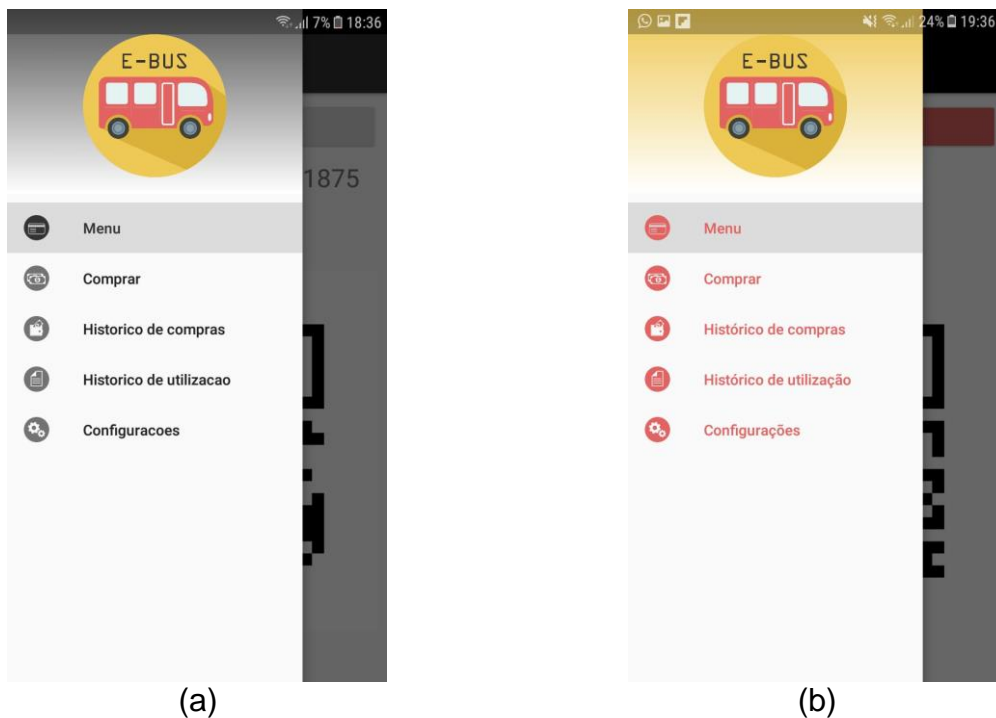
Fonte: Elaborado pelo autor (2018)

3.5 Tela menu navegação

A figura 14 apresenta o protótipo da tela “Menu de navegação” na imagem (a), e na imagem (b), a versão final da tela no aplicativo composta por:

- **Opção Menu:** Redireciona o usuário para a tela *home* do aplicativo;
- **Opção Comprar:** Redireciona o usuário para a tela onde ele irá realizar as compras de crédito;
- **Opção Histórico de compras:** Redireciona o usuário para a tela onde ele irá ver o seu histórico de compras mais recentes;
- **Opção Histórico de utilização:** Redireciona o usuário para a tela onde ele irá ver o seu histórico de utilizações mais recentes;
- **Opção Configurações:** Redireciona o usuário para a tela onde ele pode alterar algumas configurações pessoais da sua conta, como por exemplo: Nome, senha, nome da mãe.

Figura 17 - Tela menu navegação



Fonte: Elaborado pelo autor (2018)

3.6 Tela comprar

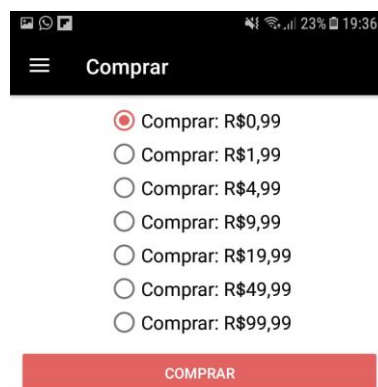
A figura 15 apresenta o protótipo da tela “Comprar” na imagem (a), e na imagem (b), a versão final da tela no aplicativo composta por:

- **Ícone Menu:** Abre o menu de navegação;
- **Radio button com opção de valores:** O usuário pode selecionar a opção de valor que deseja comprar;
- **Botão Comprar:** Realiza a compra com o valor selecionado.

Figura 18 - Tela comprar



(a)



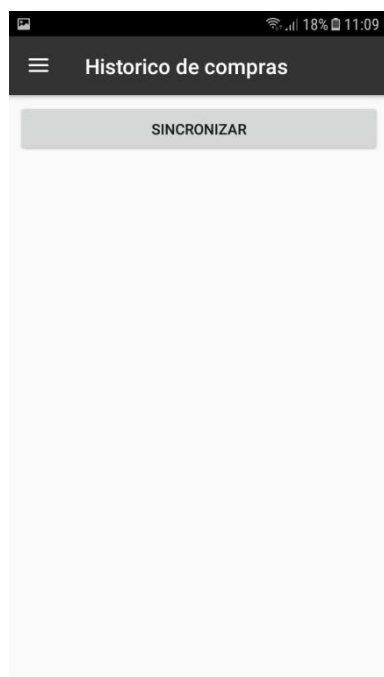
(b)

Fonte: Elaborado pelo autor (2018)

3.7 Tela histórico de compra

A figura 16 apresenta o protótipo da tela “Histórico de compra” na imagem (a), e na imagem (b), a versão final da tela no aplicativo composta por:

- **Ícone Menu:** Abre o menu de navegação;
- **Botão atualizar:** Atualiza o histórico de compra do usuário, que será exibido na tela.

Figura 19 - Tela histórico de compra

(a)



(b)

Fonte: Elaborado pelo autor (2018)

3.8 Tela histórico de utilização

A figura 17 apresenta o protótipo da tela “Histórico de utilização” na imagem (a) e na imagem (b), a versão final da no aplicativo composta por:

- **Ícone Menu:** Abre o menu de navegação;
- **Botão atualizar:** Atualiza o histórico de utilização do usuário, que será exibido na tela.

Figura 20 - Tela histórico de utilização



(a)



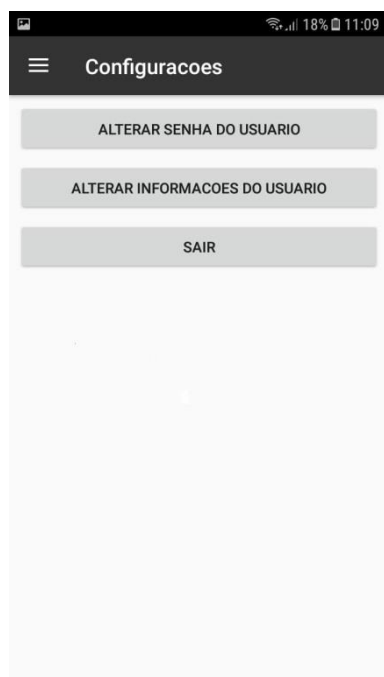
(b)

Fonte: Elaborado pelo autor (2018)

3.9 Tela configuração

A figura 18 apresenta o protótipo da tela “Configuração” na imagem (a), e na imagem (b), a versão final da tela no aplicativo composta por:

- **Ícone Menu:** Abre o menu de navegação;
- **Botão Alterar senha do usuário:** redireciona o usuário para a tela “Alterar senha”;
- **Botão alterar informações do usuário:** redireciona o usuário para a tela “Alterar informações”;
- **Botão sair:** redireciona o usuário para a tela “Home”.

