UNIOESTE - Universidade Estadual do Oeste do Paraná
CCET - Centro de Ciências Exatas e Tecnológicas
Colegiado de Informática
Curso de Bacharelado em Informática

PROJETO DA DISCIPLINA

PES II – Processo De Engenharia de Software II

CASCAVEL
2009

Alessandro Rodrigo Franco
Fernando Luiz Grando
Fernando Martins

SISTEMA - FARMÁCIA

Parte 01
Especificação de Requisitos e Modelagem Orientada a Objeto

Professor: *Victor Francisco Araya Santander*

CASCAVEL
2009

ÍNDICE

1 – MOTIVAÇÃO

Tendo em vista que atualmente o computador é uma ferramenta de trabalho quase que indispensável, a informatização de sistemas se faz necessária em vários ramos da atividade humana. Torna-se indispensável para qualquer organização o uso de um sistema informatizado para o controle de seus dados.

Esse projeto tem como objetivo informatizar o sistema de uma Farmácia, criando ferramentas que facilitem o seu trabalho diário, de modo a aprimorar o seu desempenho de maneira geral, tal como a diminuição de gastos e tempo, mas também permitir a possibilidade de um controle mais eficiente de estoques, informações, serviços etc.

Esperamos cumprir nossas metas e prazos ao longo do ano como descrito na documentação deste trabalho.

2 – INTRODUÇÃO

A empresa escolhida pelo grupo para ter como base do projeto de prática de Processo de Engenharia de Software II foi a Farmácia São Lucas, localizada em Capanema – PR, e tem como proprietário o Sr. Valdir Antonio Stokmann.

A Farmácia ainda não possui nenhum sistema de informação implantado. Toda a atividade de operação como venda, controle de estoque, arquivo de produtos, informação de funcionários, entre outras, são realizadas manualmente. Sendo assim, propõe-se a implantação de um sistema que otimize todas as funções realizadas dentro da empresa.

Fomos até a Farmácia do Sr. Valdir e através de uma longa entrevista chegamos às seguintes conclusões:

- O sistema precisa gerenciar os cadastros de clientes, de produtos e de funcionários da empresa.

- A Farmácia necessita ter um controle de estoque eficiente.

- Quanto ao pagamento realizado no caixa, o sistema terá que possuir opções para pagamento em dinheiro, e em cheque. Este último deverá ser autorizado apenas após verificação de pendências no estabelecimento.

- O sistema precisa ter boa usabilidade, para o fácil aprendizado de novos usuários.

- A segurança e a confiabilidade dos dados são requisitos indispensáveis para esse projeto. Visto isso, o sistema possuirá uma hierarquia de privilégios entre os usuários que forem acessar os dados cadastrados na Farmácia.

- O desempenho é de grande importância ao sistema, pois geralmente em uma Farmácia os clientes precisam de atendimento rápido e eficaz.

Com base nas informações adquiridas na entrevista, elaboramos esse documento que possui a descrição bem detalhada dos requisitos do sistema. Nosso projeto será modelado com base nos seguintes estudos: Modelagem Organizacional utilizando a Técnica I* (diagramas SD e SR), a Modelagem NFR Framework (grafo SIG), Diagrama de Casos de Uso e Diagrama de Classes.

3 – CRONOGRAMA

ATO_01 – *Início: 02/03/09 – Término: 09/03/09*

Discussão sobre o sistema a ser implementado.

ATO_02 – *Início: 10/03/09 – Término: 17/03/09*

Obtenção do conhecimento sobre o que vai ser desenvolvido.

ATO_03 – *Início: 18/03/09 – Término: 25/03/09*

Reuniões para escolha da metodologia de engenharia de software adotada.

ATO_04 – *Início: 26/03/09 – Término: 30/04/09*

Elicitação e Análise de Requisitos: Realizado junto à empresa através de questionário e entrevista profunda e precisa, identificando os fatos que compõem os requisitos do sistema, de forma a prover o mais correto e mais completo entendimento do que é demandado o software.

ATO_05 – *Início: 01/05/09 – Término: 15/05/09*

Debate a respeito da linguagem de programação a ser utilizada e as possíveis tecnologias adicionais.

ATO_06 – *Início: 16/05/09 – Término: 01/06/09*

Protótipo Não-Funcional: Interface gráfica principal.

ATO_07 – *Início: 02/06/09 – Término: 16/06/09*

Modelagem do banco de dados (Modelo Lógico e Conceitual).

ATO_08 – *Início: 17/06/09 – Término: 19/07/09*

Projeto de Software: Transformação dos resultados da Análise de Requisitos em um conjunto de documentos capazes de serem interpretados diretamente pelo programador.

ATO_09 – *Início: 01/07/09 – Término: 31/08/09*

Implementação: Elaboração e preparação dos módulos necessários para a execução do sistema.

ATO_09.1 – *Início: 01/07/09 – Término: 15/07/09*
Implementação do MVC (**Model-View-Controller**) para os funcionários.
Implementação do MVC (**Model-View-Controller**) para os clientes.

ATO_09.2 – *Início: 16/07/09 – Término: 31/07/09*
Implementação do MVC (**Model-View-Controller**) para os produtos.
Implementação do MVC (**Model-View-Controller**) para o estoque.

ATO_09.3 – *Início: 01/08/09 – Término: 15/08/09*
Implementação do MVC (**Model-View-Controller**) para as vendas.
Implementação do MVC (**Model-View-Controller**) para os pagamentos.

ATO_09.4 – *Início: 16/08/09 – Término: 31/08/09*
Revisão de todos os módulos implementados.

