

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DO RIO
GRANDE DO NORTE

**CLIENTE HABITUAL: UM SOFTWARE PARA MELHORAR O CONTROLE E
GERENCIAMENTO DAS INADIMPLÊNCIAS NAS VENDAS A LONGO PRAZO EM
ESTABELECIMENTOS COMERCIAIS**

RAIMUNDO HENRIQUE DA SILVA NETO

LAJES/RN, 2020

RAIMUNDO HENRIQUE DA SILVA NETO

**CLIENTE HABITUAL:UM SOFTWARE PARA MELHORAR O CONTROLE E
GERENCIAMENTO DAS INADIMPLÊNCIAS NAS VENDAS A LONGO PRAZO EM
ESTABELECIMENTOS COMERCIAIS**

Trabalho de Conclusão de Curso apresentado ao Curso Técnico de Nível Médio em Informática do Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Norte, em cumprimento às exigências legais como requisito parcial à obtenção do título de Técnico de Nível Médio em Informática.

Orientador: Prof. Me. Dannylo Johnathan Bernardino Egídio

RESUMO

Uma parte significativa dos estabelecimentos negligenciam o gerenciamento e controle das inadimplências nas vendas a longo prazo, tornando desgastante o registro e cobrança das mesmas, principalmente por parte dos clientes, alguns desses que esquecem a data definida para prestação de contas. O software desenvolvido a partir deste projeto na plataforma Android para dispositivos mobile tem como objetivo amenizar esse problema trazendo melhora no gerenciamento, no controle e uma forma de cobrar a dívida por meio digital pela internet.

Palavras-chave: Software de controle. Controle de inadimplência. Software para controle de vendas.

ABSTRACT

A significant part of the establishments neglect the management and control of defaults in sales in the long term, making the registration and collection of them exhausting, mainly by customers, some of them who forget the defined date for accountability. The software developed from this project on the Android platform for mobile devices aims to alleviate this problem by improving management, control and a way to collect debt digitally over the internet.

Keywords: Control software. Default control. Sales control software.

AGRADECIMENTOS

Agradeço a Deus por ter dado saúde e força para superar as dificuldades encontradas ao longo deste projeto, a minha família pelo apoio e motivação nos estudos a esta instituição, ao professor orientador Dannylo Johnathan por sua ótima orientação mesmo com as adversidades enfrentadas e por fim aos meus amigos que também me incentivaram a desenvolver o sistema do projeto.

SUMÁRIO

1. INTRODUÇÃO.....	7
1.1 OBJETIVO GERAL.....	8
1.2 OBJETIVOS ESPECÍFICOS.....	8
1.3 JUSTIFICATIVA.....	8
2. REFERENCIAL TEÓRICO.....	9
2.1 CONTROLE DE INADIMPLÊNCIA.....	9
2.2 DESENVOLVIMENTO ANDROID COM JAVA.....	9
2.3 REQUISITOS NÃO FUNCIONAIS.....	10
2.4 DIAGRAMA DE CASOS DE USO.....	11
2.5 DIAGRAMA DE CLASSE.....	12
3. METODOLOGIA.....	14
4. APLICABILIDADE DA SOLUÇÃO.....	15
4.1 PROBLEMA.....	15
4.2 MODELAGEM.....	16
4.2.1 REQUISITOS FUNCIONAIS E NÃO FUNCIONAIS.....	16
4.2.2 DIAGRAMA DE CASO DE USO.....	17
4.2.3 DIAGRAMA DE CLASSES.....	18
4.2.4 DESENVOLVIMENTO.....	19
4.2.4.1 TELA INICIAL.....	19
4.2.4.2 TELA DE PRODUTOS.....	21
4.2.4.3 TELA INADIMPLÊNCIA.....	22
4.2.4.4 FUNÇÃO ENVIAR E-MAIL.....	24
5. CONSIDERAÇÕES FINAIS.....	26
6. REFERENCIAL BIBLIOGRÁFICO.....	27

1. Introdução

Um problema que foi negligenciado até a atualidade é a falta de controle e gerenciamento nas vendas a longo prazo vindo a gerar inadimplência por parte dos clientes, essa inadimplência gera um déficit no caixa do estabelecimento ou no bolso do usuário que não possui loja física.

Uma vez que levantado o problema sabe-se que esse tipo de inadimplência não pode ser totalmente extinto apenas oferecendo meios de cobrança pois não temos apenas o esquecimento da dívida por parte dos clientes.

Muitas das vezes a própria condição financeira do indivíduo que o impede de prestar contas mas para os que esquecem seu débito esse meio de cobrança pode reduzir o problema.

Atualmente não há ferramentas específicas na plataforma Android que sejam exclusivamente voltadas para esse controle de forma mais rigorosa ou que forneça meios para cobrança digitalmente.

A humanidade está evoluindo cada vez mais para um mundo digital, essa é uma necessidade vista após presenciar esse tipo de problema e seus impactos na economia.

A escolha de ser um aplicativo android foi tomada visando o gasto para manter o sistema ativo.

