

**INSTITUTO FEDERAL DE MINAS GERAIS
CAMPUS SÃO JOÃO EVANGELISTA**

**JOHN PAUL SILVA LEITE; MARCELO FERREIRA DE ALMEIDA;
THIAGO DA SILVA BELIZÁRIO**

**PROPOSTA DE DESENVOLVIMENTO DE SOFTWARE ERP PARA A GESTÃO
DIRECIONADA PARA MICRO E PEQUENAS EMPRESAS COM ÊNFASE NO
MÓDULO FINANCEIRO**

**SÃO JOÃO EVANGELISTA
2014**

**JOHN PAUL SILVA LEITE; MARCELO FERREIRA DE ALMEIDA;
THIAGO DA SILVA BELIZÁRIO**

**PROPOSTA DE DESENVOLVIMENTO DE SOFTWARE ERP PARA A GESTÃO
DIRECIONADA PARA MICRO E PEQUENAS EMPRESAS COM ÊNFASE NO
MÓDULO FINANCEIRO**

Trabalho de conclusão de curso apresentado ao Instituto Federal de Minas Gerais – Campus São João Evangelista, como exigência parcial para obtenção do título de Bacharel em Sistemas de Informação.

Orientador: Edson Batista Sena
Coorientador: Esp. Bruno de Souza Toledo

**SÃO JOÃO EVANGELISTA
2014**

Proposta de desenvolvimento de software ERP para a gestão direcionada para micro e pequenas empresas com ênfase no módulo financeiro

Trabalho de conclusão de curso apresentado ao Instituto Federal de Minas Gerais – Campus São João Evangelista, como exigência parcial para obtenção do título de Bacharel em Sistemas de Informação.

Aprovado em / /

BANCA EXAMINADORA



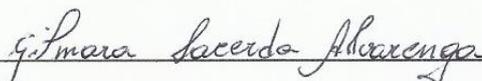
Orientador Prof. Edson Batista Sena

Instituição: Instituto Federal de Minas Gerais – Campus São João Evangelista



Coorientador Prof. Esp. Bruno de Souza Toledo

Instituição: Instituto Federal de Minas Gerais – Campus São João Evangelista



Prof. Gilmar Lacerda Alvarenga (Convidado) – IFMG-SJE

Instituição: Instituto Federal de Minas Gerais – Campus São João Evangelista

RESUMO

A gestão organizacional é um ponto fundamental para todas as empresas que deseja ganhar vantagens competitivas em relação as suas concorrentes, assim se faz necessário à informatização das mesmas, possibilitando a implantação de sistemas especializados para gerir os dados e transforma-los em informações auxiliando o gestor na tomada de decisão. Muitas das micro e pequenas empresas não apresentam nenhum meio informatizado de gestão da informação, devido ao seu alto custo e funcionalidades desnecessárias. Após a constatação desse problema foi criado o módulo financeiro de um sistema ERP para micro e pequenas empresas, com a finalidade de auxilia-las na gestão organizacional. Nesse trabalho foram realizadas pesquisas bibliométricas a respeito de projetos e artigos relacionados a sistemas ERP em micro e pequenas empresas e sobre métodos e práticas de desenvolvimento de *software*. Esse sistema auxilia o setor financeiro, proporcionando informações mais precisas, confiáveis e com maior rapidez, ajudando assim o gestor na tomada de decisão.

Palavras-chave: ERP. Modo financeiro. Micro e pequenas empresas.

ABSTRACT

The organizational management is a key issue for all companies that want to gain competitive advantages over its competitors, this way computerization of the same is necessary, enabling the deployment of specialized systems to manage data and transform it into information assisting the manager in decision-making. Many of micro and small enterprises do not have any means of computerized information management, due to its high cost and unnecessary features. After finding, this problem a financial module of an ERP system was created for micro and small enterprises, in order to assist them in organizational management. This study conducted bibliometric research on projects and articles related to ERP systems in micro and small enterprises and on methods and software development practices. This system helps the financial sector, providing more accurate, reliable and faster, helping the manager in decision-making.

Keywords: ERP, Financial mode, Micro and small enterprises.

LISTA DE FIGURAS

Figura 1 – Modelo cascata	20
Figura 2 – Desenvolvimento evolucionário	20
Figura 3 – Engenharia de <i>software</i> baseada em componente	21
Figura 4 – Ciclo de vida do <i>software</i>	22
Figura 5 – Controle de clientes da empresa entrevistada	29
Figura 6 – Controle da gestão organizacional da empresa entrevistada	30
Figura 7 – Diagrama de caso de uso de clientes	32
Figura 8 – Diagrama de caso de uso de caixa	37
Figura 9 – Diagrama de caso de uso de depósito bancário	42
Figura 10 – Diagrama de caso de uso de diária do caixa	47
Figura 11 – Diagrama de caso de uso de fornecedor	52
Figura 12 – Diagrama de caso de uso de contas pagas	57
Figura 13 – Diagrama de caso de uso de contas recebidas	62
Figura 14 – Diagrama de caso de uso de tipos	67
Figura 15 – Diagrama de classes do sistema	72
Figura 16 – Tela de cadastro de cliente	73
Figura 17 – Tela de cadastro de cliente	75
Figura 18 – Tela de consulta de clientes	76
Figura 19 – Tela de cadastro de caixa	77
Figura 20 – Tela de cadastro de contas a pagar	78
Figura 21 – Tela de relatórios	79
Figura 22 – Modelo entidade relacional	81

LISTA DE QUADROS

Quadro 1 – Cadastro de cliente	33
Quadro 2 – Exclusão de clientes	34
Quadro 3 – Consultar cliente	35
Quadro 4 – Atualizar cliente	36
Quadro 5 – Cadastro de caixa	38
Quadro 6 – Excluir caixa	39
Quadro 7 – Consultar caixa	40
Quadro 8 – Atualizar caixa	41
Quadro 9 – Cadastro depósito bancário	43
Quadro 10 – Excluir depósito bancário	44
Quadro 11 – Consultar depósito bancário	45
Quadro 12 – Atualizar depósito bancário	46
Quadro 13 – Cadastro de diária de caixa	48
Quadro 14 – Excluir diária de caixa	49
Quadro 15 – Consultar diária de caixa	50
Quadro 16 – Atualizar diária de caixa	51
Quadro 17 – Cadastro fornecedor	53
Quadro 18 – Excluir fornecedor	54
Quadro 19 – Consultar fornecedor	55
Quadro 20 – Atualizar fornecedor	56
Quadro 21 – Cadastrar pagas	58
Quadro 22 – Excluir pagas	59
Quadro 23 – Consulta pagas	60
Quadro 24 – Atualizar pagas	61
Quadro 25 – Cadastrar recebidas	63
Quadro 26 – Excluir recebidas	64
Quadro 27 – Consultar recebidas	65
Quadro 28 – Atualizar recebidas	66
Quadro 29 – Cadastrar tipo	68
Quadro 30 – Excluir tipo	69
Quadro 31 – Consultar tipos	70

LISTA DE SIGLAS

C# – C Sharp

CASE – *Computer-Aided Software Engineering*

CRUD – *Create Read Update e Delete*

ERP – *Enterprise Resource Planning*

LTDA – Limitada

RF – Requisitos funcionais

RNF – Requisitos não funcionais

MER – Modelo Entidade Relacionamento

SGBD – Sistemas Gerenciadores de Banco de Dados

UML – *Unified Modeling Language*

WEB – *World Wide Web*

SUMÁRIO

1	INTRODUÇÃO.....	12
1.1	OBJETIVOS	13
1.1.1	Objetivo geral.....	13
1.1.2	Objetivos específicos.....	13
1.2	MOTIVAÇÃO.....	13
1.3	PROBLEMA.....	14
1.4	SOLUÇÃO PROPOSTA	14
2	REFERENCIAL TEÓRICO	15
2.1	SISTEMA DE INFORMAÇÃO	15
2.2	SISTEMA ERP	16
2.3	ENGENHARIA DE SOFTWARE	17
2.3.1	Processo de Software	18
2.3.2	Modelos de Processos de Software	19
2.3.3	Ciclo de Vida do Software.....	21
2.3.4	Ciclo de Vida de um Sistema ERP	22
2.4	REQUISITOS DE SOFTWARE	23
2.5	REQUISITOS FUNCIONAIS E NÃO FUNCIONAIS	23
2.6	TRABALHOS CORRELACIONADOS	24
3	METODOLOGIA.....	25
4	ANÁLISE E PROJETO DO SISTEMA	27
4.1	LEVANTAMENTO DE REQUISITOS.....	27
4.2	ANÁLISE DOS REQUISITOS	28
4.3	DIAGRAMAS UML	31
4.3.1	Definição de diagrama de caso de uso.....	31
4.3.2	Definição de diagrama de classes	31
4.3.3	Definição de diagrama de modelo entidade relacionamento (MER)	32
4.3.4	Diagramas de caso de uso.....	32
4.3.5	Diagrama de classes	72

4.4	PROTOTIPAÇÃO	73
4.5	MODELAGEM DE DADOS	80
5	RESULTADOS OBTIDOS	82
6	CONCLUSÃO	83
7	TRABALHOS FUTUROS	83
	REFERÊNCIAS	85

1 INTRODUÇÃO

A maioria das micro e pequenas empresas originam-se de um negócio informal, geralmente ocorrido de um indivíduo que se empenhou bastante e adquiriu confiança para competir no mercado. Diante desse contexto, o micro e pequeno empreendedor eventualmente precisará fazer um controle administrativo e financeiro dos negócios, assim se faz necessário à utilização de recursos manuais para o controle da empresa, como livros de anotações, ou o uso de planilhas eletrônicas.

Ambos os métodos que os micro e pequenos empreendedores usualmente adotam para a gestão organizacional de suas empresas, funcionam inicialmente, entretanto conforme o negócio cresce, esses métodos mostram ser ineficientes e obsoletos, devido ao aumento do número de funcionários e de operações, conseqüentemente irão acontecer diferenças nos dados financeiros, produção, vendas, entre outros.

Diante desse cenário, as micro e pequenas empresas vêm procurando ferramentas para auxiliá-las em suas gestões, visando ganhar vantagens competitivas no mercado.

O Sistema Integrado de Gestão (ERP) possui funções, como por exemplo, a estruturação dos custos de uma empresa que são importantes para a organização de integração dos métodos de trabalho para melhoria no processo de gestão da empresa. Segundo Guedes (2006), um sistema ERP (*Enterprise Resource Planning*) baseia-se em funções que atendem a um ou mais departamentos.

Os *softwares* ERP, possibilitam o suporte para a tomada de decisão e estão cada vez mais frequentes nas micro e pequenas empresas. De acordo com Oliveira (2010), o crescimento das micro e pequenas empresas, as forçam a investir em ferramentas integradas de auxílio à gestão para solucionar os problemas gerenciais decorrente de sua expansão.

Os *softwares* de gestão de apoio às decisões são ferramentas frequentemente utilizadas por grandes empresas, pois há necessidade de informar ao gestor ou responsável, o que ocorre na empresa em tempo real, ainda é necessário que os mesmos cheguem de maneira rápida e correta, tornando a empresa mais competitiva no mercado.

O sistema proposto neste trabalho tem como principal finalidade atender as micro e pequenas empresas que buscam melhorar sua estrutura organizacional e introduzir um método de trabalho mais eficaz.

O estudo para o levantamento de requisitos foi realizado na micro empresa, Supermercado Novo Cruzeiro, empresa do ramo varejista, localizada no município de Guanhães - MG, para entender e demonstrar quais as principais dificuldades nos processos de gestão organizacional, principalmente se tratando do controle financeiro. Este projeto apresenta como proposta, o desenvolvimento do módulo financeiro de um *software* ERP de gestão organizacional direcionado para micro e pequenas empresas.

1.1 OBJETIVOS

O objetivo deste trabalho é propor o desenvolvimento de uma ferramenta de *software* ERP para micro e pequenas empresas, possibilitando-as gerenciar os seus processos produtivos, de maneira que permita melhorias em suas estruturas organizacionais, consequentemente deixando-as mais flexíveis e competitivas no mercado atual.

1.1.1 Objetivo geral

O objetivo geral deste projeto é o desenvolvimento de um sistema ERP voltado para o módulo financeiro, focando a gestão de micro e pequenas empresas.

1.1.2 Objetivos específicos

- a) Fornecer informações gerenciais mais precisas e confiáveis.
- b) Garantir a segurança das informações.
- c) Otimizar os custos da empresa e aumentar a velocidade dos registros.

1.2 MOTIVAÇÃO

É viável a utilização de um *software* ERP de gerenciamento em todas as empresas, entretanto a proposta aqui realizada foi direcionada para micro e pequenas empresas, nas quais a utilização de registros manuais dos dados referentes a ela é utilizada como único

recurso para a qualidade da gerência. O *software* proposto poderá fornecer ao micro e pequeno empreendedor mais facilidade em extrair relatórios, reduzir custos, melhorar eficiência e até mesmo o controle de todos os gastos da empresa, com maior confiabilidade e agilidade.

1.3 PROBLEMA

Com o crescimento do mercado atual, as empresas possuem um grande número de informações, em que se faz necessário o gerenciamento das mesmas. A partir deste âmbito, as micro e pequenas empresas, buscam ferramentas para auxiliá-las na gestão, entretanto, tais ferramentas são de custo elevado e podem possuir funcionalidades desnecessárias, ou seja, módulos que são direcionados para grandes empresas.

Além dos problemas citados, as micro e pequenas empresas ainda enfrentam outras dificuldades, como por exemplo; a falta de informatização dos dados, pois a maioria das empresas desse porte não utilizam ferramentas ou *softwares* que possibilite o gerenciamento das informações. A falta de padronização de dados e a ausência de um sistema de ERP de gestão organizacional deixam as micro e pequenas empresas sujeitas a vários riscos, como perda de informações, e conseqüentemente prejuízo deixando-as menos competitivas no mercado.

1.4 SOLUÇÃO PROPOSTA

A proposta deste projeto é o desenvolvimento de um sistema ERP de gestão para micro e pequenas empresas, no qual possui a finalidade de apoio à decisão para o gestor, além da substituição dos registros manuais dos dados das empresas por uma ferramenta automatizada, assim integrando e otimizando todos os seus setores. Salientando que, foi contemplado ao fim desse projeto, apenas o módulo financeiro de acordo com os requisitos analisados.

2 REFERENCIAL TEÓRICO

Este capítulo descreve as definições, conceitos, histórico, além de vantagens e desvantagens de sistema de informação e *Softwares* ERPs, entre outros tópicos relacionados à Engenharia de Software, modelos de processos de software. Foi feito o estudo bibliográfico por meio de livros e artigos científicos de estudiosos da área de sistemas ERP.

2.1 SISTEMA DE INFORMAÇÃO

Sistema de Informação pode ser definido como um conjunto de componentes inter-relacionados que são coletados e armazenados com informações destinadas para a tomada de decisão, para o controle e a manutenção de uma empresa. Além de fornecer dados para tomada de decisão; auxilia os gestores a analisar os problemas e criar novos produtos (LAUDON e LAUDON, 2011).

Sistema de informação é composto por várias formas de organização, desde o chão de fábrica até o nível estratégico. Essas informações compreendem todos os dados úteis das pessoas e principalmente da empresa, como exemplo, o fluxo de caixa de uma pequena empresa que processa milhões de dados de pessoas e custos de produtos (LAUDON e LAUDON, 2011).

A função de um sistema de informação é estruturar as informações em dados relevantes para a empresa, bem como a tomada de decisão para a sua melhoria e o seu desempenho da empresa, sendo administrado pelo gestor que verifica quais decisões tomar para melhoria e solucionar um problema que esteja sofrendo (LAUDON e LAUDON, 2011). Não só a tecnologia é importante para as empresas, mas também pessoas qualificadas para desempenhar os trabalhos dentro das mesmas, pois não basta possuir um sistema bem estruturado e não ter pessoas qualificadas para a sua utilização.

Os sistemas de informação são compostos por dimensões que auxiliam na estrutura do sistema que, de acordo com Laudon e Laudon (2011) são:

- Organizações – partes fundamentais do sistema, a partir delas são estruturados sistemas próprios para as melhorias e organização das empresas onde são feitos todos os serviços prestados.

- Pessoas – uma empresa é tão boa quanto às pessoas que a formam, para ter um bom sistema e uma boa organização as pessoas são fundamentais para essa meta. A qualificação das mesmas serve como apoio na tomada de decisão e posteriormente no sucesso da empresa.
- Tecnologia – a tecnologia da informação é uma das ferramentas que os gerentes utilizam para enfrentar as mudanças. O conjunto de *hardware* que é a parte física usada para a entrada e o *software* que são instruções detalhadas para coordenar os componentes do hardware de um sistema de informação.

As três restrições compõem um sistema de informação, sem elas o sistema é inútil, pois uma depende da outra e assim o sucesso é garantido para o crescimento da empresa. As ferramentas são fundamentais para o desempenho das empresas, visando sempre crescer e ampliar seu mercado para futuras competições entre as adversárias (LAUDON e LAUDON, 2011).

Segundo Laudon e Laudon (2011), não é apenas a tecnologia que compõem um sistema de informação, mas também é composto por organizações e pelos papéis das pessoas que são fundamentais para o sucesso do sistema.

Cada empresa possui seu sistema de trabalho, com o qual busca ser competitiva, mas para ter qualidade em um sistema é necessário possuir pessoas qualificadas e tecnologia adequada para o desenvolvimento dos seus trabalhos com competência e eficácia (LAUDON e LAUDON, 2011).

2.2 SISTEMA ERP

O Sistema Integrado de Gestão Empresarial (ou ERP) integra todos ou a maioria dos setores de uma empresa com a finalidade de estruturar e organizar os métodos de trabalho, na busca de agilidade e eficiência no serviço (SOMMERVILLE, 2010).

De acordo com Souza e Saccol (2009), ERP (*Enterprise Resource Planning*) é definido como um sistema de informação integrado, adquiridos através de pacotes comerciais, a fim de lidar com todas as operações da empresa. Assim, o ERP auxilia a empresa usando o processo organizacional, sendo o mesmo, composto por módulos com a finalidade de apoiar a tomada de decisão, tornando a empresa mais flexível e competitiva no mercado.

De acordo com Guedes (2006), os sistemas ERP baseiam-se em grupos de componentes de *software* completamente integrados que agrupam funções que atendem a um ou mais departamento. Para Souza (2000), ERP é definido como um sistema de informação integrado que possui a função de auxiliar todas ou maior parte das operações de uma empresa.

Os sistemas ERP são baseados em vários subsistemas (módulos), que são divisões de um sistema que integram todos os diferentes setores da empresa, desenvolvidos de acordo com modelos de processos, no qual tem grande extensão operacional. Os principais processos de negócios em que o ERP está ligado nas empresas são: compras, contabilidade, financeiro, vendas, controle de produção, recursos humanos, qualidade de serviço, entre outros (GUEDES, 2006).

