

INTRODUÇÃO AO PROCESSO DE SOFTWARE PRAXIS

Prof. Paulo Vinícius Moreira Dutra

Sumário

1.	Praxis 3.0	3
1.1	Visão geral	3
1.1.1	Introdução	3
1.2	Conceitos	3
1.3	Padrões e recursos	8
1.4	Ciclo de Vida	12
2.	Modelagem do Problema	13
2.1	Visão geral	13
2.1.1	Introdução	13
2.2	Organização do modelo do problema	13
2.3	Visão de Requisitos	13
2.3.1.1	Objetivos	14
2.3.1.2	Escopo	14
2.3.1.3	Materiais de Referência	15
2.3.1.4	Definições e siglas	16
2.3.2	Descrição Geral	16
2.3.2.1	Diagrama de Contexto	16
2.3.2.2	Perspectiva do Produto	17

1. Praxis 3.0

1.1 Visão geral

1.1.1 Introdução

Esta seção define o **Praxis**, um processo destinado a suportar projetos didáticos em disciplinas de Engenharia de Software de cursos de Informática. A sigla **Praxis** significa **PR**ocesso para **Ap**licativos **eX**tensíveis **I**nterativo**S**, refletindo uma ênfase no desenvolvimento de aplicativos gráficos interativos, baseados na tecnologia orientada a objetos.

O Praxis é desenhado para suportar projetos de seis meses a um ano de duração, realizados individualmente ou por pequenas equipes. Com isto, pretende-se que ele seja utilizável para projetos de fim de curso, ou projetos de aplicação de disciplinas de engenharia de software.

A UML é a notação de modelagem utilizada no Praxis, em todos os passos em que for aplicável. Os padrões incluídos procuram ser conformes com os padrões correspondentes do IEEE [IEEE94].

1.2 Conceitos

O Praxis é composto de diversos elementos em um projeto, que são chamados de disciplinas. Ao todo são 10 disciplinas, que são classificadas em quatro grupos: **Especificação, Solução, Gestão e Ambiente**. As disciplinas ajudam a estruturar a modelagem do projeto, melhorar o entendimento entre as partes envolvidas e também o planejamento do projeto. A tabela 1.1 lista todas as 10 disciplinas que envolvem o Praxis.

Grupo	Disciplina	Sigla	Objetivo
<i>Especificação</i>	Requisitos	RQ	Visa a obter um conjunto de requisitos de um produto, acordado entre cliente e fornecedor.
	Análise	AN	Visa a detalhar, estruturar e validar os requisitos de um produto, em termos de um modelo conceitual do problema.
<i>Solução</i>	Desenho	DS	Definir uma estrutura implementável para um produto de software que atenda aos requisitos especificados para ele.
	Teste	TS	Verificar os resultados da implementação, através do planejamento, desenho e realizações de testes.
	Implementação	IM	Realizar o desenho de um sistema em termos diversos tipos de componentes de código e de documentação de uso, conforme as tecnologias escolhidas.

<i>Gestão</i>	Gestão de Qualidade	GQ	Verificar e garantir a qualidade em projetos e produtos de software.
	Gestão de projetos	GP	Planejar e controlar os projetos de desenvolvimento de software.
	Gestão de alterações	GA	Administrar as alterações em requisitos e artefatos dos projetos e produtos.
<i>Ambiente</i>	Engenharia de Processos	EP	Dar suporte e promover melhorias nos próprios processos de software
	Engenharia de Sistemas	ES	Desenvolver o ambiente de sistema em que um produto está incluído.

Tabela 1.1 – Disciplinas Praxis

As disciplinas Praxis possuem quatro elementos importantes: **Tarefa, Papel, Produto de Trabalho e Orientação**.

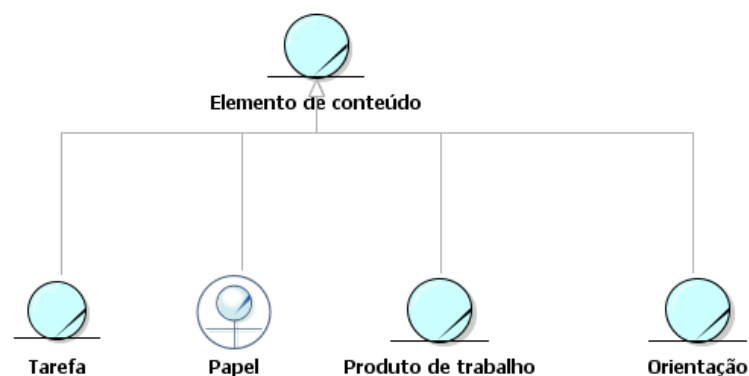


Figura 1.1 – Elementos principais das disciplinas Praxis.

Cada disciplina contém processos, ou seja, a descrição de como um trabalho deve ser realizado. Os processos de cada uma das disciplinas são decompostos em **atividades** que representam um agrupamento de tarefas. A figura 1.1 ilustra que uma disciplina possui uma ou mais atividades, uma atividade pode possuir zero ou mais subatividades, e por último, uma atividade possui uma ou mais tarefas a serem realizadas.

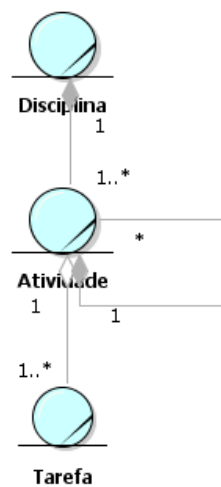


Figura 1.2 – Decomposição das disciplinas no Praxis

Atividades

É um elemento de definição de trabalho. Define as atividades básicas de trabalho a serem realizadas dentro de uma disciplina. A figura 1.3 ilustra um exemplo da atividade utilizada na disciplina de requisitos. Nesta sequência lógica da figura 1.3 é realizada as seguintes atividades: **Identificação dos requisitos, Levantamento dos requisitos, Detalhamento dos requisitos, Inspeção dos requisitos e Publicação dos requisitos.**

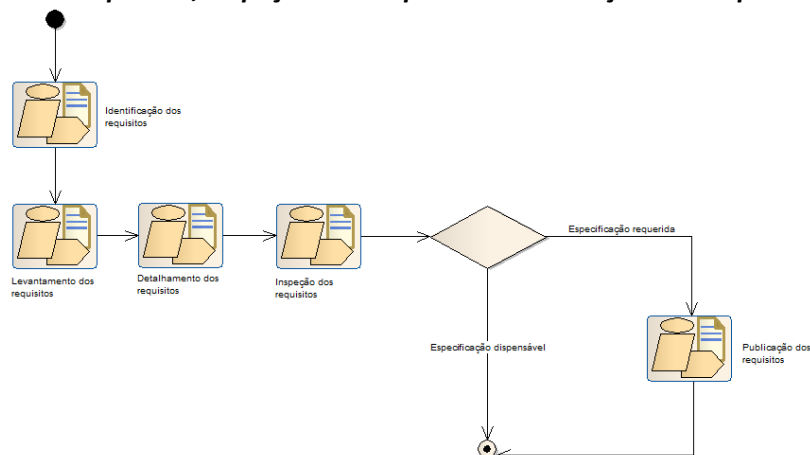


Figura 1.3 – Exemplo de atividade da disciplina de requisitos.

Papéis

É um conjunto de responsabilidades e competências desempenhado por uma ou mais pessoas dentro de um projeto de software. A tabela 1.2 apresenta alguns papéis que fazem parte de um projeto utilizando Praxis, no qual descreve uma descrição e em qual disciplina o papel é utilizado. Um papel pode fazer parte em mais de uma disciplina.