Figura 21 - Tela configuração

(a)



(b)

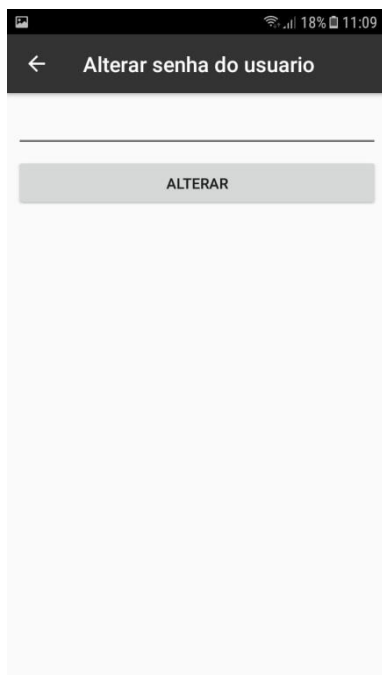
Fonte: Elaborado pelo autor (2018)

3.10 Tela alterar senha

A figura 19 apresenta o protótipo da tela “Alterar senha” na imagem (a), e na imagem (b), a versão final da tela no aplicativo composta por:

- **Ícone Menu:** Abre o menu de navegação;
- **Campo para digitar a nova senha:** campo onde o usuário deverá digitar a nova senha;
- **Campo para confirmar a nova senha:** campo onde o usuário deverá repetir a nova senha;
- **Botão alterar:** executará a alteração;

Figura 22 - Tela alterar senha



(a)



(b)

Fonte: Elaborado pelo autor (2018)

3.11 Tela alterar informações do usuário

A figura 20 apresenta o protótipo da tela “Alterar informações do usuário” na imagem (a), e na imagem (b), a versão final da tela no aplicativo composta por:

- **Ícone Menu:** Abre o menu de navegação;
- **Mensagem:** O usuário deverá preencher apenas os campos que deseja alterar, conforme texto informativo presente na tela em questão;
- **Campo nome:** Para que usuário digite o nome com a alteração desejada;
- **Campo data de nascimento:** Para que o usuário digite a data com a alteração desejada;
- **Campo telefone:** Para que o usuário digite o telefone com a alteração desejada;
- **Campo e-mail:** Para que o usuário digite o endereço de e-mail com a alteração desejada;
- **Radio button Sexo:** Para que o usuário escolha a sexo, conforme alteração desejada;

- **Nome da mãe:** Para que o usuário digite o nome da mãe, com a alteração desejada;
- **Botão alterar:** Executa as alterações.

Figura 23 - Tela alterar informações do usuário

Alterar informações do usu...

Deixa em branco os campos que você não deseja alterar

CPF

RG

Nome

Data de nascimento

Telefone

Email

Sexo: Macho Femea

Nome da mae

ALTERAR CONTA

(a)

Alterar informações do usu...

Deixa em branco os campos que você não deseja alterar

Nome

Data de Nascimento

Telefone

E-mail

Sexo: Masculino Feminino

Nome da mãe

ALTERAR CONTA

(b)

Fonte: Elaborado pelo autor (2018)

4 METODOLOGIA

A metodologia utilizada para o desenvolvimento do sistema foi *framework* o *Scrum*, método que surgiu na década de 1980, criado por Ken Schwaber e Jeff Sutherland (SHUTHERLAND, 2014).

Insatisfeitos com o modelo tradicional em cascata, cujo processo era lento, com atrasos na entrega e orçamentos estourados que eram comuns, e o resultado final quase nunca era o que o cliente esperava, Schwaber e Sutherland desenvolveram em conjunto o *Scrum*.

O *framework Scrum* defende uma abordagem interativa entre as equipes, com entregas e testes incrementais ao final de cada ciclo (*Sprint*) cujos pilares são: a transparência em todos os aspectos do projeto a inspeção constante em tudo que está sendo executado e a adaptação, que se refere não somente ao processo, mas também às mudanças no produto final.

Para a realização de um projeto com *Scrum*, uma equipe precisa cobrir três posições fundamentais: *scrum master*, *product owner* e *dev team*.

- **Scrum master.** Tem o papel de facilitador, com a finalidade de ajudar e orientar a equipe para a eliminação de qualquer dificuldade que venha surgir, que é sua maior responsabilidade.
- **Product owner.** Será a voz do usuário dentro do projeto. É sua responsabilidade identificar quais funcionalidades agregam mais valor ao projeto segundo as necessidades do cliente. A partir dessa análise ele irá definir a ordem de prioridade das entregas das pendências.
- **Dev team.** É a equipe que de fato irá executar as tarefas. Possui autonomia para decidir quantas pendências conseguem finalizar em um ciclo de trabalho.

Não há hierarquias entre eles. Todos têm autonomia para apontar qualquer falha ou mudança a ser executada desde que esteja respaldado de argumentos que convençam os demais integrantes do time.

Este projeto conta com 4 integrantes e 1 professor orientador. Foi com base nesse fator que se tornou possível dividir as funções que seriam exercidas por cada membro do grupo.

- **Scrum Master.** Professor orientador.

- **Product owner:** O *Product owner* mora em Campinas desde a infância, por isso conhece os pontos fortes e fracos do transporte público da cidade, além de ter maior facilidade para manter um contato direto com o cliente.
- **Dev team:** Todos os integrantes do grupo irão trabalhar juntos para que seja possível resolver as pendências de cada *Sprint*. O grupo deve se auto-organizar e dividir as atividades a serem feitas, e manter o andamento das entregas transparente.

O *Scrum* conta com três artefatos durante o seu processo:

- **Product Backlog:** São as pendências do produto. Nessa fase é necessário identificar as funcionalidades ou tarefas que devem ser executadas, mas ao invés de uma lista cheia de itens a serem feitos tem-se histórias, sempre curtas, que devem conter no mínimo um personagem, um desejo e uma motivação, o que ajuda a identificar o seu valor dentro do projeto, também é nesse momento em que classificar-se a dificuldade para a realização de cada história. Vale ressaltar que as funcionalidades levantadas como pendências não são imutáveis, ou seja, na medida em que o projeto evolui, é que se conhece mais sobre o usuário, entendendo o que, segundo a sua concepção, geraria mais valor ao produto, mas essas pendências são refinadas, fazendo com que sejam mais assertivas e que agreguem mais valor ao produto.
- **Sprint Backlog:** Pendências do *product backlog* que serão realizadas durante o *Sprint*, cuja quantidade será decidida pelo time que irá executá-las.
- **Incremento:** É a soma de todas as pendências executadas no *Sprint Backlog* prontas. Devem atender a definição de “prontas” que será previamente estabelecida.