Na plataforma Android ele pode funcionar totalmente gratuito sem a necessidade hospedar o sistema em servidores e contratar serviços de *DNS(Domain Name System)* diferente de sistemas web, onde é essencial esses serviços.

1.1 Objetivo geral

O presente projeto tem como objetivo o desenvolvimento de um aplicativo Android para auxiliar o controle de inadimplência das vendas a prazo dentro dos estabelecimentos comerciais.

1.2 Objetivos Específicos

Tornar obsoleto o uso do caderno de notas no balcão dos estabelecimentos comerciais, mapear os requisitos necessários para desenvolver um aplicativo de controle de inadimplência, aplicá-lo em comércios e disponibilizar o software em plataformas para o seu Download.

1.3 Justificativa

Atualmente, em 2020, com a evolução das tecnologias e da programação de software que é uma das áreas que vem crescendo muito nos últimos anos com soluções para vários problemas, o controle e gerenciamento de inadimplências nas vendas a prazo em estabelecimentos comerciais é um área que tem um déficit de atenção resultando em: alta nos atrasos dos pagamentos, falta em meios de cobranças mais rápidas e dificuldade na procura e organização de um cliente em meio a um amontoado de clientes nos cadernos de notas tornando cansativo e demorado com o tempo.

Para amenizar esse problema o Cliente Habitual é uma opção, a criação desse software para fazer a substituição e modernização dos meios até hoje usados.

Para esse fim o aplicativo vai notificar o funcionário no caso da expiração na data de pagamento estabelecida para fazer a cobrança via e-mail do pagamento ao cliente, identificar mais rapidamente o cliente para verificar o débito, data estabelecida para pagamento e dar baixa total ou parcialmente na sua pendência.

Esse projeto me ajudou dando experiência em outra plataforma, ensinando a fazer o planejamento e mapeamento para criação de projetos passo-a-passo, melhorando o trabalho e mostrando novos horizontes com possibilidades para impulsionar minha carreira profissional como desenvolvedor não apenas se contentando com o nível técnico.

2. Referencial teórico

Nessa seção do documento será exposto o referencial teórico, onde serão apresentados o diagrama de caso de uso da *UML(Unified Modeling Language)*, requisitos funcionais e não funcionais de um sistema, desenvolvimento android em java e um pouco sobre os problemas do controle de inadimplência.

2.1 Controle de inadimplência

Um dos principais problemas enfrentados pelos pequenos comércios e empresas é o mau gerenciamento de inadimplências nas vendas a longo prazo, para diminuir esse problema devemos ficar atentos não só as inadimplências mas também aos clientes em quem confiamos nossas vendas para minimizar esse problema com os devedores duvidosos e os “esquecidos”.

Segundo Macário (2009, p 29):

Um dos fatos que diminuem o saldo de caixa é a Provisão para devedores duvidosos - Que é uma conta redutora do ativo e a princípio não afeta o caixa, porém a constatação de uma perda afetará o caixa, principalmente a perda com provisão para devedores duvidosos, a qual terá como contrapartidas, resultado e caixa.

Tendo assim esses devedores duvidosos sempre temos a possibilidade de ser gerada inadimplência.

2.2 Desenvolvimento android com java

O sistema Android em alta desde seu lançamento é um sistema aberto sendo assim qualquer empresa pode criar, modificar e comercializar seus próprios androids tornando-o assim uma plataforma bem mais acessível à comunidade que veio a provocar sua evolução com o passar dos anos. Essa plataforma foi escolhida pelo fato de podermos criar, testar e postar nossos aplicativos sem custo e mantê-lo circulando para o público gratuitamente.

Michael Burton e Donn Felker (2014, p 9) reforçam a ideia do por que desenvolver para android:

Se você quiser que seu aplicativo esteja disponível para milhões de usuários em todo o mundo, se deseja publicar aplicativos assim que terminar de

escrevê-los e testá-los, ou se quiser desenvolver em uma plataforma aberta, já tem a sua resposta.

Nessa plataforma temos uma liberdade maior para fazer, testar e postar nossas aplicações desenvolvidas independente do nível do aplicativo sendo mais básico como uma simples calculadora ou mais robusto como um sistema de E-commerce de uma empresa.

2.3 Requisitos não funcionais

A etapa de levantamento dos requisitos em projetos é uma das principais para garantir a qualidade do produto final, esses requisitos definem o que o sistema deve suprir para atender o público alvo para diminuir os problemas que aparecem no decorrer de um projeto. De acordo com Denise e Edson (2003):

A Engenharia de Requisitos tem se tornado cada vez mais necessária para resolver os problemas encontrados nas organizações com relação à definição de sistemas.

O requisitos não funcionais são aqueles que não são implementados pelos desenvolvedores, tendo a visão de Denise e Edson (2003) há vários tipos de requisitos não funcionais dentre eles podemos citar o de tempo de resposta, *throughput* e de temporização sendo que esses são focados no desempenho do sistema.

Por fim para evitar problemas no desenrolar do desenvolvimento por causa da negligência na etapa de requisitos devemos, segundo Denise e Edson (2003):

Primeiramente devemos entender muito bem o domínio da aplicação e a interface que fica entre o mundo real e o mundo dos sistemas.