Papel	Disciplina	Responsabilidade
Engenheiro de Requisitos	RQ	Papel com a responsabilidade de levantar, detalhar e especificar requisitos.
Programador	IM	Responsável pela implementação do código dos produtos
Suporte aos usuários	ES	Responsável por realizar suporte aos usuários
Usuário		Utilização do produto, serviço ou resultado do projeto.
Gerente Geral		Planejamento, direção, controle e avaliação das atividades gerais da organização.
Programador de testes	TS	Desenvolvedor responsável pelos scripts de testes de um produto.
Engenheiro de processos	GQ	Responsável técnico por atividades relativas aos processos.

Tabela 1.2 – Exemplos de alguns Papéis do Praxis.

Tarefas

É uma unidade de Trabalho que pode ser atribuída a um papel. Cada tarefa possui um papel responsável, que é chamado de papel principal. O trabalho de uma tarefa é dividido em

passos, que é uma parte do trabalho descrito por uma tarefa. A figura 1.4 ilustra como é o relacionamento entre o papel e a tarefa.

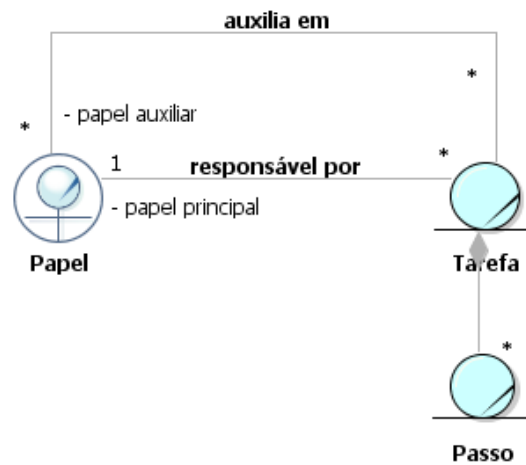


Figura 1.4 – Tarefas, passos e papéis

A figura 1.5 ilustra o exemplo de diversas tarefas executadas na disciplina de requisitos, mas especificamente na atividade de **Identificação dos requisitos**.

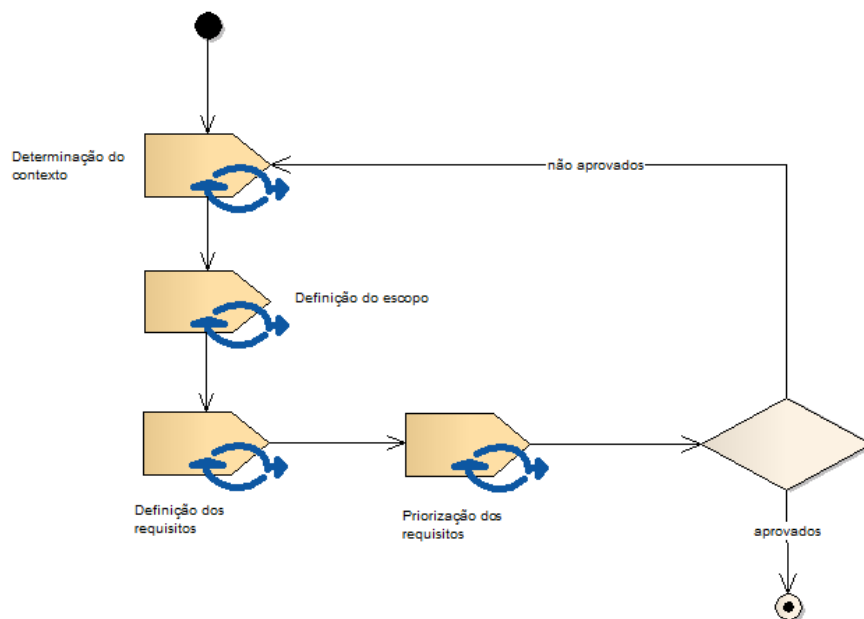


Figura 1.5 – Tarefas da atividade de Identificação dos Requisitos.

Abaixo são descritos exemplos de passos a serem realizados pela tarefa **Determinação do contexto**:

- Definição da missão do produto;
- Definição dos limites do produto;
- Levantamento dos benefícios do produto;
- Levantamento das referências aplicáveis;
- Determinação da expectativa de custo e prazo.

Produto de trabalho

Os produtos de trabalho são os artefatos consumidos, gerados ou modificados por cada tarefa. A tabela 1.3 abaixo descreve alguns tipos de artefatos.

Tipo de produto de trabalho	Descrição
Documento	Artefato produzido por ferramenta de processamento de texto ou hipertexto, para fins de documentação.
Código fonte	Código fonte desenvolvido para o produto
Executável	Códigos gerados para aplicação. Por exemplo, pastas, .class, .jar, .exe.
Plano	Artefato de planejamento de instâncias de atividades e tarefas, possivelmente com os respectivos recursos, responsabilidades e riscos.
Repositório	Repositório ou biblioteca que abrange coleção organizada de vários tipos de produtos de trabalho.

Tabela 1.3 – Exemplos de alguns artefatos

A figura 1.6 ilustra que uma tarefa consome zero ou mais produtos de trabalho (insumos obrigatórios), produz zero ou mais produtos de trabalho (resultados), consome opcionalmente zero ou mais produtos de trabalho (insumos opcionais).

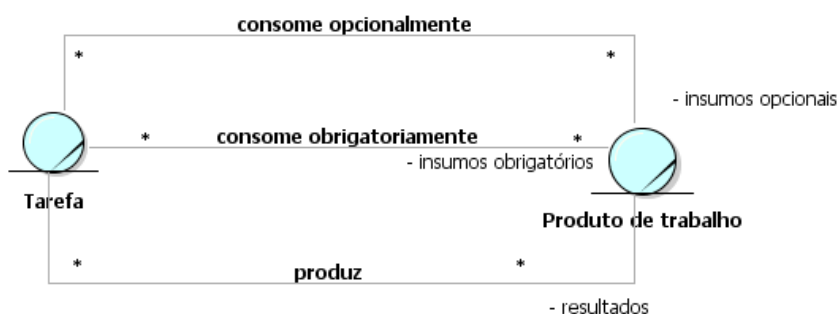


Figura 1.6 – Tarefas e produtos de trabalho

Orientações

Orientações proveem informação adicional, orientações normativas, de como realizar uma determinada tarefa ou como desempenha um Papel dentro de uma disciplina. As orientações possuem os seguintes itens: **Padrão, Diretriz, Técnica e Prática**.

Orientação	Definição
Padrão	Documento que expressa requisitos obrigatórios, empregado e cobrado para prescrever uma abordagem uniforme e disciplinada para atividades e produtos de trabalho.
Diretriz	Orientação que provê recomendações, regras e detalhes adicionais sobre produtos de trabalho e sobre execução de tarefas.
Técnica	Um procedimento sistemático definido usado por um recurso humano para realizar uma atividade a fim de produzir um produto ou resultado ou

	oferecer um serviço, e que pode empregar uma ou mais ferramentas.
Prática	Maneira ou estratégia comprovada de trabalho para atingir uma meta, que tem um impacto positivo na qualidade de um produto de trabalho ou processo.