As definições das pendências do produto foram realizadas levando em consideração o projeto como um todo, desde o contato com a prefeitura de Campinas para apresentar-lhes a ideia do aplicativo e solicitar a colaboração da empresa responsável, passando pela elaboração e escrita do relatório, até o

desenvolvimento do aplicativo, sempre levando em consideração o nível de dificuldade de cada item. Alguns dos exemplos das histórias:

- “Como cliente, quero carregar meu aplicativo sem precisar me deslocar até um ponto de revenda.”
- “Como cliente, quero poder ver o histórico de recargas realizadas, afinal preciso ter um controle do que gasto.”
- “Como cliente, quero ver quanto possuí de saldo no aplicativo a qualquer momento, não quero ser pego de surpresa”.
- “Como cliente, quero não precisar pagar mais caro em um *ticket* quando esquecer meu cartão de passe”.
- “Como cliente, não quero pagar mais caro em um *ticket* quando estou temporariamente em uma cidade”.

Para ajudar a manter os pilares fundamentais, o *Scrum* possui eventos que ocorrem no decorrer do projeto. Eles ajudam na transparência, inspeção e adaptação, e em nenhuma hipótese, podem ser ignorados ou negligenciados.

- ***Sprint Planning***: *Sprint* é uma quantidade de tempo definida pelo grupo para a realização de um conjunto de funcionalidades, ou seja, é o tempo que a equipe terá para realizar as pendências selecionadas. Sua duração será sempre fixa. No planejamento serão decididas quantas e quais pendências serão realizadas dentro do *Sprint*, e uma vez que a equipe se compromete com essas entregas, deve trabalhar em conjunto para entregar tudo o que foi combinado.
- ***Daily Scrum***: Reuniões diárias, com duração máxima de 15 minutos, cujo objetivo é expor o andamento do projeto e as dificuldades encontradas. É importante ter em mente três perguntas fundamentais:
O que eu fiz ontem?
O que vou fazer hoje?
Que obstáculos a equipe tem enfrentado?

Essa reflexão irá ajudar a equipe a identificar os problemas no momento em que eles aparecem e já se organizar para resolvê-los antes que eles afetem de maneira grave as entregas a serem realizadas.

- ***Demonstração e Retrospectiva da Sprint***: Ao final de cada *Sprint*, a equipe irá entregar as pendências ou tarefas que estão prontas,

possibilitando um *feedback* imediato, que vai mostrar se o que está sendo feito é o que se era esperado, e se é satisfatório. Feito isso, o time deverá analisar quais procedimentos deram certo ou errado, e acabaram facilitando ou dificultando a realização das tarefas. É importante ressaltar que esse processo busca o aprimoramento da equipe e não culpados para apontar.

4.1 Utilizando *Scrum*

Para este projeto, foi definido que cada *Sprint* teria duração de 28 dias. As reuniões diárias são feitas na faculdade, e a cada fim de um *Sprint* há uma reunião na casa de um integrante do grupo, ou através de uma videoconferência para analisar as entregas feitas, conversar sobre as dificuldades enfrentadas e planejar o próximo *Sprint*.

4.1.1 *Sprint* um

O *Sprint* um foi definido através de uma reunião, na qual houve participação de todos os membros. Nesta reunião o *Product owner* apresentou as atividades e seus níveis de importância e juntamente com o *dev team* definiu as seguintes atividades: Escrita da introdução, escrita da metodologia e escrita do resumo. A Tabela 13 apresenta os prazos das atividades, seu nível de dificuldade, sua importância e seu status ao final do *Sprint*.

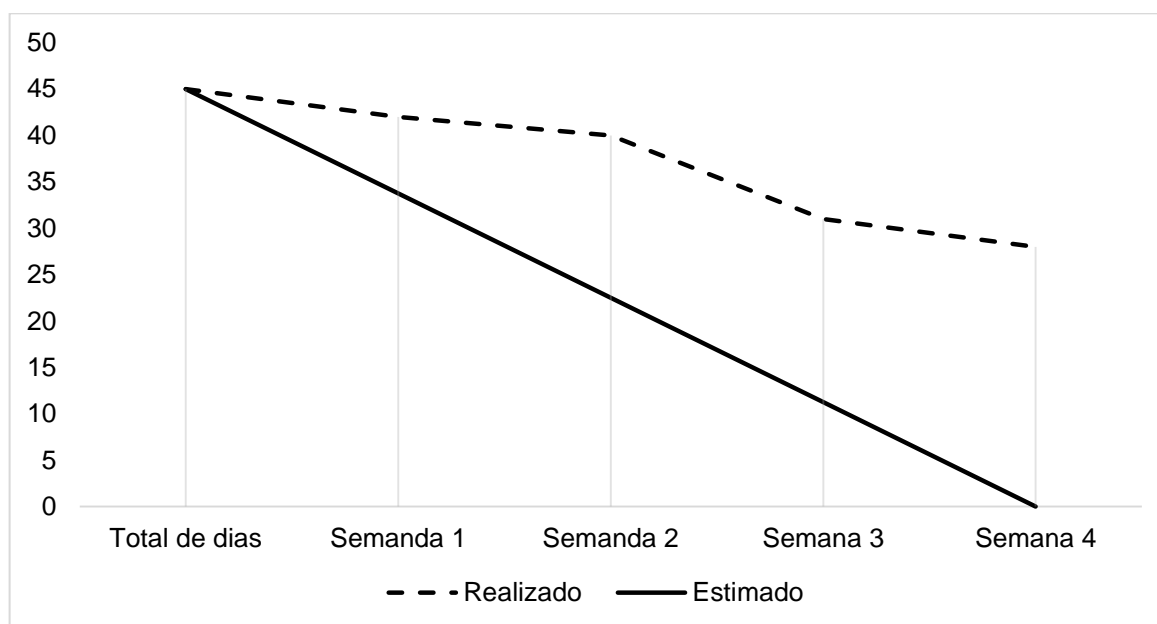
Tabela 13 - Demonstração *Sprint* um

Nome da Tarefa	Dias	Dificuldade	Importância	Status
<i>Sprint</i> 1: 04/09/17 até 30/09/17	28	13	27	Completa
Escrita Introdução	7	5	9	Completa
Escrita metodologia	7	5	9	Completa
Escrita Resumo	3	3	9	Completa
<u>Reunião com Cliente</u>	<u>28</u>	<u>10</u>	<u>10</u>	<u>Não concluída</u>

Fonte: Elaborado pelo autor (2018)

Com base na Tabela 13, foi projetado o gráfico de *burndown* (Figura 21). Neste gráfico é apresentada, no eixo Y, a soma do total de dias necessários para realizar as tarefas escolhidas, e o eixo X, a quantidades de semanas que a Sprint possui. O gráfico mostra o andamento do *Sprint* a cada semana.

Figura 24 - Gráfico *Burndown Sprint* um



Fonte: Elaborado pelo autor (2018)

A tabela 14 apresenta uma visão geral da retrospectiva do *Sprint*. Todos os membros da equipe estavam presentes na reunião. Ao avaliar nosso desempenho procuramos encontrar os pontos de acertos e erros, e juntos, propor ações de melhorias para o desempenho da equipe.