Assim sendo, quanto mais entendermos essas duas partes para separar esses requisitos e chegar cada vez mais perto de produzir um sistema melhor com menos falhas.

2.4 Diagrama de casos de uso

Guedes (2011) afirma que dentre os diagramas da *UML(Unified Modeling Language)*, o diagrama de caso de uso é considerado o mais geral e informal. Este diagrama tem como objetivo demonstrar para os usuários o comportamento externo segundo a visão do usuário do software do sistema, mostrando as funções e serviços oferecidos e quais poderão ser utilizados pelos usuários em cada funcionalidade. É uma linguagem simples e de fácil compreensão para que os usuários possam ter uma ideia geral de como o sistema deve se comportar. Abaixo, segue uma breve explicação sobre os elementos utilizados em caso de uso no diagrama.

Atores: Com o papel do usuário que irá utilizar as funcionalidades e os serviços do sistema. Os atores são representados por desenhos de “bonecos de palito”, contendo o nome ou tipo do ator logo abaixo, pois o mesmo pode assumir vários papéis como o de gerente, funcionário, cliente ou até mesmo um sistema integrado (Guedes, 2011, p. 53)

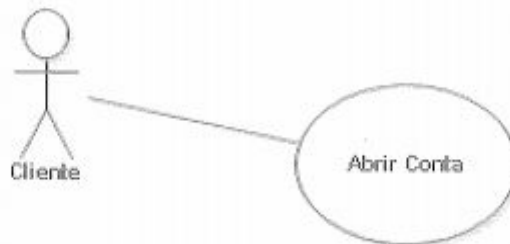


Figura 1 – Modelo de diagrama de caso de uso ação de um usuário

Associações: É representada por uma linha ligando o ator ao caso de uso. As linhas podem conter setas, que indicam o sentido em que as informações são transmitidas, além de poder conter descrição em forma de anotações para esclarecer a informação que está sendo passada (Guedes, 2011).

Generalização: Segundo Guedes (2011), essa é uma forma de associação que indica pequenas diferenças entre os casos de uso.

Costuma-se definir um caso de uso geral que descreve as características compartilhadas por todos os casos de uso em questão e então relacioná-lo com todos os casos de uso envolvidos.

A generalização é representada por uma linha com uma seta mais grossa, indicando qual o caso de uso geral para onde a seta aponta e quais os casos de uso especializados de onde saem as setas apontando para o caso de uso geral.

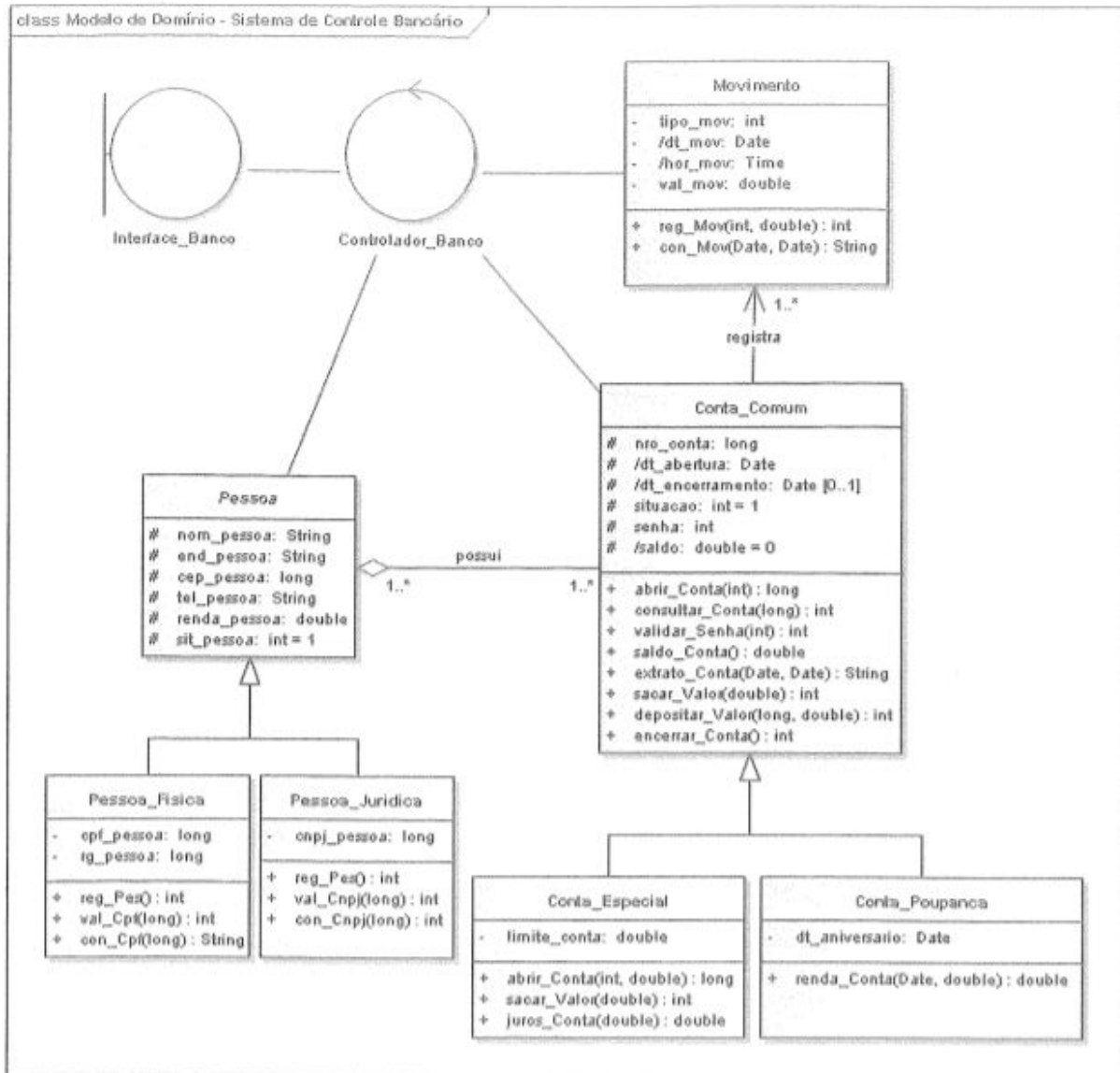
Inclusão: Quando um caso de uso está ligado a outro por meio de inclusão, a execução do primeiro acarreta na execução do segundo, afirma Guedes (2011). Esta associação é representada por uma linha tracejada com uma seta em uma de suas extremidades que aponta para o caso de uso incluído, ou seja quando o caso de uso “A” que inclui o caso de uso “B” for chamado, o caso de uso “B” também será executado.