Os conceitos propostos sobre os sistemas ERP geralmente são utilizados por empresas que querem desenvolver seus aplicativos internamente, entretanto o único conceito para esse sistema é ligado ao fato de serem pacotes comprados. Segundo Souza (2000), os ERPs possuem características que o diferem de outros tipos de pacote e de sistemas feitos internamente em suas empresas, tais características são de suma importância para analisar os aspectos de sucesso na sua implantação e os benefícios da sua utilização.

Segundo Lovato (2009), o sistema ERP possibilita para as empresas que decidem adotá-lo, maior confiabilidade das informações que estão sendo trabalhadas e disponibilizadas aos gestores em tempo real, além de diminuir o trabalho repetitivo feito nos registros manuais. Mas deve-se salientar que essas características apenas podem ser alcançadas através do comprometimento dos funcionários, que são responsáveis em alimentar o sistema com informações.

2.3 ENGENHARIA DE SOFTWARE

A engenharia de *software* contribui para a construção de *softwares* qualificados e com maior rapidez, economizando assim tempo e custo. Algo que nem sempre foi assim, anteriormente o setor de *software* passava por diversos problemas, não estavam preparados para a construção de programas complexos e a demanda por eles só aumentava, o que ocasionou uma grande crise, devido ao não cumprimento dos prazos estabelecidos para as entregas e a má qualidade do mesmo, assim para solucionar esses problemas surgiu o que hoje conhecemos como engenharia de *software* (OLIVEIRA, 2009).

Segundo Sommerville (2010), engenharia de *software* é uma disciplina que envolve todo o processo de desenvolvimento de um *software*, desde sua criação até sua fase de manutenção. Não envolvendo apenas os processos de desenvolvimento e sim todos os aspectos que estão envolvidos no *software*, como os métodos de gerenciamento do projeto, ferramentas que serão utilizadas, procedimentos e teorias que auxiliam todos os aspectos de produção de *software*.

O principal objetivo da engenharia de *software* é construir *softwares* qualificados dentro dos prazos e custos que foram estabelecidos, utilizando métodos, ferramentas e procedimentos, que são elementos essenciais na criação de um sistema (SOMMERVILLE, 2010).

2.3.1 Processo de Software

Processo de *software* é um conjunto de atividades, métodos e procedimentos realizados para desenvolver um determinado *software*. Esse processo de desenvolvimento é complexo, para auxiliar o desenvolvimento, existem ferramentas de engenharia de *software*, como a CASE (*Computer-Aided Software Engineering*) cujo seu significado é Engenharia de *Software* Auxiliada por Computador, utilizado para dar apoio aos processos de desenvolvimento, análise de requisitos, projeto e teste (SOMMERVILLE, 2010).

Para desenvolver um *software* não há um processo ideal, devido a sua grande diversidade e necessidade de mudança, porém segundo Sommerville (2010), existem quatro atividades fundamentais em qualquer processo de desenvolvimento:

1. Especificação de *software*: define funcionalidade e restrições do sistema;
2. Projeto e implementação de *software*: período de desenvolvimento que atenda todas as especificações exigidas;
3. Validação de *software*: é necessário que o *software* seja validado demonstrando que ele faz aquilo que realmente o cliente especificou;
4. Evolução de *software*: o *software* deve evoluir para atender as mudanças e as necessidades do cliente.

2.3.2 Modelos de Processos de Software

Sommerville (2010) apresenta a seguinte definição para modelos de processos de *software*:

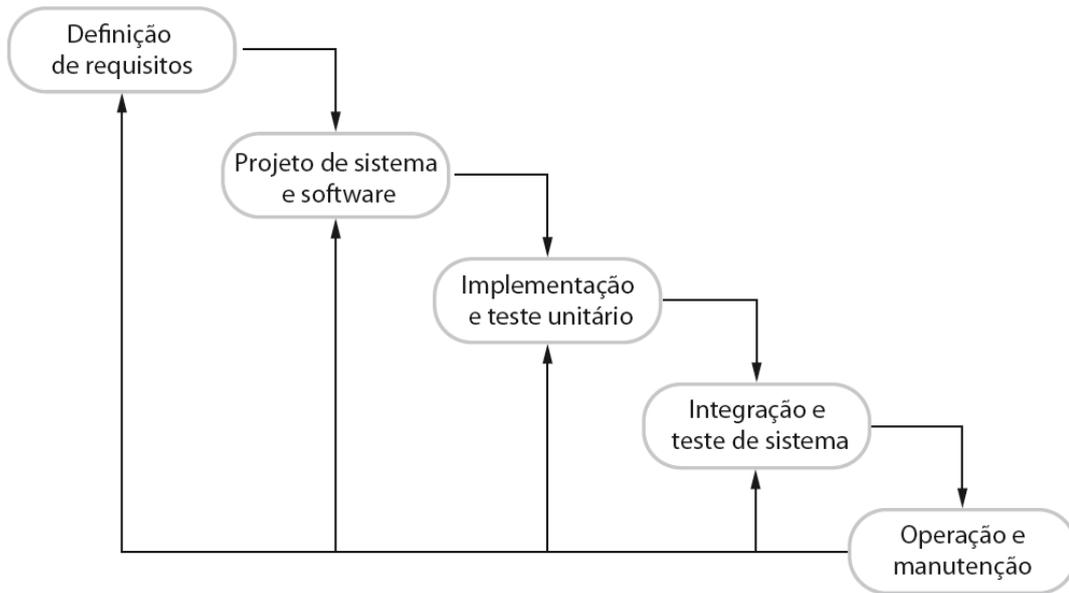
Um modelo de processo de *software* é uma representação abstrata de um processo de *software*. Cada modelo de processo representa um processo sob determinada perspectiva e, dessa forma, fornece somente informações parciais sobre esse processo (SOMMERVILLE, 2010, p. 43).

Existem vários modelos de processos de *softwares*, os mais utilizados são: modelo cascata, desenvolvimento evolucionário e a engenharia de *software* baseado em componentes.

Segundo Sommerville (2010), o modelo cascata é dirigido a planos, ou seja, todas as atividades do projeto são planejadas antecipadamente e nele as partes de especificação e desenvolvimento são separadas. O modelo cascata é dividido em fases, como descritos a seguir:

- Análise e definição de requisitos: serviços, restrições e objetivo do sistema que são definidos pelo usuário;
- Projeto de sistema e *software*: processo que faz a definição divide os requisitos em sistema de *software* ou *hardware*, identifica e faz a descrição das abstrações fundamentais do sistema e suas restrições;
- Implementação e teste de unidade: realização de um conjunto de programas onde são verificadas suas especificações;
- Integração e teste de sistema: as unidades individuais são integradas e testadas como um sistema completo para verificar se os requisitos foram atendidos;
- Operação e manutenção: fase mais longa do ciclo de vida de um *software*, onde são feitas as correções de erros não detectados no estágio anterior.

Figura 1- Modelo cascata.

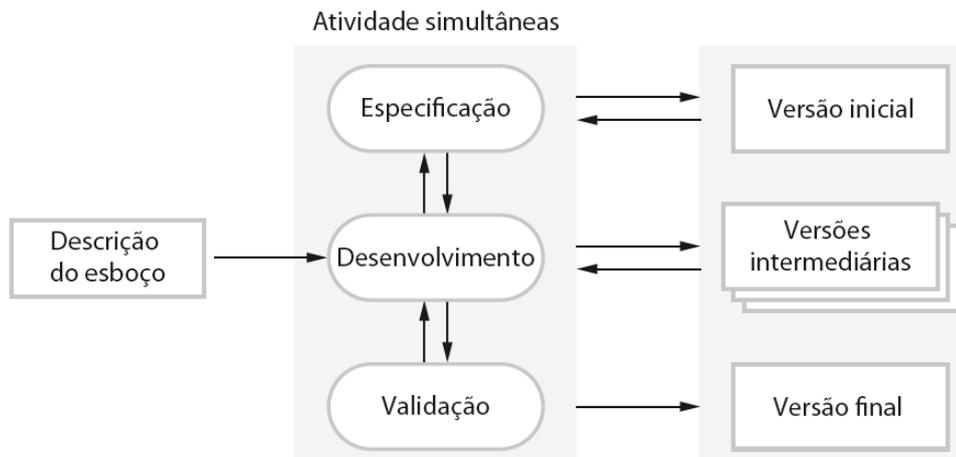


Fonte: SOMMERVILLE (2010, p. 44).

Cada uma dessas fases depende do término da outra, pois não é possível começar outra fase sem terminar a anterior, como mostra a Figura 1.

Segundo Sommerville (2010), desenvolvimento evolucionário pode ser dirigido a planos ou a métodos ágeis e as fases de especificação, desenvolvimento e validação são intercaladas. Nesse processo o desenvolvimento de *software* é realizado em diversas versões, até chegar ao ponto que o cliente realmente deseja e ser entregue a versão completa do sistema, conforme a Figura 2.

Figura 2- Desenvolvimento evolucionário.



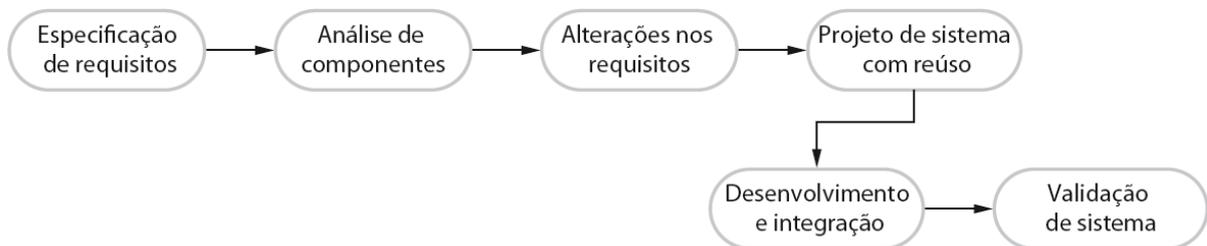
Fonte: SOMMERVILLE (2010, p. 46)

Engenharia de *software* baseado em componentes reutiliza os *softwares* existentes para o desenvolvimento de outros sistemas. E nesse processo de desenvolvimento há outros estágios de especificação (SOMMERVILLE, 2010).

- Análise de componentes: é feita uma análise e especificação de requisitos onde são buscados componentes para implementação dessas especificações;
- Modificação de requisitos: são verificados os componentes de requisitos para realizar possíveis soluções alternativas;
- Projeto de sistema com reuso: durante o estágio os projetistas consideram os componentes reusados, organizando o framework para eles;
- Desenvolvimento e integração: *software* que não pode ser adquirido externamente é desenvolvido e é integrado para criação de um novo sistema, onde a integração é parte do processo de desenvolvimento.

A Figura 3 a seguir, representa a engenharia de *software* baseada em componentes.

Figura 3 - Engenharia de *software* baseada em componentes



Fonte: SOMMERVILLE (2010, p. 46)

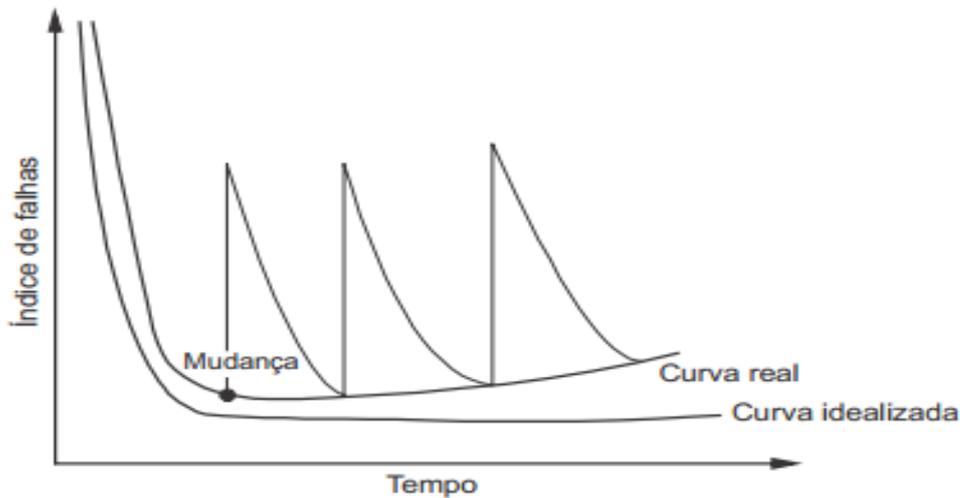
Esses três modelos de processos de *software* possuem características distintas, mas quando são desenvolvidos grandes sistemas, esses processos são utilizados em conjunto.

2.3.3 Ciclo de Vida do Software

Ciclo de vida de um *software* envolve todas as fases que o *software* passa, desde sua criação até o seu momento de descarte. Nesse ciclo estão incluídas as fases de levantamento de requisitos, escopo do projeto, implementação, teste e manutenção. O estágio final de um *software* é quando há a necessidade de substituí-lo por sistemas mais modernos e que atendam

as especificações dos clientes (OLIVEIRA, 2009). A Figura 4 a seguir, demonstra o ciclo de vida do *software*.

Figura 4 - Ciclo de vida do *software*



Fonte: Pressman (2010)

2.3.4 Ciclo de Vida de um Sistema ERP

O ciclo de vida de um sistema ERP é diferente de um ciclo de vida de um sistema tradicional. Para conhecer o ciclo de vida de um sistema ERP é necessário entendermos o ciclo de vida de pacotes comerciais de *software*. Esse ciclo de vida tem como principal característica um processo de desenvolvimento externo, onde sua construção depende do fornecedor e não da empresa que irá utilizá-lo (OLIVEIRA, 2009).

Assim como o ciclo de vida de pacotes comerciais possuem características distintas do ciclo de vida de *softwares* tradicionais, o ciclo de vida de um sistema ERP também possui características que os distinguem dos demais ciclos, tanto de *softwares* habituais, como os de pacotes comerciais.

Segundo Oliveira (2009), o ciclo de vida de um sistema ERP se difere dos demais pela parte funcional, pois o foco está na implantação do sistema, onde poderá ser implantadas fases de adaptação de tecnologias e de processos.

2.4 REQUISITOS DE SOFTWARE

Requisitos são descrições dos serviços fornecidos pelo sistema, onde são refletidas a necessidades do cliente para resolver seus problemas. Os requisitos de *software* são definidos como um processo de descobrir, analisar, documentar e verificar todas as informações que são necessárias para o cliente e posteriormente na produção do sistema para sua especificação (SOMMERVILLE, 2010).

A descrição dos requisitos é fundamental para a produção do sistema, sem tais condições a produção de um sistema de alta qualidade para o cliente não é clara, além de ter a possibilidade de demorar a ser desenvolvido pelos programadores, por falta de entendimento das funções programadas para a divisão das responsabilidades (SOMMERVILLE, 2010).

2.5 REQUISITOS FUNCIONAIS E NÃO FUNCIONAIS

Os requisitos de sistema de *software* são classificados de acordo com Sommerville (2010) em:

- Requisitos funcionais (RF) – são as declarações que o sistema deve fornecer para o serviço, como ele deve reagir e também o que o sistema pode fazer ou não;
- Requisitos não funcionais (RNF) – são as restrições do sistema ou função que são oferecidas para o serviço do sistema.

Na confecção de um sistema, os requisitos são fundamentais, pois é o meio no qual será modelado o *software* de acordo com o que o cliente necessita, essa é a primeira fase de produção de um programa.

Os requisitos funcionais são usados para descrever o que o sistema pode fazer e dar autonomia ao acesso do cliente. Os requisitos não funcionais são as respostas do sistema, seja no armazenamento ou a confiabilidade.

A seguir estão mostrados os Requisitos Funcionais (RF) e Requisitos Não funcionais (RNF):

- a) Alterar dados de cadastros no banco de dados (RF);
- b) Disponibilidade de impressão de relatórios (RF);
- c) Exclusão de cadastros no banco de dados (RF);

- d) Funcionar com sistemas Windows 7 ou superior (RNF);
- e) Inserção de dados fornecidos no banco de dados (RF);
- f) Leituras de dados fornecidos pelo usuário (RF);
- g) Pesquisa de cadastros no bando de dados (RF);
- h) Possuir velocidade aceitável para a sua utilização do sistema (RNF).

2.6 TRABALHOS CORRELACIONADOS

Com o avanço da tecnologia no mundo, cada área realiza seus próprios experimentos, em busca dos benefícios oferecidos pelos sistemas ERP. Assim, torna-se relevante a divulgação de trabalhos científicos desenvolvidos na área que permeia este estudo, com objetivo de localizar referências a trabalhos semelhantes.

Pamplona (2010) desenvolveu um sistema de informação para controle financeiro de uma microempresa via *web*, que possibilita facilidade no acesso, pois como o sistema é *web*, ele pode ser acessado a qualquer momento desde que haja conexão com o servidor. Esse sistema foi desenvolvido para uma tesouraria, abrangendo funcionalidades como cadastros básicos, tipos de movimentação para controles de contas a receber, contas a pagar e fluxo de caixa.

Cruz (2010) analisou os benefícios de informação financeira, contábil e operacional, após a implantação de sistema ERP, demonstrando avanços com a implantação do *software*, entretanto demonstra algumas dificuldades no processo de sua implantação, ressaltando a importância de um estudo aprofundado da empresa antes de realizar a migração dos recursos manuais para recursos tecnológicos.

Como se pode perceber, os autores evidenciam a necessidade da utilização de um sistema ERP em empresas de pequeno porte, em que sua implantação apresenta melhorias consideráveis, referente à gestão organizacional.

3 METODOLOGIA

Para a realização deste trabalho foi utilizado o método de pesquisa bibliométrica, que é definido por Jr, Maia, Viola (2011), é o ato de pesquisa de análise crítica de trabalhos que já foram publicados sobre projetos relacionados. O projeto foi realizado com base em uma análise em artigos científicos e livros que descrevem sobre as características dos sistemas ERP em micro e pequenas empresas, além de métodos e práticas de desenvolvimento de *softwares*.

Para o desenvolvimento deste projeto, foi empregado o método de estudo de caso de caráter qualitativo, no qual é definido como “[...] uma relação dinâmica entre o mundo real e o sujeito, isto é, um vínculo indissociável entre o mundo objetivo e a subjetividade do sujeito que não pode ser traduzido em números.” (SILVA e MENEZES, 2005, p. 20). A escolha dessa metodologia é justificada por ser fundamentada no desenvolvimento de um sistema ERP que visa à melhoria da gestão dos processos de micro e pequenas empresas.

Para o desenvolvimento do sistema solicitado, foi usada a linguagem de programação orientada a objetos, C# (C Sharp), que é definida por Robinson *et al.* (2004), como uma linguagem associada a o .NET framework, uma plataforma projetada para desenvolver qualquer tipo de aplicação através de um novo ambiente. Além desta característica, o C# pode ser de fácil entendimento para os programadores que já produzem códigos em C, C++ e Java.