Tabela 14 - Retrospectiva *Sprint* um

O que ficou Pronto Na Sprint	Resumo, Introdução e metodologia a serem entregues no TCC 1.
O Que foi bom?	Houve uma boa comunicação entre os integrantes. Dessa forma todos estavam cientes das dificuldades enfrentadas.
O que foi ruim?	Não houve iniciativa dos integrantes para ajudar a resolver os problemas identificados.
Ações de Melhorias?	Embora o nosso “cliente” estivesse sem horários disponíveis para uma reunião presencial, poderia ter sugerido reuniões por <i>Skype</i> , ou ter trocados mais e-mail para tirar a principais dúvidas com relação ao projeto, de forma a minimizar os impactos, ou seja, expandir a gama de possibilidades e alternativas para contornar o problema.
Observações:	O Gerente de T.I da Transurc não tinha disponibilidade para reunião para os próximos meses.

Fonte: Elaborado pelo autor (2018)

4.1.2 *Sprint* dois

O *Sprint* dois contou com a participação de todos os membros. Neste *Sprint*, realizou-se uma videoconferência, o *Product owner* apresentou as atividades restantes e seus níveis de importância para juntamente com o *Dev Team* definir as atividades da *Sprint* dois, cuja atividades definidas foram: Definição dos Requisitos com cliente, Documentação do ambiente de desenvolvimento, Documentação do Sistema – UML, Diagrama Entidade Relacionamento, Criação Banco de dados, Escrita do documento e Formatação. A Tabela 15 apresenta os prazos das atividades, seu nível de dificuldade, sua importância e seu status ao final do *Sprint*.

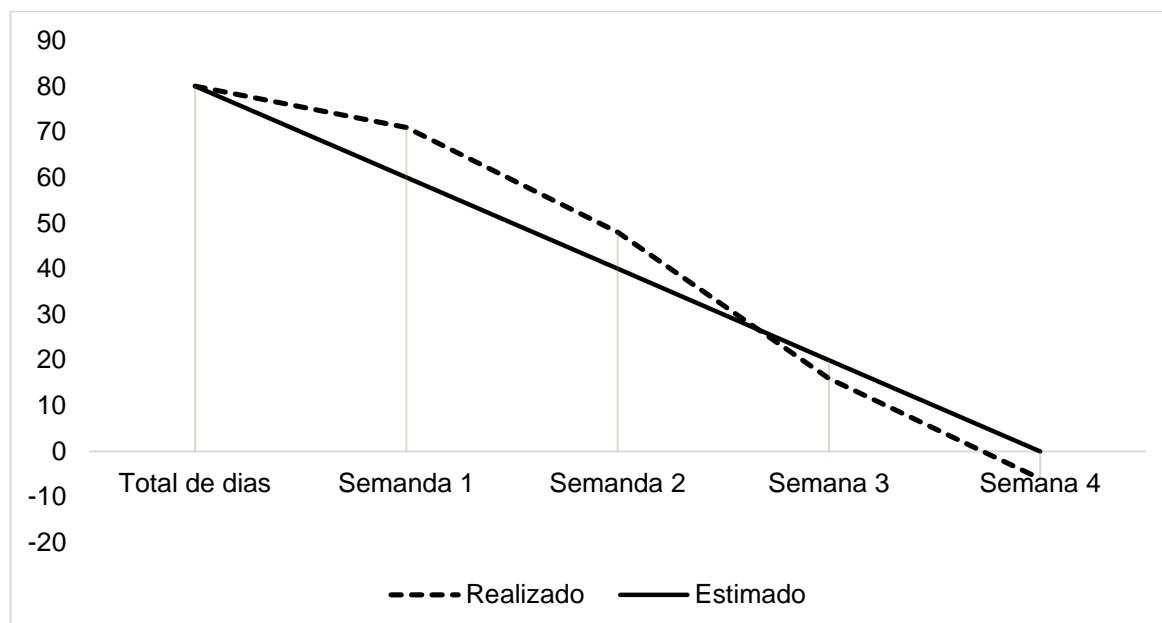
Tabela 15 - Demonstração *Sprint* dois

Nome da Tarefa	Dias	Dificuldade	Importância	Status
<i>Sprint</i> 2: 02/10/17 até 29/10/17	28	32	54	Completa
Definição dos Requisitos com cliente	10	10	10	Completa
Documentação do ambiente de desenvolvimento	2	2	4	Completa
Documentação do Sistema – UML	8	6	9	Completa
Diagrama Entidade Relacionamento	3	4	9	Completa
Criação Banco de dados	3	3	8	Completa
Escrita do documento	4	5	9	Completa
Formatação	3	2	5	Completa

Fonte: Elaborado pelo autor (2018)

Com base na Tabela 15, foi projetado o gráfico de *burndown* (Figura 22). Neste gráfico é apresentada, no eixo Y, a soma do total de dias necessários para realizar as tarefas escolhidas, e o eixo X, a quantidade de semanas que o *Sprint* possui. O gráfico mostra o andamento do *Sprint* a cada semana.

Figura 25 - Gráfico *Burndown Sprint dois*



Fonte: Elaborado pelo autor (2018)

A tabela 16 apresenta uma visão geral da retrospectiva do *Sprint*. Todos os membros da equipe estavam presentes na reunião. Ao avaliar nosso desempenho procurou-se encontrar os pontos de acertos e erros, e junto, propor ações de melhorias para o desempenho da equipe.

Tabela 16 – Retrospectiva *Sprint dois*

O que ficou Pronto Na Sprint	Todas entregar necessárias para apresentação TCC1
O Que Foi Bom?	Houve mais proatividade dos integrantes para auxiliar as atividades que ficaram para o final do <i>Sprint</i>
O que foi ruim?	Todas as atividades a serem realizadas dependiam das repostas do cliente aos e-mails. O cliente demorou a responder e, por isso, no final do <i>Sprint</i> havia muitas atividades acumuladas.
Ações de Melhorias?	Cautela ao distribuir muitas atividades dependentes de uma única coisa. Poderia ter adiantado partes do projeto que não eram tão dependentes da reposta do cliente e passaria menos tempo esperando.
Observações:	Apesar da demora, o cliente respondeu a todas as duvidas mais importantes para o início do projeto e agendou-se a reunião para janeiro.

Fonte: Elaborado pelo autor (2018)

4.1.3 *Sprint três*

O *Sprint* três contou com a participação de todos os membros. Nesta reunião o *Product owner* apresentou as atividades restantes e seus níveis de importância para, juntamente com o *Dev Team*, definir as atividades do *Sprint*, as atividades definidas foram: Nome/Logo do Aplicativo, Protótipo *App*, Tela de *Login*, Tela de Cadastro, Tela Compra, Tela de Histórico, Tela Consulta Saldo, Tela altera Cadastro. A Tabela 17 apresenta os prazos das atividades, seu nível de dificuldade, sua importância e seu status ao final do *Sprint*.

