

CAPÍTULO 1 – INTRODUÇÃO

1.1 - Contexto

O motivo da escolha desse tema para o trabalho de final curso tem como base os mais de dez anos que atuo nesta área. Neste dez anos vi a grande evolução da informática nos grandes atacados de material de construção e isso trouxe uma grande facilidade tanto para o cliente, como para os vendedores e ao dono de empresa.

No entanto, nem sempre a informática é empregada de forma correta, desenvolvendo-se sistemas que não atendem às necessidades dos clientes e dos usuários. Isto se deve em parte à falta de um desenvolvimento adequado do software para esses sistemas. Por isso venho por meio deste trabalho, mostrar como se deve fazer um projeto bem elaborado para um sistema.

1.2 - Proposta

Neste trabalho deseja-se fazer o projeto de um sistema que possa ser empregado em atacados em material de construção, elétrico e hidráulico. Sendo somente abordada neste trabalho a área de vendas deste sistema. Para tanto, será feito um estudo do paradigma da Orientação a Objeto, do padrão UML(Unified Modeling Language) e a metodologia RUP (Rational Unified Process), aplicando os conceitos adquiridos na disciplina engenharia de software I e II, feita no decorrer do curso de ciência da computação.

Em relação ao sistema, será produzido análise e projeto, mas será abordado com uma maior ênfase a parte de análise, onde o analista de sistema buscará todas as informações com o cliente para que possa estar descrevendo tudo aquilo que o cliente necessita.

Dispostos de todas as informações o analista de sistema partirá para a elaboração dos diagramas, com a elaboração do mesmo, será desenvolvido o projeto, mas essa parte não será

abordada com tanta ênfase neste trabalho. Com a elaboração da análise e projeto do sistema, um programador terá uma facilidade maior em se desenvolver este projeto.

1.3 - Objetivo

Os principais objetivos gerais e específicos desse trabalho são:

- Realizar um estudo sobre o desenvolvimento de software orientado a objetos, o padrão UML e a metodologia RUP;
- Identificar as atividades que compõem o processo de controle de vendas em uma loja de material de construção; e
- Desenvolver a análise do software de sistema de vendas de uma loja de material de construção, empregando os conceitos estudados.

1.4 - Organização da Monografia

Neste documento, onde são apresentados os resultados do desenvolvimento deste trabalho, está organizado da seguinte forma:

- Capítulo 1: apresenta-se uma introdução aos motivos que levaram ao desenvolvimento deste trabalho;
- Capítulo 2: será apresentada as definições das tecnologias RUP e UML utilizadas no trabalho, serão apresentadas todas as fases de cada tema citado e de como elas elaboradas;
- Capítulo 3: será apresentada uma introdução ao sistema, aonde será mostrado como a empresa era antes da informatização e o que será feito;
- Capítulo 4: será apresentada a modelagem do sistema, onde constará todo o desenvolvimento descrito no capítulo 2;
- Capítulo 5: será apresentada a conclusão do trabalho e sua futura evolução.

CAPÍTULO 2 – PESQUISA BIBLIOGRÁFICA

Neste capítulo será apresentados os conceitos de RUP e UML, afim de trazer um entendimento melhor sobre o que sendo mostrado pelo trabalho. Com esse capítulo poderá se ter uma idéia de como será conduzido o trabalho seguindo as metodologias citadas acima.

2.1 – UML

Para a produção de novos sistemas utilizando orientação a objeto nas fases de análise de requisito, análise de sistemas e design existem grandes problemas, pois não há uma padronização eficaz que abranja qualquer aplicação. A confusão se torna grande, quando cada simbologia utiliza-se de um tipo de notação, como: seus próprios conceitos, gráficos e terminologias.

A UML não é simplesmente uma padronização de notação, mas sim um forma de que se tenha o conhecimento para que se aprenda a modelar utilizando-se orientação a objeto no estado da arte. [Booth ET AL,1999]

A UML foi desenvolvida por três amigos, Grady Booch, James Rumbaugh e Ivar Jacobson, eles possuem um grande conhecimento na área de orientação a objeto, pois as três grandes metodologias orientado a objeto foram eles que desenvolveram e a UML é a junção do que havia de melhor nestas três metodologias.

2.1.1 – UML – A unificação dos métodos para a criação de um novo padrão

A UML é uma tentativa de padronizar a modelagem orientada a objetos de uma forma que qualquer sistema, seja qual for o tipo, possa ser corretamente, com consistência, fácil de se comunicar com outras aplicações, simples de ser atualizado e compreensível. [Booth ET AL,1999]

Existiam várias metodologias de modelagem orientada a objeto cada uma com sua própria notação que até o surgimento da UML causavam uma guerra entre os desenvolvedores de orientado a objeto. A UML acabou com a guerra pegando as melhores idéias de cada metodologia e mostrando como seria a mudança de cada uma para a UML. [Booth ET AL,1999]

Será mostrado logo abaixo as principais metodologias que se tornaram populares nos anos 90: [Booth ET AL,1999]

- Booch – O método de Grady Booch para desenvolvimento orientado a objeto está disponível em muitas versões. Booch definiu a noção de que um sistema é analisado a partir de um número de visões, onde cada visão é descrita por um número de modelos e diagramas. O método de Booch também trazia uma simbologia complexa de ser desenhada a mão, continha também o processo pelo quais sistemas são analisados por macro e micro visões.
- OMT – Técnica de Modelagem de Objetos é um método desenvolvido pela GE (General Electric) onde James Rumbaugh trabalhava. O método é especialmente voltado para o teste dos modelos, baseado nas especificações da análise de requisitos do sistema. O modelo total do sistema baseado no método OMT é composto pela junção dos métodos de objetos, funcional e use-cases.
- OOSE/Objectory – Os métodos OOSE e o Objectory foram desenvolvidos baseados no mesmo ponto de vista formado por Ivar Jacobson. O método OOSE é a visão de Jacobson de um método orientado a objeto, já o Objectory é usado para a construção de sistemas tão diversos quanto eles forem. Ambos os métodos são baseados na utilização de use-case, que definem os requisitos iniciais do sistema, vistos por um ator externo. O método Objectory também foi adaptado para a engenharia de negócios, onde é usado para modelar e melhorar os processos envolvidos no funcionamento de empresas.

Os métodos possuem cada um sua própria notação (seus próprios símbolos para representar modelos orientados a objeto), processos (que atividades são desenvolvidas em diferentes partes do desenvolvimento) e ferramentas (as ferramentas CASE que suportam cada uma destas notações e processos). [Booth ET AL,1999]

Com a diversidade de conceitos, “os três amigos” , Grady Booch, James Rumbaugh e Ivar Jacobson, resolveram criar a Linguagem de Modelagem Unificada. Primeiramente eles

distribuíram diversas versões da UML (uma versão beta) para os desenvolvedores e com isso eles obtiveram novas idéias sugeridas pelos desenvolvedores que melhorou ainda mais a linguagem.

Os objetivos da UML são: [Booth ET AL,1999]

- A modelagem de sistemas (não apenas de software) usando os conceitos da orientação a objeto;
- Estabelecer uma união fazendo com que métodos conceituais sejam também executáveis;
- Criar uma linguagem de modelagem usável tanto pelo homem quanto pela máquina.

2.1.2 – Uso da UML

A UML pode ser utilizada para desenvolver qualquer tipo de sistema, e todas as características do sistema são abrangidas em seus diagramas. A UML é aplicada em todas as fases, desde da análise de requisitos até a fase de teste.

O objetivo principal da UML é descrever qualquer tipo de sistema, em forma de diagrama orientados a objeto. A UML também é usada para representar sistemas mecânicos sem nenhum software. Aqui estão alguns tipos diferentes de sistemas com suas características mais comuns: [Booth ET AL,1999]

- Sistemas de Informação: Armazenar, pesquisar, editar e mostrar informações para os usuários. Manter grandes quantidades de dados com relacionamento complexos, que são guardados em bancos de dados relacionais ou orientados a objetos.
- Sistemas Técnicos: Manter e controlar equipamentos técnicos como de telecomunicações, equipamentos militares ou processos industriais. Eles devem possuir interfaces especiais do equipamento e menos programação de software de que os sistemas de informação. Sistemas técnicos são geralmente sistemas real-time.
- Sistemas Real-Time Integrados: Executados em simples peças de hardware integrado a telefones celulares. Carros, alarmes, etc. Estes sistemas implementam programação de baixo nível e requerem suporte real-time.
- Sistemas Distribuídos: Distribuídos em máquinas a onde os dados são transferidos facilmente de uma máquina para outra. Eles requerem mecanismos de comunicação

sincronizados para garantir a integridade dos dados e geralmente são construídos em mecanismos de objetos como CORBA, DCOM ou Java Beans/RMI.

- **Sistemas de Software:** Definem uma infra-estrutura técnica que outros softwares utilizam. Sistemas Operacionais, bancos de dados e ações de usuários que executam ações de baixo nível no hardware, ao mesmo tempo em que disponibilizam interfaces genéricas de uso de outros softwares.
- **Sistemas de Negócios:** Descrevem os objetos, especificações (pessoas, computadores, etc), as regras (leis, estratégias de negócios, etc) e o atual trabalho desempenhado nos processos do negócio.

É importante argumentar que os sistemas não possuem apenas umas destas características citadas acima, mas várias delas ao mesmo tempo. Como por exemplo, os sistemas de informações de hoje, utilizam-se tanto da característica distribuída como da real-time.

2.1.3 – Visões

Desenvolver um sistema complexo não é uma tarefa fácil. Pois o bom seria se o sistema inteiro pudesse ser representado somente em um gráfico e que descrevesse por completo as reais intenções do sistema, sendo facilmente interpretável. Mas isso não é possível, pois um único gráfico é incapaz de capturar todas as informações necessárias para descrever o sistema. [Booth ET AL,1999]

Um sistema é composto por vários aspectos: Funcional (que tem sua estrutura estática e suas interações dinâmicas), não Funcional (requisitos de tempo, confiabilidade, desenvolvimento, etc.) e aspectos Organizacionais (organização do trabalho, mapeamento dos módulos de código, etc.). O sistema é composto por um certo número de visões, cada uma representando uma projeção da descrição completa e mostrando aspectos particulares do sistema. [Booth ET AL,1999]

Cada visão possui um número de diagramas que contém informações que são ênfase aos aspectos particulares do sistema. Também existem alguns casos de sobreposição entre os

diagramas, que significa que um deste pode fazer parte de outra visão. As visões que compõem o sistema são: [Booth ET AL,1999]

- Visão “use-case”: Descreve a funcionalidade do sistema desempenhada pelos atores externos do sistema (usuários). A visão use-case é central, já que seu conteúdo é base do desenvolvimento das outras visões do sistema. Essa visão é montada sobre os diagramas de use-case e eventualmente diagramas de atividade.
- Visão Lógica: Descreve como a funcionalidade do sistema será implementada. É feita principalmente pelos analistas e desenvolvedores. Em contraste com a visão use-case, a visão lógica observa e estuda o sistema internamente. Ela descreve e especifica a estrutura estática do sistema (classes, objetos e relacionamento) e as colaborações dinâmicas quando os objetos enviarem mensagens uns para os outros para realizarem as funções do sistema. Propriedade como persistência e concorrência são definidas nesta fase, bem como as interfaces e as estruturas de classes. A estrutura estática é descrita pelos diagramas de classes e objetos. O modelamento dinâmico é descrito pelos diagramas de estado, seqüência, colaboração e atividade.
- Visão de Componentes: É uma descrição da implementação dos módulos e suas dependências. É principalmente executado por desenvolvedores, e consiste nos componentes dos diagramas.
- Visão de Concorrência: Trata a divisão do sistema em processos e processadores. Este aspecto, que é uma propriedade não funcional do sistema, permite uma melhor utilização do ambiente onde o sistema se encontrará, se o mesmo possui execuções paralelas, e se existe dentro do sistema um gerenciamento de eventos assíncronos. Uma vez dividido o sistema em linhas de execução de processos concorrentes (threads), esta visão de concorrência deverá mostrar como se dá a comunicação e a concorrência destas threads. A visão de concorrência é suportada pelos diagramas dinâmicos, que são os diagramas de estado, seqüência, colaboração e atividade, e pelos diagramas de implementação, que são os diagramas de componente e execução.
- Visão de Organização: Finalmente, a visão de organização mostra a organização física do sistema, os computadores, os periféricos e com eles se conectam entre si. Esta visão será executada pelos desenvolvedores, integradores e testadores, será representada pelo diagrama de execução.

2.1.4 – Diagramas

Todos os sistemas possuem uma estrutura estática e um componente dinâmico. A UML suporta modelos estáticos (estruturas estática), dinâmicos (comportamento dinâmico) e funcional. A modelagem estática é composta pelo diagrama de classes e de objetos, que possuem as classes e os relacionamentos. Os relacionamentos podem ser de associações, herança, dependências ou refinamentos. Os modelos dinâmicos são suportados pelos diagramas de estado, seqüência, colaboração e atividade. E o modelamento funcional é suportada pelos diagramas de componente e execução. Em sistema utilizando UML podem ser necessários de se utilizar de todos os diagramas. Abordaremos agora cada um destes tipos de diagrama, que são divididos em nove e também logo abaixo de cada citação será mostrado um exemplo: [Booth ET AL,1999]

- Diagrama Caso de uso: É uma técnica usada para descrever e definir os requisitos funcionais de um sistema. Eles são escritos em termos de atores externos, use-cases e o sistema modelo. Os atores representam o papel de uma entidade externa ao sistema como um usuário, um hardware, ou outro sistema que interage com o sistema modelado. Os atores iniciam a comunicação com o sistema através dos uses-cases, onde o use-case representa uma seqüência de ações executadas pelo sistema e recebe do ator que lhe utiliza dados tangíveis de um tipo ou formato já conhecido, e o valor de resposta da execução de um use-case (conteúdo) também já é de um tipo conhecido, tudo isso é definido juntamente com o use-case através de texto de documentação.