Extensão: Esses são usados para descrever cenários opcionais de um caso de uso diferente da inclusão, ou seja, descrevem somente cenários que ocorrem em situações específicas, se uma determinada condição for satisfeita. Segundo Guedes (2011), relacionamentos de extensão representam funções que não ocorrem sempre, porém não são incomuns, sendo assim quando o caso de uso “A”, que estende o caso de uso “B” for usado, o caso de uso “B” poderá ser executado também se as condições forem satisfeitas.

2.5 Diagrama de classe

Diagramas de classes estão entre os mais úteis da *UML (Unified Modeling Language)* pois mapeiam de forma clara a estrutura de um determinado sistema ao modelar suas classes, seus atributos, operações e relações entre objetos. De acordo com Guedes (2011), o diagrama de classes é um dos mais importantes e utilizados da *UML (Unified Modeling Language)*, já que ele tem foco em apresentar as classes, os atributos, métodos, a cardinalidade e por fim o relacionamento entre esses objetos compondo a estrutura do sistema. De acordo com a imagem a seguir com um exemplo de diagrama de um sistema bancário:

Figura 2 -Exemplo de diagrama de classe



3. Metodologia

Nos últimos anos a tecnologia vem evoluindo bastante, trazendo cada vez mais aparelhos compactos com mais recursos como por exemplo o smartphone. Na atualidade esse aparelho já pode suprir sistemas para amenizar os problemas que surgem no cotidiano e esse é o aparelho que irá utilizar a aplicação desenvolvida em Java no sistema Android que é a plataforma da aplicação.

O Android é uma plataforma aberta ao público com uma comunidade muito presente na web. A escolha dessa plataforma foi devido a facilidade em poder criar, testar e publicar um aplicativo grátis em que não é necessário investimento por parte dos criadores e usuários no momento de disponibilizarmos o aplicativo para download e mantê-lo ativo nos dispositivos móveis dos usuários.

O Cliente Habitual(sistema proposto) traz um meio de auxiliar o controle e gerenciamento de inadimplências nas vendas a longo prazo para os estabelecimentos comerciais, porém o aplicativo não se limita apenas a estabelecimentos comerciais ele pode estar em prática para qualquer pessoa que precise anotar débitos pendentes com qualquer pessoa.

Nesta pesquisa exploratória para o projeto foi tomado como base o levantamento do problema na área onde será aplicado o aplicativo, como estabelecimentos comerciais, trabalhos exploratórios já elaborados, e para complementar pesquisas em livros e na internet para reforçar a necessidade desse aplicativo perante as inadimplências.

No comercial Victor Neto situado em Lajes/RN, foi feito o levantamento dos requisitos essenciais para o sistema, questionando o responsável do estabelecimento.

Os dados obtidos foram avaliados qualitativamente o número de clientes que frequentam o estabelecimento observado e geram essas inadimplências, bem como os resultados obtidos dos testes deste aplicativo em prática, para gerenciar e notificar os clientes.

4. Aplicabilidade da solução

Neste capítulo será apresentado a oportunidade observada para criar esse projeto, a modelagem onde explicamos como foi feita a base do software com o levantamento dos requisitos e diagramas e por fim no desenvolvimento que terá *screenshots* da tela do aplicativo em uso.