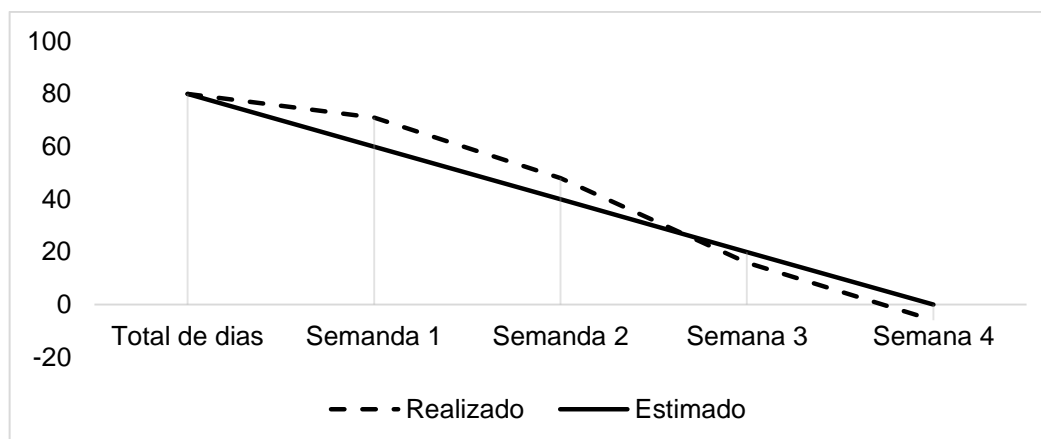
Tabela 17 - Demonstração *Sprint* três

Nome da Tarefa	Dias	Dificuldade	Importância	Status
<i>Sprint</i> 3: 05/03/18 até 01/04/18	28	41	63	Completa
Nome/Logo do Aplicativo	10	6	3	Completa
Protótipo <i>App</i>	10	5	6	Completa
Tela de <i>Login</i>	3	5	10	Completa
Tela de Cadastro	4	5	10	Completa
Tela compra	3	5	10	Completa
Tela de Histórico	4	5	8	Completa
Tela Consulta Saldo	3	5	8	Completa
Tela altera Cadastro	4	5	8	Completa

Fonte: Elaborado pelo autor (2018)

Com base na Tabela 17, foi projetado o gráfico de *burndown* (Figura 23). Neste gráfico é apresentada, no eixo Y, a soma do total de dias necessários para realizar as tarefas escolhidas, e o eixo X, a quantidades de semanas que o *Sprint* possui. O gráfico mostra o andamento do *Sprint* a cada semana.

Figura 26 - Gráfico *Burndown* *Sprint* três



Fonte: Elaborado pelo autor (2018)

A tabela 18 apresenta uma visão geral da retrospectiva do *Sprint*, todos os membros da equipe participaram. Ao avaliar nosso desempenho, procurou-se encontrar os pontos de acertos e erros, e juntos propor ações de melhorias para o desempenho da equipe.

Tabela 18 - Retrospectiva *Sprint* três

O que ficou Pronto No <i>Sprint</i>	Nome, logo e protótipo dos aplicativos. Todas as telas foram desenvolvidas de forma a estarem prontas para receber suas funcionalidades.
O Que Foi Bom?	Com o desenvolvimento da parte estética do aplicativo, conseguiu-se identificar a possibilidade de melhoria referente a usabilidade do produto.
O que foi Ruim?	A equipe tinha capacidade de executar mais pendências do que as contidas nos <i>Sprints</i> .
Ações de Melhorias?	Elaborar um <i>Sprint</i> mais regular com relação às atividades que cada integrante deve executar
Observações:	

Fonte: Elaborado pelo autor (2018)

4.1.4 *Sprint* quatro

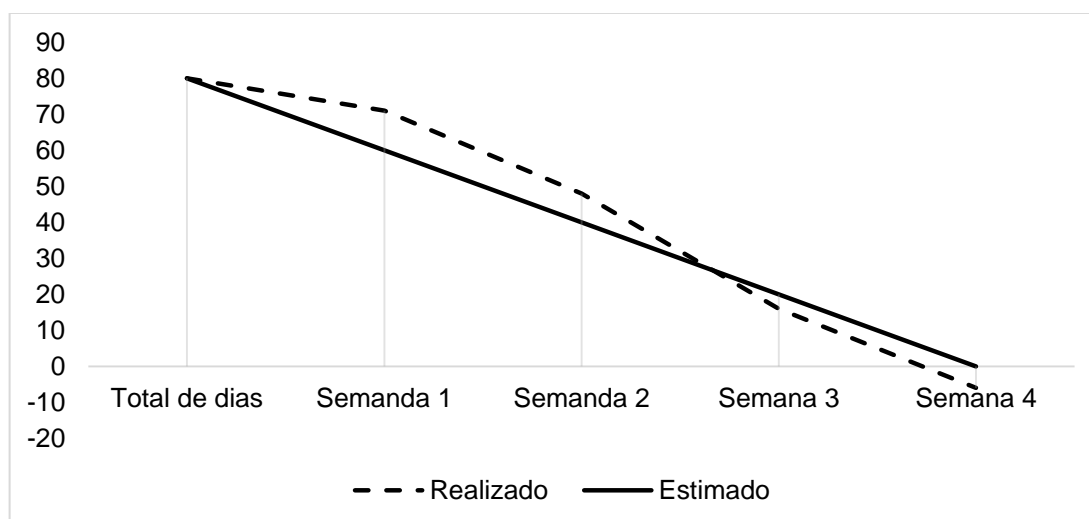
O *Sprint* quatro contou com a participação de todos os membros. Nesta reunião, o *Product owner* apresentou as atividades restantes e os níveis de importância para juntamente com o *Dev Team*, definir as atividades do *Sprint*, que foram: atualizar a introdução, atualizar o resumo, atualizar a Metodologia, correção de diagramas, criar o dicionário de dados, atualizar a funcionalidade compra, atualizar a Funcionalidade *login*, atualizar a Funcionalidade cadastro. A Tabela 19 apresenta os prazos das atividades, seu nível de dificuldade, sua importância e seu status ao final do *Sprint*.

Tabela 19 - Demonstração *Sprint* quatro

Nome da Tarefa	Dias	Dificuldade	Importância	Status
<i>Sprint</i> 4: 03/04/18 até 30/04/18	28	51	55	Completa
Atualizar a Introdução	7	5	5	Completa
Atualizar o Resumo	7	4	5	Completa
Atualizar a Metodologia	7	7	5	Completa
Correção de Diagramas	7	5	5	Completa
Criar Dicionário de dados	7	6	5	Completa
Atualizar a Funcionalidade compra	15	8	10	Completa
Atualizar a Funcionalidade <i>login</i>	15	8	10	Completa
Atualizar a Funcionalidade cadastro	15	8	10	Completa

Fonte: Elaborado pelo autor (2018)

Com base na Tabela 19, foi projetado o gráfico de *burndown* (Figura 24). Neste gráfico é apresentada no eixo Y a soma do total de dias necessários para realizar as tarefas escolhidas, e o eixo X, a quantidades de semanas que o *Sprint* possui. O gráfico mostra o andamento do *Sprint* a cada semana.

Figura 27 - Gráfico de *Burndown* *Sprint* quatro

Fonte: Elaborado pelo autor (2018)

A tabela 20 apresenta uma visão geral da Retrospectiva do *Sprint*, todos os membros da equipe estavam presentes na reunião. Ao avaliar nosso desempenho procurou-se encontrar os pontos de acertos e erro, e juntos, propor ações de melhorias para o desempenho da equipe.