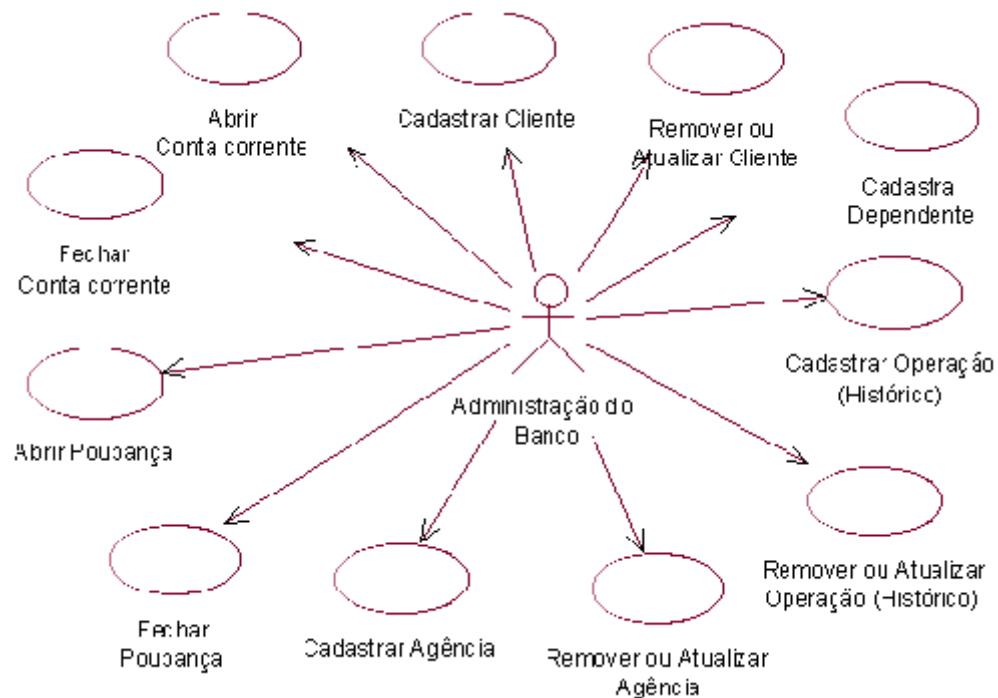


Figura 2.1: Exemplo de Diagrama de caso de uso

- **Diagrama de Classes:** Demonstra a estrutura estática das classes de um sistema onde estas representam as “coisas” que são gerenciadas pela aplicação modelada. Classes podem se relacionar com outras através de diversas maneiras; associação (conectadas entre si), dependência (uma classe depende ou usa outra classe), especialização (uma classe é uma especialização de outra classe), ou em pacotes (classes agrupadas por características similares). Todos estes relacionamentos são mostrados no diagrama de classes juntamente com as suas estruturas internas, que são os atributos e operações. O diagrama de classes é considerado estático já que a estrutura descrita é sempre válida em qualquer ponto do ciclo de todas as classes que estão inseridas em um único diagrama e uma certa classe pode participar de vários diagramas de classes.

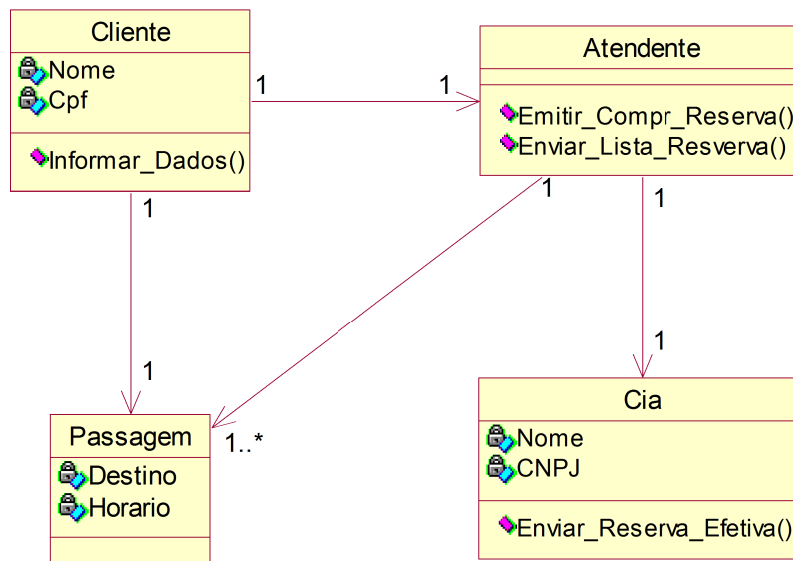


Figura 2.2: Exemplo de Diagrama de Classe

- **Diagrama de Objetos:** É uma variação do diagrama de classes e utiliza quase a mesma notação. A diferença é que o diagrama de objetos mostra os objetos que foram instanciados das classes. O diagrama de objetos é como se fosse o perfil do sistema em um certo momento de sua execução. A mesma notação do diagrama de classes é utilizada com duas exceções; os objetos são escritos com seus nomes sublinhados e todas as instancias num relacionamento são mostradas. Os diagramas de objetos não são tão importantes como os diagramas de classes, mas eles são muito úteis para exemplificar diagramas complexos de classes ajudando muito em sua compreensão. Diagramas de objetos também são usados como parte dos diagramas de colaboração, onde a colaboração dinâmica entre os objetos no sistema é mostrada.
- **Diagrama de Estado:** É tipicamente um complemento para a descrição das classes. Este diagrama mostra todos os estados possíveis que objetos de uma classe podem se encontrar e mostram também quais são os eventos do sistema que provocam tais mudanças. Os diagramas de estado não são escritos para todas as classes de um sistema, mas apenas para aquelas que possuem um número definido de estados conhecidos e onde o comportamento das classes é afetado e modificado pelos diferentes estados.

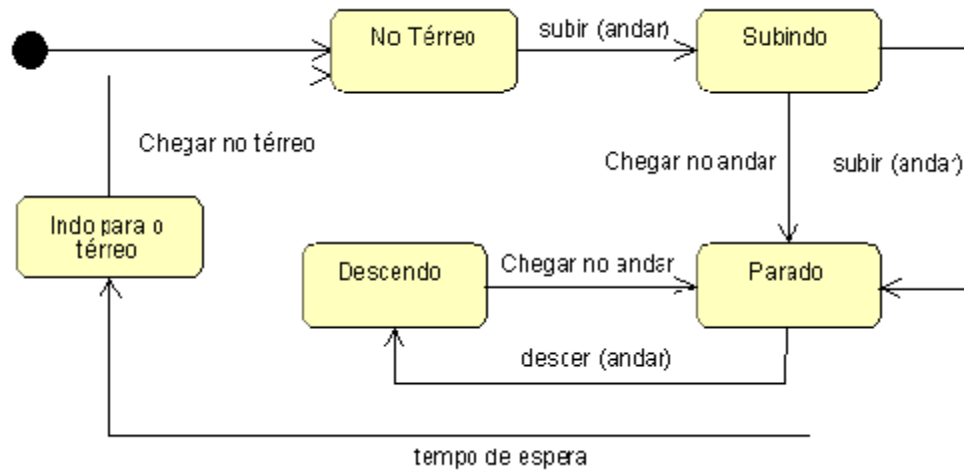


Figura 2.3: Exemplo de Diagrama de Estado

- **Diagrama de Seqüência:** Mostra a colaboração dinâmica entre os vários objetos de um sistema. O mais importante aspecto deste diagrama é que a partir dele percebe-se a seqüência de mensagens enviadas entre os objetos. Ele mostra a interação entre os objetos, alguma coisa que acontecerá em um ponto específico da execução do sistema. O diagrama de seqüência consiste em um número de objetos mostrado em linhas verticais. O decorrer do tempo é visualizado observando-se o diagrama no sentido vertical de cima para baixo. As mensagens enviadas por cada objeto são simbolizadas por setas entre os objetos que se relacionam.

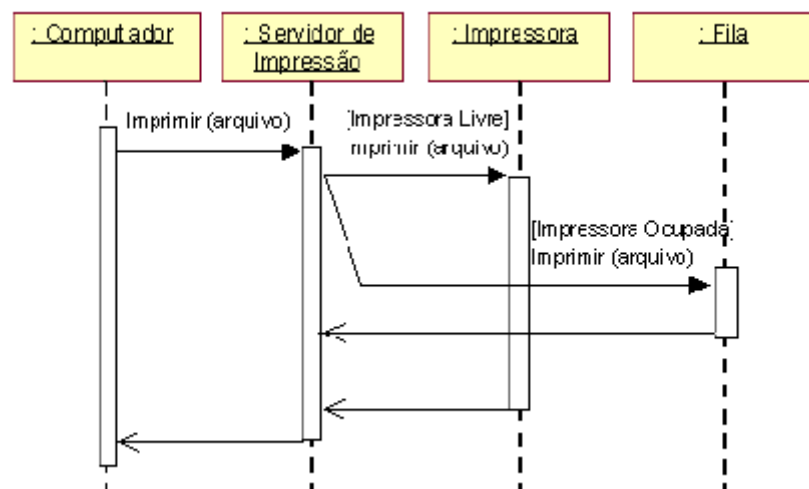


Figura 2.4: Exemplo de Diagrama de Seqüência

- Diagrama de Colaboração: Mostra de maneira semelhante ao diagrama de seqüência, a colaboração dinâmica entre os objetos. Normalmente pode-se escolher entre utilizar o diagrama de colaboração ou diagrama de seqüência. Além de mostrar a troca de mensagens entre os objetos, percebe-se também os objetos com os seus relacionamentos. A interação de mensagens é mostrada em ambos os diagramas. Se a ênfase do diagrama for decorrer do tempo, é melhor escolher o diagrama de seqüência, mas se a ênfase for o contexto do sistema, é melhor dar prioridade ao diagrama de colaboração.

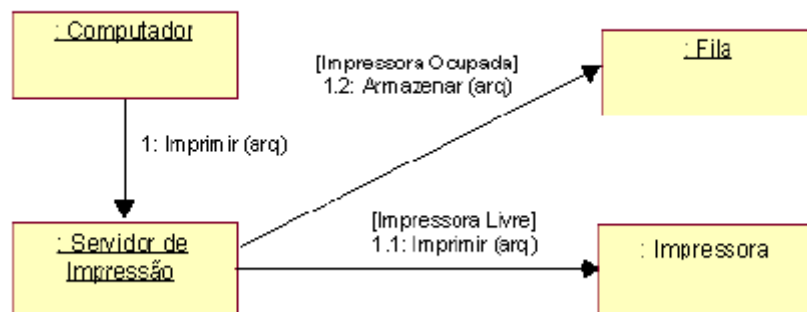


Figura 2.5: Exemplo de Diagrama de Colaboração

- Diagrama de Atividade: Capturam ações e seus resultados. Eles focam o trabalho executando na implementação de uma operação (método), e suas atividades numa instancia de um objeto. O diagrama de atividade é uma variação do diagrama de estado e possui um propósito um pouco diferente do diagrama de estado, que é o de capturar ações (trabalho e atividades que serão executados) e seus resultados em termos das mudanças de estado dos objetos.

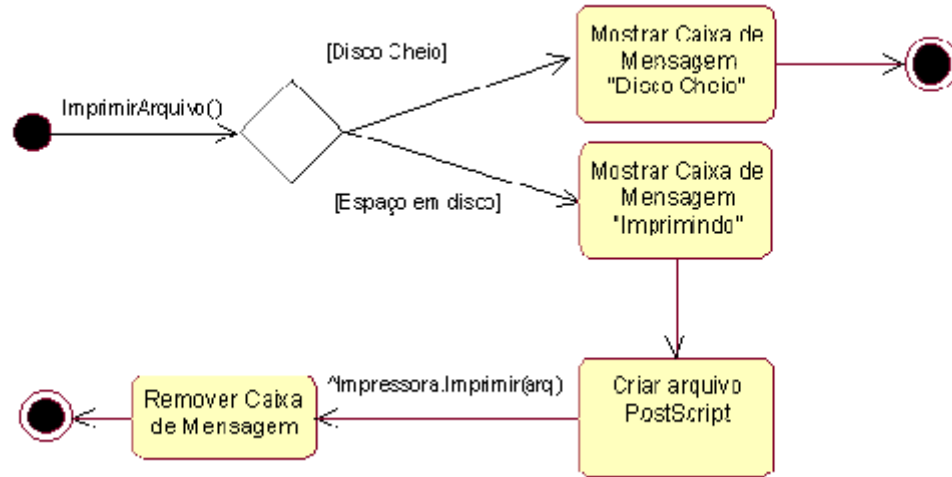


Figura 2.6: Exemplo de Diagrama de Atividade

- Diagrama de Componente: Juntamente com o de execução, são diagramas que mostram o sistema por um lado funcional, expondo as relações entre seus componentes e a organização de seus módulos durante sua execução. Ele também descreve os componentes de software e suas dependências entre si, representando a estrutura do código gerado. Os componentes são implementação na arquitetura física dos conceitos e da funcionalidade definidos na arquitetura lógica (classes, objetos e seus relacionamentos). Eles são tipicamente os arquivos implementados no ambiente de desenvolvimento.