Tabela 1.4 – Orientações normativas

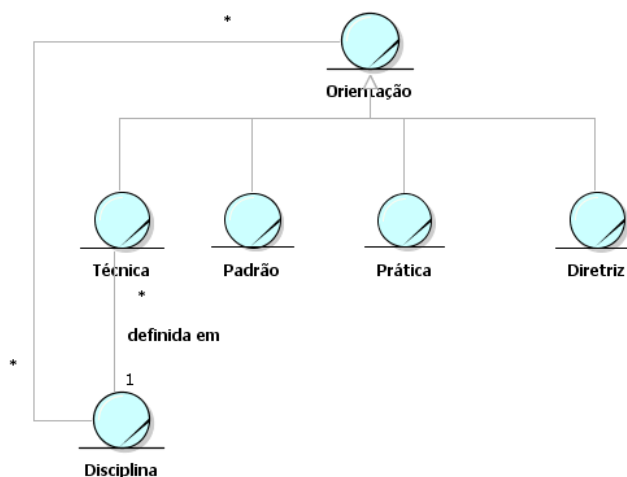


Figura 1.7 – Orientações normativas

1.3 Padrões e recursos

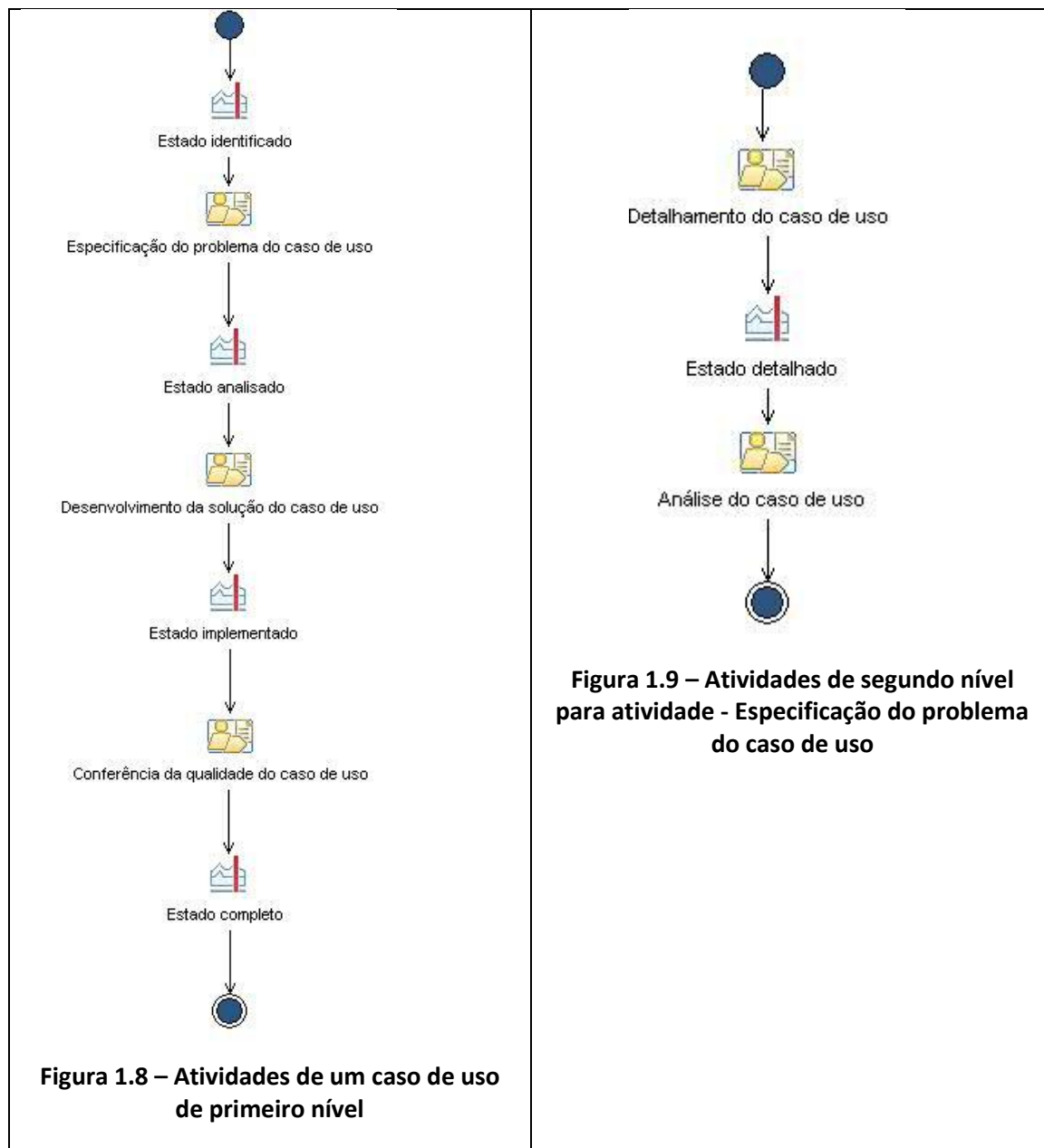
Padrão de recursos é um processo especial que descreve um aglomerado reutilizável de atividades em uma área de processo, fornecendo uma abordagem de desenvolvimento consistente para problemas comuns. Dentro do Praxis 3.0, existem os seguintes tipos de **Padrão de recursos** reutilizáveis.

- **Disciplinas:** correspondente a cada uma das disciplinas descritas na tabela 1.1;
- **Processos subsidiários:** Atendem a propósitos específicos, dentro de um processo de entrega. Processo de entrega descreve uma abordagem completa e integrada para desempenhar um tipo específico de projeto, inclusive o que é produzido, como é produzido e a equipe necessária para participar de todo o ciclo de vida do projeto, como os processos de aquisição (Gestão de projetos), manutenção (Gestão de alterações) e inovação técnica (Engenharia de Processos).
- **Subprocessos:** processos parciais usados como blocos de construção dos Processos de Entrega, por exemplo o desenvolvimento de caso de uso.
- **Desenvolvimento de caso de uso:** Demonstra atividades e tarefas utilizadas na maioria das disciplinas para construção de casos de uso.
- **Iteração genérica:** Mostra como instâncias do desenvolvimento de caso de uso são combinadas com atividades de abertura, suporte e fechamento de uma iteração genérica, dentro do modelo de ciclo de vida em espiral.

Desenvolvimento de caso de uso

Os casos de uso no processo Praxis são extremamente importantes, já que fazem parte na maioria das disciplinas e são utilizados para identificar as funcionalidades do sistema (figura 1.8). Para cada disciplina os casos de uso seguem uma hierarquia de atividades e tarefas que possuem ciclo de vida de desenvolvimento. A figura 1.8 ilustra o exemplo de atividades para

criação dos casos de uso. O caso de uso parte de um estado inicial, chamado de **Estado identificado**, atingido na primeira iteração de um projeto, quando os casos de uso recebem um nome e uma descrição sucinta. A partir daí executa-se um conjunto de atividades de primeiro nível. Cada atividade de primeiro nível são compostas de atividades de segundo nível, oriundas das varias disciplinas.



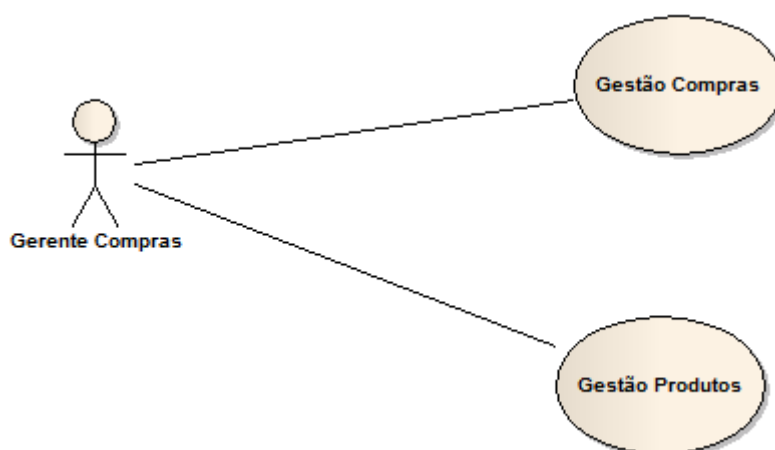


Figura 1.10 – Exemplo de caso de uso

Na maioria das atividades de segundo nível, quando o material desenvolvido é aprovado, o caso de uso atinge um novo estado de desenvolvimento, no qual passam por um critério de aprovação. A tabela 1.5 mostra a lista dos estados de caso de uso.