O ambiente de desenvolvimento foi o Visual Studio 2010, que é um *software* pago, sendo ele um conjunto de ferramentas e serviços que auxilia na criação de vários aplicativos, além de atender às necessidades para o desenvolvimento de um *software*, disponibilizando para o usuário numerosos recursos, além dos desenvolvedores estarem familiarizado com a utilização da ferramenta (MICROSOFT, 2014).

Utilizou-se o *Crystal Reports* no Visual Studio 2010 que é uma ferramenta adicional que auxilia na geração de relatórios mais precisos e com maior rapidez. Disponibilizado pela empresa SAP criadora de *software* de gestão de negócio através do download gratuito, sem a necessidade de registrá-lo (WHEADON, 2009).

Para o armazenamento de informações, foi adotado o Sistema Gerenciador de Banco de Dados (SGBD), que significa “[...] o *software* que trata de todo o acesso ao banco de dados” (DANTE, 2003, p. 37). O banco de dados utilizado foi o Firebird versão 2.5.2.26540, que funciona nos principais sistemas operacionais da atualidade, oferecendo alta performance

e suporta procedimentos de armazenados e gatilhos, além de ser gratuito para todos os usuários, podendo o mesmo ser personalizado por qualquer um (Firebird Foudation Incorporated, 2014).

Esta opção é justificada pelo fato do banco de dados executar nos principais sistemas operacionais no mercado atual, além de ser confiável, possui código aberto e não necessita de licença para seu uso (gratuito), dessa forma ela é livre para a utilização, seja ela comercial ou não (WENER, 2004).

A modelagem do banco foi feita através do *software* gratuito IBExpert versão 2012.02.21.1 que segundo WENER (2004), é considerado uma ferramenta de administração de banco de dados Interbase/Firebird gerador de documentação de banco de dados, capaz de exportar dados para outras plataformas possui interface agradável e de fácil uso.

A ferramenta aplicada para a modelagem das classes e os diagramas (caso de uso e classe) é o Astah, que é uma ferramenta CASE de diagramas UML, que permite a criação de diagramas de forma rápida e fácil, auxiliando no processo de desenvolvimento do *software*. Possui um processo de modelagem bem simples, dinâmico e poderoso e, além disso, é gratuito (ASTAH, 2014).

O levantamento dos requisitos foi realizado através de uma pesquisa em uma micro empresa, Supermercado Novo Cruzeiro, no qual foram feitas perguntas aos proprietários, a fim de obter tais informações pertinentes ao projeto.

Para auxiliar o emprego das ferramentas citadas e para o desenvolvimento do programa, usou-se a rede mundial de computadores.

4 ANÁLISE E PROJETO DO SISTEMA

Este capítulo apresenta as definições e aplicações de levantamento de requisitos, UML (*Unified Modeling Language*), diagrama MER (Modelo Entidade Relacionamento), diagrama de classe e diagrama de caso de uso, além da apresentação das funcionalidades das telas do protótipo do sistema.

4.1 LEVANTAMENTO DE REQUISITOS

O levantamento de requisitos é definido por Sommerville (2010), como um processo de agrupamento das informações sobre o sistema proposto, para obter os requisitos do programa. A cada fase de produção, são obtidas várias informações, sendo que, a primeira fase é a busca dos requisitos a partir de uma reunião com o cliente que o deseja, no qual são feitas perguntas de como o sistema resolverá seus problemas e ouvir as propostas de ideias que são sempre apontadas pelos mesmos.

A fase dois, análise de todas as informações para decidir as responsabilidades de cada desenvolvedor e qual será a finalidade do sistema. Todos da equipe de produção buscam dar ideias para a resolução dos problemas que o sistema tem a finalidade de solucionar e a agilidade no desenvolvimento do sistema para não perderem tempo e custo (SOMMERVILLE, 2010).

As entrevistas formais ou informais são fundamentais para o processo de engenharia de *software*. Os requisitos são derivados das respostas das questões abordadas na entrevista. De acordo com Sommerville (2010), pode ser classificada em dois tipos:

- Entrevista fechada – onde o cliente responde perguntas predefinidas;
- Entrevista aberta – nesta etapa tem um roteiro predefinido.

As entrevistas são úteis para saber como é o funcionamento da empresa do cliente, e quais são os problemas enfrentados por ela.

4.2 ANÁLISE DOS REQUISITOS

A análise de requisitos, como citado anteriormente, foi feita através de uma entrevista com os proprietários do supermercado Novo Cruzeiro, situado na cidade de Guanhães, no mês de julho, a fim de adquirir conhecimentos das reais necessidades do sistema.

Após a análise dos requisitos concluída, constatou-se que na empresa utiliza-se de recursos manuais e planilhas Excel, que é um *software* que permite a criação de planilhas eletrônicas, disponibilizado pela empresa Microsoft (*Microsoft Corporation*, 2014), entretanto a utilização dos dois apresentou ineficiência, baixa confiabilidade e grande demora em acessar as informações.

A seguir, a Figura 5 apresenta como é a forma do controle de clientes na empresa entrevistada.

Figura 5 – Controle de clientes da empresa entrevistada

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
950																
951	25-LAIZ RODRIGUES															PAGAMENTO
952		JANEIRO	FEVEIRO	MARÇO	ABRIL	MAIO	JUNHO	JULHO	AGOSTO	SETEMBRO	OUTUBRO	NOVEMBRO	DEZEMBRO			DATA
953	DEB. RESTANTE	R\$ 113,48														
954	1															
955	2			R\$ 47,26												
956	3															11/abr R\$ 200,00
957	4					R\$ 11,49										13/ago R\$ 480,00
958	5															08/out R\$ 120,00
959	6															
960	7															
961	8				R\$ 31,84											
962	9															
963	10				R\$ 54,96											
964	11															
965	12															
966	13															
967	14	R\$ 17,25				R\$ 19,11			R\$ 27,84		R\$ 15,70					
968	15															
969	16															
970	17										R\$ 34,57					
971	18															
972	19															
973	20					R\$ 82,22		R\$ 37,00	R\$ 24,36							
974	21															
975	22				R\$ 13,77											
976	23															
977	24															
978	25															
979	26									R\$ 31,17						
980	27				R\$ 20,41											
981	28				R\$ 40,92											
982	29					R\$ 82,70				R\$ 18,32						
983	30			R\$ 119,85	R\$ 6,05											
984	31															
985	TOTAL	R\$ 130,73	R\$ 0,00	R\$ 167,11	R\$ 167,95	R\$ 195,52	R\$ 0,00	R\$ 37,00	R\$ 52,20	R\$ 49,49	R\$ 50,27	R\$ 0,00	R\$ 0,00			R\$ 850,27
986	RECEBIMENTO		R\$ 200,00	R\$ 480,00	R\$ 120,00											R\$ 800,00
987	RESTANTE	R\$ 130,73	-R\$ 200,00	-R\$ 312,89	R\$ 47,95	R\$ 195,52	R\$ 0,00	R\$ 37,00	R\$ 52,20	R\$ 49,49	R\$ 50,27	R\$ 0,00	R\$ 0,00			R\$ 50,27
988																

Fonte: Elaborada pelos autores

Na Figura 5, é informado ao usuário o nome do cliente, o somatório mensal de valores a receber, recebidos e restantes (que é uma subtração do primeiro pelo segundo respectivamente). A planilha também oferece um somatório de todos os valores mensais de dados recebidos, a receber e saldo.

O usuário deve digitar os valores a pagar nos seus respectivos campos de data para que seja feita a soma. Para os dados de contas recebidas o usuário deve escrever a data do recebimento e o valor.

Constatou-se a inviabilidade desse meio de controle pelo fato de não possuir dados completos do cliente na planilha, fórmulas de somatórios são facilmente excluídos e dados inseridos erroneamente não tratados.

A seguir é apresentado como é feito o controle das diárias de caixa, contas pagas e recebimentos de forma manual.

Figura 6 – Controle da gestão organizacional da empresa entrevistada

	DÉBITO	CAI
06/20	Saldo	9648
21	Caixa 20-06	781933
21	Agroalves 109075-1	51333 222083
21	pagamento a +	70753 156300
21	Caixa 21-06	173300
21	Posto	3000 326600
21	Feito	1200 325400
21	pgto Guleia	1900 323500
22	Caixa 22-06	109600
22	Posto	5000 428100
23	ernio 145	19745 40835
23	ernio 145 A	2830 40552
23	Rede tel	51701 35382
23	Megafone 2310522-1	66913 286911
23	Anchieta	26529 260382
23	05	25309 235073
23	Deposito 9234-7	195500 39573
23	Caixa 23-06	129700
23	Posto	5000 164273
23	pgto oues	7300 156973
23	Casa Rubens	1000 155973
24	cate Jiquit. 6285	15000 140973
24	Y. Soares 166793-3-3	19517 121456
24	Luiza Cruz 695748	8239 38217
24	Sofnos 25360001	15373 22844
24	pgto bomba	12000 10844
24	Caixa 24-06	211154

Fonte: Elaborado pelos autores

Na Figura 6, é apresentada uma breve descrição, valor e data da movimentação do dinheiro, seja um valor recebido ou pago. Ao término de cada dia é realizado o cálculo manual, fechamento do dia (“saldo”).

Observando o cenário ilustrado na Figura 6, é evidente que esse método de gestão organizacional é obsoleto e ineficiente, pois podem ser inseridos valores e somatórios de forma errônea.

4.3 DIAGRAMAS UML

UML (*Unified Modeling Language*) é uma linguagem que ajuda ao desenvolvedor a documentar e modelar os sistemas orientados a objetos, no qual é definida uma série de elementos que os auxiliam (BOOCH; RUMBAUCH; JACOBSON, 2005).

Segundo Booch, Rumbauch e Jacobson (2005), a UML ajuda a visualizar o sistema (como ele está) e como desejamos que ele se torne, além de permitir a especificação do comportamento do mesmo. Sendo a UML um guia para o desenvolvimento de programas, ele também documenta todas as decisões tomadas.

A UML é essencial para sistemas orientados a objeto, as empresas que não dão atenção a essa fase do projeto têm grande probabilidade de ocasionar erros na construção do *software*.

4.3.1 Definição de diagrama de caso de uso

Segundo Booch, Rumbauch e Jacobson (2005, p. 98), diagrama de caso de uso “mostra um conjunto de casos de uso e atores (um tipo especial de classe) e seus relacionamentos”. Esses diagramas são importantes para à organização, visualização e especificação de um elemento.

4.3.2 Definição de diagrama de classes

O diagrama de classe, segundo Booch, Rumbauch e Jacobson (2005), mostra um conjunto de classes, interfaces, colaborações e seus relacionamentos. O diagrama é importante para visualização, especificação e documentação de modelos estruturais.

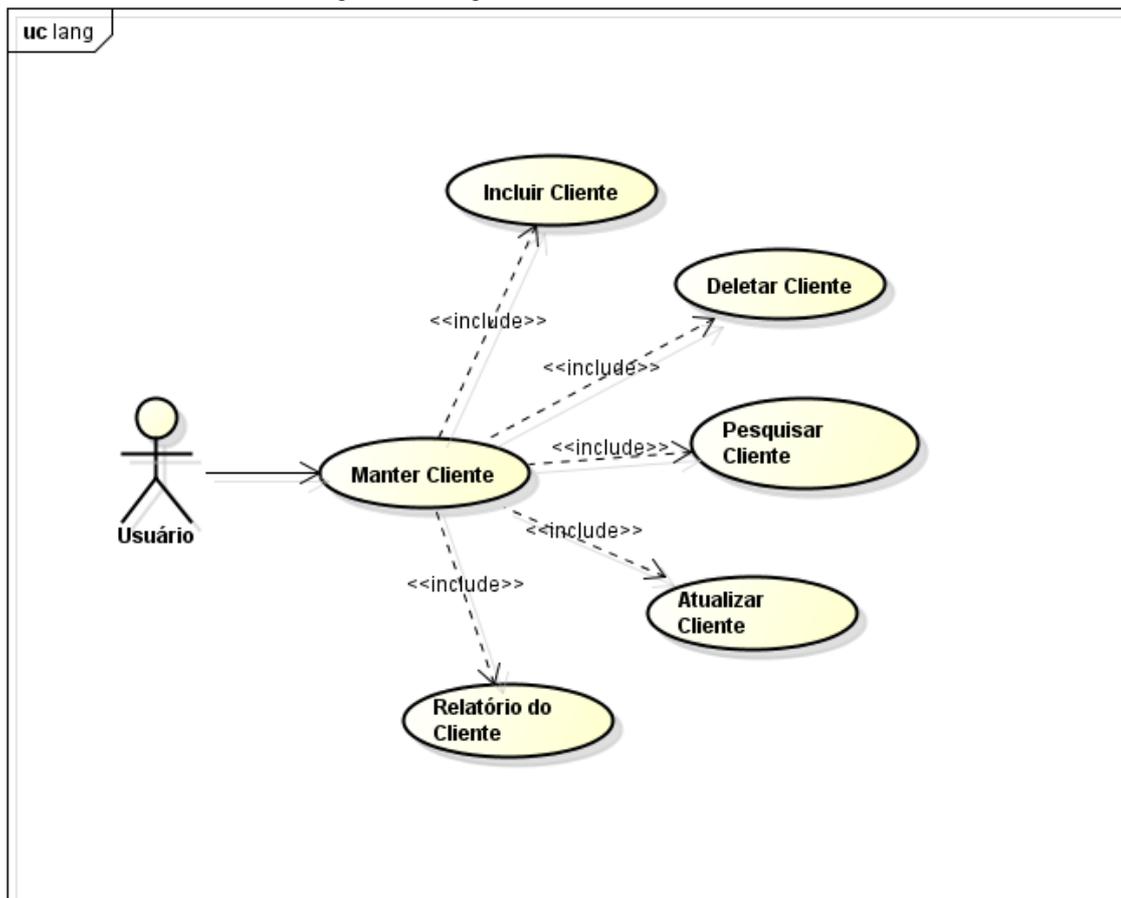
4.3.3 Definição de diagrama de modelo entidade relacionamento (MER)

Silberschatz (1999) define o diagrama de modelo relacional como:

O modelo entidade-relacionamento (E-R) tem por base a percepção de que o mundo real como um conjunto de objetos básicos, chamados entidades e do relacionamento entre eles. Uma entidade é uma “coisa” ou um “objeto” do mundo real que pode ser identificado por outros objetos (SILBERSCHATZ; KORTH; SUDARSHAN, 1999, p. 21).

4.3.4 Diagramas de caso de uso

Figura 7 – Diagrama de caso de uso de clientes



Fonte: Elaborado pelos autores

A Figura 7 apresenta o diagrama de caso de uso do módulo cliente, utilizado no sistema.

Quadro 1 – Cadastro de cliente

Cenário: Cadastro de clientes
Resumo: Este cenário é responsável em descrever como é o processo de inclusão de um cliente no sistema.
Atores: Usuário
Pré-Condições: Não existem pré-condições.
<p>Fluxo de eventos primário:</p> <ol style="list-style-type: none"> 1. O caso de uso se inicia no momento em que o usuário seleciona “Cliente”. 2. O usuário fornece ao sistema dados do cliente, tais como nome, data de nascimento, endereço (logradouro), número, bairro, cidade, complemento, CEP, telefone fixo, celular, limite da conta e tipo do cliente. 3. Se o tipo do cliente for “Jurídico”, cliente deve fornecer dados de CNPJ e razão social. 4. Se o tipo do cliente for “Físico”, cliente deve fornecer dados de CPF e RG. 5. O usuário clica em “Salvar” 6. Usuário submete dados ao sistema. 7. O sistema verifica todas as informações fornecidas e insere as informações do cliente no banco de dados. 8. Sistema retorna uma mensagem de “Cliente salvo com sucesso”.
<p>Fluxo de eventos secundário:</p> <ul style="list-style-type: none"> • A qualquer momento antes de submeter às informações, o usuário pode selecionar cancelar. Os dados não são salvos e o caso de uso termina. • No caso 7, se alguma informação obrigatória não estiver preenchida ou incorreta o sistema retorna uma mensagem informando que deve ser alterado ou preenchido de forma correta os campos incorretos.
Pós-condição: Os dados devem ser salvos no banco com todos os dados obrigatórios preenchidos.

Fonte: Elaborado pelos autores

O Quadro 1 apresenta o cenário de cadastro dos diagramas de caso de uso do módulo cliente, utilizado no sistema.

Quadro 2 – Exclusão de clientes

Cenário: Excluir cliente
Resumo: Este cenário é responsável em descrever como é o processo de exclusão de um cliente no sistema.
Atores: Usuário
Pré-Condições: Selecionar o cliente que deve ser excluído dos registros.
<p>Fluxo de eventos primário:</p> <ol style="list-style-type: none"> 1. O caso de uso se inicia no momento em que o usuário seleciona a opção “Deletar”. 2. Sistema retorna mensagem de confirmação. 3. Usuário confirma exclusão do cliente. 4. Sistema deleta dados do cliente no banco de dados. 5. Sistema retorna mensagem “Cliente deletado com sucesso”.
<p>Fluxo de eventos secundário:</p> <ul style="list-style-type: none"> • No caso 2, é a última opção que o usuário pode cancelar a exclusão do cliente.
Pós-condição: Não existem pós-condições.

Fonte: Elaborado pelos autores

O Quadro 2 apresenta o cenário de excluir dos diagramas de caso de uso do módulo cliente, utilizado no sistema.

Quadro 3 – Consultar cliente

Cenário: Consultar cliente
Resumo: Este cenário é responsável em descrever como é o processo de pesquisa de clientes no sistema.
Atores: Usuário
Pré-Condições: Pesquisar cliente desejado no campo de “Pesquisa” para a consulta nos registros.
<p>Fluxo de eventos primário:</p> <ol style="list-style-type: none"> 1. O caso de uso se inicia no momento em que o usuário escolhe o campo de pesquisa e informa os dados que devem ser requisitados. 2. Sistema consulta informações no banco de dados ao mesmo tempo em que os dados são inseridos. 3. Sistema retorna lista de registros.
<p>Fluxo de eventos secundário:</p> <ul style="list-style-type: none"> • No caso 3, se não houver nenhum registro encontrado não irá retornar nenhuma informação. • O sistema irá pesquisar dados referentes ao nome do cliente e CPF ou CNPJ.
Pós-condição: Não existem pós-condições.

Fonte: Elaborado pelos autores

O Quadro 3 apresenta o cenário de consultar dos diagramas de caso de uso do módulo cliente, utilizado no sistema.