Tabela 20 – Retrospectiva *Sprint* quatro

O que ficou Pronto No <i>Sprint</i>	Finalizadas as funcionalidades mais importantes do aplicativo, finalizada também a introdução, resumo, metodologia, diagramas e dicionários de dados da versão final do TCC.
O Que Foi Bom?	Foi o <i>Sprint</i> em que se conseguiu aproximar da execução que foi planejada
O que foi ruim?	Devido a problemas no desenvolvimento da funcionalidade de compra, houve atraso na entrega.
Ações de Melhorias?	Executar as tarefas mais técnicas logo no início do <i>Sprint</i> , dessa forma, em caso de problemas técnicos, tem-se um tempo maior para resolver o incidente sem que haja qualquer atraso.
Observações:	

Fonte: Elaborado pelo autor (2018)

4.1.5 *Sprint* cinco

O *Sprint* Cinco contou com a participação de todos os membros. Nesta reunião, o *Product owner* apresentou as atividades finais e os níveis de importância para, juntamente com o *Dev Team* definir as atividades do *Sprint*, que foram: atualizar a funcionalidade histórico, Atualizar a funcionalidade consultar saldo, atualizar a funcionalidade alterar cadastro, desenvolvimento plano de testes, realização dos testes, escrita da implementação, escrita das considerações finais e formatação de acordo com a norma ABNT. A Tabela 21 apresenta os prazos das atividades, seu nível de dificuldade, sua importância e seu status ao final do *Sprint*.

Tabela 21 – Demonstração *Sprint* cinco

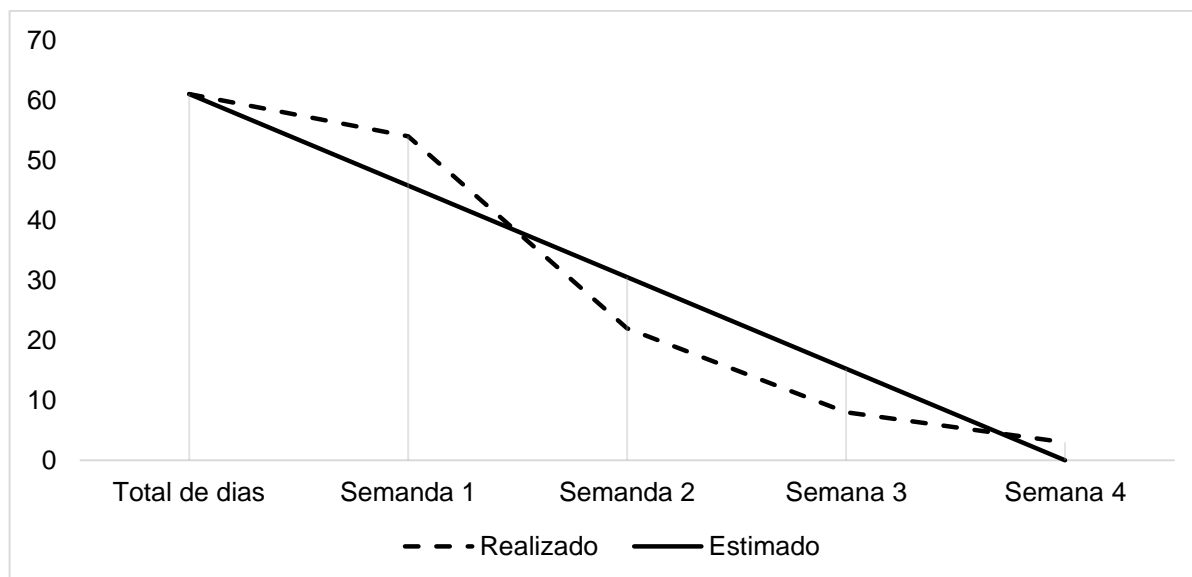
Nome da Tarefa	Dias	Dificuldade	Importância	Status
<i>Sprint</i> 5: 02/05/18 até 29/05/18	61	49	70	Completa
Atualizar funcionalidade de histórico	15	9	8	Completa
Atualizar a funcionalidade consultar saldo	15	9	10	Completa
Atualizar a funcionalidade alterar cadastro	15	5	7	Completa
Desenvolvimento plano de testes	7	7	10	Completa
Realização dos testes	4	4	10	Completa
Escrita da Implantação	2	6	9	Completa
Escrita das considerações finais	2	5	9	Completa
Formatação ABNT	1	4	7	Completa

Fonte: Elaborado pelo autor (2018)

Com base na Tabela 21, foi projetado o gráfico de *burndown* (Figura 25). Neste gráfico é apresentada, no eixo Y, a soma do total de dias que levaremos para

realizar as tarefas escolhidas, e o eixo X, a quantidade de semanas que o *Sprint* possui. O gráfico mostra o andamento do *Sprint* a cada semana.

Figura 28 - Gráfico de *Burndown Sprint* cinco



Fonte: Elaborado pelo autor (2018)

A tabela 22 apresenta uma visão geral da retrospectiva do *Sprint*, todos os membros da equipe estavam presentes na reunião. Ao avaliar nosso desempenho procurou-se encontrar os pontos de acertos e erro, e juntos propor ações de melhorias.

Tabela 22 - Retrospectiva *Sprint* cinco

O que ficou Pronto No <i>Sprint</i>	Finalizado o restante das funcionalidades do aplicativo, realizado os testes e as devidas correções e finalizada a documentação do ambiente necessário para implantação do aplicativo e considerações finais.
O Que Foi Bom?	Foi um <i>Sprint</i> em que nosso andamento foi melhor do que o planejado na maior parte do <i>Sprint</i> .
O que foi ruim?	Ao realizar os testes, identificaram-se vários erros de acentuação no aplicativo, e houve um retrabalho para corrigir.
Ações de Melhorias?	Se atentar e se planejar de forma igualitária para todas as etapas do projeto, desde atividades mais complexas, até atividades consideradas mais fáceis.
Observações:	

Fonte: Elaborado pelo autor (2018)

5 IMPLEMENTAÇÃO

Este capítulo apresenta a implementação do aplicativo, o plano de testes com a descrição do teste, o passo e o resultado esperado, e em seguida o resultado real obtido dos testes.