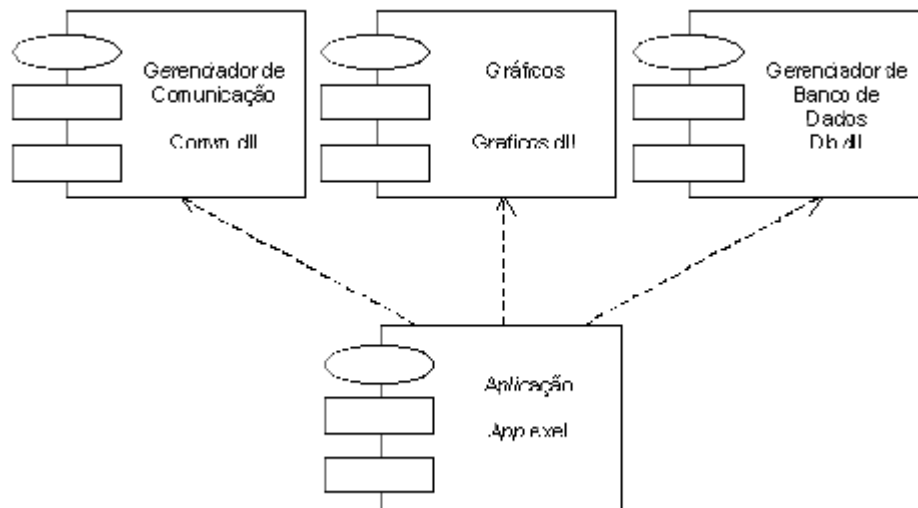


Figura 2.7: Exemplo de Diagrama de Componente

- Diagrama de Distribuição: É composto por componentes, que possuem a mesma simbologia dos componentes do diagrama de componentes, nodes, que significam objetos físicos que fazem parte do sistema, podendo ser uma maquina cliente numa LAN, uma maquina servidora, uma impressora, um roteador, etc, e conexões entre nodes e componentes que juntos compõem toda a arquitetura física do sistema.

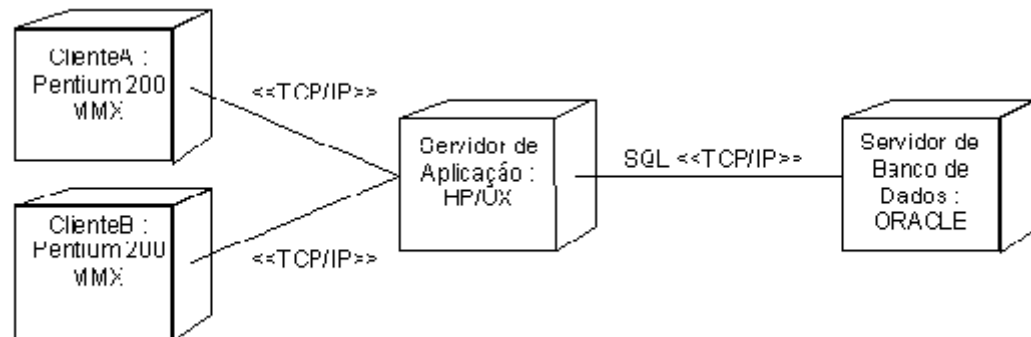


Figura 2.8: Exemplo de Diagrama de Distribuição

2.2 – METODOLOGIA DE DESENVOLVIMENTO DE SOFTWARE

A cada dia as organizações se vêem mais dependentes dos sistemas de informação. Alguns sistemas de informação são qualificados pelas empresas como uma missão crítica, pois pode gerar enormes prejuízos financeiros caso haja eventuais problemas com os mesmos. [Kruchten 2003]

O desenvolvimento de software é uma atividade de alto risco, que gera grandes prejuízos. Atualmente muitos projetos de desenvolvimento de software são iniciados e não são terminados, e outros são terminados consumindo prazos e orçamentos bem acima do que foi estipulado no início do projeto e também muitos softwares desenvolvidos possuem um nível baixo de qualidade.

Conforme Kruchten, um produto de qualidade deve ter ausência de defeitos e, principalmente, deve atender aos propósitos desejados. Alguma coisa com qualidade deve fazer o

que as pessoas querem que elas façam. Se alguma coisa é livre de defeitos, mas não faz o que as pessoas querem que ela faça, essa coisa é um produto desnecessário.

A avaliação da qualidade de software não pode ser de maneira isolada. Softwares são desenvolvidos pelas organizações através de procedimentos. Um método pobre ou a ausência de uma metodologia pode ser a causa da baixa qualidade. A qualidade do software está relacionada diretamente com a qualidade de processos e metodologia utilizada. [Kruchten 2003]

2.2.1 – Metodologias de desenvolvimento

As metodologias de desenvolvimento e as estruturas de avaliação de processos podem ser comparadas sob duas dimensões: de um lado, o vértice pouca formalidade / muita formalidade e de outro, o método cascata / método iterativo, exemplificado através da Figura 2.9.

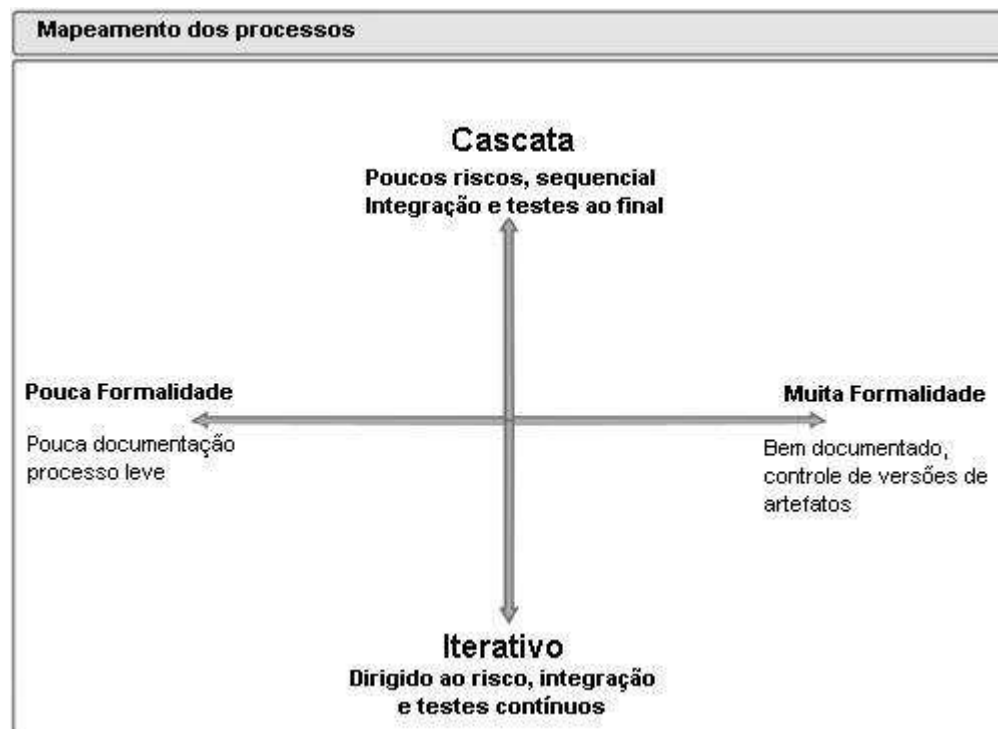


Figura 2.9: Mapa de processos. [Kruchten 2003]

Com pouca formalidade, os processos são produzidos com o mínimo de documentação possível e procedimentos de trabalho bastante informais. Os formais contêm maior documentação, mantém o histórico dos artefatos gerados e possuem gerenciamento de mudanças.

O método cascata é um procedimento linear, onde a integração e os testes são feitos no fim do ciclo de desenvolvimento. O método iterativo é guiado pelo risco, ou seja, é voltado para a eliminação e minimização de riscos. A implementação da arquitetura, a integração e os testes são realizados desde o início do ciclo de vida do aplicativo.

2.2.1.1 – Método cascata

Esse método ainda é muito utilizado e também é conhecido como sequencial ou linear. Segundo [Kroll e Kruchten 2003], esse processo se baseia nos seguintes passos:

- Entender completamente o problema a ser resolvido, seus requisitos e suas restrições;
- Projetar uma solução que atenda todos os requisitos e restrições. Examinar o projeto cuidadosamente e ter certeza que todas as partes interessadas concordam que essa é a solução certa;
- Fazer a implementação do projeto, usando as melhores técnicas de engenharia;
- Verificar se a solução atende aos requisitos estabelecidos;
- Distribuir o produto.

Esse processo é igual à forma em que são construídos as pontes e edifícios. Algumas coisas devem ser feitas dessa maneira. Em um projeto com dois meses de duração, esta metodologia poderia ser usada. Mas normalmente, softwares não devem ser desenvolvidos dessa forma.

2.2.1.2 – Método iterativo e incremental

O método iterativo foi criado para suprir as dificuldades que se encontravam no método cascata. Já que o modelo cascata pode ser usado com sucesso em projetos pequenos, onde o domínio do problema é bem conhecido, a solução encontrada foi dividir estes grandes projetos em projetos menores.

Dessa maneira, alguns requisitos e riscos podem ser identificados com maior clareza. Um projeto pode ser realizado, uma implementação pode ser construída para esse projeto, validada e testada. Esse processo se repete com outras partes do sistema até que o sistema inteiro seja finalizado. Isso é chamado de modo iterativo. [Kruchten 2003]

No sistema são feitas em pequenas partes uma iteração. A iteração segue o modelo sequencial tradicional, com identificação de necessidades, análise, projeto, implementação e testes. A cada iteração o sistema é incrementado até que o ciclo de desenvolvimento do aplicativo termine. Nesse ponto, um novo ciclo de desenvolvimento pode ser iniciado. [Kruchten 2003]

No desenvolvimento de projetos que possuem várias iterações que são incrementando até que chegue ao seu objetivo é chamando de método iterativo e incremental. Atualmente esse paradigma de desenvolvimento é bem aceito e vem sendo utilizado por várias metodologias de desenvolvimento de software. [Kruchten 2003]

2.2.2 – Rational unified process (RUP)

Conforme [Kroll e Kruchten 2003], pode ter três definições para o RUP:

- O RUP é uma maneira de desenvolvimento de software que é iterativa, centrada à arquitetura e guiada por casos de uso. É descrita em vários livros e artigos. Uma das maiores fontes de informações é o próprio produto IBM RUP, que contém guias detalhados, exemplos e modelos cobrindo todo o ciclo de vida do software;
- O RUP é um processo de engenharia de software bem definido e bem estruturado. O RUP define claramente quem é responsável pelo que, como as coisas devem ser feitas e quando fazê-las. O RUP também provê uma estrutura bem definida para o ciclo de vida de um projeto RUP, articulando claramente os marcos essenciais e pontos de decisão;
- O RUP é também um produto de processo que oferece uma estrutura de processo customizável para a engenharia de software. O produto IBM RUP suporta a customização e autoria de processos, e uma vasta variedade de processos, ou configuração de processos, podem ser montadas nele. Essas configurações do RUP podem ser criadas para suportar equipes grandes e pequenas, e técnicas de desenvolvimento disciplinadas ou menos

formais. O produto IBM RUP contém várias configurações e visões de processos prontas que guiam analistas, desenvolvedores, testadores, gerentes de projeto, gerentes de configuração, analistas de dados, e outros membros da equipe de desenvolvimento em como desenvolver o software. Ele tem sido utilizado por muitas companhias em diferentes setores da indústria.

O RUP utiliza a UML, para modelar, especificar e documentar artefatos. A UML tornou-se um padrão empresarial para a modelagem orientada a objeto.

Por ser flexível e configurável, o RUP pode ser utilizado em projetos de pequeno, médio e grande porte. [Kroll e Kruchten 2003] mostra como o RUP pode ser utilizado em um projeto de uma semana com uma equipe de uma pessoa.

2.2.2.1 – Os princípios do RUP

Não existe um modo exato para se usar o RUP, pois ele pode ser utilizado de varias formas e será diferente em cada projeto e organização. Mas existe alguns princípios que podem caracterizar e diferenciar o RUP de outros métodos iterativos: [Kruchten 2003]

- Atacar os riscos cedo e continuamente;
- Certificar-se de entregar algo de valor ao cliente;
- Focar no software executável;
- Acomodar mudanças cedo;
- Liberar um executável da arquitetura cedo;
- Construir o sistema com componentes;
- Trabalhar junto como um time;
- Fazer da qualidade um estilo de vida, não algo para depois.

2.2.2.2 – Elementos do RUP

O RUP possui cinco elementos principais que são: papéis, atividades, artefatos, fluxos de trabalho e disciplinas.

Um papel ou perfil tem como característica o comportamento e a responsabilidade de um determinado indivíduo ou grupo unidos num trabalho de equipe. Um indivíduo pode assumir um determinado papel, como: [Kruchten 2003]

- Analista de sistema – O indivíduo que assume este papel coordena a obtenção dos requisitos e a modelagem dos casos de uso identificando funcionalidades do sistema e estabelecendo limites do sistema;
- Projetista – Esse indivíduo define responsabilidades, operações, atributos, relacionamentos de uma ou mais classes e determina como elas devem ser ajustadas para serem implementadas no ambiente;
- Projetista de testes – Responsável pelo planejamento, projeto, implantação e avaliação de testes, incluindo a geração de plano e modelo de teste, implementando procedimentos de testes e avaliando a abrangência dos testes, resultados e a efetividade.