Número	Nome	Definição	Disciplina	Critérios de aprovação
10	Identificado	Recebeu nome da descrição sucinta	RQ	Revisão gerencial
20	Detalhado	Os fluxos de eventos estão completos e consistentes com os atores e interfaces de usuário e de sistema	RQ	Inspeção de requisitos; revisão gerencial.
30	Analísado	Os fluxos de eventos foram realizados como interações das colaborações de análise	AN	Inspeção de análise
40	Desenhado	As colaborações de uso derivadas estão completamente desenhadas	DS	Inspeção de desenho externo.
50	Especificado	As colaborações de testes derivadas estão completamente especificados.	TS	Inspeção de desenho dos testes.
60	Realizado	As colaborações de desenho interno derivadas foram realizadas, tendo as classes participantes de entidade e controle sido implementadas.	IM	Teste de unidade; testes de regressão; inspeção de implementação.
70	Implementado	As colaborações de desenho interno estão completamente implementadas, integradas e documentadas.	IM	Inspeção de desenho interno; testes de integração (automatizados, de regressão e manuais).
80	Testado	Os testes de aceitação foram executados com sucesso por uma equipe	TS	Teste de sistemas (automatizados, inclusive regressão, e

		independente e os defeitos encontrados foram todos corrigidos.		testes manuais destrutivos).
90	Validado	Foi aprovado na avaliação dos usuários.	GQ	Avaliação de uso.
100	Completo	A linha de base correspondente foi aprovada em auditoria da qualidade	GQ	Auditoria da qualidade revisão gerencial.

Tabela 1.5 – Estados de desenvolvimento dos casos de uso

Iterações

Uma iteração é um grupo de atividades que são repetidas mais de uma vez, utilizadas para organizar o trabalho em ciclos repetitivos. O grupo básico de um ciclo de vida em espiral é a Iteração genérica, no qual há um conjunto de instâncias de desenvolvimento de caso de uso, executadas em paralelo.

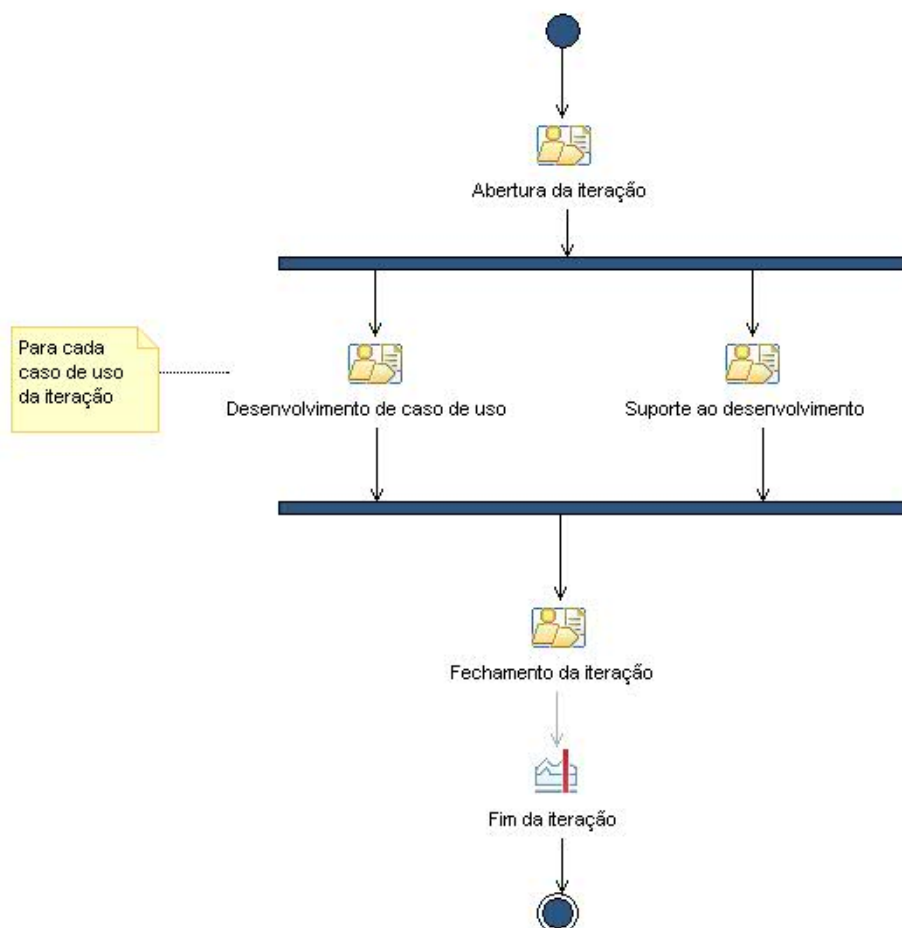


Figura 1.11 – Iteração genérica de desenvolvimento do caso de uso

1.4 Ciclo de Vida

O ciclo de vida padrão do Praxis 3.0 é formado por fases, figura 1.12, e são elas: **Iniciação, Elaboração, Construção e Transição**. O ciclo de vida padrão segue um modelo quase-espiral, figura 1.13. A espiral é executada dentro das fases de Elaboração e Construção, no qual são formadas iterações. Fora da espiral ficam as fases de **Iniciação e Transição**.

Fase	Descrição
Iniciação	Fase na qual necessidades dos usuários e conceitos da aplicação são analisados o suficiente para justificar a especificação de um produto de software, resultando em uma proposta de especificação. Dependendo da opção adotada, ela tem quatro iterações.
Elaboração	Fase na qual a especificação do produto é detalhada o suficiente para modelar conceitualmente o domínio do problema, validar os requisitos em termos deste modelo conceitual e permitir um planejamento apurado da fase de construção.
Construção	Fase na qual é desenvolvida (desenhada, implementada e testada) uma liberação completamente operacional do produto, que atende aos requisitos especificados.
Transição	Fase na qual o produto é colocado à disposição de uma comunidade de usuários para testes finais, treinamento e uso inicial.