5.1 Plano de testes

Tabela 23 - Plano de testes

Caso de Teste	Nome do Teste	Descrição	Passos	Resultado Esperado
CT 1	Criar cadastro	Criar um cadastro para acesso ao aplicativo ao clicar no botão: "Criar Conta"	1- Clicar no botão "Criar Conta"	Caso todos os dados estejam digitados corretamente, o cadastro será realizado gerando uma tela de confirmação.
			2- Preencher os dados solicitados	
			3- Clicar novamente em "Criar conta"	
CT 2	Recuperar Senha	Recuperar uma senha esquecida	1- Clicar no botão "recuperar senha"	Enviar um e-mail para o proprietário da conta, caso o CPF esteja correto
			2- Digitar o CPF cadastrado	
			3- Receber um e-mail com a senha	
CT 3	QR Code	Tela inicial chamada de "Menu" onde se encontra o QR code e o saldo atual do cartão. Existe também um botão de sincronização para atualizar o saldo.	Ao entrar na conta você já é diretamente direcionado a tela de Menu	Mostrar o QR code
CT 4	Tela Comprar	Tela Comprar se encontra na aba lateral e já tem valores pré-definidos para a escolha do cliente. A compra já é realizada diretamente na Google play	1- Clicar na barra de compra, localizada no menu lateral	1 - Abrir a tela da Google play
			2 - Selecionar um valor desejado	2 - Confirmar a compra
			3 - Clicar em comprar	3 - Atualizar o saldo do cliente
CT 5	Histórico de Compras	Tela de Histórico de compras se encontra no menu lateral e é onde ficam registrados todas as compras realizadas pelo cliente	1- Clicar no ícone "Histórico de Compras", localizado no menu lateral	1 - Mostrar data e hora das compras realizadas.
			2- Clicar em sincronizar, para atualizar até o histórico mais recente	2 - Atualizar os dados ao clicar no botão de sincronizar.

Caso de Teste	Nome do Teste	Descrição	Passos	Resultado Esperado
CT 6	Histórico de Utilização	Tela de Histórico de utilizações de saldo do cliente	1- Clicar no ícone "Histórico de utilização", localizado no menu lateral	1 - Mostrar data e hora das utilizações do saldo
			2- Clicar em sincronizar para atualizar até o histórico mais recente	2- Atualizar os dados ao clicar no botão sincronizar
CT 7	Configurações "Alterar Senha"	Botão de Alterar senha se encontra, dentro das configurações	1- Clicar em "Configurações", localizado na barra lateral	Confirmar para o usuário que a senha foi alterada com sucesso
			2- Clicar no botão alterar senha	
			3- Digitar a senha e repetir para realizar a alteração	
			4- Clicar no botão alterar	
CT 8	Configurações "Alterar informações"	Botão de Alterar Informações se encontra, dentro das configurações	1- Clicar em "Configurações", localizado na barra lateral	Salvar as informações alteradas na base de dados
			2- Clicar no botão alterar informações	
			3- Digitar apenas o que deseja alterar	
			4- Clicar em alterar conta para salvar	
CT 9	Configurações "Botão Sair"	Botão de Sair se encontra na barra lateral, dentro de configurações	1- Clicar em "Configurações", localizado na barra lateral	Sair da conta do usuário
			2- Clicar no botão sair	

Fonte: Elaborado pelo autor (2018)

5.2 Resultados dos testes

Caso de teste 1 (CT 1): Foi elaborado um teste de criação de conta onde o usuário digita os dados para realizar seu cadastro no sistema. Caso o usuário deixe de digitar alguma informação em um dos campos obrigatórios, ele será notificado com um erro para que faça a correção. Caso tenha algum dado digitado

incorretamente, será informado para corrigi-lo. O aplicativo conta com máscara para validar RG, CPF, telefone, e-mail e senha, fazendo com que alimente o banco de dados com informações corretas. Se por algum motivo o usuário estiver sem acesso à internet o aplicativo exibirá uma notificação com um erro de conexão. Para os dados serem armazenados no banco e gerar um cadastro, todas as exigências do aplicativo devem ser realizadas corretamente. Feito isso, o arquivo será atualizado e aparecerá um ícone de carregamento até que os dados sejam salvos, assim gerando uma tela de confirmação.

Caso de teste 2 (CT 2): Ao clicar no botão de recuperação de senha, será direcionado a outra tela para informar o CPF, que será localizado no banco de dados e verificar se ele existe ou não. Caso o CPF não exista no banco de dados ou foi digitado incorretamente o usuário será informado com uma mensagem de erro. Se estiver tudo correto, o sistema enviará um e-mail para o endereço que está registrado junto ao CPF com a senha.

Caso de teste 3 (CT 3): Logo ao entrar no sistema com usuário e senha, abrirá uma tela com o QR *code* já gerado para uso e a informação de saldo no topo. Na tela inicial existe um botão de sincronizar que é para atualizar o saldo disponível no QR *code*. Caso não seja possível iniciar o sistema por algum motivo, o usuário será informado. Caso os dados para entrar no sistema estejam corretos, o usuário será direcionado diretamente para a tela inicial do sistema.

Caso de teste 4 (CT 4): Ao clicar na opção de compra, o usuário será direcionado para a área correspondente, onde encontrará valores pré-definidos para realizar a recarga de seu QR *code*. Esta parte é simples e ágil, onde o comprador pode apenas selecionar o valor e clicar no botão comprar, cujo valor será debitado diretamente da carteira do Google *play*.

Caso de teste 5 (CT 5): Selecionando-se a opção de histórico de compras, aparecerá um botão de sincronizar, que, ao ser clicado atualizará e mostrará as compras realizadas, contendo data, hora e valor.

Caso de teste 6 (CT 6): O histórico de utilização mostra as informações registradas no banco de dados, com data, hora e valor de cada vez que o usuário utilizou o QR *code*.

Caso de teste 7 (CT 7): Ao entrar na opção de configurações, aparecerão três botões. O primeiro deles é para alterar senha que, estando conectado a uma sessão e selecionando-se o mesmo, será possível alterar a senha instantaneamente digitando duas vezes para confirmação. Os campos de digitação de nova senha contêm máscara e validador. Caso haja algum erro aparece uma mensagem informando o motivo do problema.

Caso de teste 8 (CT 8): Caso a segunda opção seja escolhida (alterar informações do usuário), vários campos de informações serão carregados em branco como nome, data de nascimento e etc. Para realizar uma alteração no seu cadastro basta digitar o que se deseja que substitua na informação anterior e clicar em alterar conta. Caso não queira alterar uma determinada informação, basta deixar o campo em branco. Ao realizar a tentativa de alterar alguma informação, o sistema entrará no banco de dados para salvar as novas informações, mas para isto, tudo que o usuário digitou necessita estar correto. Para isto existem máscaras que farão o serviço de garantir que os dados foram digitados corretamente. Caso exista algum erro, o usuário será informado com uma mensagem de erro.

Caso de teste 9 (CT 9): Ao sair da conta, o usuário será direcionado à tela de *login*, onde poderá simplesmente fechar o aplicativo, realizar um novo cadastro ou entrar com uma conta diferente.

6 CONSIDERAÇÕES FINAIS

Este trabalho teve como objetivo final o desenvolvimento de um aplicativo *Android* que possibilitasse que quaisquer pessoas com um *smartphone* pudessem se cadastrar e fazer de seu aparelho um cartão de passe único comum.

O aplicativo foi concluído com êxito sendo capaz de realizar ações, como cadastro de novos usuários, realizar compras, acompanhar o histórico de compra e utilização, além de poder alterar algumas informações básicas e recuperar a senha, caso o usuário tenha algum problema com ela. Todo aparelho *Android*, a partir da versão 4.2, tem condições de rodar o aplicativo.