A atividade executa um determinado trabalho e se obtém um resultado muito importante para o contexto do projeto, e ela é dividida em passos, são elas:

- Planejar uma iteração: realizada pelo papel gerente de projeto;
- Encontrar casos de uso e atores: realizada pelo papel analista de sistemas;
- Rever o projeto: realizada pelo papel revisor de projeto;
- Executar um teste de performance: realizado pelo papel testador de performance.

Um artefato é uma pequena informação que é produzida, modificada ou utilizada em um processo. O artefato é tudo aquilo que é produzido durante o desenvolvimento do projeto. O artefato pode ter várias formas, como:

- Um modelo, como de caso de uso, de projeto;
- Um elemento de um modelo, como uma classe, um caso de uso, um sub-sistema;
- Um documento, como um caso de negócio, glossário, visão;
- Código fonte;
- Executáveis.

A enumeração de atividades, papéis e artefatos não constituem um processo, pois é preciso saber a sequência do desenvolvimento das atividades para que se possa produzir o

artefato de valor para o projeto. O fluxo de trabalho é uma seqüência de atividades que são executadas para obter resultados valiosos na produção do projeto. O RUP possui três tipos de fluxos de trabalho, são eles: [Kruchten 2003]

- Fluxos de trabalho principais, associados com cada disciplina (figura 2.10);
- Fluxos de trabalho de detalhe, para detalhar cada fluxo de trabalho principal (figura 2.11);
- Planos de iteração, que mostram como a iteração deverá ser executada.

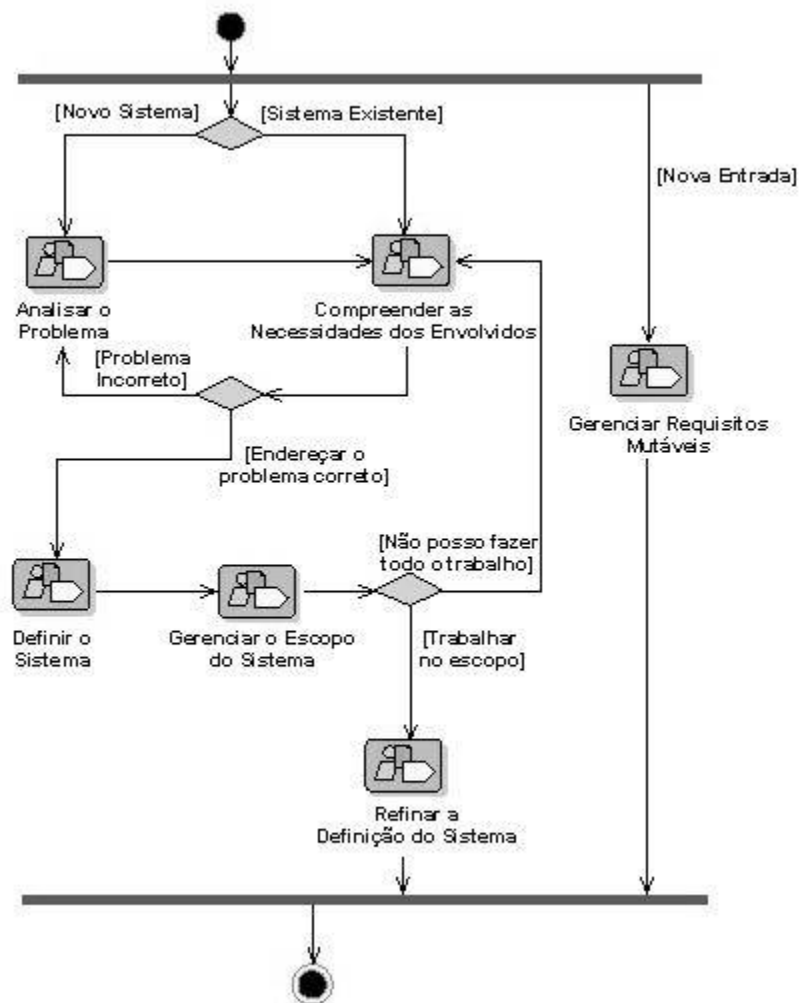


Figura 2.10: Fluxo de trabalho principal: requisitos.

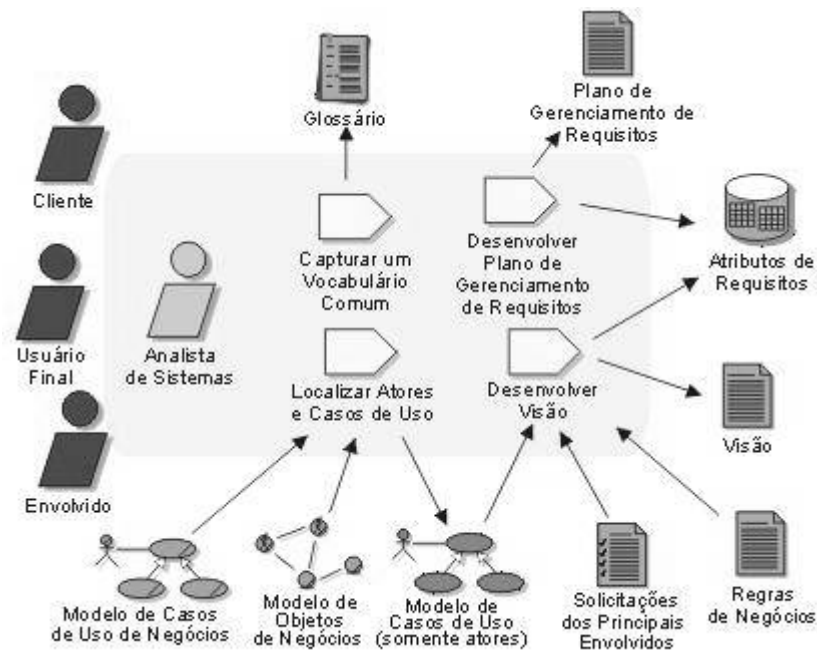


Figura 2.11: Detalhamento de fluxo de trabalho: analisar o problema.

Uma atividade possui diversas atividades relacionadas que fazem parte de um determinado projeto. A separação das atividades em disciplinas torna a compreensão das atividades mais fácil, porém dificulta mais o planejamento das atividades. O RUP é dividido em nove disciplinas, são elas, contida na figura abaixo: [Kruchten 2003]

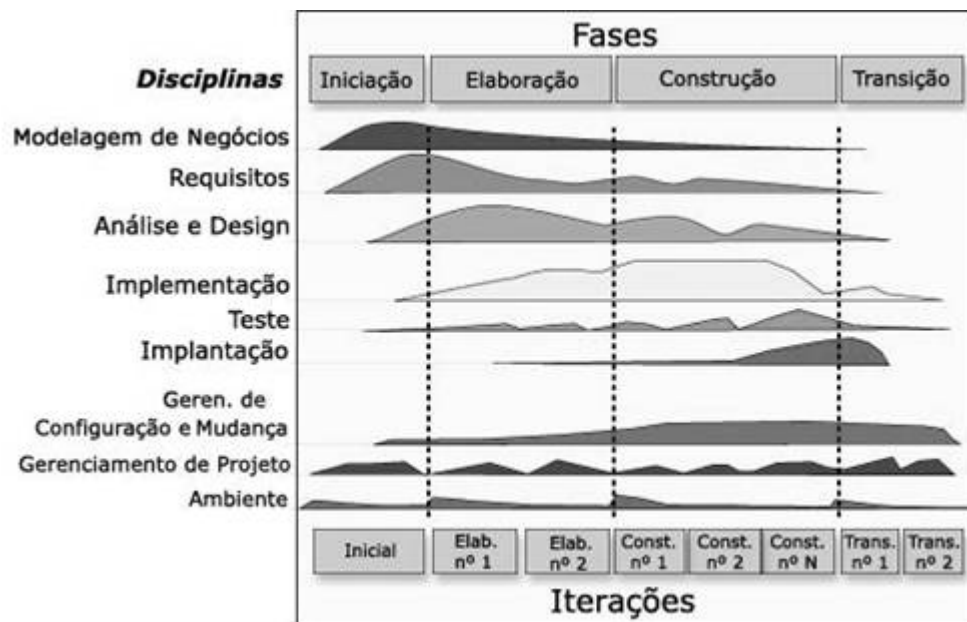


Figura 2.12: Arquitetura geral do RUP.

Conforme mostra a figura 2.12, o RUP possui duas dimensões:

- O eixo horizontal representa o tempo e mostra os aspectos do ciclo de vida do processo à medida que se desenvolve. Representa o aspecto dinâmico do processo. É expresso em termos de fases, disciplinas e marcos.
- O eixo vertical representa as disciplinas, que agrupam as atividades de maneira lógica, por natureza. Representa o aspecto estático do processo. É descrito em termos de componentes, disciplinas, atividades, fluxos de trabalho, artefatos e papéis do processo.

2.2.2.3 – Ciclo de vida de um projeto RUP

O ciclo de desenvolvimento no RUP possui quatro fases: iniciação, elaboração, construção e transição. Cada fase possui um intervalo de tempo entre dois marcos principais, como é mostrado na figura 2.13.

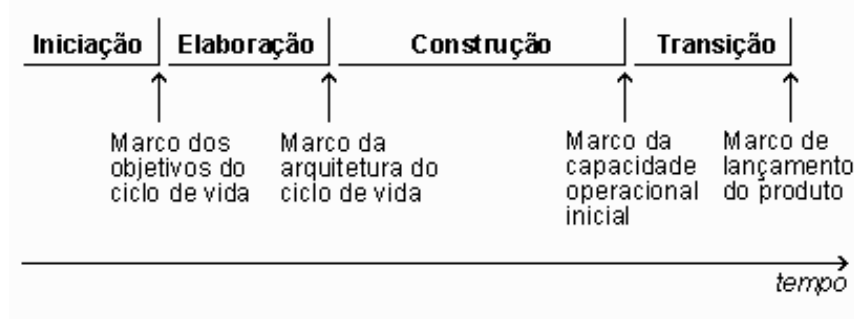


Figura 2.13: As fases e os marcos de um projeto.

Com o termino do ciclo de desenvolvimento, já se obtém uma versão completa do software em questão.

As fases definem os estados do projeto. A fase de iniciação, foca no tratamento de riscos relacionados com o caso de negócio. Deve ser verificado se o projeto é viável e se é financeiramente possível. Na fase elaboração, o foco deve ser nos riscos técnicos e arquiteturais. O escopo deve ser revisado e os requisitos devem estar mais compreendidos. Durante a construção, a atenção será voltada para os riscos “lógicos”, e a maior parte do trabalho será

realizada. Na fase de transição, serão tratados os riscos associados com a logística de distribuição do produto para a base de usuários. [Kruchten 2003]

Com a variação de empresa e projeto, um ciclo de desenvolvimento para um projeto de tamanho médio, contém uma distribuição de esforço e programação, como é visto na tabela 2.1 e na figura 2.14.

	Iniciação	Elaboração	Construção	Transição
Esforço	~5%	20%	65%	10%
Programação	10%	30%	50%	10%

Tabela 2.1: Distribuição de esforço e programação em projetos de médio porte.

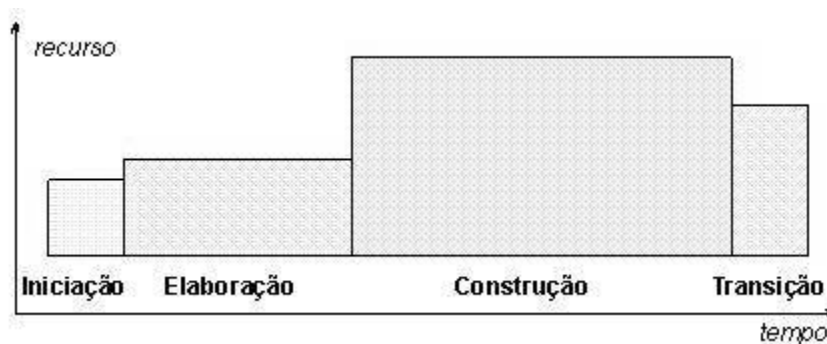


Figura 2.14: Distribuição de esforço e programação em projetos de médio porte.

Como foi dito no documento do RUP, a passagem pelas quatro fases, obtém-se uma geração do software. E se o resultado final que obteve não era o esperado, irá se repetir as fases de iniciação, elaboração, construção e transição, obtendo novas gerações. Este ciclo subsequente é chamado de ciclo de evolução.

O ciclo de evolução pode sofrer mudanças, quanto a: melhorias sugeridas pelos usuários, mudanças no contexto do usuário, mudanças na tecnologia subjacente reação a concorrência e entre outros mais.

3 – SISTEMA DE CONTROLE DE ATACADO DE MATERIAL DE CONSTRUÇÃO

3.1 – Descrição do sistema

Um grande atacado na área de material de construção, elétrico e hidráulico, que atua no mercado a mais de dez anos, que reside na cidade de Ouro Branco, possui mais de cem funcionários, sendo boa parte deste exercendo a função de vendedor. (Este trabalho tratará somente da área de vendas)

Esta empresa possui distribuído entre a região sudeste entre 3 a 5 vendedores por estado, que semanalmente visitam seus clientes; estes clientes são varejo em matérias de construção e matérias elétricos e hidráulico. Cada vendedor possui dois catálogos, um com os produtos e seus códigos e outro com os preços referentes a cada produto.