Figura 1.12 – Fases do Ciclo de vida padrão

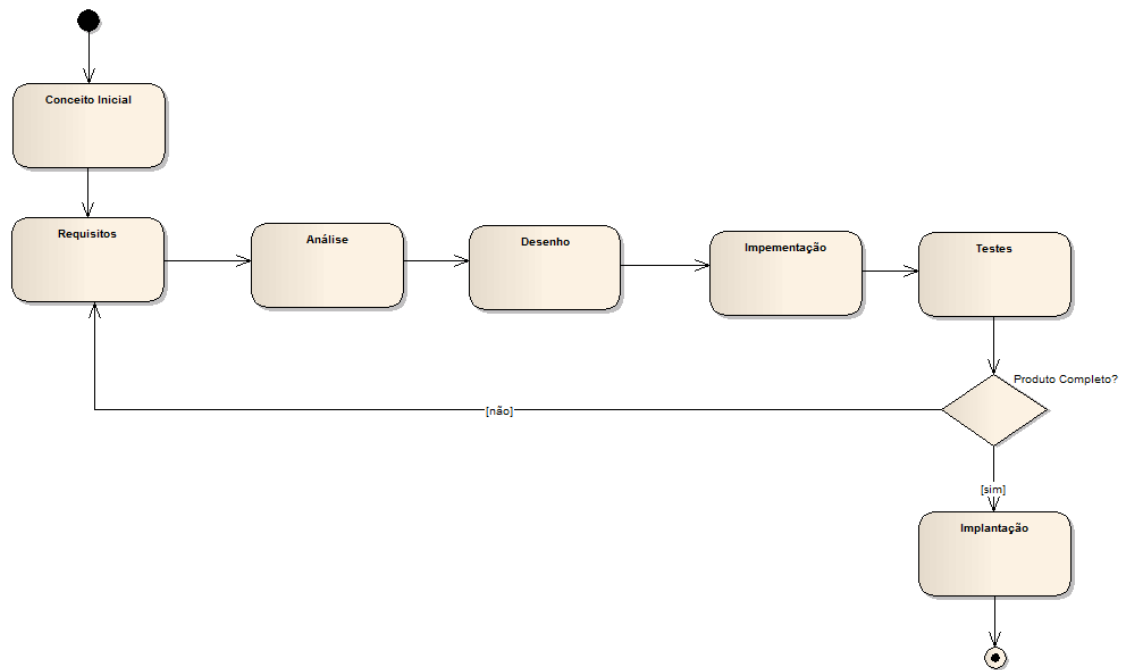


Figura 1.12 – Ciclo de vida do modelo Quase-Espiral

2. Modelagem do Problema

2.1 Visão geral

2.1.1 Introdução

A modelagem do problema tem o objetivo de descrever os requisitos do problema que deve ser resolvido por um produto, sistema ou aplicativo, de maneira mais completa e precisa possível. Este é um artefato que é construído ao longo das atividades das disciplinas de **Requisitos** e **Análise**.

O modelo do problema cobre apenas aspectos relacionados a requisitos do sistema, no qual gera um artefato chamado de **Especificação de Requisitos**.

2.2 Organização do modelo do problema

O modelo do problema possui duas visões: **Visão de Requisitos** e **Visão de análise**.

2.3 Visão de Requisitos

Descreve o problema do ponto de vista dos usuários. Seu conteúdo segue a organização recomendada pela norma IEEE-830. Esta organização possui os seguintes itens do documento:

- **Introdução:** Provê uma visão geral dos requisitos; suas subseções descrevem os objetivos, escopo, limites e benefícios esperados do produto a ser desenvolvido.
- **Descrição Geral:** Descreve uma visão geral do sistema, sem detalhar os requisitos funcionais e não-funcionais, mas fornecendo uma forma que facilitará o entendimento. Neste item devem conter diagramas de contexto, restrições de memória, modos de operação, requisitos de adaptação ao ambiente, funções do sistema e descrição dos usuários.
- **Requisitos específicos:** Neste item devem ser especificados os requisitos funcionais, requisitos não-funcionais e requisitos de interfaces externas.

A seguir serão descritos os passos para especificar a visão de requisitos. Estas especificações fazem parte do artefato **Especificação de Requisitos**.

2.3.1 Introdução à documentação da visão de requisitos

2.3.1.1 Objetivos

No campo **Objetivos deste artefato**, descrever o propósito do **Modelo de Problema**. No campo **Público-alvo**, especificar o público desse artefato. Tabela 2.1

Objetivos deste artefato	Descrever e especificar os requisitos que devem ser atendidas pelo produto Merci, de forma a satisfazer as necessidades de seus clientes, bem como definir o produto a ser feito, para os desenvolvedores da UnitedHackers Informática.
Público-alvo	Cliente, usuários e desenvolvedores do projeto Merci.

Tabela 2.1 - Objetivos

2.3.1.2 Escopo

No item escopo deve ser informado a **Missão do Produto**, **Limites do produto**, **Benefícios do produto** e **Referências aplicáveis**.

Missão do Produto

Descreve sucintamente que valor o produto acrescenta para o cliente e os usuários.

Nome do produto	Merci 1.5
Componentes principais	Merci (Componente único)
Missão do produto	Apoio informatizado ao controle de vendas e de compras de pequenas mercearias, implementado com tecnologia do estado da arte na época da contratação.

Tabela 2.2 – Missão do Produto

Limites do produto

Se necessário, esclarecer os limites do produto, ou seja, funções que o produto não terá. Isso evita falsas expectativas por parte do cliente e usuários.

Número	Limite
1	O Merci só fará a Emissão de Nota Fiscal durante a Operação de Venda.
2	O Merci não fará vendas parcelas e só receberá dinheiro ou cheque.
3	O Merci não manterá um cadastro de clientes da mercearia.
4	O backup e a recuperação das bases de dados do sistema ficam a cargo da administração de dados do cliente e não serão providos pelo Merci.
5	O Merci não terá ajuda on-line, mas apenas um manual de uso.

Tabela 2.3 – Limites do Produto

Benefícios do Produto

Identificar os benefícios que se espera obter com o produto e o valor deles para o cliente. Os benefícios devem ser classificados em três categorias: **Essencial**, **desejável** e **opcional**.

- **Essencial:** benefício sem cujo atendimento o produto é inaceitável;
- **Desejável:** benefício cujo atendimento aumenta o valor do produto, mas cuja ausência pode ser relevada em caso de necessidade, traz uma perda maior de valor do produto.
- **Opcional:** benefício cuja ausência traz uma perda menor de valor.

Número	Benefício	Valor para o cliente
1	Diminuição de erros na venda de mercadorias.	Essencial
2	Qualidade na emissão da nota fiscal e tíquete de venda, em relação à emissão manual.	Essencial
3	Identificação de distorções entre o vendido e o estoque.	Essencial
4	Agilidade na compra de mercadorias.	Desejável
5	Economia de mão-de-obra.	Desejável
6	Diminuição do custo de estocagem.	Desejável
7	Identificação de produtos mais e menos vendidos.	Desejável
8	Conhecimento do mercado de fornecedores.	Opcional
9	Indicação de promoções.	Opcional

Tabela 2.4 – Benefícios do Produto

2.3.1.3 Materiais de Referência

Identificar e catalogar todos os materiais cuja consulta possa ser necessária para melhor entendimento dos requisitos.

Número	Tipo do material	Referência bibliográfica
1	Entrevistas	Ata de entrevistas, que podem ser conseguidas com a secretaria da United Hackers Ltda.
2	Manual	United Hackers. <i>Manual de Usuário do Sistema Financeiro Grana-2007</i> . Fornecido pela United Hackers Ltda.

3	Relatório	United Hackers. <i>Enunciado do trabalho – Projeto Merci 1.5</i> . Fornecido pela United Hackers Ltda.
4	Padrão	Metódio Prudente. <i>Manual de Processos de Software v. 3.0</i> . RT – UHI - 015/2007. Fornecido pela United Hackers Ltda.
5	Padrão	IEEE. IEEE Std. 830 — 1998. <i>IEEE Recommended Practice for Software Requirements Specifications</i> . IEEE Software Standards, 2003.

Tabela 2.5 – Materiais de referência

2.3.1.4 Definições e siglas

Descrever todas as definições de todas as siglas, abreviações e termos usados na **Especificação dos Requisitos**. Deve se supor que a **Especificação de Requisitos** será lida tanto por desenvolvedores e por usuários, e por isso deve conter as definições relevantes, tanto em termos da área de aplicação quando de informática usados na **Especificação de Requisitos** e que não sejam do conhecimento do público geral.

Número	Sigla	Definição
1	HD	Disco Rígido
2	RAM	Memória principal do computador
3	Leitor de Código Barras	Equipamento responsável por ler o código de barra do produto

Tabela 2.6 – Definições e siglas

Visão geral deste documento

Descrever neste item o restante da **Especificação de Requisitos**, indicando sua estrutura básica. Veja o exemplo abaixo.

<Informar número do tópico> Visão geral deste documento

De acordo com o Padrão para Especificação de Requisitos, ou seja:

- Parte 2: Descrição geral do produto
- Parte 3: Requisitos específicos
- Parte 4: Informação de suporte

2.3.2 Descrição Geral

2.3.2.1 Diagrama de Contexto

Incluir aqui um diagrama de casos de uso que mostre as fronteiras do produto com seu ambiente de aplicação, inclusive os diversos usuários e outros sistemas do cliente com os quais o produto deve interagir, esse diagrama deve identificar os limites entre o produto e o restante do sistema.