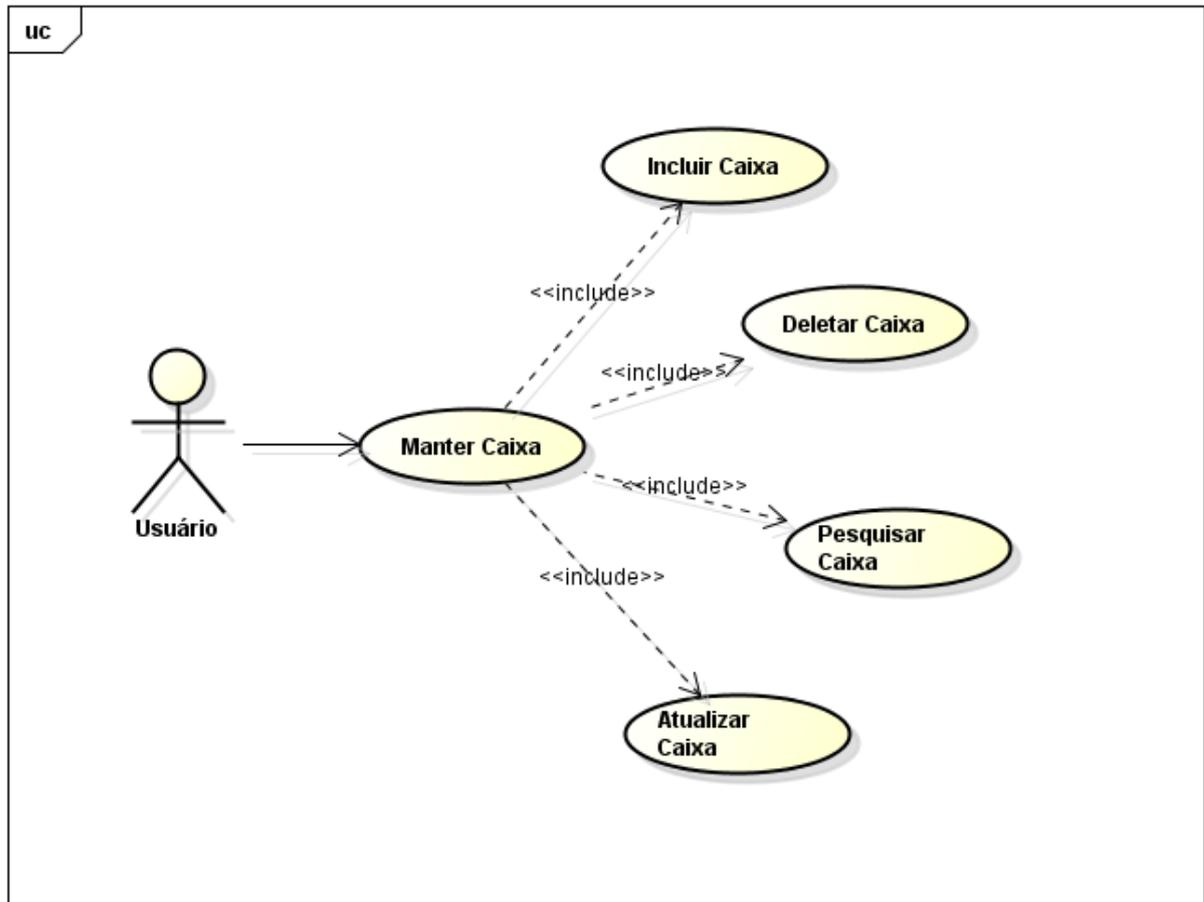
Quadro 4 – Atualizar cliente

Cenário: Atualizar cliente
Resumo: Este cenário é responsável em descrever como é o processo de edição de um cliente no sistema.
Atores: Usuário
Pré-Condições: Selecionar o cliente que deve ser alterado dos registros.
<p>Fluxo de eventos primário:</p> <ol style="list-style-type: none"> 1. O caso de uso se inicia no momento em que o usuário seleciona a opção “Alterar”. 2. Sistema busca dados no banco. 3. Sistema preenche os dados do cliente nos campos referentes aos mesmos. 4. O usuário fornece ao sistema dados do cliente, tais como nome, data de nascimento, endereço (logradouro), número, bairro, cidade, complemento, CEP, telefone fixo, celular, limite da conta e tipo do cliente. 5. Se o tipo do cliente for “Jurídico”, cliente deve fornecer dados de CNPJ e razão social. 6. Se o tipo do cliente for “Físico”, cliente deve fornecer dados de CPF e RG. 7. Usuário submete dados ao sistema. 8. O sistema verifica todas as informações fornecidas e altera as informações do cliente no banco de dados. 9. Sistema retorna uma mensagem de “Cliente alterado com sucesso”.
<p>Fluxo de eventos secundário:</p> <ul style="list-style-type: none"> • A qualquer momento antes de submeter às informações, o usuário pode selecionar cancelar. Os dados não são alterados e o caso de uso termina. • No caso 8, se alguma informação obrigatória não estiver preenchida ou incorreta o sistema retorna uma mensagem informando que deve ser alterado ou preenchido de forma correta os campos incorretos.
Pós-condição: Os dados devem ser salvos no banco com todos os dados obrigatórios preenchidos.

Fonte: Elaborado pelos autores

O Quadro 4 apresenta o cenário de atualizar dos diagramas de caso de uso do módulo cliente, utilizado no sistema.

Figura 8 – Diagrama de caso de uso de caixa



Fonte: Elaborado pelos autores

A Figura 8 apresenta o diagrama de caso de uso do módulo caixa, utilizado no sistema.

Quadro 5 – Cadastro de caixa

Cenário: Cadastro de caixa
Resumo: Este cenário é responsável em descrever como é o processo de inclusão de um caixa no sistema.
Atores: Usuário
Pré-Condições: Não existem pré-condições.
<p>Fluxo de eventos primário:</p> <ol style="list-style-type: none"> 1. O caso de uso se inicia no momento em que o usuário seleciona “Caixa”. 2. O usuário fornece ao sistema o nome do caixa. 3. O usuário clica em “Salvar”. 4. Usuário submete dados ao sistema. 5. O sistema verifica todas as informações fornecidas e insere as informações do caixa no banco de dados. 6. Sistema retorna uma mensagem de “Caixa salvo com sucesso”.
<p>Fluxo de eventos secundário:</p> <ul style="list-style-type: none"> • A qualquer momento antes de submeter às informações, o usuário pode selecionar cancelar. Os dados não são salvos e o caso de uso termina. • No caso 4, se alguma informação obrigatória não estiver preenchida ou incorreta o sistema retorna uma mensagem informando que deve ser alterado ou preenchido de forma correta os campos incorretos.
Pós-condição: Os dados devem ser salvos no banco com todos os dados obrigatórios preenchidos.

Fonte: Elaborado pelos autores

O Quadro 5 apresenta o cenário de cadastro dos diagramas de caso de uso do módulo caixa, utilizado no sistema.

Quadro 6 – Excluir caixa

Cenário: Excluir caixa
Resumo: Este cenário é responsável em descrever como é o processo de exclusão de um caixa no sistema.
Atores: Usuário
Pré-Condições: Selecionar o caixa que deve ser excluído dos registros na parte de consulta.
<p>Fluxo de eventos primário:</p> <ol style="list-style-type: none"> 1. O caso de uso se inicia no momento em que o usuário seleciona a opção “Deletar”. 2. Sistema retorna mensagem de confirmação. 3. Usuário confirma exclusão do caixa. 4. Sistema deleta dados do caixa no banco de dados. 5. Sistema retorna mensagem “Caixa deletado com sucesso”.
Fluxo de eventos secundário: No caso 2, é a última opção que o usuário pode cancelar a exclusão do caixa.
Pós-condição: Não existem pós-condições.

Fonte: Elaborado pelos autores

O Quadro 6 apresenta o cenário de excluir dos diagramas de caso de uso do módulo caixa, utilizado no sistema.

Quadro 7 – Consultar caixa

Cenário: Consultar caixa
Resumo: Este cenário é responsável em descrever como é o processo de pesquisa de caixas no sistema.
Atores: Usuário
Pré-Condições: Pesquisar caixa desejado no campo de “Pesquisa” para a consulta nos registros.
<p>Fluxo de eventos primário:</p> <ol style="list-style-type: none"> 1. O caso de uso se inicia no momento em que o usuário escolhe o campo de pesquisa e informa os dados que devem ser requisitados. 2. Sistema consulta informações no banco de dados ao mesmo tempo em que os dados são inseridos. 3. Sistema retorna lista de registros.
<p>Fluxo de eventos secundário:</p> <ul style="list-style-type: none"> • No caso 3, se não houver nenhum registro encontrado não irá retornar nenhuma informação. • O sistema irá pesquisar os nomes dos caixas.
Pós-condição: Não existem pós-condições.

Fonte: Elaborado pelos autores

O Quadro 7 apresenta o cenário de consulta dos diagramas de caso de uso do módulo caixa, utilizado no sistema.

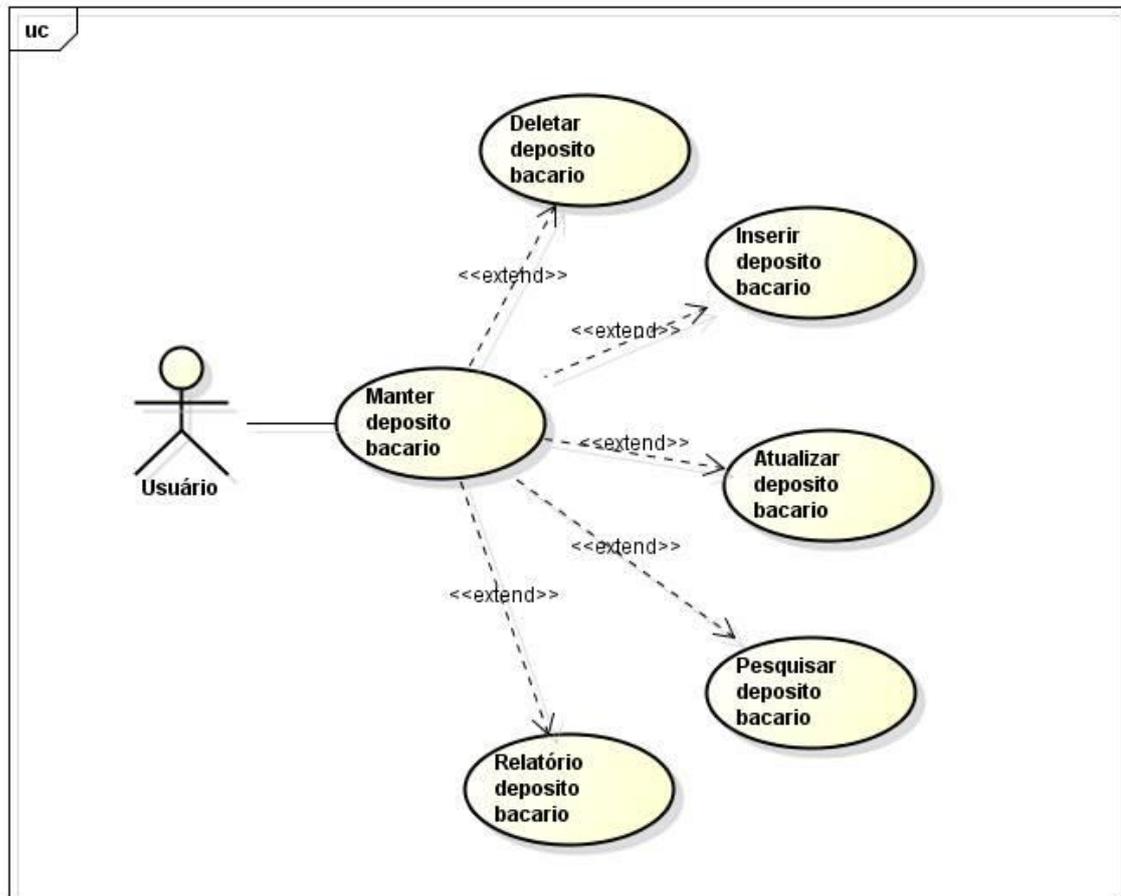
Quadro 8 – Atualizar caixa

Cenário: Atualizar caixa
Resumo: Este cenário é responsável em descrever como é o processo de edição de um caixa no sistema.
Atores: Usuário
Pré-Condições: Selecionar o caixa que deve ser alterado dos registros.
<p>Fluxo de eventos primário:</p> <ol style="list-style-type: none"> 1. O caso de uso se inicia no momento em que o usuário seleciona a opção “Alterar”. 2. Sistema busca dados no banco. 3. Sistema preenche o nome do caixa selecionado. 4. O usuário fornece ao sistema o nome do caixa. 5. Usuário submete dados ao sistema. 6. O sistema verifica todas as informações fornecidas e altera as informações do caixa no banco de dados. 7. Sistema retorna uma mensagem de “Caixa alterado com sucesso”.
<p>Fluxo de eventos secundário:</p> <ul style="list-style-type: none"> • A qualquer momento antes de submeter às informações, o usuário pode selecionar cancelar. Os dados não são alterados e o caso de uso termina. • No caso 6, se alguma informação obrigatória não estiver preenchida ou incorreta o sistema retorna uma mensagem informando que deve ser alterado ou preenchido de forma correta os campos incorretos.
Pós-condição: Os dados devem ser salvos no banco com todos os dados obrigatórios preenchidos.

Fonte: Elaborado pelos autores

O Quadro 8 apresenta o cenário de atualizar dos diagramas de caso de uso do módulo caixa, utilizado no sistema.

Figura 9 – Diagrama de caso de uso de depósito bancário



Fonte: Elaborado pelos autores

A Figura 9 apresenta o diagrama de caso de uso do módulo depósito bancário, utilizado no sistema.

Quadro 9 – Cadastro depósito bancário

Cenário: Cadastro de depósitos bancários
Resumo: Este cenário é responsável em descrever como é o processo de inclusão de um depósito bancário no sistema.
Atores: Usuário
Pré-Condições: Não existem pré-condições.
<p>Fluxo de eventos primário:</p> <ol style="list-style-type: none"> 1. O caso de uso se inicia no momento em que o usuário seleciona “Depósitos bancários”. 2. O usuário fornece ao sistema dados do depósito bancário, tais como nome do banco, agência bancária, número da conta, tipo da conta, data do depósito e o valor do depósito. 3. Usuário submete dados ao sistema. 4. O sistema verifica todas as informações fornecidas e insere as informações do depósito bancário no banco de dados. 5. Sistema retorna uma mensagem de “Depósito bancário salvo com sucesso”.
<p>Fluxo de eventos secundário:</p> <ul style="list-style-type: none"> • A qualquer momento antes de submeter às informações, o usuário pode selecionar cancelar. Os dados não são salvos e o caso de uso termina. • No caso 4, se alguma informação obrigatória não estiver preenchida ou incorreta o sistema retorna uma mensagem informando que deve ser alterado ou preenchido de forma correta os campos incorretos.
Pós-condição: Os dados devem ser salvos no banco com todos os dados obrigatórios preenchidos.

Fonte: Elaborado pelos autores

O Quadro 9 apresenta o cenário de cadastro dos diagramas de caso de uso do módulo depósito bancário, utilizado no sistema.

Quadro 10 – Excluir depósito bancário

Cenário: Excluir depósito bancário
Resumo: Este cenário é responsável em descrever como é o processo de exclusão de um depósito bancário no sistema.
Atores: Usuário
Pré-Condições: Selecionar o depósito bancário que deve ser excluído dos registros.
<p>Fluxo de eventos primário:</p> <ol style="list-style-type: none"> 1. O caso de uso se inicia no momento em que o usuário seleciona a opção “Deletar”. 2. Sistema retorna mensagem de confirmação. 3. Usuário confirma exclusão do depósito bancário. 4. Sistema deleta dados do tipo no banco de dados. 5. Sistema retorna mensagem “Depósito bancário deletado com sucesso”.
<p>Fluxo de eventos secundário:</p> <ul style="list-style-type: none"> • No caso 2, é a última opção que o usuário pode cancelar a exclusão do depósito bancário.
Pós-condição: Não existem pós-condições.

Fonte: Elaborado pelos autores

O Quadro 10 apresenta o cenário de excluir dos diagramas de caso de uso do módulo depósito bancário, utilizado no sistema.

Quadro 11 – Consultar depósito bancário

Cenário: Consultar depósito bancário
Resumo: Este cenário é responsável em descrever como é o processo de pesquisa de depósitos bancários no sistema.
Atores: Usuário
Pré-Condições: Pesquisar depósito bancário desejado no campo de “Pesquisa” para a consulta nos registros.
<p>Fluxo de eventos primário:</p> <ol style="list-style-type: none"> 1. O caso de uso se inicia no momento em que o usuário escolhe o campo de pesquisa e informa os dados que devem ser requisitados. 2. Sistema consulta informações no banco de dados ao mesmo tempo em que os dados são inseridos. 3. Sistema retorna lista de registros.
<p>Fluxo de eventos secundário:</p> <ul style="list-style-type: none"> • No caso 3, se não houver nenhum registro encontrado não irá retornar nenhuma informação. • O sistema irá pesquisar o nome do banco e número da conta.
Pós-condição: Não existem pós-condições.

Fonte: Elaborado pelos autores

O Quadro 11 apresenta o cenário de consulta dos diagramas de caso de uso do módulo depósito bancário, utilizado no sistema.