Estes pedidos são enviados à empresa por telefone, onde o vendedor liga a um 0800 e passa o CNPJ do cliente, os códigos dos produtos comprados e as condições de pagamento a um funcionário da empresa, para que possa enviar este pedido para o setor de armazém, onde será separado o pedido.

Nesta época havia muita dificuldade em se controlar a empresa, pois tudo era feito manualmente, desde o envio de pedido até o controle de estoque. Para se ter uma idéia, se gastava em torno de 3 a 4 dias para se despachar um pedido para a entrega, por ser tudo manual, quando o funcionário recebia um pedido por telefone do vendedor, ele anotava o pedido num rascunho e levava-o para o outro funcionário que era responsável por separar as mercadorias, até que separasse a mercadoria já tinha passado um dia, logo ele entregaria o rascunho para outro funcionário para que pudesse tirar a nota fiscal a manuscrito, também se gastava um bom tempo, pois até que esse funcionário fosse ao departamento de cadastro para pegar os dados do cliente. Logo depois da mercadoria ser separada e a nota fiscal fosse tirada, o funcionário que tirou a nota fiscal levava-a para que outro funcionário pudesse emitir o boleto bancário, que também era feito manuscrito, logo gastava mais de meio dia até que preenche-se os boletos. Depois de tudo feito, ligava para a transportadora para que ela pudesse recolher o pedido.

O controle dos clientes que não efetuavam o pagamento dos boletos, também eram feitos manualmente, pois um funcionário tinha que todos os dias comunicar se com banco, para que ele o envia-se um fax com a relação dos boletos quitados. O funcionário gastava bastante tempo para se separar os clientes em débitos relacionados a cada vendedor, logo o funcionário ligava para o vendedor para que ele pudesse entrar em contato com o cliente para se fazer o pagamento do boleto em aberto, ou se não ele estaria bloqueado para fazer pedido. Com a quitação do boleto, o cliente deveria enviar um fax para a empresa do boleto pago para que a empresa pudesse fazer a liberação para fazer pedido.

Uma outra dificuldade era o controle de estoque, pois se tinha que colocar uns cinco funcionários para verificar a quantidade em estoque de cada produto, e se gastava muito tempo, pois eram muitos produtos. Depois de se contar todos os produtos, um funcionário pegava os relatórios e levava para o departamento de compra, para a reposição dos produtos em faltam ou em baixa no estoque. Também tinha um funcionário que fiscalizava os funcionários que atuavam na área de estoque, para que não houve se o furto de mercadoria e se tinha mensalmente balanço, para se ver se estava tendo furto de mercadorias.

A empresa observando que estava tendo muitas reclamações em torno de que estava vindo mercadorias erradas, preços errados, faltando mercadoria, entre outras coisas mais, e também estava vendo que seus concorrentes estavam informatizando suas empresas, decidiram contratar uma empresa de informática especializada em desenvolvimento de software comerciais, que pudessem desenvolver um sistema para tentar amenizar entre 80 a 90% destes erros.

Com a informatização da empresa, todos os clientes receberão um comunicado para que possam entrar em contato com a empresa para fazer um novo cadastro, e neste deverão informar, a razão social, cnpj, endereço, telefone, o nº da agencia e conta corrente do banco que eles trabalham e o nome e telefone de dois a cinco fornecedores que eles comprem. Caso não haja nenhuma pendência, os clientes passaram a receber visitas semanalmente dos vendedores para que possam fazer seus pedidos.

Com a informatização, os vendedores receberam um palm com um sistema de vendas, e neste contaram com as seguintes funções; quantidade em estoque, produtos novos e em promoção, clientes novos e com pendências e preços.

Como funcionará o sistema do vendedor; O cliente deverá informar ao vendedor o CNPJ, logo o vendedor terá todas as informações do cliente, como, razão social, endereço, telefone,

últimos pedidos realizados pelo cliente e de no máximo cinco e também terá informação do débito do cliente, caso o tenha, pois se estiver em debito com a empresa o sistema não autorizará que vendedor faça vendas a este cliente até que ele regularize suas pendências. Caso não haja débitos, o cliente poderá fazer seus pedidos. Para fazer um pedido, inicialmente ele deverá informar as condições de pagamentos, pois para cada tipo de condição de pagamento o preço sofrerá pequenas alterações, o cliente poderá alterar as condições de pagamento em qualquer hora do pedido. Logo o cliente informará o produto que esteja precisando, no sistema o vendedor poderá encontrar o produto informando sua descrição ou informando seu código, com isso o vendedor terá as seguintes informações sobre o produto, o preço para a condição de pagamento escolhida, desconto caso o produto esteja em promoção, a disponibilidade do produto em estoque, pois para o vendedor existirem três condições, o produto com estoque alto, neste caso o cliente poderá pedir o produto, pois o estoque na empresa desse determinado produto estará alto, a outra e a do produto com estoque médio, nessa condição o cliente poderá fazer o pedido deste produto, mas não poderá ter certeza se ele será entregue, pois nesta condição o estoque já está baixo, como os pedidos são despachados em ordem de chegada, o pedido que chegar primeiro com esse produto será despachado até que o estoque deste produto zere e a ultima condição e a do produto com estoque baixo, não haverá o produto em estoque. Com o fechamento do pedido é informado ao cliente o total do pedido e este é armazenado juntamente com os outros, até que no final do dia o vendedor envie todos os pedidos e também atualize seus preços, estoque e pendências de clientes, conectando-se o palm a internet.

Com o pedido enviado e recebido na empresa pelo setor de vendas, o funcionário responsável tem como função distribuir os pedidos aos setores de pagamento e de armazém de mercadoria, que respectivamente serão feitos os boletos bancários e os produtos serão separados e também será emitida a nota fiscal, para que a transportadora entregue o pedido ao seu destino.

O sistema de venda da empresa terá o controle total dos pedidos, pois ele trabalhará em forma de parceria com o sistema do armazém de mercadoria, então no sistema de vendas encontrará o status dos pedidos, pois esta informação será passada para o sistema dos vendedores quando eles fizerem atualizações, para isso o status será dividido em três partes, como: imprimindo nota fiscal, nesta opção estará sendo impressa a nota fiscal e aguardando que o boleto bancário seja entregue pelo setor de pagamento, a segunda opção é a separando pedido, nesse caso a nota fiscal já estará impressa juntamente com o boleto, para que um certo funcionário

separe as mercadorias contidas no pedido e por ultimo enviando para transportadora, nessa opção o pedido estará separado e embalado, para que a transportadora possa entregá-lo no destino correspondente à nota fiscal.

O sistema de venda também receberá informações do sistema de pagamento, sobre o pagamento ou não dos boletos que se encontram em aberto na empresa, para que o sistema de vendas possa passar estas informações para o sistema dos vendedores no ato que eles fizerem atualizações do sistema, caso o cliente encontre-se com boletos em aberto com a empresa, será feito o bloqueio deste cliente ate que regularize suas pendências.

4 – DESENVOLVIMENTO

Nesta fase do trabalho serão apresentados os documentos gerados nas etapas do RUP descritas no capítulo 2, iniciação, elaboração, construção e transição. Com a elaboração do mesmo poderá ter uma idéia de como comportará o sistema. Neste trabalho não constará a fase de transição, pois não haverá a implementação do mesmo.

Neste capítulo os documentos e diagramas serão apresentados de acordo com as fases do RUP onde ocorre maior concentração no seu desenvolvimento. No entanto, ressaltar que um diagrama de classes, por exemplo, não é feito somente na construção, mas durante toda a aplicação da metodologia.

4.1 – Fase Iniciação

Esta fase será feita de acordo com as especificações citadas no capítulo 2, de acordo com as fases do RUP.

A proposta deste trabalho tem como objetivo a elaboração de dois sistemas, um que estará instalado na empresa e outro para os vendedores, logo o sistema do vendedor terá a possibilidade de conectar-se a empresa através da internet, para que ele possa enviar os pedidos e atualizar seu sistema.

Mas para este trabalho estaremos trabalhando somente com a análise e projeto, baseando se nas informações citadas pelo o cliente, que esta descrita no capítulo 3. Mas só as informações citadas pelo cliente não será suficiente, pois o analista poderá acrescentar mais algumas funções que seja útil ao sistema e que não seja visível ao cliente.

4.2 – Fase Elaboração

4.2.1 – Requisitos

4.2.1.1 – Requisitos Funcionais

Os requisitos funcionais descrevem aquilo que o sistema deve fazer e são adquiridas nas informações passadas pelo cliente. Estas informações estão contidas no capítulo 3.

Requisito Funcional 1: O sistema deve permitir ao cliente informar seu CNPJ para o vendedor (nome, código).

Requisito Funcional 2: O sistema deve bloquear os clientes, caso haja pendências de informações de dados de cadastro.

Requisito Funcional 3: O sistema deve liberar os clientes para que possam fazer pedidos (código, código funcionário).

Requisito Funcional 4: O sistema deve exibir informações sobre o cliente (razão social, cnpj, endereço, telefone).

Requisito Funcional 5: O sistema deve exibir informações dos últimos pedidos realizado pelo cliente e de no máximo cinco.

Requisito Funcional 6: O sistema deve exibir informações de débito do cliente.

Requisito Funcional 7: O sistema deve permitir ao cliente informar as condições de pagamento (dias).

Requisito Funcional 8: O sistema deve alterar os preços dos produtos conforme a condição de pagamento escolhida pelo cliente.

Requisito Funcional 9: O sistema deve permitir ao cliente informar o produto (descrição, código) que esteja precisando ao vendedor.

Requisito Funcional 10: O sistema deve permitir ao vendedor informar as condições do produto (preço, quantidade, desconto, disponibilidade no estoque) ao cliente.

Requisito Funcional 11: O sistema deve exibir para o vendedor qual é a condição de estoque (estoque alto, estoque médio, estoque baixo) que o produto se encontra.

Requisito Funcional 12: O sistema deve calcular o total do pedido do cliente.

Requisito Funcional 13: O sistema deve permitir que vendedor distribua os pedidos aos setores de pagamento e de armazém de mercadoria.

Requisito Funcional 14: O sistema deve permitir a impressão do boleto bancário referente ao pedido recebido pelo setor de pagamentos.

Requisito Funcional 15: O sistema deve permitir que seja separado o produto relacionado aos pedidos recebidos.

Requisito Funcional 16: O sistema deve permitir que o setor de armazém de mercadoria emita a nota fiscal.

Requisito Funcional 17: O sistema deve permitir que o setor de armazém de mercadoria informe ao setor de vendas o status (imprimindo nota fiscal, separando pedido, enviando para transportadora, entregue) que o pedido se encontra.

Requisito Funcional 18: O sistema deve permitir que o setor de pagamento informe ao setor de vendas os clientes que se encontram com boletos em aberto.

Requisito Funcional 19: O sistema deve bloquear os clientes que se encontre com boletos em aberto.

4.3 – Fase Construção

Nesta fase serão apresentados os diagramas desenvolvidos de acordo com as notações da UML, esses diagramas serão necessários devido as características do sistema.

4.3.1 – Fluxo de análise

Logo abaixo serão apresentados os diagramas de caso de uso e classe e também apresentado a especificação de requisito de caso de uso, havendo a explicação de somente um, pois o restante será apresentado no anexo A.