Normalmente, são indicados nesse diagrama apenas os casos de uso de base, excluindo-se os casos de uso de inclusão e extensão. Se o número de casos de uso for

pequeno, podem ser incluídos no diagrama de contexto os casos de uso de extensão mais importantes e que tenham interação significativa com atores. Se por outro lado, o número de casos de uso de base for grande o suficiente para prejudicar a legibilidade do diagrama de contexto, pode-se adotar uma das seguintes providências:

- Partir o diagrama de contexto em vários diagramas;
- Agrupar os casos de uso em pacotes e colocar esses pacotes no diagrama de contexto;

2.3.2.2 Perspectiva do Produto

Restrições de memória

Descreve os limites requeridos de memória primária e secundária. Esses limites só devem ser especificados quando forem requisitos que serão exigidos para aceitação do produto. Para cada tipo de memória, por exemplo, RAM ou HD, declaram-se os limites aplicáveis.

Número	Tipo	Limites aplicáveis
1	HD	O produto deve ocupar no máximo 200MB (sem considerar base de dados)
2	RAM	O produto deve executar em 256MB.

Tabela 2.7 – Restrições de memória

Modos de Operação

Descreve os modos requeridos de operação se esses modos tiverem diferenças significativas de funcionalidade. Diferenças mais simples, como de permissão de acesso, podem ser representadas apenas por diferentes atores. Por exemplo, por existir um modo normal, em que a maioria das funções é executada de forma interativa, e modos especiais, caracterizados por dedicação de certas funções, operação em lote ou automatizada, execução de funções de suporte, backups e recuperação etc.

Número	Tipo	Identificação	Limites aplicáveis
1	Interativa	MODO DE GESTÃO	Modo de operação do Merci , no qual o sistema está disponível para a Gestão de Mercadorias, Gestão Manual de Estoque, Gestão de Pedidos de Compras, Gestão de Fornecedores, Emissão de Relatórios, Gestão de Usuários e Abertura do Caixa.
2	Interativa	MODO DE VENDA	Modo de operação do Merci , no qual o sistema está liberado apenas para a Operação de Venda, Emissão de Nota Fiscal e Fechamento do Caixa.

Tabela 2.8 – Modos de Operação

Requisitos de adaptação ao ambiente

Descreve os possíveis requisitos de adaptação do produto aos ambientes particulares em que ele será implantado, baseado nos modos de operação. Por exemplo, parâmetros e métodos de configuração requeridos para ambientes específicos devem ser descritos aqui. Para cada requisito de adaptação ao ambiente, declaram-se o Tipo do requisito (por exemplo, o que deverá ser configurável) e seus Detalhes (como essa configuração deverá ser feita).

Número	Requisito	Limites aplicáveis
1	Configuração da impressão do ticket de venda e da Nota Fiscal	As dimensões desses relatórios deverão ser configuráveis.

Tabela 2.9 – Requisitos de adaptação ao ambiente.

Usuários e Sistemas Externos

Deverão ser incluídos os atores que representam os usuários humanos da aplicação e os sistemas externos com os quais ela interage.

Número	Ator	Definição
1	Caxeiro	Funcionário operador comercial de caixa.
2	Gerente	Funcionário responsável pela abertura e fechamento do caixa, além do cadastramento de usuários.
3	Gestor de Compras	Funcionário responsável pela gestão dos cadastros de mercadorias e fornecedores, e pela emissão e acompanhamento de pedidos de compra.
4	Gestor de Estoque	Funcionário responsável pela manutenção da consistência entre o estoque físico da mercearia e o estoque cadastrado no Merci .
5	Sistema Financeiro	Sistema de gestão financeira, que recebe os detalhes financeiros das transações diárias, para utilização posterior pela administração financeira da mercearia.

Tabela 2.10 – Usuários e sistemas externos

Restrições

Descrevem aspectos técnicos e gerenciais que possam limitar as opções dos desenvolvedores, tais como:

- Restrições legais;
- Limitações de hardware;
- Restrições relativas a interfaces com outros produtos;
- Restrições quanto a linguagens de programação;
- Requisitos de auditoria;
- Restrições de desempenho;
- Restrições de confiabilidade;
- Restrições de segurança;

Essas restrições, quando detalhadas, normalmente correspondem a requisitos não-funcionais. Para restrição, declaram-se um Tipo, correspondente a um item da lista apresentada anteriormente, e uma Descrição.

Número	Tipo	Definição
1	Ambiente	O ambiente operacional a ser utilizado é o Windows XP (ou compatível).
2	Ambiente	O sistema deverá executar em um Pentium 500 MHz, com impressora de tecnologia laser ou de jato de tinta, a ser usada para impressão de todos os relatórios, exceto os tickets de venda.
3	Ambiente	Será utilizada uma impressora específica para a emissão dos tickets de venda, configurável como impressora suportada pelo ambiente operacional.
4	Expansibilidade	O produto deve ser desenvolvido levando-se em consideração que poderá ser expandido para mais de um caixa.
5	Legal	O produto deverá estar de acordo com as leis e regulamentos vigentes na época de sua aquisição.
6	Segurança	O produto deverá restringir o acesso através de senhas individuais para cada usuário.

Tabela 2.11 – Restrições

Hipóteses de trabalho

Descrevem fatores que não restrições limitativas da solução, como no item **Restrições**, correspondem a uma condição que deve ser satisfeita pelo cliente. Por exemplo, pode-se refletir aqui o acerto sobre qual sistema de gerência de banco de dados ou sistema operacional será usado na plataforma do produto, se essas escolhas influenciarem no desenvolvimento.

Número	Hipótese de trabalho	Dependências
1	Será utilizado um sistema de gerenciamento de banco de dados compatível com o padrão ODBC	O cliente deve adquiri-lo, instalá-lo e povoá-lo.
2	Será utilizado um sistema de gerência de bancos de dados compatível com o padrão JPA.	O cliente deve adquiri-lo, instalá-lo e povoá-lo.

Tabela 2.12 – Hipóteses de Trabalho

2.3.3 Requisitos específicos

Requisitos de Interfaces Externas

Este item serve para facilitar a localização dos requisitos de interfaces externas, reunindo essas interfaces em diagramas que representam diretamente os requisitos de navegação. Cada requisito de navegação é representado por uma associação direcional com o estereótipo <<navigate>>, cujo nome é, normalmente, o nome da operação (da classe de origem) que causa a navegação. Para cada função, o diagrama deve mostrar a navegação entre telas internas à função, mas apenas as telas externas de onde se entra a função e para onde se vai sair dela.