A metodologia *Scrum* possibilitou a validação do desenvolvimento do projeto. Ao final de cada *Sprint* houve um *feedback* das atividades realizadas. Esse processo foi de extrema importância para identificar os requisitos que agregariam mais valor ao aplicativo. Ao perceber as prioridades, também foi possível definir melhorias futuras que não foram realizadas nesse trabalho.

Devido ao limitado tempo de desenvolvimento, algumas melhorias pensadas não puderam ser aplicadas, como por exemplo, o desenvolvimento do aplicativo em outras plataformas, embora a grande maioria da população utilize *Android* no *smartphone*, também há uma parcela que utilizam *IOS*.

A parte de segurança também não foi explorada no desenvolvimento, para tornar o *QR code* mais seguro. Uma opção é criptografar o processo de codificação e decodificação do *QR code*. Outra medida de segurança importante é a criptografia da senha.

REFERÊNCIAS

BEIGHLEY, L; MORRISON, M. **Use a cabeça! PHP e MySQL**. Marcelo Santos, Rio de Janeiro: Alta Books, 2010. 770p.

BOOCH, G; RUMBAUGH, J; JACOBSON, I. **UML**, Guia do Usuário. Fábio Freitas da Silva, Rio de Janeiro: Campus , 2005. 238p.

PRESSMAN, Roger S. **Software Engineering: a Practitioner's Approach**. 6. ed. McGraw-Hill, 2005

SETZER, V; SILVA, F. **Banco de dados**, Aprenda o que são melhores conhecimentos e construa o seu. São Paulo: Edgard Blucher, 2005. 380p.

SIERRA, K; BATES B. **Use a cabeça! Java**. Aldir José Coelho, Rio de Janeiro: Alta Books, 2005. 664p.

SOMMERVILLE, I. **Engenharia de software**. 9ª edição. São Paulo: Pearson Education, 2011. 548p.

SHUTHERLAND, J. **Scrum**, a arte de fazer o dobro do trabalho na metade do tempo. Natalie Gerhardt. São Paulo: Leya, 2014. 158p.

APÊNDICE A – Script usado na criação das tabelas do banco de dados

```
create table usuario
(
  id int auto_increment primary key,
  nome varchar(80) not null,
  rg varchar(15) not null unique,
  cpf varchar(15) not null unique,
  sexo char(1) not null,
  data_nasc date not null,
  nome_mae varchar(80) not null,
  senha varchar(30) not null
);
create table cartao
(
  id int auto_increment primary key,
  numero_cartao varchar(20) not null unique,
  saldo decimal(10,2) not null,
  status char not null,
  id_usu int not null,
  foreign key (id_usu) references usuario(id)
);
create table utilizacao
(
  id int auto_increment primary key,
  valor decimal(10,2) not null,
  numero_onibus varchar(15) not null,
  data datetime not null,
  id_car int not null,
  foreign key (id_car) references cartao(id)
);
create table movimento
(
  id int auto_increment primary key,
  valor decimal(10,2) not null,
  data datetime not null,
  id_car int not null,
  foreign key (id_car) references cartao(id)
);
delimiter |
create trigger SomaSaldo after insert
on saldo
```

```
for each row
begin
    update cartao set saldo=saldo+new.valor where cartao.id=new.id_car;
end;
|
delimiter ;SomaSaldo
```

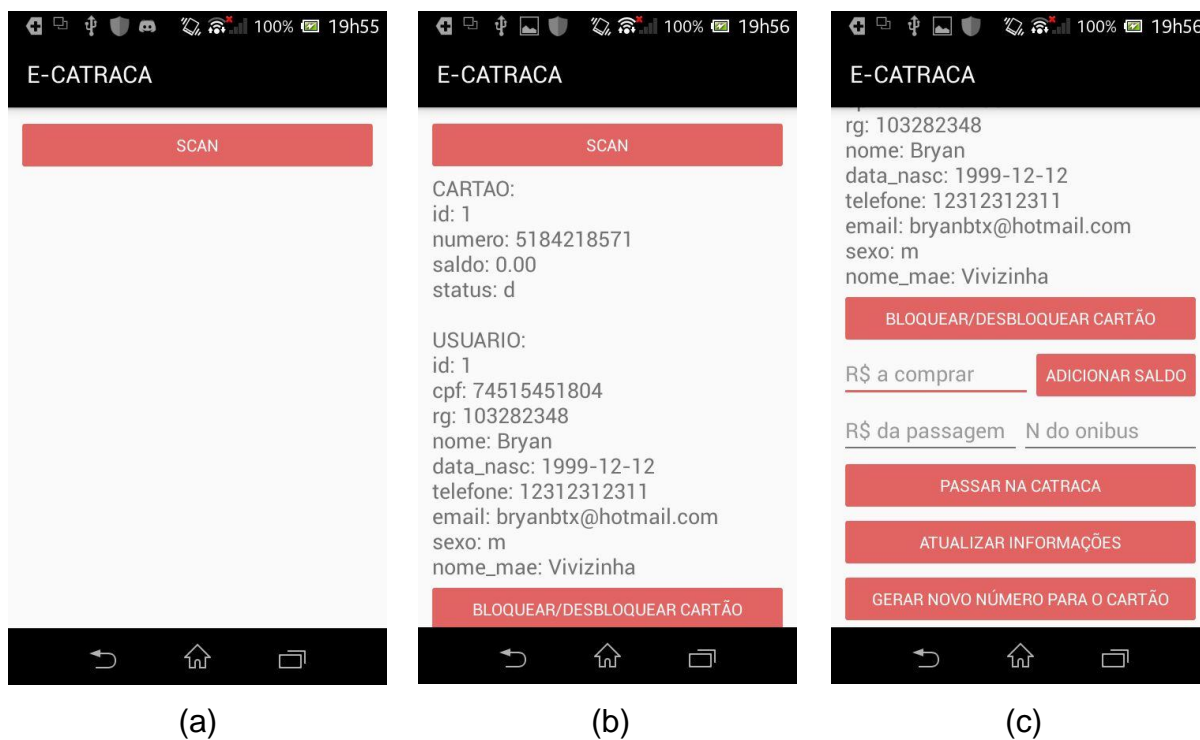
APÊNDICE B – Simulador catraca

Para simular a comunicação com a catraca e as ações que seriam executadas pela empresa de transporte público, foi desenvolvido um aplicativo com as seguintes funcionalidades:

- **Scan:** Abre a câmera do celular para ler o QR code do aplicativo, após o que, carrega todas as informações do usuário;
- **Bloquear/desbloquear cartão:** Altera o *status* do cartão, de desbloqueado para bloqueado e vice-versa;
- **Adicionar saldo:** Adiciona saldo ao cartão;
- **Passar na catraca:** Libera a catraca e subtrai o valor da passagem do saldo;
- **Atualizar informações:** Atualiza as informações do usuário;
- **Gerar novo número para o cartão:** Gera um novo número para o cartão.

A Figura 26 apresenta a tela inicial do aplicativo. As Figuras 26.b e 26.c demonstram a tela com as informações de cada usuário e as ações que podem ser efetuadas com seu cartão.

Figura 29 - Tela simulador catraca



Fonte: Elaborado pelo autor (2018)