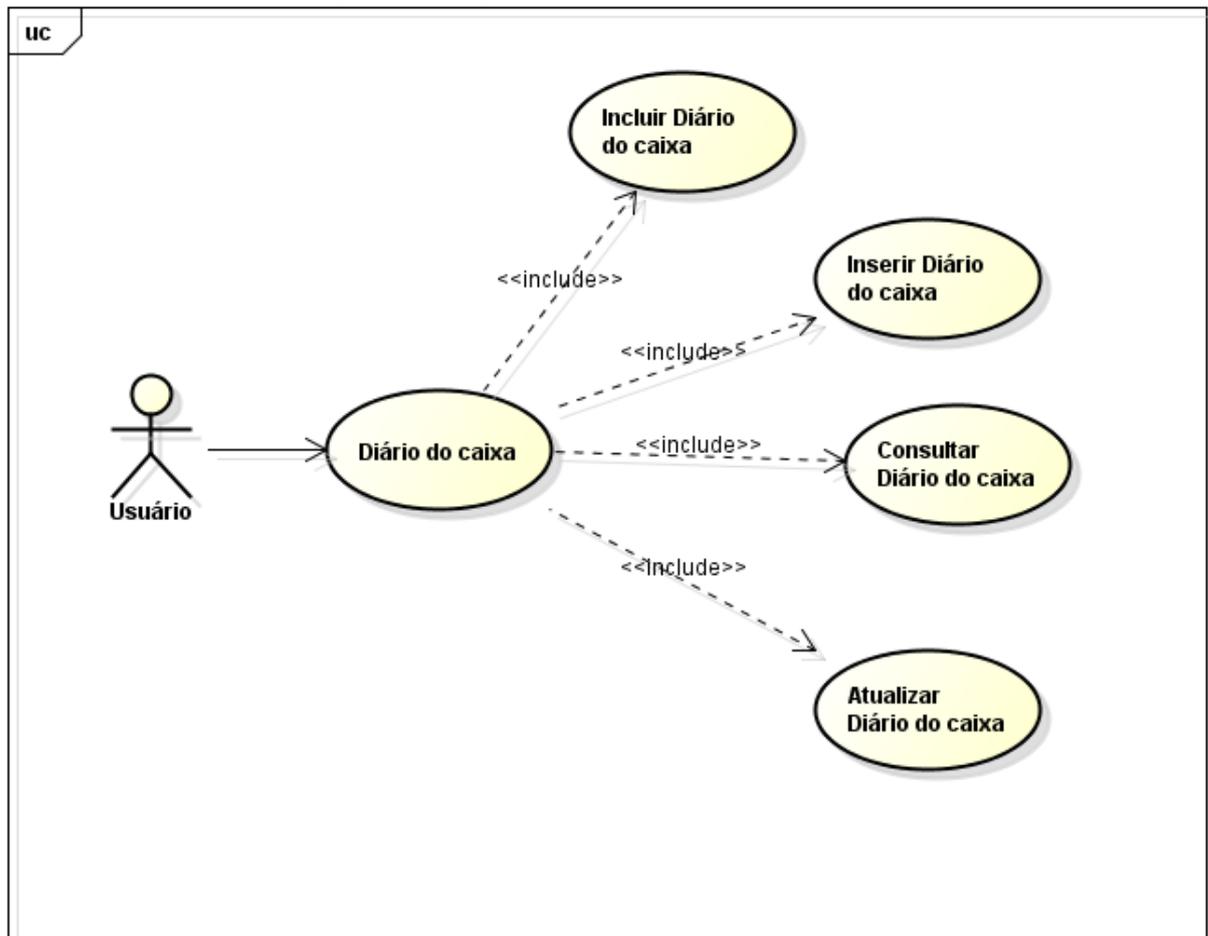
Quadro 12 – Atualizar depósito bancário

Cenário: Atualizar depósito bancário
Resumo: Este cenário é responsável em descrever como é o processo de edição de um depósito bancário no sistema.
Atores: Usuário
Pré-Condições: Selecionar o depósito bancário que deve ser alterado dos registros.
<p>Fluxo de eventos primário:</p> <ol style="list-style-type: none"> 1. O caso de uso se inicia no momento em que o usuário seleciona a opção “Alterar”. 2. Sistema busca dados no banco. 3. Sistema preenche dados do depósito bancário referente aos mesmos. 4. O usuário fornece ao sistema dados do depósito bancário, tais como nome do banco, agência bancária, número da conta, tipo da conta, data do depósito e o valor do depósito. 5. Usuário submete dados ao sistema. 6. O sistema verifica todas as informações fornecidas e altera as informações do depósito bancário no banco de dados. 7. Sistema retorna uma mensagem de “Depósito bancário alterado com sucesso”.
<p>Fluxo de eventos secundário:</p> <ul style="list-style-type: none"> • A qualquer momento antes de submeter às informações, o usuário pode selecionar cancelar. Os dados não são alterados e o caso de uso termina. • No caso 6, se alguma informação obrigatória não estiver preenchida ou incorreta o sistema retorna uma mensagem informando que deve ser alterado ou preenchido de forma correta os campos incorretos.
Pós-condição: Os dados devem ser salvos no banco com todos os dados obrigatórios preenchidos.

Fonte: Elaborado pelos autores

O Quadro 12 apresenta o cenário de atualizar dos diagramas de caso de uso do módulo depósito bancário, utilizado no sistema.

Figura 10 - Diagrama de caso de uso de diário do caixa



Fonte: Elaborada pelos autores

A Figura 10 apresenta o diagrama de caso de uso do módulo diário do caixa, utilizado no sistema.

Quadro 13 – Cadastro de diário do caixa

Cenário: Cadastro de diário do caixa
Resumo: Este cenário é responsável em descrever como é o processo de inclusão de um diário do caixa no sistema.
Atores: Usuário
Pré-Condições: Para o cadastro de um diário do caixa deve haver pelo menos um registro de caixa.
<p>Fluxo de eventos primário:</p> <ol style="list-style-type: none"> 1. O caso de uso se inicia no momento em que o usuário seleciona “Diário do caixa”. 2. O usuário fornece ao sistema dados do controle de caixa, tais como caixa, valor, data do diário do caixa. 3. Usuário submete dados ao sistema. 4. O sistema verifica todas as informações fornecidas e insere as informações do controle de caixa no banco de dados. 5. Sistema retorna uma mensagem de “Diário do caixa salvo com sucesso”.
<p>Fluxo de eventos secundário:</p> <ul style="list-style-type: none"> • A qualquer momento antes de submeter às informações, o usuário pode selecionar cancelar. Os dados não são salvos e o caso de uso termina. • No caso 4, se alguma informação obrigatória não estiver preenchida ou incorreta o sistema retorna uma mensagem informando que deve ser alterado ou preenchido de forma correta os campos incorretos.
Pós-condição: Os dados devem ser salvos no banco com todos os dados obrigatórios preenchidos.

Fonte: Elaborado pelos autores

O Quadro 13 apresenta o cenário de cadastro dos diagramas de caso de uso do módulo diário do caixa, utilizado no sistema.

Quadro 14 – Excluir diário do caixa

Cenário: Excluir diário do caixa
Resumo: Este cenário é responsável em descrever como é o processo de exclusão de um diário do caixa no sistema.
Atores: Usuário
Pré-Condições: Selecionar o diário do caixa que deve ser excluída dos registros.
Fluxo de eventos primário: <ol style="list-style-type: none"> 1. O caso de uso se inicia no momento em que o usuário seleciona a opção “Deletar”. 2. Sistema retorna mensagem de confirmação. 3. Usuário confirma exclusão do Diário do caixa. 4. Sistema deleta dados do diário do caixa no banco de dados. 5. Sistema retorna mensagem “Diário do caixa deletada com sucesso”.
Fluxo de eventos secundário: <ul style="list-style-type: none"> • No caso 2, é a última opção que o usuário pode cancelar a exclusão do diário do caixa.
Pós-condição: Não existem pós-condições

Fonte: Elaborado pelos autores

O Quadro 14 apresenta o cenário de excluir dos diagramas de caso de uso do módulo diário do caixa, utilizado no sistema.

Quadro 15 – Consultar diário do caixa

Cenário: Consultar diário do caixa
Resumo: Este cenário é responsável em descrever como é o processo de pesquisa do diário do caixa no sistema.
Atores: Usuário
Pré-Condições: Pesquisar controle de caixa desejado no campo de “Pesquisa” para a consulta nos registros.
<p>Fluxo de eventos primário:</p> <ol style="list-style-type: none"> 1. O caso de uso se inicia no momento em que o usuário escolhe o campo de pesquisa e informa os dados que devem ser requisitados. 2. Sistema consulta informações no banco de dados ao mesmo tempo em que os dados são inseridos. 3. Sistema retorna lista de registros.
<p>Fluxo de eventos secundário:</p> <ul style="list-style-type: none"> • No caso 3, se não houver nenhum registro encontrado não irá retornar nenhuma informação. • O sistema irá pesquisar os nomes dos caixas.
Pós-condição: Não existem pós-condições.

Fonte: Elaborado pelos autores

O Quadro 15 apresenta o cenário de consultar dos diagramas de caso de uso do módulo diário do caixa, utilizado no sistema.

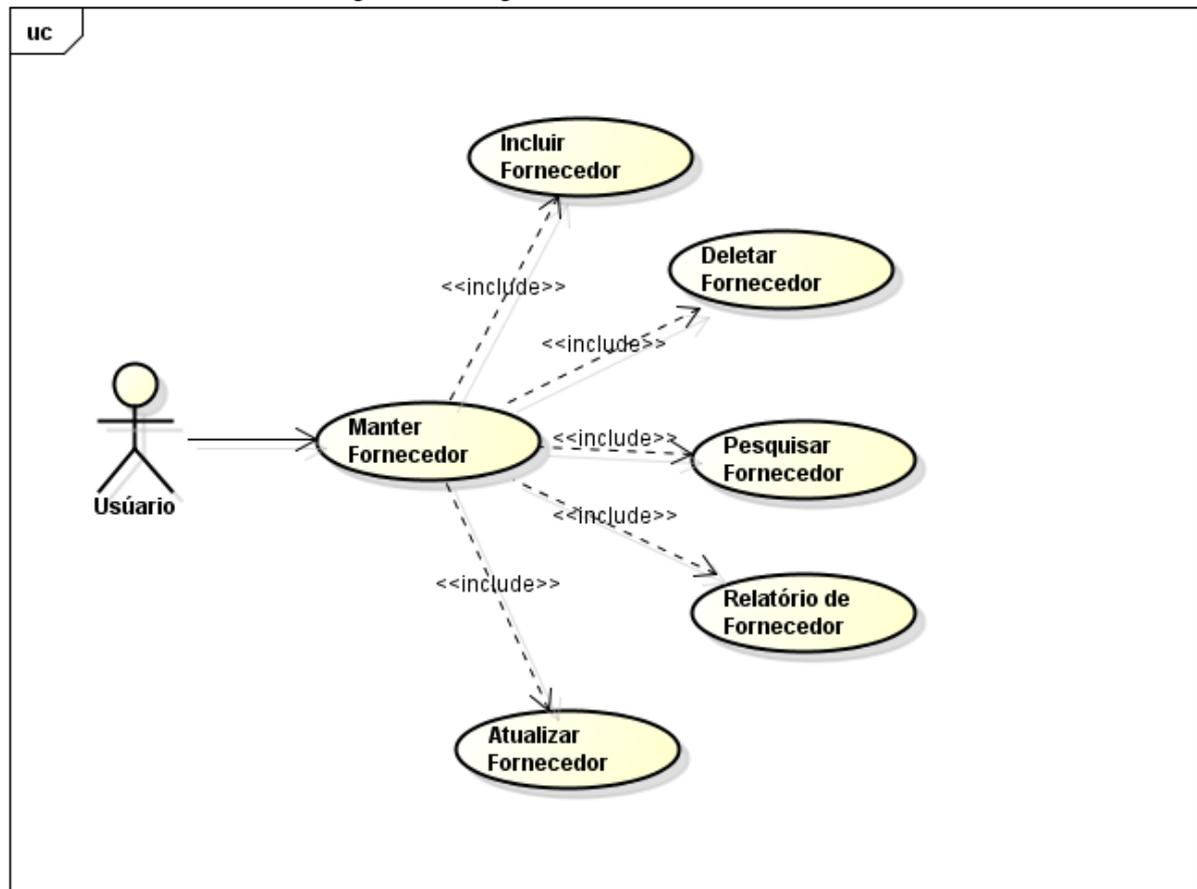
Quadro 16 – Atualizar diário do caixa

Cenário: Atualizar diário do caixa
Resumo: Este cenário é responsável em descrever como é o processo de edição de um diário do caixa no sistema.
Atores: Usuário
Pré-Condições: Selecionar o diário do caixa que deve ser alterado dos registros.
<p>Fluxo de eventos primário:</p> <ol style="list-style-type: none"> 1. O caso de uso se inicia no momento em que o usuário seleciona a opção “Alterar”. 2. Sistema busca dados no banco. 3. Sistema preenche os dados da conta nos campos referentes aos mesmos. 4. O usuário fornece ao sistema dados do controle de caixa, tais como caixa, valor, data do controle do caixa. 5. Usuário submete dados ao sistema. 6. O sistema verifica todas as informações fornecidas e insere as informações do diário do caixa no banco de dados. 7. Sistema retorna uma mensagem de “Diário do caixa alterada com sucesso”.
<p>Fluxo de eventos secundário:</p> <ul style="list-style-type: none"> • A qualquer momento antes de submeter às informações, o usuário pode selecionar cancelar. Os dados não são alterados e o caso de uso termina. • No caso 6, se alguma informação obrigatória não estiver preenchida ou incorreta o sistema retorna uma mensagem informando que deve ser alterado ou preenchido de forma correta os campos incorretos.
Pós-condição: Os dados devem ser salvos no banco com todos os dados obrigatórios preenchidos.

Fonte: Elaborado pelos autores

O Quadro 16 apresenta o cenário de atualizar dos diagramas de caso de uso do módulo diário do caixa, utilizado no sistema.

Figura 11 – Diagrama de caso de uso de fornecedor



Fonte: Elaborada pelos autores

A Figura 11 apresenta o diagrama de caso de uso do módulo fornecedor, utilizado no sistema.

Quadro 17 – Cadastro fornecedor

Cenário: Cadastro de fornecedor
Resumo: Este cenário é responsável em descrever como é o processo de inclusão de um fornecedor no sistema.
Atores: Usuário
Pré-Condições: Não existem pré-condições.
<p>Fluxo de eventos primário:</p> <p>9. O caso de uso se inicia no momento em que o usuário seleciona “Fornecedor”.</p> <p>10. O usuário fornece ao sistema dados do fornecedor, tais como nome, CNPJ, razão social, endereço (logradouro), número, bairro, cidade, complemento, CEP, telefone fixo, celular.</p> <p>11. O sistema verifica todas as informações fornecidas e insere as informações do fornecedor no banco de dados.</p> <p>12. Sistema retorna uma mensagem de “Fornecedor salvo com sucesso”.</p>
<p>Fluxo de eventos secundário:</p> <ul style="list-style-type: none"> • A qualquer momento antes de submeter às informações, o usuário pode selecionar cancelar. Os dados não são salvos e o caso de uso termina. • No caso 3, se alguma informação obrigatória não estiver preenchida ou incorreta o sistema retorna uma mensagem informando que deve ser alterado ou preenchido de forma correta os campos incorretos.
Pós-condição: Os dados devem ser salvos no banco com todos os dados obrigatórios preenchidos.

Fonte: Elaborado pelos autores

O Quadro 17 apresenta o cenário de cadastro dos diagramas de caso de uso do módulo fornecedor, utilizado no sistema.

Quadro 18 – Excluir fornecedor

Cenário: Excluir fornecedor
Resumo: Este cenário é responsável em descrever como é o processo de exclusão de um fornecedor no sistema.
Atores: Usuário
Pré-Condições: Selecionar o fornecedor que deve ser excluído dos registros em consultas.
<p>Fluxo de eventos primário:</p> <ol style="list-style-type: none"> 1. O caso de uso se inicia no momento em que o usuário seleciona a opção “Deletar”. 2. Sistema retorna mensagem de confirmação. 3. Usuário confirma exclusão do fornecedor. 4. Sistema deleta dados do fornecedor no banco de dados. 5. Sistema retorna mensagem “Fornecedor deletado com sucesso”.
<p>Fluxo de eventos secundário:</p> <ul style="list-style-type: none"> • No caso 2, é a última opção que o usuário pode cancelar a exclusão do fornecedor.
Pós-condição: Não existem pós-condições.

Fonte: Elaborado pelos autores

O Quadro 18 apresenta o cenário de excluir dos diagramas de caso de uso do módulo fornecedor, utilizado no sistema.

Quadro 19 – Consultar fornecedor

Cenário: Consultar fornecedor
Resumo: Este cenário é responsável em descrever como é o processo de pesquisa de fornecedores no sistema.
Atores: Usuário
Pré-Condições: Pesquisar fornecedor desejado no campo de “Pesquisa” para a consulta nos registros.
<p>Fluxo de eventos primário:</p> <ol style="list-style-type: none"> 1. O caso de uso se inicia no momento em que o usuário escolhe o campo de pesquisa e informa os dados que devem ser requisitados. 2. Sistema consulta informações no banco de dados ao mesmo tempo em que os dados são inseridos. 3. Sistema retorna lista de registros.
<p>Fluxo de eventos secundário:</p> <ul style="list-style-type: none"> • No caso 3, se não houver nenhum registro encontrado não irá retornar nenhuma informação. • O sistema irá pesquisar nome, CNPJ e razão social de registros no banco.
Pós-condição: Não existem pós-condições.

Fonte: Elaborado pelos autores

O Quadro 19 apresenta o cenário de consultar dos diagramas de caso de uso do módulo fornecedor, utilizado no sistema.

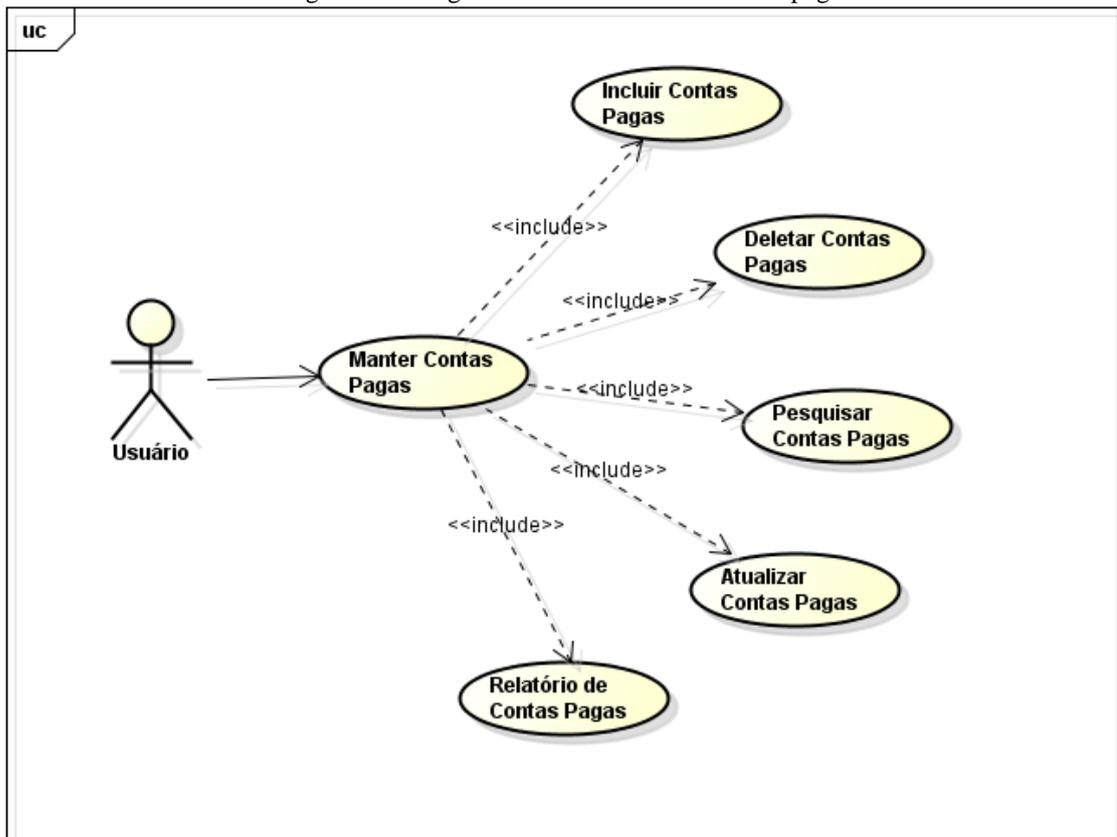
Quadro 20 – Atualizar fornecedor

Cenário: Atualizar fornecedor
Resumo: Este cenário é responsável em descrever como é o processo de edição de um fornecedor no sistema.
Atores: Usuário
Pré-Condições: Selecionar o fornecedor que deve ser alterado dos registros.
<p>Fluxo de eventos primário:</p> <ol style="list-style-type: none"> 1. O caso de uso se inicia no momento em que o usuário seleciona a opção “Alterar”. 2. Sistema busca dados no banco. 3. Sistema preenche os dados do fornecedor nos campos referentes aos mesmos. 4. O usuário fornece ao sistema dados do fornecedor, tais como nome, CNPJ, razão social, endereço (logradouro), número, bairro, cidade, complemento, CEP, telefone fixo, celular. 5. Usuário submete dados ao sistema. 6. O sistema verifica todas as informações fornecidas e altera as informações do fornecedor no banco de dados. 7. Sistema retorna uma mensagem de “Fornecedor alterado com sucesso”.
<p>Fluxo de eventos secundário:</p> <ul style="list-style-type: none"> • A qualquer momento antes de submeter às informações, o usuário pode selecionar cancelar. Os dados não são alterados e o caso de uso termina. • No caso 6, se alguma informação obrigatória não estiver preenchida ou incorreta o sistema retorna uma mensagem informando que deve ser alterado ou preenchido de forma correta os campos incorretos.
Pós-condição: Os dados devem ser salvos no banco com todos os dados obrigatórios preenchidos.