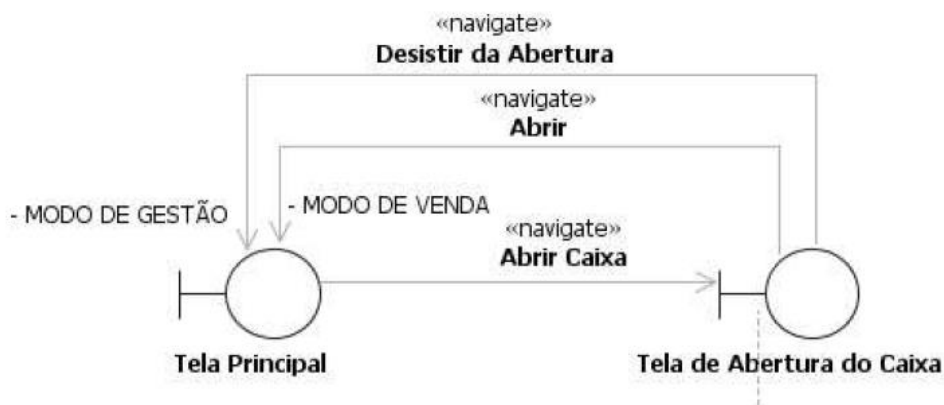


Figura 2.1 – Navegação entre telas

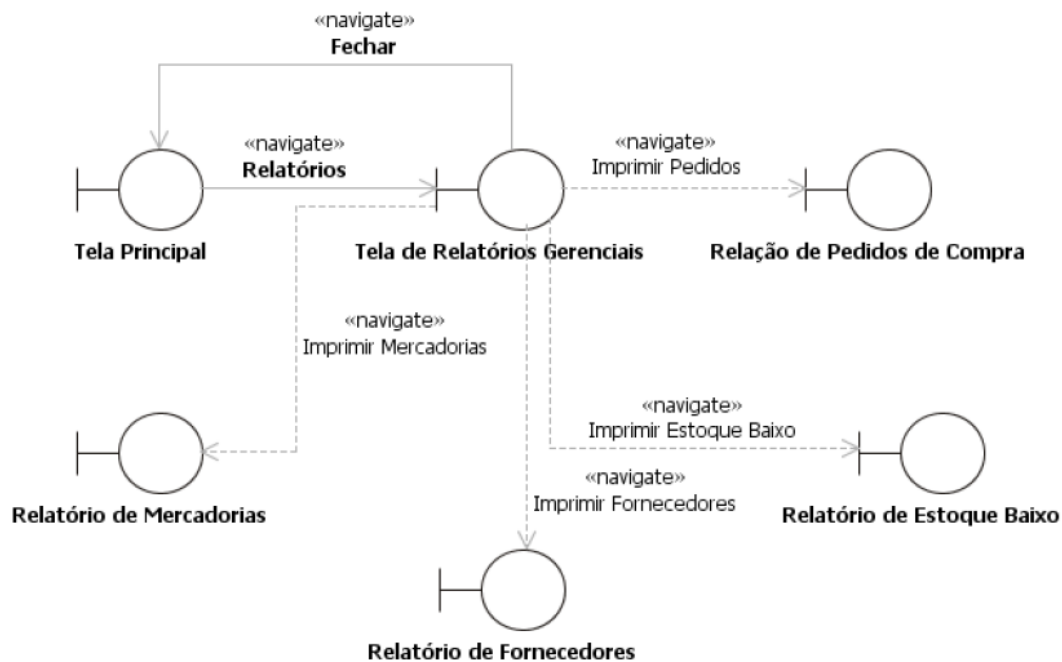


Figura 2.2 – Navegação entre telas

Requisitos funcionais

Este item deverá possuir os casos de uso que modelam as funções da aplicação, agrupadas em grupos funcionais, figura 2.3.

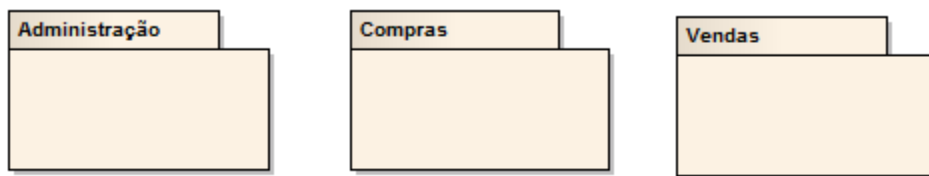


Figura 2.3 – Organização dos requisitos funcionais em grupos

Para cada grupo funcional, mostrar os casos de uso que participam desse grupo, inclusive os casos de uso de inclusão, extensão especialização/generalização.

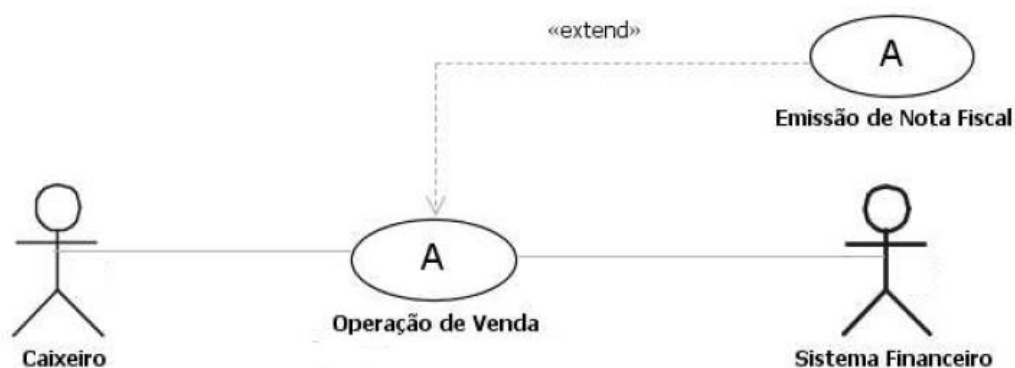


Figura 2.3 – Requisitos funcionais – exemplo de diagrama de caso de uso

A documentação de um caso de uso deve compreender uma descrição resumida da função que o caso de uso representa. O comportamento detalhado do caso de uso pode ser descrito de variis maneiras, neste caso, utiliza-se o diagrama de atividades, outros diagramas podem ser utilizados, tais como, diagrama de estados e sequência.

Nome do Caso de Uso	Operação de Venda
Caso de Uso Geral	
Ator Principal	Caixeiro
Atores Secundários	Sistema Financeiro
Resumo	Operação de Venda ao cliente da mercearia. Durante a operação, é possível incluir, alterar e excluir Itens de Venda de Mercadorias especificadas. Ao término da operação, o Ticket de Venda é emitido, e o saldo no Caixa e os níveis de estoque das Mercadorias dos Itens de Venda são atualizados.
Pré-condições	Descrição
1	O Merci está no MODO DE VENDAS e o Caixeiro invocou a função Vendas.
Pós-condições	

1		Foi impresso um Ticket de Venda completo, cujos campos refletem corretamente as regras de negócio aplicáveis.
2		O Ticket de Venda reflete corretamente a Venda efetuada, conforme registrado no Merci.
3		O saldo do Caixa foi atualizado com o Total a pagar, conforme a regra de negócio aplicável.
4		Após cada Venda, o Merci continua no MODO DE VENDAS.
Atividades		
Número	Agente	Descrição
1	Merci	Exibe a Tela de Venda.
2	Caixeiro	Invoca a operação Nova.
3	Merci	Gera o Número da Operação de venda, conforme a regra de negócio aplicável, e coloca a Tela de Venda no estado ABERTA.
4	Caixeiro	Executa-se o subfluxo Inserção de Novo Item de Venda.
5	Caixeiro	Informa a Forma de pagamento e invoca a operação Encerrar.
6	Merci	Coloca a Tela de Venda no estado FECHADA.
7	Merci	Executa-se o subfluxo Impressão do Ticket de Venda.
8	Merci	Atualiza o valor do Caixa e notifica o Sistema Financeiro, conforme as regras de negócio aplicáveis.

Tabela 2.13 – Documentação de Caso de Uso

Requisitos não-funcionais

Os requisitos não-funcionais globais, aplicando-se ao produto como um todo. Um requisito não-funcional pode ser específico de um caso de uso; por exemplo, a duração máxima de uma transação, ou um requisito de usabilidade. A figura 2.4 lista alguns tipos de requisitos não-funcionais.