4.3.1.1 – Diagrama de caso de uso

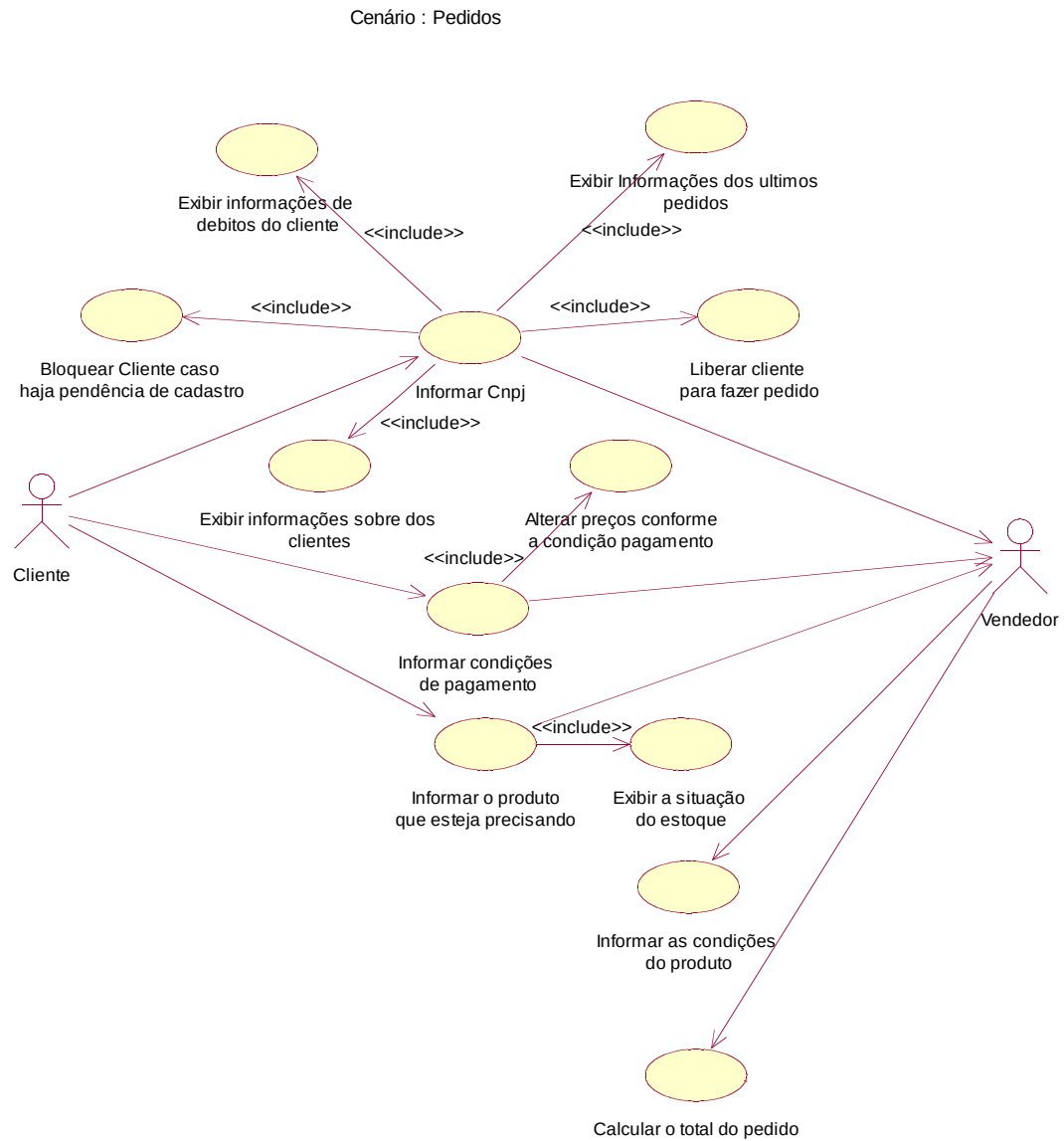


Figura 4.1: Diagrama de Caso de Uso para o Cenário de Pedido

4.3.1.2 – Especificação de Caso de Uso

Será apresentada somente uma especificação para um estado do diagrama de caso de uso, pois o restante estará contido em um documento no anexo A.

CASO DE USO 01	Informar CNPJ
1 - ATOR PRINCIPAL	Vendedor
2 - SUMÁRIO	Este caso de uso é feito pelo vendedor quando ele vai ao cliente para efetuar um pedido, informando o CNPJ do mesmo. O objetivo deste caso de uso é possibilitar que se exiba informação do cliente e que ele possa fazer pedido.
3 - PRÉ-CONDIÇÕES	Não aplicável
4 - FLUXO 4.1 - FLUXO PRINCIPAL	a) O vendedor solicita no sistema a opção cliente. b) O sistema solicita o CNPJ do cliente, com a informação do mesmo, logo serão exibidas informações sobre o cliente, os últimos pedidos realizados pelo mesmo e os débitos, boletos em que vão vencer e que já estão vencidos. c) Caso tenha alguma pendência, o seu CNPJ será bloqueado até que se regularize sua situação. d) Também encontrará a opção de novo pedido.
4.2 - FLUXO ALTERNATIVO	a) O vendedor poderá fazer uma pesquisa indo a opção últimos pedidos, lá ele encontrará a seguinte informação, a data dos últimos pedidos e de no máximo dez. b) Ainda na opção últimos pedidos, o vendedor poderá entrar no pedido para informar ao cliente os produtos que ele comprou, o preço e a quantidade comprada. c) O vendedor também poderá fazer uma pesquisa indo a opção débitos, lá terá duas opções, uma é boletos em aberto, nesta opção o vendedor terá informação das duplicatas que irão vencer e a outra opção é boletos vencidos, já nesta opção o vendedor terá informação das duplicatas vencidas. d) Ainda na opção débitos, o vendedor poderá entrar na duplicata em aberto ou vencida para informar ao cliente a que pedido está relacionado à duplicata.

4.3 - FLUXO DE EXECUÇÃO	a) O CNPJ não foi preenchido ao entrar na opção cliente. Logo aparecerá uma mensagem e retornará a opção. b) O CNPJ foi preenchido errado. Logo aparecerá uma mensagem e retornará a opção. c) O CNPJ foi preenchido corretamente, mas o cliente apresenta alguma pendência. Logo aparecerá uma mensagem e a opção novo pedido será desabilitada.
5 - REQUISITOS NÃO FUNCIONAIS	Não aplicável.
6 - REGRA DE NEGÓCIO	RN1: O campo CNPJ é obrigatório seu preenchimento. RN2: Na opção últimos pedidos, encontrará no máximo 10 pedidos, ordenados pela data.

4.3.1.3 – Diagrama de classe

Serão apresentados as entidades que compõem o sistema juntamente com seus atributos e métodos que são apresentados no digrama de caso de uso.

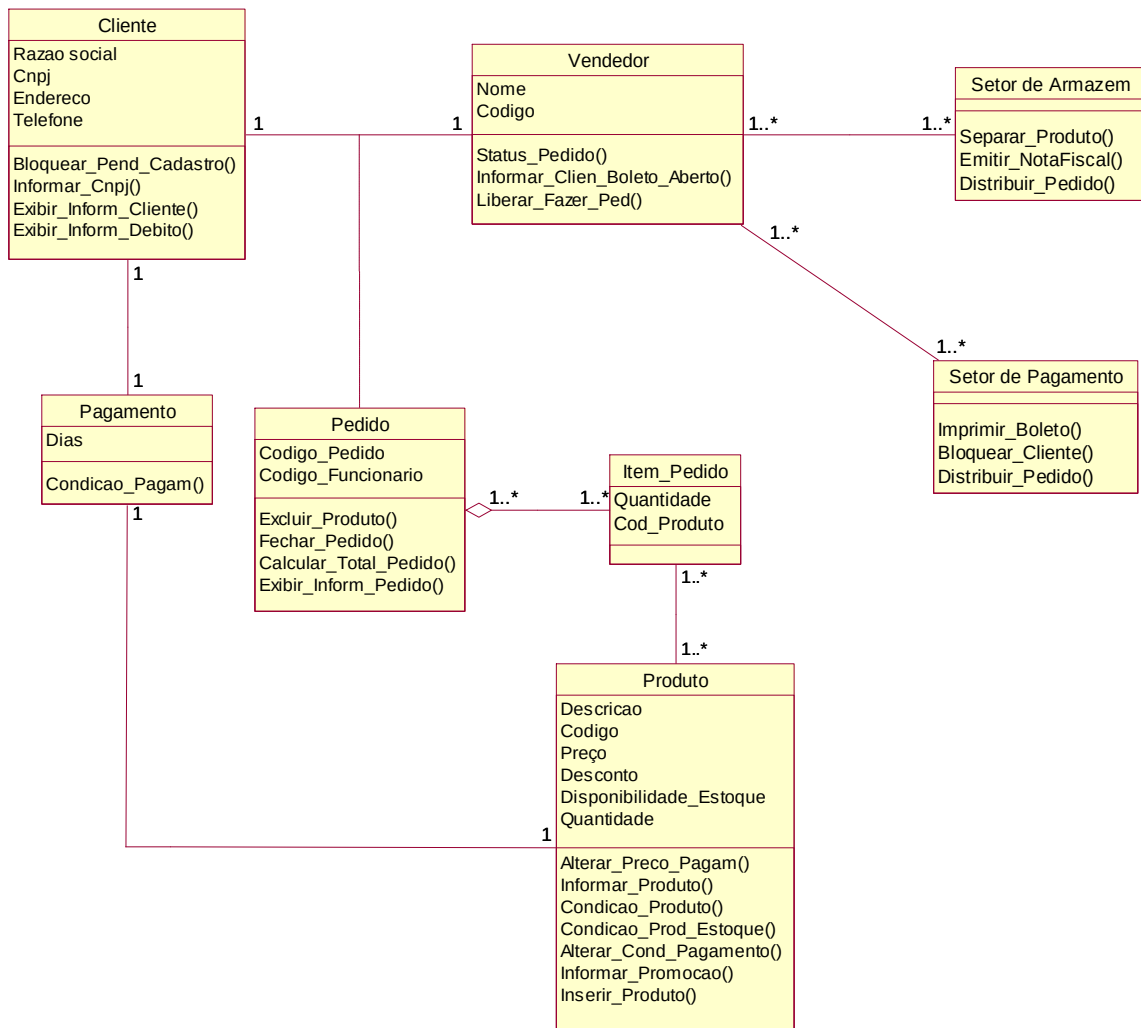


Figura 4.2: Diagrama de Classe do Projeto

4.3.2 – Fluxo de projeto

4.3.2.1 – Diagrama de Projeto de Classe

O diagrama de projeto de classe é uma copia do diagrama de classe, porém acrescentado dos tipos dos atributos e visibilidade e os parâmetros de entrada e saída dos métodos e também com a sua visibilidade.

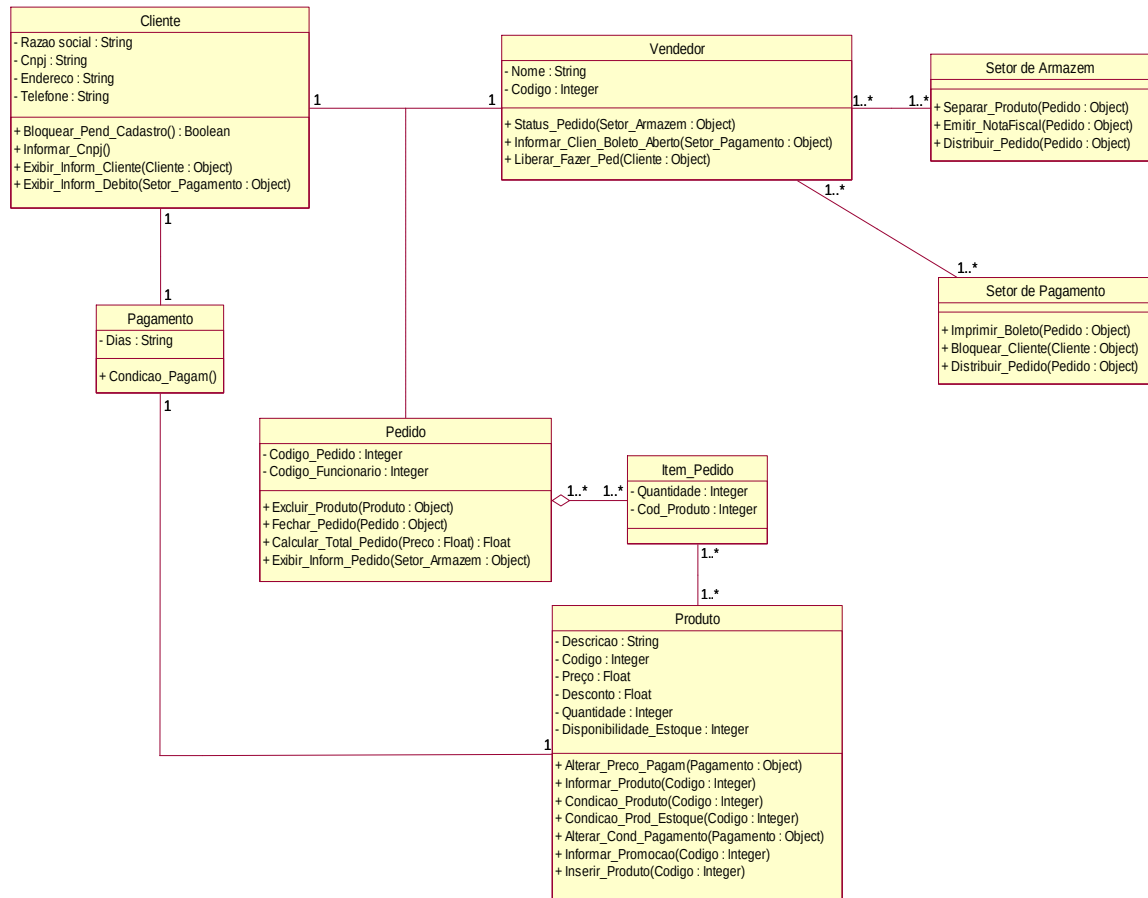


Figura 4.3: Diagrama de Projeto de Classe do Projeto

4.3.2.2 – Descrição de método

Será apresentado logo abaixo o pseudo-código para método descrito no diagrama de projeto de classe e também serão apresentados alguns outros em um documento em anexo.