Fonte: Elaborado pelos autores

O Quadro 20 apresenta o cenário de atualizar dos diagramas de caso de uso do módulo fornecedor, utilizado no sistema.

Figura 12 - Diagrama de caso de uso de contas pagas



Fonte: Elaborado pelos autores

A Figura 12 apresenta o diagrama de caso de uso do módulo contas a pagar, utilizado no sistema.

Quadro 21 – Cadastrar pagas

Cenário: Cadastro de pagas
Resumo: Este cenário é responsável em descrever como é o processo de inclusão de uma conta a pagar no sistema.
Atores: Usuário
Pré-Condições: Para o cadastro de Contas a pagar deve haver pelo menos um registro de cliente e um registro de tipo.
<p>Fluxo de eventos primário:</p> <ol style="list-style-type: none"> 1. O caso de uso se inicia no momento em que o usuário seleciona “Contas a Pagar”. 2. O usuário fornece ao sistema dados da conta, tais como fornecedor, tipo, valor, número do documento, data do pagamento, data do cadastro e status. 3. Usuário submete dados ao sistema. 4. O sistema verifica todas as informações fornecidas e insere as informações da conta a pagar (ou recebida) no banco de dados. 5. Sistema retorna uma mensagem de “Conta salva com sucesso”.
<p>Fluxo de eventos secundário:</p> <ul style="list-style-type: none"> • A qualquer momento antes de submeter às informações, o usuário pode selecionar cancelar. Os dados não são salvos e o caso de uso termina. • No caso 4, se alguma informação obrigatória não estiver preenchida ou incorreta o sistema retorna uma mensagem informando que deve ser alterado ou preenchido de forma correta os campos incorretos. • O sistema irá preencher automaticamente o campo data de cadastro com a data atual em que o usuário estará fazendo a interação com o programa.
Pós-condição: Os dados devem ser salvos no banco com todos os dados obrigatórios preenchidos.

Fonte: Elaborado pelos autores

O Quadro 21 apresenta o cenário de cadastro dos diagramas de caso de uso do módulo contas a pagar, utilizado no sistema.

Quadro 22 – Excluir pagas

Cenário: Excluir pagas
Resumo: Este cenário é responsável em descrever como é o processo de exclusão de uma conta a pagar no sistema.
Atores: Usuário
Pré-Condições: Selecionar a conta a pagar que deve ser excluído dos registros.
<p>Fluxo de eventos primário:</p> <ol style="list-style-type: none"> 1. O caso de uso se inicia no momento em que o usuário seleciona a opção “Deletar”. 2. Sistema retorna mensagem de confirmação. 3. Usuário confirma exclusão da conta. 4. Sistema deleta dados da conta a pagar no banco de dados. 5. Sistema retorna mensagem “Conta deletada com sucesso”.
<p>Fluxo de eventos secundário:</p> <ul style="list-style-type: none"> • No caso 2, é a última opção que o usuário pode cancelar a exclusão da conta a pagar.
Pós-condição: Não existem pós-condições.

Fonte: Elaborado pelos autores

O Quadro 22 apresenta o cenário de excluir dos diagramas de caso de uso do módulo contas a pagar, utilizado no sistema.

Quadro 23 – Consulta pagas

Cenário: Consultar pagas
Resumo: Este cenário é responsável em descrever como é o processo de pesquisa de contas a pagar no sistema.
Atores: Usuário
Pré-Condições: Pesquisar contas a pagar desejado no campo de “Pesquisa” para a consulta nos registros.
<p>Fluxo de eventos primário:</p> <ol style="list-style-type: none"> 1. O caso de uso se inicia no momento em que o usuário escolhe o campo de pesquisa e informa os dados que devem ser requisitados. 2. Sistema consulta informações no banco de dados ao mesmo tempo em que os dados são inseridos. 3. Sistema retorna lista de registros.
<p>Fluxo de eventos secundário:</p> <ul style="list-style-type: none"> • No caso 3, se não houver nenhum registro encontrado não irá retornar nenhuma informação. • O sistema irá pesquisar o nome do fornecedor, tipo da conta e número da conta.
Pós-condição: Não existem pós-condições.

Fonte: Elaborado pelos autores

O Quadro 23 apresenta o cenário de consultar dos diagramas de caso de uso do módulo contas a pagar, utilizado no sistema.

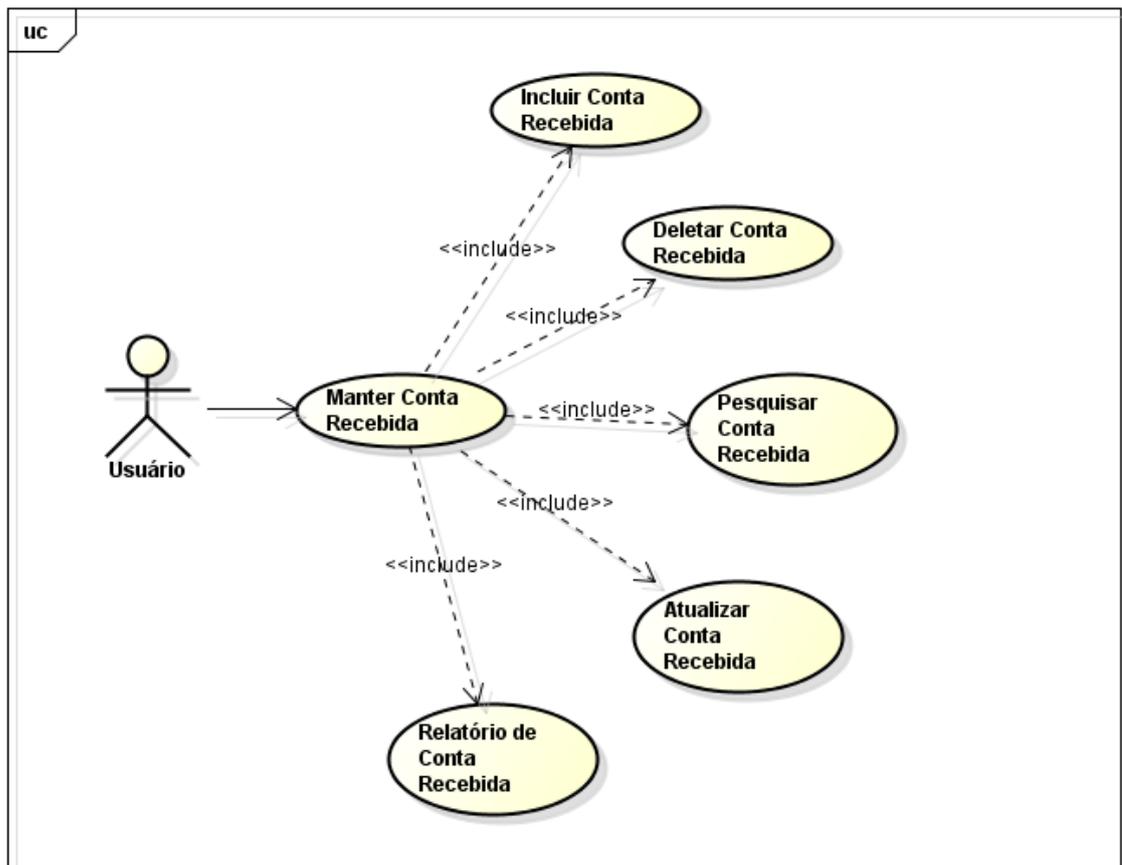
Quadro 24 – Atualizar pagas

Cenário: Atualizar pagas
Resumo: Este cenário é responsável em descrever como é o processo de edição de uma conta a pagar no sistema.
Atores: Usuário
Pré-Condições: Selecionar a conta que deve ser alterado dos registros e deve haver pelo menos um registro de cliente e um registro de tipo.
<p>Fluxo de eventos primário:</p> <ol style="list-style-type: none"> 1. O caso de uso se inicia no momento em que o usuário seleciona a opção “Alterar”. 2. Sistema busca dados no banco. 3. Sistema preenche os dados da conta nos campos referentes aos mesmos. 4. O usuário fornece ao sistema dados da conta, tais como fornecedor, tipo, valor, número do documento, data do pagamento, data do cadastro e status. 5. Usuário submete dados ao sistema. 6. O sistema verifica todas as informações fornecidas e insere as informações da conta a pagar no banco de dados. 7. Sistema retorna uma mensagem de “Conta alterada com sucesso”.
<p>Fluxo de eventos secundário:</p> <ul style="list-style-type: none"> • A qualquer momento antes de submeter às informações, o usuário pode selecionar cancelar. Os dados não são salvos e o caso de uso termina. • No caso 6, se alguma informação obrigatória não estiver preenchida ou incorreta o sistema retorna uma mensagem informando que deve ser alterado ou preenchido de forma correta os campos incorretos. • O sistema irá preencher automaticamente o campo data de cadastro com a data atual em que o usuário estará fazendo a interação com o programa.
Pós-condição: Os dados devem ser salvos no banco com todos os dados obrigatórios preenchidos.

Fonte: Elaborado pelos autores

O Quadro 24 apresenta o cenário de atualizar dos diagramas de caso de uso do módulo contas a pagar, utilizado no sistema.

Figura 13 - Diagrama de caso de uso de contas recebidas



Fonte: Elaborado pelos autores

A Figura 13 apresenta o diagrama de caso de uso do módulo contas a receber, utilizado no sistema.

Quadro 25 – Cadastrar recebidas

Cenário: Cadastro de recebidas
Resumo: Este cenário é responsável em descrever como é o processo de inclusão de uma conta a receber no sistema.
Atores: Usuário
Pré-Condições: Para o cadastro de Contas a receber deve haver pelo menos um registro de cliente.
<p>Fluxo de eventos primário:</p> <ol style="list-style-type: none"> 1. O caso de uso se inicia no momento em que o usuário seleciona “Contas A Receber”. 2. O usuário fornece ao sistema dados da conta, tais como cliente, valor, data do recebimento, data do cadastro e status. 3. Usuário submete dados ao sistema. 4. O sistema verifica todas as informações fornecidas e insere as informações da conta a receber no banco de dados. 5. Sistema retorna uma mensagem de “Conta salva com sucesso”.
<p>Fluxo de eventos secundário:</p> <ul style="list-style-type: none"> • A qualquer momento antes de submeter às informações, o usuário pode selecionar cancelar. Os dados não são salvos e o caso de uso termina. • No caso 4, se alguma informação obrigatória não estiver preenchida ou incorreta o sistema retorna uma mensagem informando que deve ser alterado ou preenchido de forma correta os campos incorretos. • O sistema irá preencher automaticamente o campo data de cadastro com a data atual em que o usuário estará fazendo a interação com o programa.
Pós-condição: Os dados devem ser salvos no banco com todos os dados obrigatórios preenchidos.

Fonte: Elaborado pelos autores

O Quadro 25 apresenta o cenário de cadastro dos diagramas de caso de uso do módulo contas a receber, utilizado no sistema.

Quadro 26 – Excluir recebidas

Cenário: Excluir recebidas
Resumo: Este cenário é responsável em descrever como é o processo de exclusão de uma conta a receber no sistema.
Atores: Usuário
Pré-Condições: Selecionar a conta a receber que deve ser excluído dos registros.
<p>Fluxo de eventos primário:</p> <ol style="list-style-type: none"> 1. O caso de uso se inicia no momento em que o usuário seleciona a opção “Deletar”. 2. Sistema retorna mensagem de confirmação. 3. Usuário confirma exclusão da conta. 4. Sistema deleta dados da conta a receber no banco de dados. 5. Sistema retorna mensagem “Conta deletada com sucesso”.
<p>Fluxo de eventos secundário:</p> <ul style="list-style-type: none"> • No caso 2, é a última opção que o usuário pode cancelar a exclusão da conta a receber.
Pós-condição: Não existem pós-condições.

Fonte: elaborada pelos autores

O Quadro 26 apresenta o cenário de excluir dos diagramas de caso de uso do módulo contas a receber, utilizado no sistema.

Quadro 27 – Consultar recebidas

Cenário: Consultar recebidas
Resumo: Este cenário é responsável em descrever como é o processo de pesquisa de contas a receber no sistema.
Atores: Usuário
Pré-Condições: Pesquisar contas a receber desejado no campo de “Pesquisa” para a consulta nos registros.
<p>Fluxo de eventos primário:</p> <ol style="list-style-type: none"> 1. O caso de uso se inicia no momento em que o usuário escolhe o campo de pesquisa e informa os dados que devem ser requisitados. 2. Sistema consulta informações no banco de dados ao mesmo tempo em que os dados são inseridos. 3. Sistema retorna lista de registros.
<p>Fluxo de eventos secundário:</p> <ul style="list-style-type: none"> • No caso 3, se não houver nenhum registro encontrado não irá retornar nenhuma informação. • O sistema irá pesquisar pelo os nomes de clientes.
Pós-condição: Não existem pós-condições.

Fonte: Elaborado pelos autores

O Quadro 27 apresenta o cenário de consultar dos diagramas de caso de uso do módulo contas a receber, utilizado no sistema.

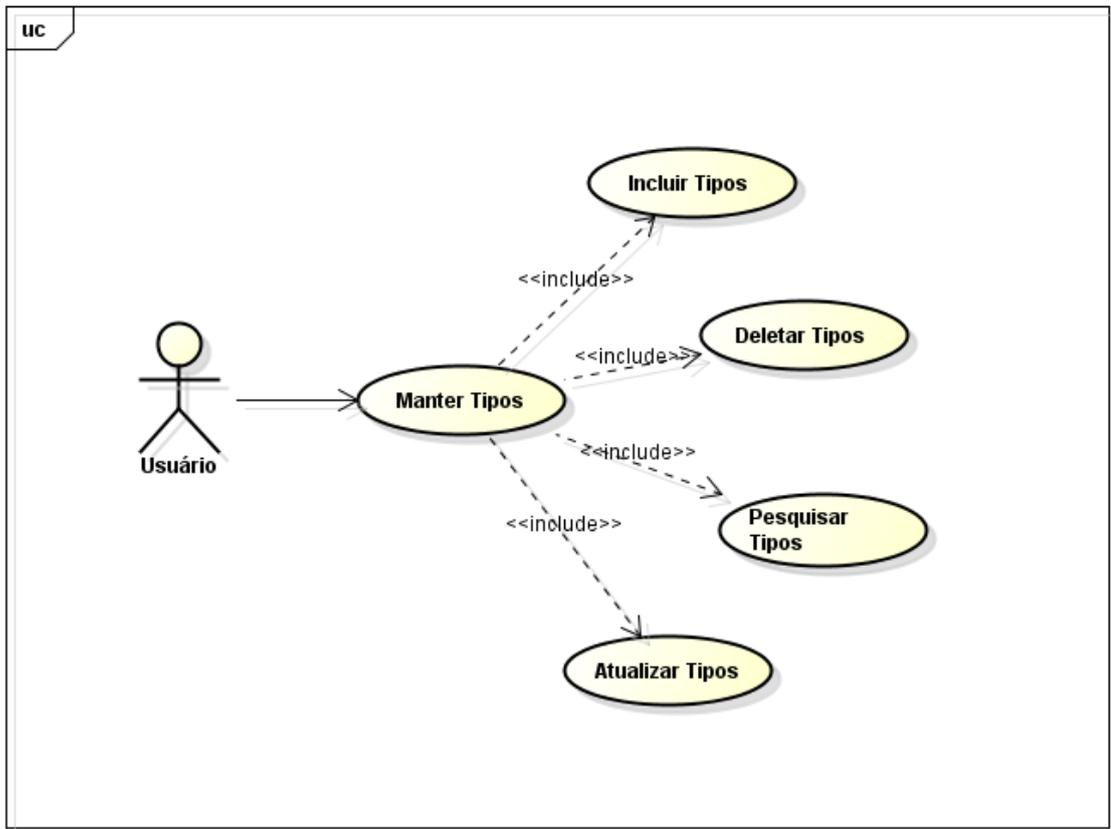
Quadro 28 – Atualizar recebidas

Cenário: Atualizar recebidas
Resumo: Este cenário é responsável em descrever como é o processo de edição de uma conta a receber no sistema.
Atores: Usuário
Pré-Condições: Selecionar a conta que deve ser alterado dos registros e deve haver pelo menos um registro de cliente.
<p>Fluxo de eventos primário:</p> <ol style="list-style-type: none"> 1. O caso de uso se inicia no momento em que o usuário seleciona a opção “Alterar”. 2. Sistema busca dados no banco. 3. Sistema preenche os dados da conta nos campos referentes aos mesmos. 4. O usuário fornece ao sistema da conta, tais como cliente, valor, data do recebimento, data do cadastro e status. 5. Usuário submete dados ao sistema. 6. O sistema verifica todas as informações fornecidas e insere as informações da conta a receber no banco de dados. 7. Sistema retorna uma mensagem de “Conta alterada com sucesso”.
<p>Fluxo de eventos secundário:</p> <ul style="list-style-type: none"> • A qualquer momento antes de submeter às informações, o usuário pode selecionar cancelar. Os dados não são salvos e o caso de uso termina. • No caso 6, se alguma informação obrigatória não estiver preenchida ou incorreta o sistema retorna uma mensagem informando que deve ser alterado ou preenchido de forma correta os campos incorretos. • O sistema irá preencher automaticamente o campo data de cadastro com a data atual em que o usuário estará fazendo a interação com o programa.
Pós-condição: Os dados devem ser salvos no banco com todos os dados obrigatórios preenchidos.

Fonte: Elaborada pelos autores

O Quadro 28 apresenta o cenário de atualizar dos diagramas de caso de uso do módulo contas a receber, utilizado no sistema.

Figura 14 - Diagrama de caso de uso de tipos



Fonte: Elaborado pelos autores

A Figura 14 apresenta o diagrama de caso de uso do módulo tipo de contas a pagar (CP), utilizado no sistema.