4.1 Problema

Quando frequentamos estabelecimentos comerciais podemos notar no balcão no mínimo um caderninho ou bloco de notas para anotar as vendas feitas a longo prazo no estabelecimento. No comercial Victor Neto situado em Lajes/RN onde há esses clientes o problema foi analisado no cotidiano com esse “caderninho” para anotar as vendas a prazo e foi feito o levantamento dos seguintes problemas:

1. Quando temos muitos clientes neste caderno o gerenciamento dos mesmos se torna demorado e cansativo.
2. O ambiente se torna deselegante pois é feito um amontoado de cadernos ou blocos de notas nos balcões.
3. Há clientes que esquecem a data do pagamento e outros que a situação financeira na atualidade não permite pagar.
4. Faltam meios para cobrar os clientes esquecidos e os duvidosos sobre sua dívida.

Na tabela a seguir temos a quantidade de clientes que frequentam o comercial Victor Neto para comprar a longo prazo e desses quantos esquecem a data ou não podem pagar por situação financeira.

Tabela 1: Clientes do comercial Victor Neto

Comercial Victor Neto	
Número de clientes que compram a longo prazo	16
Número de Clientes que geralmente esquecem a data	2 a 3

Fonte: Elaborado pelo autor

Analisando esses problemas chegamos a um consenso que é necessário melhorar o gerenciamento, controle de inadimplência nas vendas a longo prazo e

tornar obsoleto o uso do caderninho ou bloco de notas no balcão dos estabelecimentos comerciais.

4.2 Modelagem

Nesse tópico será apresentado as bases para o desenvolvimento do software, o levantamento dos requisitos funcionais e não funcionais, o diagrama de caso de uso e o diagrama de classes.

4.2.1 Requisitos funcionais e não funcionais

Inicialmente foram levantados os requisitos do sistema com a orientação do orientador separando-os em duas tabelas: os requisitos funcionais e os não funcionais, classificando-os por prioridade e atribuindo um código único de referência para cada.

A seguir serão apresentadas duas tabelas: a primeira contém os requisitos funcionais e a segunda contendo os requisitos não funcionais, ambos interferem no objetivo ou funcionamento do aplicativo de acordo com a prioridade estabelecida.

Tabela 02: requisitos funcionais.

Código	Requisito	Prioridade
RF01	Gerenciar Clientes	Média
RF02	Gerenciar Produtos	Baixa
RF03	Gerenciamento de Inadimplência	Alta
RF04	Enviar Email para o cliente da expiração na data definida para pagamento	Alta
RF05	Notificar usuário sobre expiração na data definida para pagamento	Média
RF06	Listagem de produtos que faltam na loja	Baixa

Fonte: Elaborado pelo autor

Tabela 03: requisitos não funcionais.

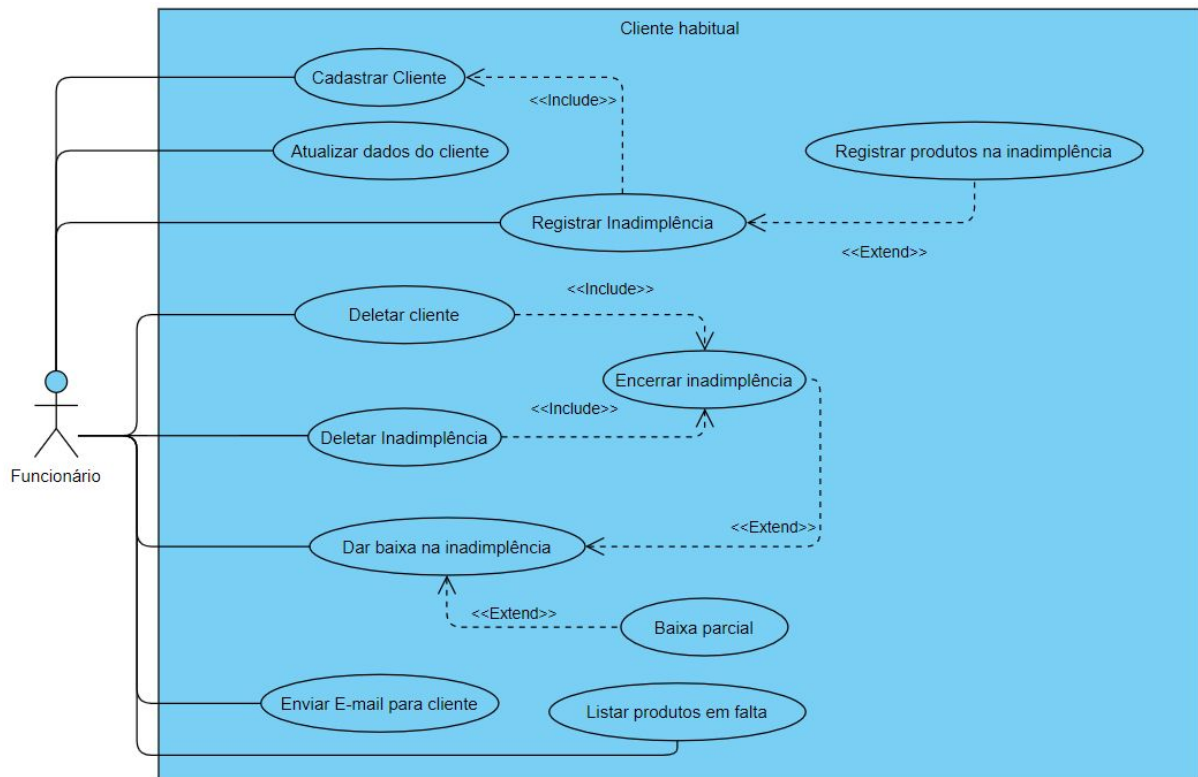
Código	Requisito	Prioridade
RNF01	O aplicativo funcionará na plataforma Android.	Alta
RNF02	Apresentar interface e funções auto-explicativas facilitando o manuseio para o usuário.	Média
RNF03	É necessário um aplicativo cliente de e-mail instalado no dispositivo para usar a função de E-mail.	Alta
RNF04	O aplicativo precisará de acesso a internet.	Alta