Informar_Cnpj ()

```
Caracter Cnpj
Mensagem("Digite o Cnpj do Cliente:")
Ler Cnpj
Se Procurar_Cnpj(Cnpj) Faça
    Exibir_Inform_Cliente
    Exibir_Inform_Pedido
    Exibir_Inform_Debito
    Se Bloquear_Pend_Castrado(Cnpj) Faça
        Mensagem("Por favor, entra em contato com a empresa para a atualização
        de seus dados")
        Terminar
    Liberar_Fazer_Ped
Senão
    Mensagem("Cnpj digitado errado")
Fim
```

4.3.2.3 – Diagrama de distribuição

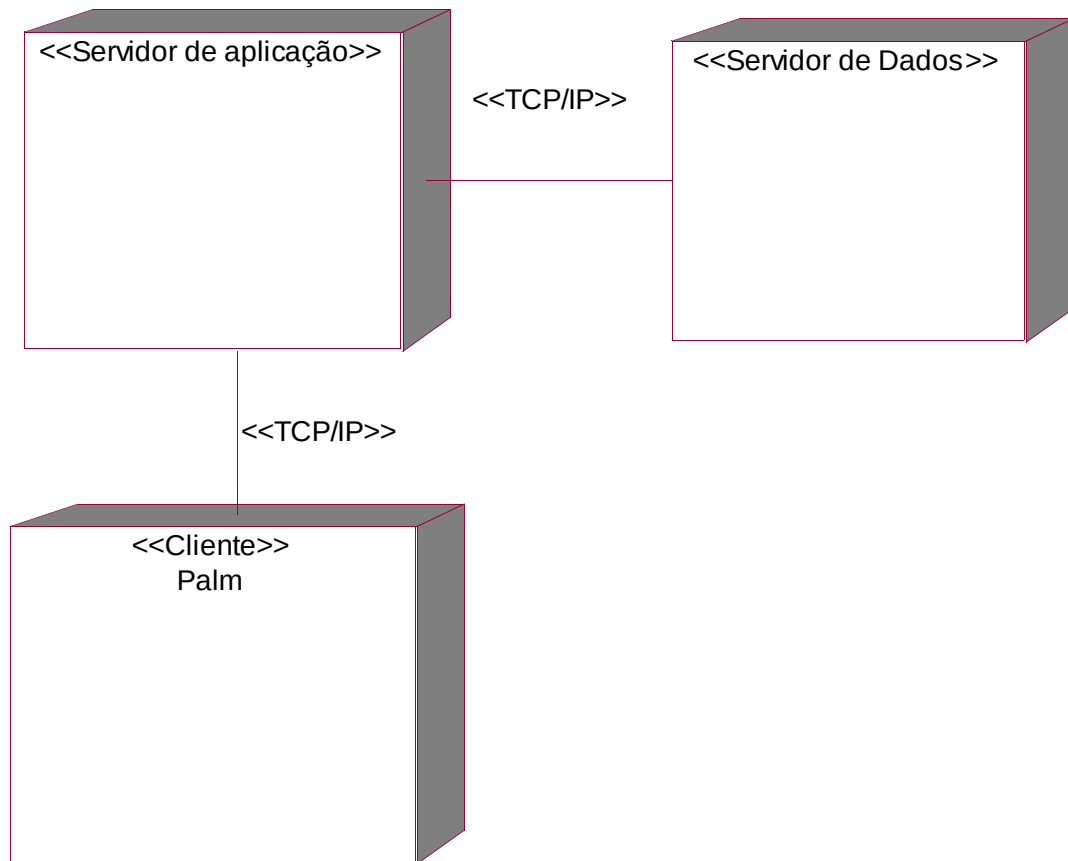


Figura 4.4: Diagrama de distribuição do projeto

5 – CONCLUSÃO

5.1 – Revisão

Este trabalho teve como objetivo de fazer projeto de um software para controle da área de vendas de um atacado de material de construção, utilizando a paradigma da Orientação a Objeto, o padrão UML e metodologia RUP.

O resultado deste trabalho foi apresentado neste documento, que apresentou os seguintes itens:

- Capítulo 1: Foi feita uma introdução do trabalho e o motivo pelo qual foi feita essa escolha.
- Capítulo 2: Foi apresentado, e com grande aprofundamento os conceitos de RUP e UML, que são bases para a análise de qualquer software.
- Capítulo 3: Foi feita uma apresentação do sistema de material de construção, contando um pouco do que seria desenvolvido.
- Capítulo 4: Foram apresentados como se desenvolveram os diagramas para a análise do sistema.

5.2 – Conclusões

A UML tem como objetivo facilitar as empresas de desenvolvimento de software a uma maior comunicação e aproveitamento dos modelos desenvolvidos pelos seus vários analistas envolvidos no processo de produção de software, já que a linguagem que será utilizada por todos será a mesma, acabando com qualquer problema de interpretação e mal entendimento de modelos criados por outros desenvolvedores.[Booth ET AL,1999]

A utilização de uma metodologia de desenvolvimento de software como o RUP, é possível obter:

- Qualidade de software;
- Produtividade no desenvolvimento, operação e manutenção de software.

- Controle sobre desenvolvimento dentro de custos, prazos e níveis de qualidade desejados;
- Estimativa de prazos e custos com maior precisão.

Apesar dos benefícios, deve-se ter a consciência que os benefícios não virão de maneira imediata. É necessário adquirir treinamento, adaptação da metodologia no contexto ao qual ela será utilizada, apoio especializado para as equipes de desenvolvimento e tempo para a absorção da metodologia.[Kruchten 2003]

Com a utilização da engenharia de software orientado a objeto, foi possível chegar ao resultado esperado, pois foram utilizados todos os recursos para se facilitar o desenvolvimento do projeto.

Com a utilização do RUP e UML, pode obter uma coerência maior de como desenvolver uma análise e projeto para qualquer tipo de sistema. Com o desenvolvimento do mesmo, qualquer profissional da área poderá implementá-lo.

5.3 - Trabalhos Futuros

Este trabalho tem como objetivos futuros:

- Emprego de padrões de projetos;
- Implementação do software utilizando-se diferentes linguagens de programação OO, com emprego de frameworks de persistência;
- Implementação do banco de dados em um SGBD OO;
- Projeto de outras atividades relacionadas com a atividade de controle de atacado de material de construção.

REFERÊNCIAS BIBLIOGRÁFICAS

- (KRUCHTEN,2003) Kruchten, Philippe. **Introdução ao RUP – Rational Unified Process**. Ed. Ciência moderna Ltda., 2003.
- (BOOCH,2000) Booch, Grady.**UML, guia do usuário / Grady Booch, James Rumbaugh, Ivar Jacobson; Tradução de Fábio Freitas da Silva**. Rio de Janeiro: Campus, 2000.
- (RONALDO,2005) Ronaldo Rezende Vilela Luiz - Sistemas de Informação – Universidade de Uberaba (UNIUBE) – 20/03/2005
<http://www.javafree.com.br/home/modules.php?name=Content&pa=showpage&pid=45>
- (FUNDÃO,2005) Fundão - 20/03/2005
<http://www2.fundao.pro.br/articles.asp?cod=194>
- (DGA,2005) DGA – Diretoria Geral de Administração – 20/03/2005
http://www.dga.unicamp.br/treinamento/apostilas/apostila_uml.pdf

ANEXO A – DIAGRAMA DE CASO DE USO

A.1 – DIAGRAMA DE CASO DE USO

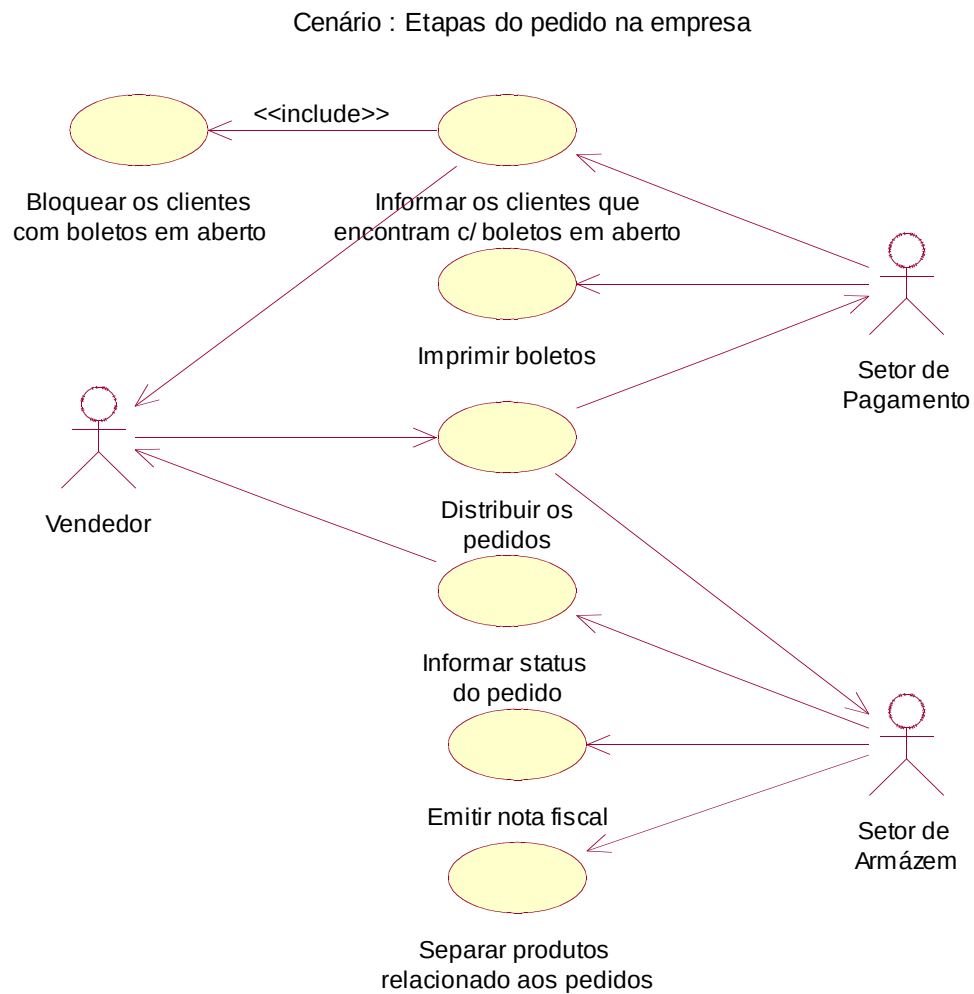


Figura A.1: Diagrama de Caso de Uso do Cenário Etapas do Pedido na Empresa

A.2 – Especificações de Caso de Uso

CASO DE USO 02	Exibir informações sobre o cliente
----------------	------------------------------------

1 - ATOR PRINCIPAL	Vendedor
2 – SUMÁRIO	Neste caso de uso, será exibida a informação sobre o cliente como, razão social, cnpj, endereço, telefone.
3 - PRE-CONDIÇÕES	Não aplicável.
4 – FLUXO	
4.1 - FLUXO PRINCIPAL	a) Exibir informações sobre o cliente.
4.2 - FLUXO DE EXECUÇÃO	a) Será exibido um quadro, e neste quadro terá as informações do cliente.
5 - REQUISITOS NÃO FUNCIONAIS	Não aplicável.
6 - REGRA DE NEGÓCIO	RN1: O campo razão social será preenchido automaticamente. RN2: O campo cnpj será preenchido automaticamente. RN3: O campo endereço será preenchido automaticamente. RN4: O campo telefone será preenchido automaticamente.

CASO DE USO 03	Exibir informação dos últimos pedidos.
1 - ATOR PRINCIPAL	Vendedor
2 - SUMÁRIO	Neste caso de uso, será exibida a informação dos últimos pedidos do cliente, onde constaram os dez últimos pedidos.
3 - PRE-CONDIÇÕES	Não aplicável.
4 – FLUXO	
4.1 - FLUXO PRINCIPAL	a) Exibir informações sobre os últimos pedidos realizado pelo cliente.
4.2 - FLUXO DE EXECUÇÃO	a) Será exibido um quadro, aonde serão mostrados os últimos dez pedidos ordenado pela data.
5 - REQUISITOS NÃO FUNCIONAIS	Não aplicável
6 - REGRA DE NEGÓCIO	RN1: O quadro de últimos pedidos será preenchido automaticamente.

CASO DE USO 04	Exibir informação de débito do cliente.
1 - ATOR PRINCIPAL	Vendedor
2 - SUMÁRIO	Neste caso de uso, será exibida a informação de débito do cliente, onde constaram as duplicatas em aberto e vencida.
3 - PRE-CONDIÇÕES	Não aplicável.
4 – FLUXO	a) Exibir informação sobre os débitos do cliente.

4.1 - FLUXO PRINCIPAL	
4.2 - FLUXO DE EXECUÇÃO	a) Será exibidos dois quadros, aonde serão mostrados os débitos em aberto e vencido ordenado pela data.
5 - REQUISITOS NÃO FUNCIONAIS	Não aplicável.
6 - REGRA DE NEGÓCIO	RN1: O quadro de débitos em aberto será preenchido automaticamente. RN2: O quadro de débitos vencidos será preenchido automaticamente.

CASO DE USO 05	Bloquear cliente caso haja pendência de cadastro.
1 - ATOR PRINCIPAL	Vendedor
2 - SUMÁRIO	Neste caso de uso, o cliente será bloqueado, caso ele tenha alguma pendência de cadastro.
3 - PRE-CONDIÇÃO	Não aplicável
4 - FLUXO 4.1 - FLUXO PRINCIPAL	a) Ao entrar com o cnpj, se o cliente tiver alguma pendência de cadastro, a opção novo pedido será bloqueado até que o cliente regularize sua situação.
5 - REQUISITOS NÃO FUNCIONAIS	Não aplicável.
6 - REGRA DE NEGÓCIO	RN1: Com a pendência de cadastro, será exibida uma mensagem

CASO DE USO 06	Liberar cliente para fazer pedido.
1 - ATOR PRINCIPAL	Vendedor
2 - SUMÁRIO	Neste caso de uso, o vendedor efetuará um novo pedido e também será exibida a forma de pagamento, nº do pedido e o código do vendedor.
3 - PRE-CONDIÇÃO	Não aplicável
4 - FLUXO 4.1 - FLUXO PRINCIPAL	a) O vendedor solicita no sistema a opção novo pedido. b) O sistema solicita a forma de pagamento, com a escolha do mesmo, serão exibidos os produtos que eventualmente será comprado pelo cliente. c) O vendedor solicitará no sistema a opção novo produto.

	d) Logo será exibido numa lista com os produtos escolhidos pelo cliente. e) O vendedor poderá escolher o subfluxo na opção excluir produto, caso tenha inserido um produto errado. f) O sistema disponibilizará o total do pedido atualizado com a inserção de novos produtos. g) Ao final do pedido, o vendedor escolherá a opção fechar pedido.
4.2 - FLUXO ALTERNATIVO	a) O vendedor poderá cancelar a operação de pedido, fechando a interface.
4.3 - SUBFLUXO	Excluir produto
4.3.1 – SUBFLUXO	a) O vendedor seleciona o produto que deseja excluir. b) O vendedor escolherá a opção excluir produto, logo o produto será excluído da lista de produto do pedido.
5 - REQUISITOS NÃO FUNCIONAIS	Não aplicável
6 - REGRA DE NEGÓCIO	RN1: O campo forma de pagamento será preenchido de acordo com a escolhida pelo cliente. RN2: O campo nº do pedido será preenchido automaticamente. RN3: O campo código do vendedor será preenchido automaticamente.