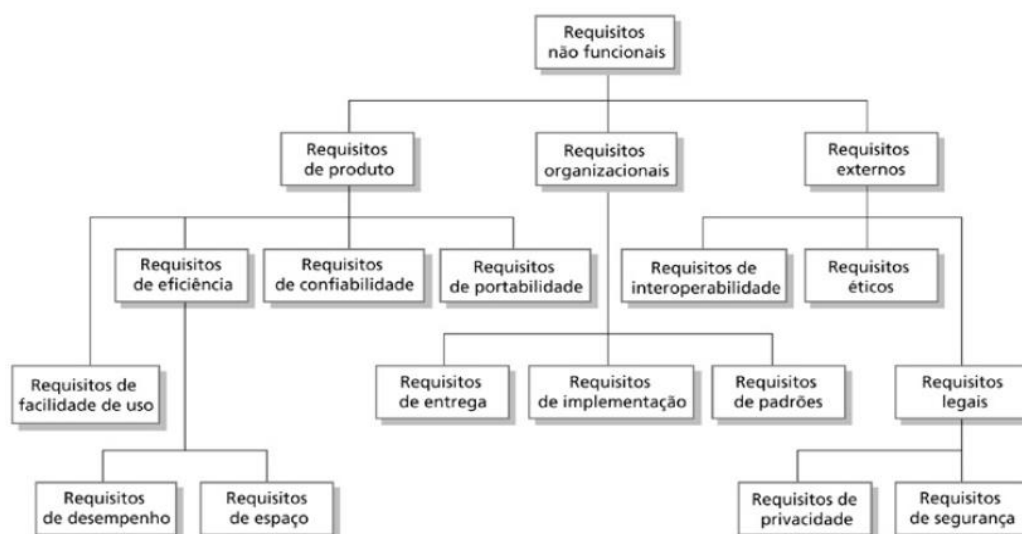


Figura 2.4 – Tipos de requisitos não-funcionais

Na modelagem do problema o tipo principal de requisito não-funcional é formado pelos Requisitos de qualidade, correspondentes a aspecto de qualidade do software descritos pela nota ISO-9126. Os requisitos de qualidade são divididos em seis características principais: Funcionalidade, Confiabilidade, Usabilidade, Manutenibilidade, Portabilidade e Desempenho. Requisitos de desempenho são requisitos numéricos a que o produto ou sistema maior deve obedecer, geralmente exercem na arquitetura. Requisitos de desempenho podem ser estáticos ou dinâmicos.

Número	Nome	Aplicação	Descrição
1	Uso de senhas	Gestão de Usuários	O acesso dos usuários às funções será restrito por senhas, conforme o respectivo grupo.
2	Proteção contra perda de dados	Gestão de Usuários, Gestão Manual de Estoques, Gestão de Mercadorias, Gestão de Fornecedores, Gestão de Pedidos de Compras, Operação de Venda, Emissão de Nota Fiscal	Nas interfaces de usuário, toda operação que possa causar perda de dados necessitará de confirmação.
3	Tempo de resposta para pesquisas no banco de dados	Gestão de Usuários, Gestão Manual de Estoques, Gestão de Mercadorias, Gestão de Fornecedores, Gestão de Pedidos de Compras, Operação de Venda, Abertura do Caixa, Fechamento do Caixa, Emissão de Relatórios	O tempo para realização de qualquer operação de pesquisa de objetos persistentes não pode ser maior do que 10 segundos.
4	Expansibilidade dos pontos de acesso	Operação de Venda	O produto deve ser expansível para mais de um terminal de caixa.

Tabela 2.14 – Exemplos de requisitos de qualidade

Número	Nome	Característica	Sub-característica	Motivação	Validação
1	Uso de senhas	Funcionalidade	Segurança de acesso	Garantia de que as funções sejam executadas por quem tenha as permissões adequadas.	Teste automatizado do mecanismo de acesso que vier a ser adotado.
2	Proteção contra perda de dados	Usabilidade	Operacionalidade	Produtividade no uso.	Emissão da mensagem de confirmação em tais casos, verificada pelo menos por teste manual.
3	Tempo de resposta para pesquisas no banco de dados	Eficiência	Tempo de resposta	Agilidade no uso do produto.	Medição do tempo para recuperação desses itens, em operações de pesquisa.

Tabela 2.15 – Propriedades dos requisitos de qualidade

Requisitos de dados persistentes são aqueles cujo valor subsiste após cada execução do produto, e estão geralmente armazenados em memória secundária, por exemplo, sistemas de arquivos ou banco de dados relacional.

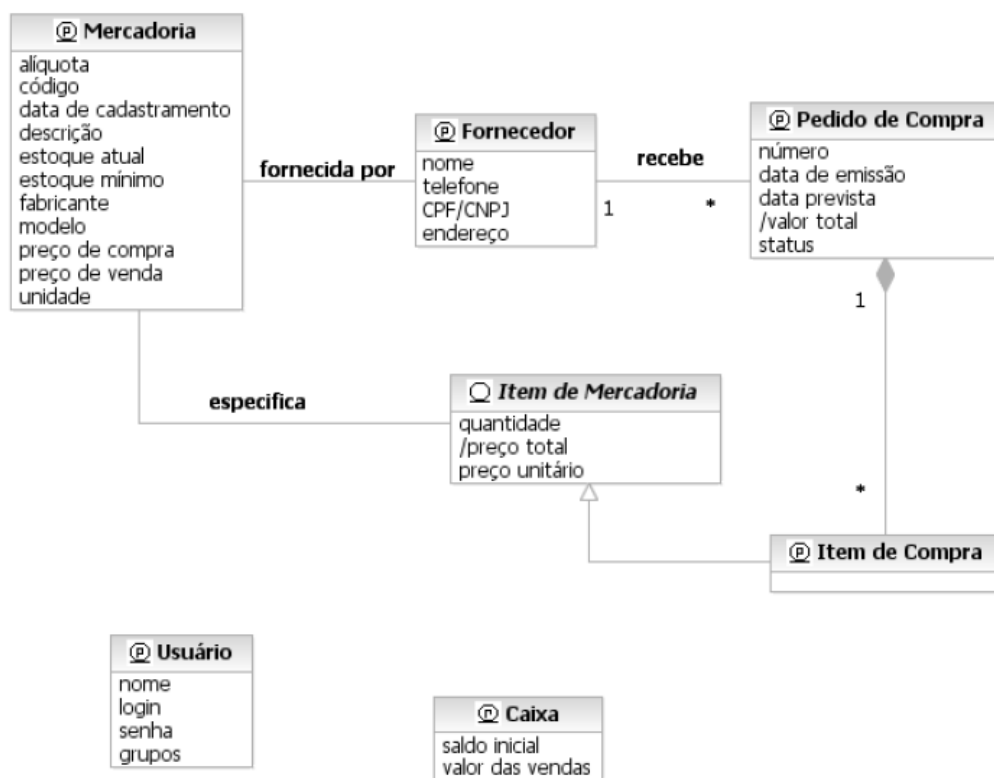


Figura 2.5 – Diagrama de dados persistentes

Os requisitos técnicos representam restrições oriundas do cliente ou de autoridades externas que restringem o espaço para desenvolvimento da solução. São requisitos que devem ser incluídos quando decorrem na imposição do cliente ou de terceiros, como padrões e restrições legais.

Número	Nome	Aplicação	Descrição
1	Implementação J2SE	Global	O produto deverá ser implementado em Java, baseado na plataforma J2SE.
2	Modelagem UML 2.0	Global	Os modelos usados deverão seguir o padrão UML 2.0.

Tabela 2.16 – Aplicação e restrição

Número	Nome	Tipo	Sub-tipo	Motivação
1	Implementação J2SE	Produto	Implementação	Compatibilidade com o padrão do produto atual.
2	Modelagem UML 2.0	Produto	Modelagem	Compatibilidade com o estado da arte.

Tabela 2.17 – Propriedades