Fonte: Elaborado pelo autor

4.2.2 Diagrama de caso de uso

O Diagrama de caso de uso foi usado para representar os requisitos funcionais do aplicativo, o ator, as funções e as relações entre elementos utilizados da forma mais simples possível. A seguir na imagem 3 temos o diagrama do Cliente Habitual:

Figura 3: Caso de uso do Cliente Habitual.



Fonte: Elaborado pelo autor

No diagrama é apresentado de forma simples o funcionamento do aplicativo de acordo com os requisitos funcionais onde: para cadastrar uma nova inadimplência temos que obrigatoriamente cadastrar ou ter cadastrado um cliente e em seguida temos a opção de registrar os produtos. O funcionário pode atualizar dados do cliente livremente (Nome e E-mail), porém o cliente obrigatoriamente deve ter um nome.

Para deletar uma inadimplência ou cliente, temos que encerrar as inadimplências ativas. Quando for dar baixa no valor da inadimplência temos as opções de dar baixa parcial ou encerrar a mesma. O funcionário pode e deve enviar um e-mail lembrete quando a data de pagamento expirar para o cliente. E por último o funcionário ou dono do comércio pode fazer uma lista de produtos que faltam na loja (função ainda não implementada pois mesmo sem ela não afeta o funcionamento do sistema).

4.2.3 Diagrama de classes

Para descrever a estrutura do software foi usado o diagrama de classe apresentando as classes com seus atributos e métodos, as operações e as relações entre as classes. No diagrama temos os seguintes elementos:

Atributos: na parte do meio, sua visibilidade “-” pois são privados e classes externas não podem modificá-las diretamente no aplicativo.

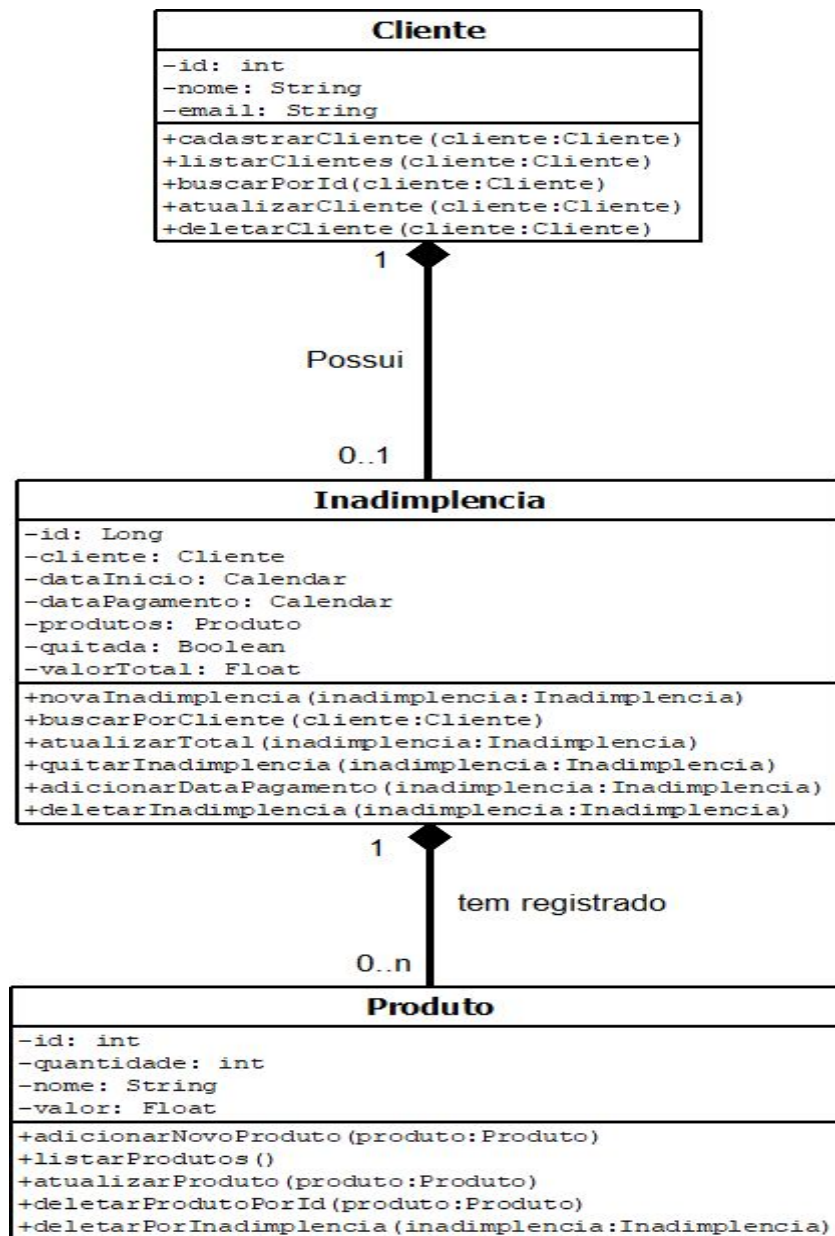
Métodos: presentes em baixo, no software são “+” que significam públicos, sendo assim podem ser usados por em classes externas .