CASO DE USO 07	Informar condições de pagamento.
1 - ATOR PRINCIPAL	Cliente
2 - SUMÁRIO	Neste caso de uso, o cliente informará a condição de pagamento para o pedido.
3 - PRE-CONDIÇÃO	Não aplicável
4 - FLUXO	
4.1 - FLUXO PRINCIPAL	a) O cliente escolherá a forma de pagamento, de acordo com as opções disponibilizadas pelo sistema.
5 - REQUISITOS NÃO FUNCIONAIS	Não aplicável
6 - REGRA DE NEGÓCIO	RN1: É obrigatório a escolha da forma de pagamento, caso contrário será exibida uma mensagem e logo voltará a

	opção de escolha.
--	-------------------

CASO DE USO 08	Alterar preços conforme a condição pagamento
1 - ATOR PRINCIPAL	Vendedor
2 - SUMÁRIO	Neste caso de uso, conforme a forma de pagamento escolhida, o preço sofrerá alguma alteração. Mas podendo ser alterada quando o cliente for adquirir um produto
3 - PRE-CONDIÇÃO	Não aplicável
4 - FLUXO	a) De acordo com a forma de pagamento escolhida, o preço
4.1 - FLUXO PRINCIPAL	do produto poderá sofrer acréscimo ou desconto.
5 - REQUISITOS NÃO FUNCIONAIS	Não aplicável

CASO DE USO 09	Informar o produto que esteja precisando
1 - ATOR PRINCIPAL	Vendedor
2 - SUMÁRIO	Neste caso de uso, o cliente informa ao vendedor o produto que ele esteja precisando, e vendedor informa a situação em que o produto se encontra em relação a preço, desconto e disponibilidade no estoque.
3 - PRE-CONDIÇÃO	Não aplicável
4 - FLUXO	
4.1 - FLUXO PRINCIPAL	a) O vendedor solicita no sistema a opção novo produto. b) O cliente informará o produto que esteja precisando, por nome ou por código, e logo o vendedor terá as seguintes informações, preço, desconto e disponibilidade no estoque. c) As condições satisfazendo o cliente, o vendedor escolherá a opção no sistema inserir produto, logo o produto será adicionado ao pedido. d) O vendedor poderá alterar a condição de pagamento, para se analisar o preço. e) O estoque será informado ao vendedor pela forma de sinal, quando o estoque estiver alto terá no sistema um sinal verde, quando estoque para acabar o sinal estará com o sinal amarelo e quando o estoque estiver zerado o sinal será vermelho.

	f) Quando o produto estiver em promoção, o sistema informará ao vendedor que o produto terá um desconto adicional.
4.2 - FLUXO ALTERNATIVO	a) O vendedor poderá cancelar a operação de novo produto, fechando a interface. b) O vendedor poderá excluir um produto, quando se minimizar a interface de novo produto e exclui lá na interface de pedido na opção excluir produto.
4.3 - FLUXO EXECUÇÃO	a) Na interface do produto, será exibido um quadro dividido nas seguintes opções, código, descrição, preço e estoque. b) Os produtos estarão contidos em uma lista, ordenadas pela descrição em ordem alfabética. c) Para a inserção de um produto na lista de pedido, basta selecionar o produto, colocar a quantidade do produto e solicitar a opção inserir produto. d) Caso o produto já esteja no pedido, o vendedor tente inserir novamente o produto, logo o sistema exibirá uma mensagem e informando que o produto já está contido no pedido. e) Quando o produto tiver algum desconto, o sistema exibirá uma mensagem e no campo desconto terá a quantidade em porcentagem. f) Caso o vendedor execute a opção inserir produto, sem que esteja um produto selecionado, o sistema exibirá uma mensagem.
5 - REQUISITOS NÃO FUNCIONAIS	Não aplicável
6 - REGRA DE NEGÓCIO	RN1: O campo desconto será preenchido automaticamente de acordo com o produto selecionado. RN2: O campo quantidade será preenchido de acordo com o cliente.

CASO DE USO 10	Exibir a situação do estoque
-----------------------	-------------------------------------

1 - ATOR PRINCIPAL	Vendedor
2 - SUMÁRIO	Neste caso de uso, o sistema informará ao vendedor a disponibilidade do estoque de cada produto.
3 - PRE-CONDIÇÃO	Não aplicável
4 - FLUXO 4.1 - FLUXO PRINCIPAL	a) De acordo com a quantidade em estoque, será exibido a informação de cada produto em estoque.
5 - REQUISITOS NÃO FUNCIONAIS	Não aplicável
6 - REGRA DE NEGÓCIO	RN1: A informação do produto será exibida automaticamente ao selecionar o produto, de acordo com a quantidade em estoque.

CASO DE USO 11	Informar as condições do produto
1 - ATOR PRINCIPAL	Vendedor
2 - SUMÁRIO	Neste caso de uso, ao se selecionar o produto será exibido as informações do mesmo.
3 - PRE-CONDIÇÃO	Não aplicável
4 - FLUXO 4.1 - FLUXO PRINCIPAL	a) Ao se selecionar o produto, será exibido na lista de produto as informações do produto relacionada a preço, disponibilidade de estoque e desconto caso o tenha.
5 - REQUISITOS NÃO FUNCIONAIS	Não aplicável
6 - REGRA DE NEGÓCIO	RN1: As informações do produto serão preenchidas na lista de produto automaticamente ao se selecionar o produto.

CASO DE USO 12	Calcular o total do pedido
1 - ATOR PRINCIPAL	Vendedor
2 - SUMÁRIO	Neste caso de uso, a cada produto inserindo na lista de pedido, o total do pedido será calculado.
3 - PRE-CONDIÇÃO	Não aplicável
4 - FLUXO 4.1 - FLUXO PRINCIPAL	a) Há cada produto inserido no pedido será calculado o total do pedido
5 - REQUISITOS NÃO FUNCIONAIS	Não aplicável
6 - REGRA DE NEGÓCIO	RN1: O campo total do pedido será preenchido

	automaticamente de acordo com inserção de produtos.
--	---

CASO DE USO 13	Informar os clientes que encontram c/ boletos em aberto
1 - ATOR PRINCIPAL	Setor de Pagamento
2 - SUMÁRIO	Neste caso de uso, o setor de pagamento informa ao vendedor todos os clientes que constarem boletos em aberto e vencidos.
3 - PRE-CONDIÇÃO	Não aplicável
4 - FLUXO	a) Enviar todas as informações relacionadas a boletos ao
4.1 - FLUXO PRINCIPAL	vendedor.
5 - REQUISITOS NÃO FUNCIONAIS	Não aplicável

CASO DE USO 14	Bloquear os clientes com boleto em aberto
1 - ATOR PRINCIPAL	Setor de Pagamento
2 - SUMÁRIO	Neste caso de uso, o setor de pagamento bloqueará todos os clientes que possuírem boletos em aberto.
3 - PRE-CONDIÇÃO	Não aplicável
4 - FLUXO	
4.1 - FLUXO PRINCIPAL	a) Serão bloqueados todos os clientes que possuírem débitos. b) O cliente terá o cnpj bloqueado para fazer pedido. c) O cliente terá o cnpj desbloqueado somente depois da quitação dos boletos em aberto.
5 - REQUISITOS NÃO FUNCIONAIS	Não aplicável
6 - REGRA DE NEGÓCIO	RN1: Caso o cliente esteja com cnpj bloqueado, será exibida uma mensagem quando o sistema pedir a inserção do cnpj.

CASO DE USO 15	Imprimir boletos
1 - ATOR PRINCIPAL	Setor de Pagamento
2 - SUMÁRIO	Neste caso de uso, serão impressos todos os boletos relacionados aos pedidos.

3 - PRE-CONDIÇÃO	Não aplicável
4 - FLUXO	
4.1 - FLUXO PRINCIPAL	a) Imprimir os boletos relacionado ao pedido.
5 - REQUISITOS NÃO FUNCIONAIS	Não aplicável

CASO DE USO 16	Distribuir os pedidos
1 - ATOR PRINCIPAL	Vendedor
2 - SUMÁRIO	Neste caso de uso, o vendedor envia os pedidos realizados para os setores de armazém e de pagamento, aonde o setor de armazém emitirá a nota fiscal e separará os produtos relacionados ao pedido e no setor de pagamento serão impresso os boletos.
3 - PRE-CONDIÇÃO	Não aplicável
4 - FLUXO	a) Enviar os pedidos para o setor de armazém.
4.1 - FLUXO PRINCIPAL	b) Enviar os pedidos para o setor de pagamento.
4.2 - FLUXO DE EXECUÇÃO	a) No sistema terá uma opção comunicação com a empresa, logo serão enviados os pedidos e o sistema será atualizado.
5 - REQUISITOS NÃO FUNCIONAIS	Não aplicável
6 - REGRA DE NEGÓCIO	RN1: Logo depois de se enviar os pedidos e sistema ser atualizado, será exibida uma mensagem.

CASO DE USO 17	Informar status do pedido
1 - ATOR PRINCIPAL	Setor de Armazém
2 - SUMÁRIO	Neste caso de uso, o setor de armazém enviará ao vendedor em que status o pedido se encontra.
3 - PRE-CONDIÇÃO	Não aplicável
4 - FLUXO	
4.1 - FLUXO PRINCIPAL	a) O setor de armazém enviará ao vendedor em que status (imprimindo nota fiscal, separando pedido, enviando para transportadora) o pedido se encontra. b) Caso o status esteja na opção imprimindo nota fiscal, o setor de armazém recebeu o pedido e já estão emitindo a nota fiscal. c) Caso o status esteja na opção separando pedido, o setor

	de armazém está separando o pedido que está relacionado à nota fiscal. d) Caso o status esteja na opção enviando para transportadora, o setor já separou o pedido e já imprimiu a nota fiscal e já pode ser recolhido pela transportadora para que possa fazer a entrega. e) Caso o status esteja na opção entregue, o cliente já recebeu o pedido, para essa confirmação o cliente deverá assinar o canhoto da nota fiscal para que a transportadora possa enviar este canhoto ao setor de armazém.
5 - REQUISITOS NÃO FUNCIONAIS	Não aplicável
6 - REGRA DE NEGÓCIO	RN1: O campo status do pedido será preenchido automaticamente, quando se exibir informações dos últimos pedidos.

CASO DE USO 18	Emitir nota fiscal
1 - ATOR PRINCIPAL	Setor de Armazém
2 - SUMÁRIO	Neste caso de uso, será emitida a nota fiscal relacionada ao pedido enviado pelo vendedor.
3 - PRE-CONDIÇÃO	Não aplicável
4 - FLUXO	
4.1 - FLUXO PRINCIPAL	a) Emitir a nota fiscal relacionada ao pedido
5 - REQUISITOS NÃO FUNCIONAIS	Não aplicável

CASO DE USO 19	Separar produtos relacionados ao pedido
1 - ATOR PRINCIPAL	Setor de Armazém
2 - SUMÁRIO	Neste caso de uso, é aonde serão separados os produtos de acordo com o pedido.
3 - PRE-CONDIÇÃO	Não aplicável
4 - FLUXO	
4.1 - FLUXO PRINCIPAL	a) Separar os produtos relacionados na nota fiscal.
5 - REQUISITOS NÃO FUNCIONAIS	Não aplicável

ANEXO B – DESCRIÇÃO DE MÉTODOS

Logo abaixo segue algumas descrições de métodos relacionados no diagrama de projeto de classe:

Liberar_Fazer_Ped (Cliente : Objeto)

 Codigo inteiro;

 Parar Condicao;

 Parar = falso

 Condicao_Pagam;

 Enquanto (Não Parar) faça

 Mensagem(“Informe o código do produto:”)

 Ler (Codigo)

 Informar_Produto(codigo);

 Calcular_Total_Pedido(Produto.preco);

```
        Mensagem (“Deseja incluir novo produto: Sim ou Não”)
        Se Nao faca
            Parar = verdadeiro;
        Fim enquanto
    Fim Método
```

```
Informar_Produto (Codigo inteiro)
    Condicao_Produto(Codigo);
    Condicao_Estoque(Codigo);
    Informar_Promoca(Codigo);
    Mensagem (“Deseja inserir o produto no pedido: Sim ou Não”)
    Se Sim faca
        Inserir_Produto(Codigo);
    Fim Metodo
```