Quadro 29 – Cadastrar tipo

Cenário: Cadastro tipos
Resumo: Este cenário é responsável em descrever como é o processo de inclusão de um tipo no sistema.
Atores: Usuário
Pré-Condições: Não existem pré-condições.
<p>Fluxo de eventos primário:</p> <ol style="list-style-type: none"> 1. O caso de uso se inicia no momento em que o usuário seleciona “Tipos de CP”. 2. O usuário fornece ao sistema nome do tipo. 3. Usuário submete dados ao sistema. 4. O sistema verifica todas as informações fornecidas e insere as informações do tipo no banco de dados. 5. Sistema retorna uma mensagem de “Tipo salvo com sucesso”.
<p>Fluxo de eventos secundário:</p> <ul style="list-style-type: none"> • A qualquer momento antes de submeter às informações, o usuário pode selecionar cancelar. Os dados não são salvos e o caso de uso termina. • No caso 4, se alguma informação obrigatória não estiver preenchida ou incorreta o sistema retorna uma mensagem informando que deve ser alterado ou preenchido de forma correta os campos incorretos.
Pós-condição: Os dados devem ser salvos no banco com todos os dados obrigatórios preenchidos.

Fonte: Elaborada pelos autores

O Quadro 29 apresenta o cenário de cadastro dos diagramas de caso de uso do módulo tipos de contas a pagar (cp), utilizado no sistema.

Quadro 30 – Excluir tipo

Cenário: Excluir tipo
Resumo: Este cenário é responsável em descrever como é o processo de exclusão de um tipo no sistema.
Atores: Usuário
Pré-Condições: Selecionar o tipo que deve ser excluído dos registros.
<p>Fluxo de eventos primário:</p> <ol style="list-style-type: none"> 1. O caso de uso se inicia no momento em que o usuário seleciona a opção “Deletar”. 2. Sistema retorna mensagem de confirmação. 3. Usuário confirma exclusão do tipo. 4. Sistema deleta dados do tipo no banco de dados. 5. Sistema retorna mensagem “Tipo excluído com sucesso”.
<p>Fluxo de eventos secundário:</p> <ul style="list-style-type: none"> • No caso 2, é a última opção que o usuário pode cancelar a exclusão do tipo.
Pós-condição: Não existem pós-condições.

Fonte: Elaborada pelos autores

O Quadro 30 apresenta o cenário de excluir dos diagramas de caso de uso do módulo tipos de contas a pagar (cp), utilizado no sistema.

Quadro 31 – Consultar tipos

Cenário: Consultar tipo
Resumo: Este cenário é responsável em descrever como é o processo de pesquisa de tipos no sistema.
Atores: Usuário
Pré-Condições: Pesquisar tipo desejado no campo de “Pesquisa” para a consulta nos registros.
<p>Fluxo de eventos primário:</p> <ol style="list-style-type: none"> 1. O caso de uso se inicia no momento em que o usuário escolhe o campo de pesquisa e informa os dados que devem ser requisitados. 2. Sistema consulta informações no banco de dados ao mesmo tempo em que os dados são inseridos. 3. Sistema retorna lista de registros.
<p>Fluxo de eventos secundário:</p> <ul style="list-style-type: none"> • No caso 3, se não houver nenhum registro encontrado não irá retornar nenhuma informação. • O sistema irá pesquisar o nome do tipo.
Pós-condição: Não existem pós-condições.

Fonte: Elaborada pelos autores

O Quadro 31 apresenta o cenário de consultar dos diagramas de caso de uso do módulo tipos de contas a pagar (cp), utilizado no sistema.

Quadro 32 – Atualizar tipo

Cenário: Atualizar tipo
Resumo: Este cenário é responsável em descrever como é o processo de edição de um tipo no sistema.
Atores: Usuário
Pré-Condições: Selecionar o tipo que deve ser alterado dos registros.
<p>Fluxo de eventos primário:</p> <ol style="list-style-type: none"> 1. O caso de uso se inicia no momento em que o usuário seleciona a opção “Alterar”. 2. Sistema busca dados no banco. 3. Sistema preenche o nome do tipo selecionado. 4. O usuário fornece ao sistema o nome do tipo. 5. Usuário submete dados ao sistema. 6. O sistema verifica todas as informações fornecidas e altera as informações do tipo no banco de dados. 7. Sistema retorna uma mensagem de “Tipo alterado com sucesso”.
<p>Fluxo de eventos secundário:</p> <ul style="list-style-type: none"> • A qualquer momento antes de submeter às informações, o usuário pode selecionar cancelar. Os dados não são alterados e o caso de uso termina. • No caso 6, se alguma informação obrigatória não estiver preenchida ou incorreta o sistema retorna uma mensagem informando que deve ser alterado ou preenchido de forma correta os campos incorretos.
Pós-condição: Os dados devem ser salvos no banco com todos os dados obrigatórios preenchidos.

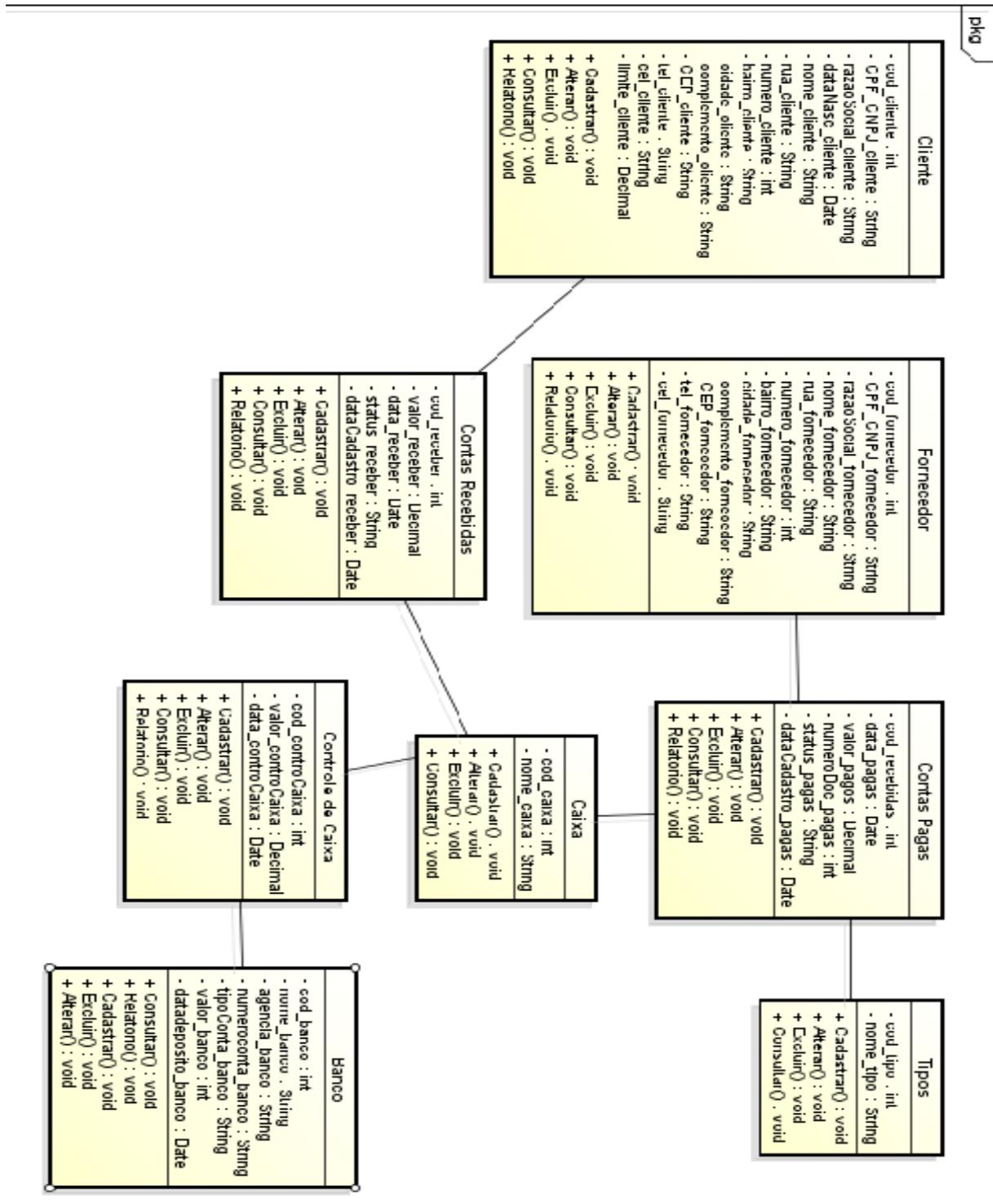
Fonte: Elaborado pelos autores

O Quadro 32 apresenta o cenário de atualizar dos diagramas de caso de uso do módulo tipos de contas a pagar (cp), utilizado no sistema.

4.3.5 Diagrama de classes

A Figura 15 apresenta o diagrama de classes do sistema desenvolvido.

Figura 15 – Digrama de classes do sistema



Fonte: Elaborado pelos autores

4.4 PROTOTIPAÇÃO

A interação do usuário com o sistema será feita por interfaces gráficas, que em conjunto da funcionalidade de gerar relatórios são as principais forma do sistema se tornar útil e bem sucedido na empresa. O sistema foi desenvolvido atendendo os requisitos funcionais focando na simplicidade, clareza e em uma interface intuitiva que melhora a interação do usuário com o programa, assim o mesmo pode executar tarefas de forma rápida e eficaz. Caso ocorra algum erro o sistema informa ao usuário permitindo a recuperação das informações.

As telas atuais do programa são constituídas por características, tais como, cadastro, consulta e geração de relatórios. No decorrer desse tópico, são expostas as principais telas que compõem o *software*.

As telas que constituem o *software* são semelhantes diferenciando-as basicamente pelos campos que existem em cada uma delas, tais telas são nomeadas como: Cliente, Fornecedor, Contas a Receber, Contas a Pagar, Caixa, Diário do Caixa, Tipo de CP, Depósitos Bancários e Relatórios. É possível realizar o CRUD (*Create Read Update e Delete*), em todas as telas citadas, são realizadas operações básicas de criar, ler, atualizar e excluir com o banco de dados (MUNHOZ, 2014).

A seguir é apresentada a tela principal, em que o usuário pode escolher o acesso que desejar nas funcionalidades do sistema.

Figura 16 – Tela de cadastro de cliente



Fonte: Elaborada pelos autores

Na Figura 16, o usuário poderá escolher a opção de ir para os diferentes cadastros e geração de relatórios. A tela é constituída de 9 botões, com as funcionalidades a seguir.

- Botão FORNECEDOR: função de abrir a tela de cadastro de fornecedores.
- Botão CLIENTE: função de abrir a tela de cadastro de clientes.
- Botão CONTAS A RECEBER: função de abrir a tela de cadastro de contas a receber.
- Botão DIÁRIO DO CAIXA: função de abrir a tela de cadastro dos diários dos caixas.
- Botão CAIXA: função de abrir a tela de cadastro de caixas.
- Botão TIPO DE CP: função de abrir outra a de cadastro de tipos de contas a pagar.
- Botão CONTAS A PAGAR: função de abrir a tela de cadastro de contas a pagar.
- Botão DEPÓSITOS BANCÁRIOS: função de abrir a tela de cadastro de depósitos bancários.
- Botão RELATÓRIOS: função de abrir a tela de relatórios.

A maioria das telas presentes no *software* está com estrutura padrão, para cadastro e consulta, com exceção das telas de “Caixa” e “Tipo de CP”. As telas possuem divisões de abas claras e coesas de consulta e cadastro para o usuário, com botões simples e interativos, facilitando a utilização do programa.

A Figura 17 ilustra a tela de cadastro de cliente, no qual é feita a gravação e alteração de registros referentes ao cliente.

Figura 17 – Tela de cadastro de cliente

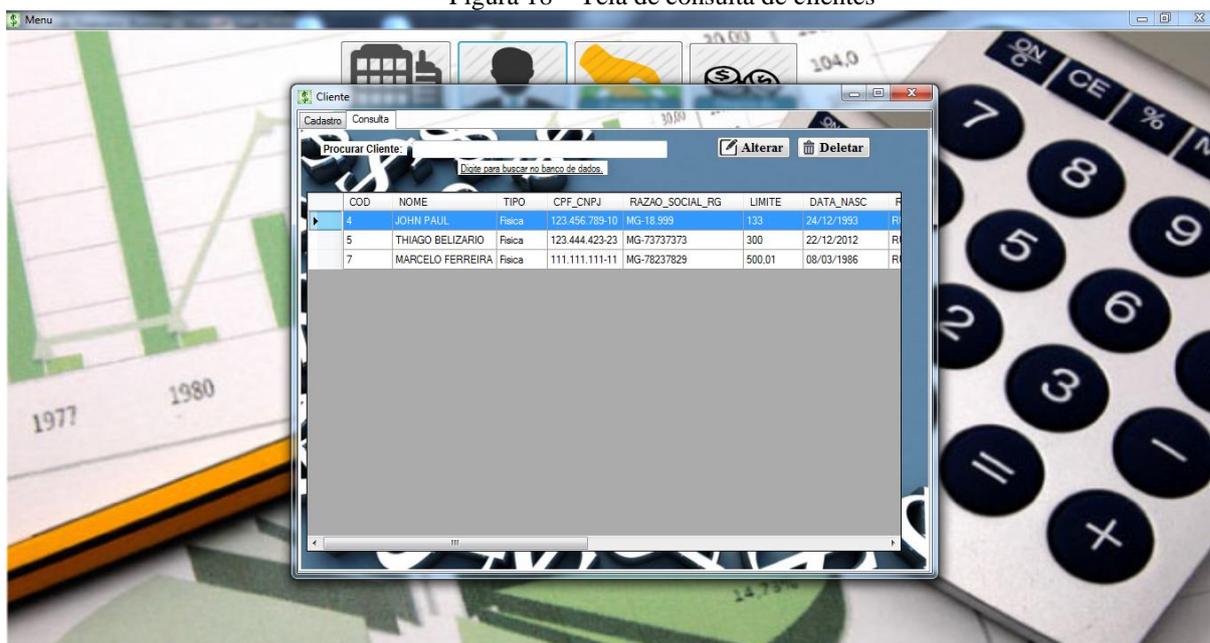
The image shows a software interface for client registration. The main window is titled 'Cliente' and has two tabs: 'Cadastro' (active) and 'Consulta'. The form is divided into two main sections: 'Dados Obrigatorios' and 'Dados Adicionais'.
Dados Obrigatorios:
- Nome: [Text Field]
- Nascimento: [Date Mask] Tipo: [Dropdown]
- CPF/CNPJ: [Text Field]
- Razão Social/RG: [Text Field]
- Limite: [Text Field]
Dados Adicionais:
- Logradouro: [Text Field]
- Numero: [Text Field]
- Bairro: [Text Field]
- Cidade: [Text Field]
- Complemento: [Text Field]
- CEP: [Text Field]
- Telefone: [Text Field]
- Celular: [Text Field]
At the bottom of the form are three buttons: 'Salvar', 'Alterar', and 'Limpar'.

Fonte: Elaborada pelos autores

- Combo Box (caixa de combinação ao lado de tipo): aparece uma caixa de combinação que deve ser escolhida entre “Física” ou “Jurídica”, no qual altera os textos, conforme a escolha e as máscaras que estão presentes nos campos de CPF/CNPJ e RAZÃO SOCIAL/RG, além de deixar os campos ativos para o usuário digitar.
- Botão Salvar: utilizado para salvar o novo registro no banco de dados na tabela de cliente (TBCLIENTE).
- Botão Alterar: utilizado para alterar um registro no banco de dados na tabela de cliente (TBCLIENTE), entretanto ele só é disponível para o uso, caso seja solicitado na aba de consulta, quando solicitado esse botão fica disponível para o uso e o botão salvar fica indisponível.
- Botão Limpar: utilizado para apagar todos os campos da tela de cadastro de cliente.

A Figura 18 mostra a tela de consulta de clientes, na qual é feita a pesquisa e exclusão de todos os clientes e também é o local onde o usuário pode selecionar o registro do cliente a ser alterado.

Figura 18 – Tela de consulta de clientes



Fonte: Elaborado pelos autores

- Campo de pesquisa: utilizado para pesquisar cliente, não tem a necessidade de clicar e outro botão para realizar a consulta, o sistema já pesquisa conforme o usuário vai digitando os caracteres.
- Botão Deletar: utilizado para excluir um registro no banco de dados na tabela de cliente (TBCLIENTE). O usuário deve escolher o registro do cliente, o programa ainda retorna uma confirmação de exclusão.
- Botão Alterar: utilizado para preencher os dados na aba de cadastro, para que os mesmos possam ser alterados de acordo com a necessidade do usuário, para que esse botão funcione deve-se escolher um registro de cliente.

As telas de “Caixa” e “Tipo de CP” diferem das demais simplesmente, pelo fato de não haver uma divisão de abas entre cadastro e consulta, ou seja, tanto o cadastro como a consulta, está presente para o usuário de uma só vez, como é ilustrado na Figura 19.

Figura 19 – Tela de cadastro de caixa

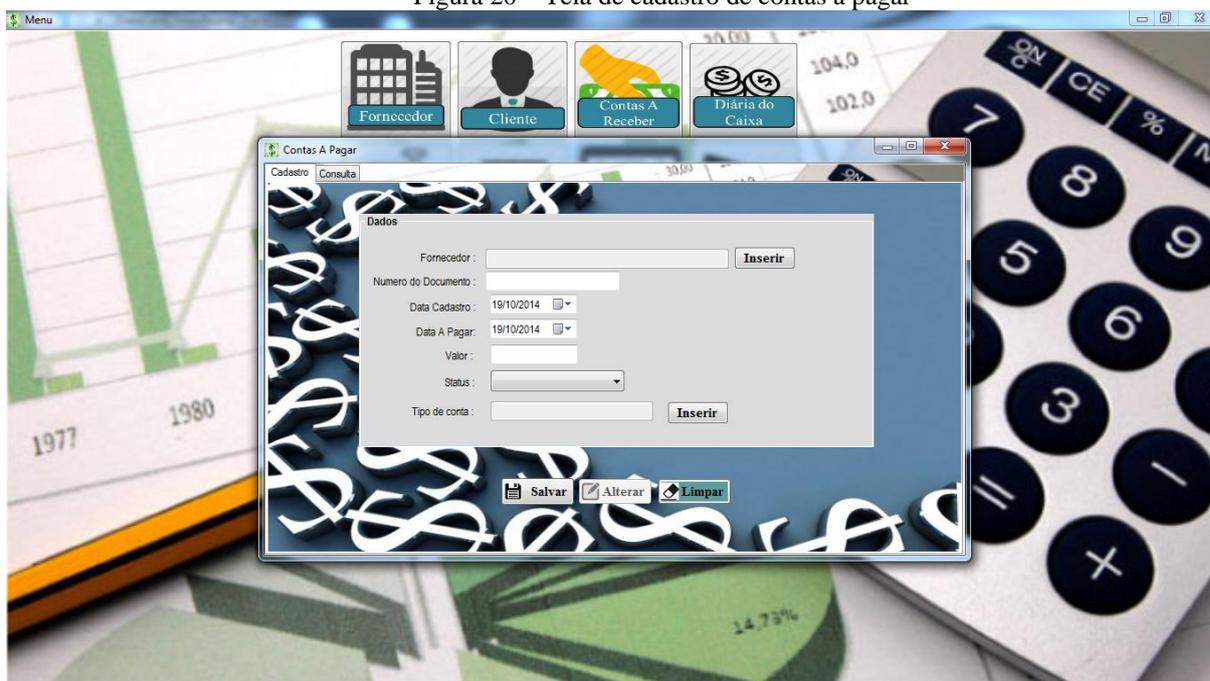


Fonte: Elaborado pelos autores

Apesar do Layout dessa tela ser um pouco diferente das demais, a utilização dos botões ainda são as mesmas. A mudança do layout da tela de CAIXA e TIPO de CP é justificada pela quantidade de dados a serem preenchidos, no caso deles apenas um campo (nome) para ser completado pelo usuário.