Relacionamento: temos apenas composições pois uma inadiplência não deve ser mantida no aplicativo sem estar ligada a um cliente e produtos devem ser atribuídos a uma inadiplência.

Cardinalidade: um cliente pode ou não ter uma inadiplência (1 ... 0) porém a mesma não existe sem um cliente (1).No caso dos produtos um produto só pode estar ligado a uma inadiplência mas a mesma pode ter nenhum ou vários produtos registrados (N..0).

A seguir veja o diagrama de classe do aplicativo desenvolvido:

Imagem 04: Diagrama de classe cliente habitual



Fonte: Elaborado pelo autor

4.2.4 Desenvolvimento

Nesta etapa temos a apresentação do desenvolvimento do projeto com *screenshots* das telas e explicação do funcionamento do aplicativo com as respectivas funcionalidades descritas nas etapas anteriores implementadas.

Tela inicial

Na tela inicial temos um campo de texto para filtrar os clientes por nome, uma lista com os clientes cadastrados e um botão flutuante para cadastrar novas inadimplências. Dentre as situações dos clientes temos a previsão para pagamento

expirada. Cliente sem dívida ou seja inadimplência quitada. Quando o cliente não tem inadimplência ativa e por fim a data de pagamento não está expirada.

Figura 05:Tela inicial



Cliente	Divida	Pagamento
Cecilia	R\$ 57,00	05/11/2020
Francisco Junio	R\$ 90,00	30/10/2020
Fernanda		Sem dividas!
Claudia do n 37	R\$ 75,00	10/01/2021
Cláudio(acentam	R\$ 22,47	30/11/2020
Joao	R\$ 3,00	12/12/2020
Nando (fazenda)	R\$ 35,50	30/11/2020
Pedro		

Fonte: Elaborado pelo autor

Quando a data expira, os dados do cliente na lista ficam vermelhos a fim de dar um destaque para tal indivíduo. Quando sem dívida ou inadimplência quitada o nome do cliente fica verde e no final temos o texto “sem dívidas!”.

Quando o cliente não possui inadimplência, o nome do mesmo na lista permanece com cor preta. E no momento em que a data ainda está dentro do prazo estabelecido vemos o nome, valor e data para pagamento na lista permanecendo com a cor preta. Após clicarmos no botão para nova inadimplência será apresentado a seguinte tela para cadastrados clientes ou inadimplências:

Figura 06: cadastro de inadimplência

A imagem mostra uma interface de usuário com um formulário para o cadastro de uma nova inadimplência. O formulário é dividido em seções por meio de abas no topo: 'Cliente', 'Divida' e 'Pagamento'. O formulário em si contém o seguinte conteúdo:

- Um campo de texto com o nome 'Nando (fazenda)'.
- Um campo de e-mail com o endereço 'nando.fzd@gmail.com'.
- Uma opção de checkbox rotulada 'Registrar produtos', que está desmarcada.
- Dois campos de entrada: 'Previsão de pagamento' com a data '30/11/2020' e 'Valor' com o valor '\$ 35.50'.
- Um botão 'SALVAR' na parte inferior direita.

Fonte: Elaborado pelo autor

Por fim, nessa tela temos a opção de registrar os produtos na inadimplência que será explicado a seguir.

Tela de produtos

Quando clicamos em um cliente na lista inicial ou optamos por registrar produtos ao cadastrar um cliente somos direcionados para a tela de produtos onde temos as informações dos produtos cadastrados, do cliente e da inadimplência quando temos produtos cadastrados na mesma podemos adicionar, editar ou remover os produtos ótimo para fazer o registro dos itens vendidos, podendo modificar o valor total da dívida adicionando ou dando baixa na dívida. Conforme o mostrado na *ScreenShot* a seguir:

Figura 07: Tela produtos



Fonte: Elaborado pelo autor

Tela de inadimplência

Diferente da anterior, essa tela só aparece quando o cliente não tem produtos cadastrados na inadimplência, ou seja, apenas o valor e data para pagamento. Nela há somente a inadimplência que podemos gerenciar seu total e a data prevista para pagamento, caso o usuário queira registrar produtos é necessário escolher a opção disponível no menu flutuante presente na tela.