As telas de Contas A PAGAR e A RECEBER, possuem uma adição de um botão para melhorar a interação do *software* com o usuário, como apresentado na Figura 20, na qual mostra a tela de contas a pagar.

Figura 20 – Tela de cadastro de contas a pagar



Fonte: Elaborado pelos autores

- Botão Inserir (ao lado de fornecedor): mostra a tela com todos os fornecedores e um campo para pesquisa para o usuário selecionar o registro desejado.
- Botão Inserir (ao lado de tipo de conta): mostra a tela com todos dos tipos de cp (contas a pagar) e um campo para pesquisa para o usuário selecionar o registro desejado.

Os botões não tem distinção, entretanto, deixa claro para o usuário de forma instintiva para a funcionalidade de cada, ou seja, que o primeiro botão “Inserir” é para adicionar um fornecedor e o segundo para a adição de um tipo de cp. Na tabela de contas a receber, possui apenas um botão com as mesmas funcionalidades, entretanto é utilizado para adicionar um cliente.

Para a emissão de relatório no *software*, elaborou-se uma tela para o usuário escolher a forma de geração de relatórios de acordo com sua necessidade, como apresentado na Figura 21.

Figura 21 – Tela de relatórios



Fonte: Elaborado pelos autores

Como pode-se observar na Figura 21, há diversas formas do usuário gerar relatórios, sendo que foi elaborado um único padrão para todas as 5 abas da tela de RELATÓRIOS, sendo composta por 2 opções de escolha no primeiro grupo de dados e 3 no segundo. Deve-se ressaltar que só tem como fazer uma escolha em cada grupo de dados.

- Todos (primeira opção do primeiro grupo): ao escolher essa opção ao gerar relatório, ele irá criar um relatório em uma tela com todos os clientes.
- Escolher: ao escolher essa opção ao gerar relatório, ele irá criar um relatório em uma tela com um cliente em específico, ele irá liberar o botão “Inserir” para a utilização.
- Botão Inserir (primeira opção do primeiro grupo): mostra a tela com todos os clientes e um campo para pesquisa para o usuário selecionar o registro desejado.
- Todos (primeira opção do segundo grupo): ao escolher essa opção ao gerar relatório, irá criar um relatório em uma tela com todas as datas, sem exceção.
- Do Dia: ao escolher essa opção ao gerar relatório, irá criar um relatório em uma tela com uma data específica, e assim, irá liberar o campo de data (ao lado) para a utilização.
- Do Mês: ao escolher essa opção ao gerar relatório, irá criar um relatório em uma tela com mês e ano específico, e assim, liberar o campo mês (ao lado) e ano (ao lado) para a utilização.

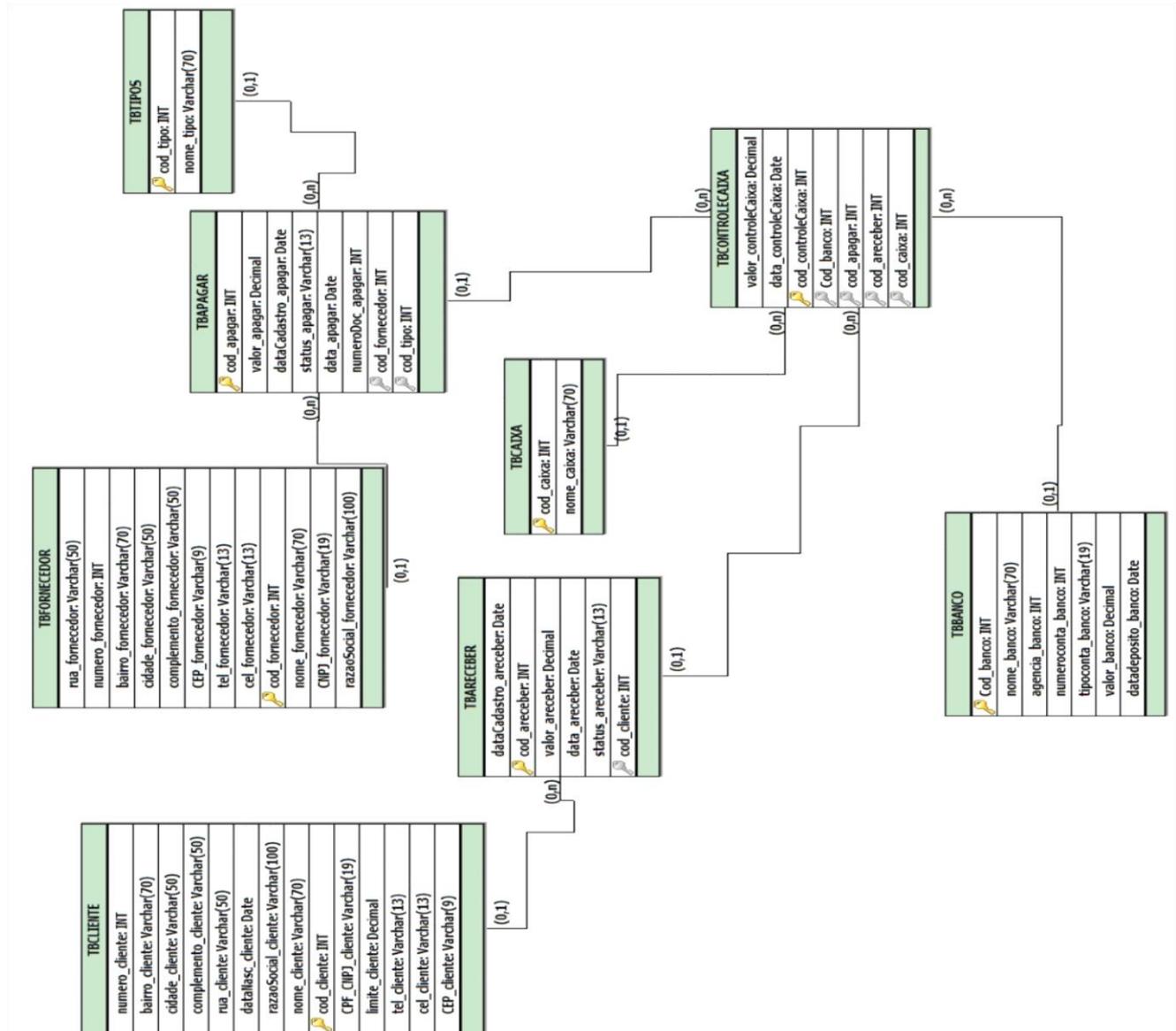
- Botão Gerar: o botão gerar é utilizado para combinar as informações dos dois grupos e chamar o relatório que irá fornecer os dados requisitados pelo usuário.

4.5 MODELAGEM DE DADOS

Para a modelagem de dados foi utilizado o Modelo Entidade Relacional (MER), pois através dele que é mostrada a estrutura lógica do banco de dados.

Considerando que todo objeto ou entidade tem propriedades únicas e particulares, sendo que tais propriedades são denominadas atributos. No MER, são apresentadas várias tabelas interligadas, dentre elas, temos a tabela cliente, no qual serão gravadas informações referentes ao cliente, assim os dados são armazenados em campos específicos (atributos) da tabela (entidade) no banco de dados.

Figura 22 – Modelo entidade relacional



Fonte: Elaborado pelos autores

Como apresentado no MER na Figura 22, o banco de dados possui oito tabelas com ligações necessárias para as principais necessidades do módulo financeiro direcionado para micro e pequenas empresas.

A tabela TIPO, é a responsável para cadastrar os diferentes tipos de contas a pagar, assim podem-se cadastrar diferentes formas de pagamentos de acordo com a necessidade do usuário, é necessário que haja pelo menos um registro dela para que possa ser realizado algum

cadastro na tabela APAGAR, além de ser necessário um fornecedor gravado na tabela FORNECEDOR.

As informações de contas a receber são inseridas na tabela ARECEBER, no qual se deve possuir pelo menos um registro de cliente na tabela CLIENTE.

A diária do caixa é a única tabela que insere dados diretamente na tabela CONTROLECAIXA e tem a necessidade de haver pelo menos um registro na tabela CAIXA, pois as demais tabelas inserem primeiramente nelas mesmas e posteriormente na tabela CONTROLECAIXA, essas tabelas são ARECEBER, APAGAR e BANCO.

A tabela CONTROLECAIXA é a responsável por interligar a maioria das tabelas, no qual armazena valores financeiros referentes às mesmas. Assim é possível fazer o controle de toda a movimentação de dinheiro na empresa.

5 RESULTADOS OBTIDOS

Os resultados obtidos do desenvolvimento do *software* de acordo com os requisitos propostos foram satisfatórios. O sistema foi aprovado pelos proprietários Sônia Aparecida de Miranda Leite e Pedro Paulo Lopes da Silva, da empresa em que foi feita a análise de requisitos, Bar e Mercearia Novo Cruzeiro LTDA.

O programa desenvolvido procurou eliminar a utilização de recursos manuais e outros meios menos convencionais para a gestão da empresa, focando na simplicidade, confiabilidade dos dados, rapidez e eficácia.

A prototipação das telas procurou melhorar a interação do usuário com o sistema, através de botões intuitivos a fim de facilitar o manuseio do *software*. Com a instalação do *software*, a interface em conjunto aos relatórios são as principais formas do *software* se tornar útil.

A aprovação do *software* se deu através do uso do sistema pelos proprietários, em que foram testadas suas funcionalidades e sua interface. Esse teste foi feito em um dos computadores dos desenvolvedores, pois o sistema não foi instalado na empresa por não possuir um computador com os requisitos mínimos para execução do *software*.

6 CONCLUSÃO

No decorrer do trabalho, observa-se que a informatização das empresas tornou-se algo essencial, uma vez que esta deseja crescer e se manter no mercado é necessário investir em sistemas que auxiliam as suas gestões organizacionais.

Acredita-se que ao empregar o sistema desenvolvido uma micro ou pequena empresa que não aproveitava nenhum sistema ou utilize um *software* obsoleto, será de bom caráter para a gestão organizacional e relação aos dados financeiros da mesma.

No desenvolvimento do *software*, houve o foco principalmente a interação do usuário com o sistema, melhorando assim a sua usabilidade, simplificando ao máximo os processos do CRUD oferecidos pelo sistema. Espera-se que o projeto, possa servir como base para o desenvolvimento de um ERP completo ou parte dele, direcionado para as micro e pequenas empresas, devido à falta de programas com essa finalidade para empresas desse porte no mercado.

Conclui-se que para uma empresa ter sucesso no mercado, é necessário um controle organizacional, a partir desse contexto o ERP se torna uma peça-chave auxiliando o gestor na tomada de decisão, mas salienta-se que um sistema com essas funcionalidades por si só, não trará resultados imediatos, pois é um meio de controle de informações da empresa, para que o gestor visualize todos os dados relacionados de forma fácil e clara.

7 TRABALHOS FUTUROS

Como possíveis trabalhos futuros, podemos apontar:

- O desenvolvimento e implantação de um módulo de vendas na empresa, uma vez que ela apresenta um sistema com a mesma finalidade, mas há viabilidade no desenvolvimento desse módulo, pois pode ser feito de acordo com as necessidades do cliente (proprietário). Esse módulo será o principal responsável em controlar as vendas feitas pela empresa, além de auxiliar o fechamento do caixa diário, fornecendo informações essenciais para o módulo financeiro desenvolvido no projeto.
- A criação e implantação de um módulo de estoque: apesar da empresa também apresentar um sistema que possui essa funcionalidade, ainda é viável o desenvolvimento dele, pois em conjunto com o módulo de vendas pode-se ter acesso a

quantidade de produtos armazenados pela empresa sem a necessidade de ir ao setor e calcular a quantidade.

- Desenvolver um módulo de compras, responsável em armazenar as compras feitas pela empresa, assim terá uma forma de controlar as compras, além de alimentar o módulo de estoque com a chegada de mercadorias. Em conjunto com os módulos citados anteriormente, fornece dados que são imprescindíveis para gestão organizacional da empresa.

REFERÊNCIAS

ASTAH. **Astah community**. Disponível em: <<http://astah.net/editions/community>>. Acesso em: 18 out. 2014.

BOOCH, Grady; RUMBAUGH; JACOBSON, Ivar. **UML: Guia Do Usuário**. Rio de Janeiro: Editora Elsevier, 2005.

CRUZ, Vera Lúcia. **Avaliação de benefícios de pós-implantação de sistemas ERP no fornecimento de informação contábil, financeira e operacional em empresas de pequeno e médio porte da região metropolitana de recife**. Disponível em: <<http://www.contabeis.ufpe.br/ppgcontabeis/images/documentos/Dissertacoes/vera%20lucia.pdf>>. Acesso em: 22 out. 2014.

DATE, C. J. **Introdução a sistemas de bancos de dados**. Rio de Janeiro: Campus, 2000. Disponível em: <<http://books.google.com.br/books?hl=pt-BR&lr=&id=xBeO9LSIK7UC&oi=fnd&pg=PP23&dq=Introdu%C3%A7%C3%A3o+a+sistemas+de+bancos+de+dados&ots=xaLzj1Af7J&sig=UVF83BtjUdjzA0fYRHkGIgC6X6AY#v=onepage&q&f=false>>. Acesso em: 28 maio 2014.

FIREBIRD FOUNDATION INCORPORATED. **Sobre o Firebird**. Disponível em: <<http://www.firebirdsql.org/en/about-firebird/>>. Acesso em: 12 jun. 2014.

GUEDES, André Luiz Azevedo. **ERP – Enterprise Resource Planning**. Disponível em: <http://sites.google.com/site/andreguedes/ERP_Enterprise_Resource_Planning.pdf>. Acesso em: 08 mar. 2014.

JR, Jorge Muniz; MAIA, Flavia Gabriele Manoel; VIOLA, Gian. **Os Principais Trabalhos Na Teoria Do Conhecimento Tácito: Pesquisa Bibliométrica 2000-2011**. Disponível em: <http://www.simpoi.fgvsp.br/arquivo/2011/artigos/E2011_T00197_PCN17366.pdf>. Acesso em: 30 maio 2014.

LAUDON, Kenneth; LAUDON, Jane. **Sistemas de informação gerenciais**. 9. ed. São Paulo: Pearson Prentice Hall, 2011.

LOVATO, André. **Pesquisa sobre a viabilidade do desenvolvimento de um sistema de ERP baseado em tecnologias de software livre/códigos aberto**. Disponível em: <<http://fateczl.edu.br/TCC/2009-1/tcc-03.pdf>>. Acesso em: 07 mar. 2014.

MICROSOFT. **Desenvolvimento de aplicativos**. Disponível em: <<http://www.visualstudio.com/pt-br/explore/application-development-vs>>. Acesso em: 12 jun. 2014.

MICROSOFT. **O que é o Excel ?**. Disponível em: <<http://office.microsoft.com/pt-br/novice/o-que-e-o-excel-HA010265948.aspx>>. Acesso em: 18 out. 2014.

MUNHOZ, Edmar. **Criando um CRUD em C#**. Disponível em: <<http://www.devmedia.com.br/criando-um-crud-em-c/30384>>. Acesso em: 19 out. 2014.

NETO, Esmerino Toscano de Brito. **Implantação de sistemas E.R.P. (Enterprise Resources Planning)** - Um estudo de caso. Disponível em: <[http://201.2.114.147/bds/bds.nsf/69a5e2bb919eaf2e832574b0004bda60/365c13f9f1a6af3383256e84006cd8ba/\\$FILE/Implata%C3%A7%C3%A3o%20de%20Sistemas%20R.%20E.%20P..pdf](http://201.2.114.147/bds/bds.nsf/69a5e2bb919eaf2e832574b0004bda60/365c13f9f1a6af3383256e84006cd8ba/$FILE/Implata%C3%A7%C3%A3o%20de%20Sistemas%20R.%20E.%20P..pdf)>. Acesso em: 15 mar. 2014.

OLIVEIRA, Gláucia Nalva Borges de. **Um Modelo Processo de Implantação de Sistemas ERP**. Disponível em: <http://www.tcc.sc.usp.br/tce/disponiveis/97/970010/tce-21052010-101612/publico/Oliveira_Glucia_Nalva_Borges_de.pdf>. Acesso em: 20 mar. 2014

PAMPLONA, Leonardo Filipe. **Sistemas de informação para controle financeiro de uma microempresa via web**. Disponível em: <<http://campeche.inf.furb.br/tccs/2010-I/TCC2010-1-16-VF-LeonardoFPamplona.pdf>>. Acesso em: 22 out. 2014.

SILBERSCHATZ, Abraham.; KORTH, Henry F.; S. SUDARSHAN. **Sistema de Banco de Dados**. 3. ed. São Paulo: Makron Books, 1999.

SILVA, Edna Lúcia da; MENEZES, Estera Muszkat. **Metodologia da Pesquisa e Elaboração de Dissertação**. 4ª edição revisada e atualizada. Disponível em: <http://www.unisc.br/portal/upload/com_arquivo/metodologia_da_pesquisa_e_elaboracao_de_dissertacao.pdf>. Acesso em: 05 abr. 2014.

SOMERVILLE, Ian. **Engenharia de Software**. 8. ed. São Paulo. Pearson Addison, 2010.

SOUZA, Cesar Alexandre de; SACCOL, Amarolinda Zanela. **Sistemas ERP no Brasil (Enterprise Resource Planning) Teoria e Casos**. São Paulo: Atlas, 2009.

SOUZA, Cesar Alexandre de. **Sistemas integrados de gestão empresarial: estudos de casos de implementação de sistemas ERP**. Disponível em: <http://www.famescbji.edu.br/famescbji/biblioteca/livros_adm/sistemas_erp.pdf>. Acesso em: 25 mar. 2014.

WENER, Carlos Eduardo. **Ferramenta de gerenciamento para o banco de dados Firebird**. Disponível em: <http://www.bc.furb.br/docs/MO/2004/279050_1_1.pdf>. Acesso em: 27 maio 2014.