Figura 08: tela inadimplência



Fonte: Elaborado pelo autor

Função enviar e-mail

Por fim, quando temos clientes com pagamento expirado, o aplicativo disponibiliza a função de enviar *e-mail* para o mesmo. Para ser usada basta apenas clicar no indivíduo na lista da tela inicial do aplicativo cujo usuário esteja com data expirada o mesmo estará com os dados em cor vermelha na lista.

Figura 09: confirmação de e-mail



Fonte: Elaborado pelo autor

Usando aplicativos de terceiros no *android* basta escolhermos o serviço cliente de *e-mail* instalado no dispositivo.

Figura 10: escolher cliente de e-mail

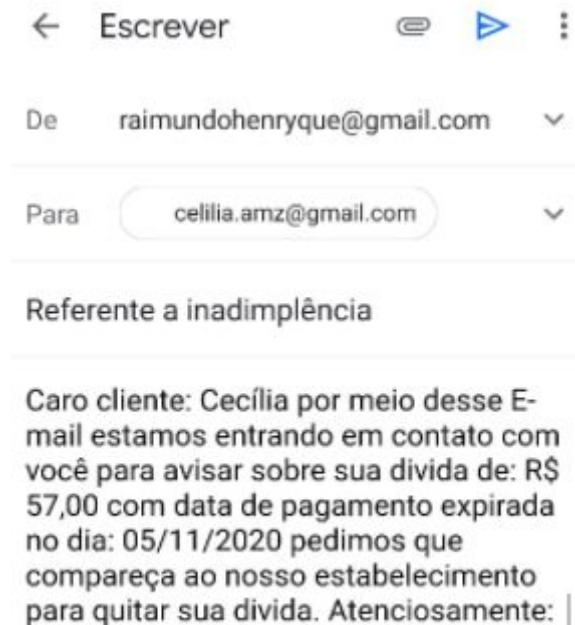


Fonte: Elaborado pelo autor

No caso de não haver *e-mail* do cliente cadastrado deve-se ser informado manualmente no momento, o corpo já vem com uma mensagem definida contendo

nome, data de expiração e valor da dívida porém detalhes adicionais podem ser adicionados a gosto como mostrado na imagem a seguir.

Figura 11: E-mail completo



Fonte: Elaborado pelo autor

Por fim, esse é o Cliente Habitual, o aplicativo que foi desenvolvido para atingir os objetivos definidos neste projeto. O código fonte do Cliente Habitual está disponível no GitHub: <https://github.com/Bill-tec/cliente-habitual>.

5. Considerações Finais

O Cliente Habitual(aplicativo proposto) foi desenvolvido para auxiliar no controle e gerenciamento das inadimplências nas vendas a longo prazo.O aplicativo cumpriu o objetivo a qual foi destinado.No estabelecimento comercial Victor Neto que foi o local de testes, o bloco de notas do balcão foi substituído pelo aplicativo agilizando assim o gerenciamento das vendas e dos clientes, removendo o amontoado de blocos de nota e melhorando o ambiente.

Os clientes com data expirada e e-mail válido foram notificados vindo a comparecer no estabelecimento para quitar a dívida, já os que não tinham condições no momento forneceram uma nova data prevista para pagamento, cumprindo assim o objetivo principal definido para o projeto e o sistema que é auxiliar no controle e gerenciamento das inadimplências.

Em plena pandemia do Covid-19, a quarentena limitou as etapas de levantamento de dados onde era necessário pesquisas e testes em outros estabelecimentos para observar os resultados do sistema em atividade, pesquisas essas para encontrar *bugs* ou possíveis mudanças que poderiam ser implementadas para melhorar a experiência do usuário no Cliente Habitual.

Visando trabalhos futuros o projeto será reformado ganhando uma versão web integrando meios de pagamento online(boleto, *PayPal*, *PicPay* entre outros disponíveis), chat entre cliente e estabelecimento que possivelmente fornece entrega de produtos por parte do vendedor, banco de dados remoto, autenticação de usuário criando uma versão de *E-commerce* segura e mais funcionalidades pensadas para melhorar a experiência dos usuários similar a sistemas populares como o Mercado Livre e a Shopee.

Na nova versão do Cliente Habitual poderá ser reaproveitado e reformados de acordo com um sistema *web* os diagramas apresentados, as tabelas de requisitos e a estrutura do projeto como as classes de modelo(objetos).

Por fim, até esse momento, por problemas financeiros e a alta do dólar, infelizmente não foi possível adquirir uma conta do *Google Play* com finalidade de disponibilizar o aplicativo para download na *Google Play Store*, mas futuramente o sistema será disponibilizado.

Referencial bibliográfico

Macário, Rodolfo Augusto Horácio. **"A importância da gestão do fluxo de caixa no controle da inadimplência."** (2009). Acesso em: 24 de Outubro de 2020.

BURTON, Michael, FELKER, Donn. **Desenvolvimento de Aplicativos Android para Leigos**. Alta Books Editora, 2014. Acesso em: 27 de Outubro de 2020.

GUEDES, Gilleanes Thorwald Araujo. **"UML 2 Uma Abordagem Prática: 2ª Edição"**. São Paulo: Novatec, 2011. acesso em: 33 de Outubro de 2020.