ATO_10 – *Início: 01/09/09 – Término: 15/09/09*
Integração: Os componentes do software separados são combinados em um todo.

ATO_11 – *Início: 16/09/09 – Término: 15/11/09*
Teste e Depuração: O software é investigado a fim de fornecer informações sobre sua qualidade em relação ao contexto em que ele deve operar. Isso inclui o processo de utilizar o produto para encontrar seus defeitos.

ATO_12 – *Início: 01/12/09 – Término: 15/12/09*
Pré-Implantação: Verificação dos equipamentos disponíveis (desktops, licenças, banco de dados, periféricos e etc.) e adequação ao requerido pelo software. Realização de testes para verificação da integridade dos dados e com os funcionários a fim de verificar se tudo está correto para a implantação.

ATO_13 – *Início: 16/12/09 – Término: 31/12/09*

Treinamento: Familiarização dos usuários com o software. Possível criação de grupos de conhecimento, ou seja, treinamento em conjunto dos funcionários de um mesmo departamento. Os programadores serão os disseminadores do treinamento para os grupos.

ATO_14 – *Início: 01/01/10 – Término: 15/01/10*

Implantação: Passagem do software para a produção. Realização das alterações na rotina (caso sejam necessárias) dos processos e importação dos dados mais atuais, deixando o software pronto para o uso em ambiente de produção.

ATO_15 – *Início: 16/01/10 – Término: 31/01/10*

Pós-implantação: Acompanhamento junto aos usuários para verificação de problemas não eliminados nas fases anteriores e ajustar pontos que não ficaram satisfatórios.

4 – METODOLOGIA

A metodologia definida por nossa equipe foi o modelo em Cascata. O modelo em cascata é um modelo de desenvolvimento de software seqüencial no qual o desenvolvimento é visto como um fluir constante para frente (como uma cascata) através das fases de análise de requisitos, projeto, implementação, integração, testes (validação) e manutenção de software. Portanto o modelo em cascata move-se para a próxima fase somente quando a fase anterior esta completa e perfeita. Desenvolvimento de fases no modelo em cascata são discretas, e não há pulo para frente, para trás ou sobreposição entre elas.

Nossa escolha se fundamenta no motivo que nosso cliente se situa em outra cidade e não teremos muitos contatos ao longo do processo de desenvolvimento do sistema. Outra forte razão é que o sistema não é complexo e possui poucos requisitos funcionais, os quais estão bem definidos, não necessitando a adoção de outra metodologia. Acreditamos também que, se houverem possíveis alterações de requisitos, essas serão mínimas, não resultando em altos custos e grande perca de tempo. Não mediremos esforços para que, como uma forma de avaliação do projeto, consigamos levá-lo até o próprio cliente, mesmo sabendo que esse método de avaliação não será freqüente, tendo em vista a dificuldade de comunicação entre as partes, conforme foi citado anteriormente. Caso apareçam inconvenientes em determinada fase do projeto, a manutenção da mesma será efetuada e, quando concluída, será apresentada novamente ao cliente, como forma de garantir a exatidão ao que foi solicitado e para que possamos seguir adiante com o processo de desenvolvimento.

Visto a baixa experiência do grupo de desenvolvimento, caso a metodologia acima citada não se encaixe de maneira eficiente durante o projeto, pretendemos adaptá-la o quanto necessário. Para isso, poderemos modificar a idéia para um desenvolvimento mais ágil, criando iterações adicionais que não estavam no planejamento inicial.

5 – ANÁLISE DE REQUISITOS

5.1 – Requisitos Funcionais

Os Requisitos Funcionais são aqueles que descrevem o comportamento do sistema, suas ações para cada entrada, ou seja, é aquilo que descreve o que tem que ser feito pelo sistema. Devem ser o cérebro do projeto, já que descrevem as funcionalidades que o sistema deve dispor. Abaixo estão listados e descritos os requisitos funcionais do sistema, referenciados da seguinte maneira:

[RF-'id'] 'Nome do Requisito Funcional'

'Descrição'

[RF-01] Controle de Funcionários

O sistema deve permitir o cadastro, a edição, a remoção e a busca de funcionários.

[RF-02] Cadastrar Funcionário

O sistema deve permitir o cadastro de funcionário a partir de seus dados pessoais: nome, RG, CPF, cargo, salário, data de nascimento, data de admissão, nome de usuário e senha.

[RF-03] Remover Funcionário

O sistema deve excluir o registro do funcionário especificado.

[RF-04] Editar Funcionário

O sistema deve permitir alterações nos dados do funcionário escolhido.

[RF-05] Buscar Funcionário

O sistema deve permitir a busca de um funcionário através de seu nome ou CPF.

[RF-06] Verificação de Cadastros

O sistema deve verificar se um registro de cadastro já existe no banco de dados.

[RF-07] Controle de Produtos

O sistema deve permitir o cadastro, a edição, a remoção e a busca de produtos.

[RF-08] Cadastrar Produto

O sistema deve permitir ao ator incluir produtos farmacêuticos no banco de dados do sistema, a partir dos seguintes dados: nome do produto, preço de custo, preço de venda, lote, fornecedor, quantidade, data de fabricação e data de validade.

[RF-09] Remover Produto

O sistema deve permitir a exclusão de um produto do banco de dados do sistema.

[RF-10] Editar Produto

O sistema deve permitir alteração nos dados do produto.

[RF-11] Buscar Produto

O sistema deve permitir consulta do produto pelo nome.

[RF-12] Controle de Clientes

O sistema deve permitir o cadastro, a edição, a remoção e a busca de clientes.

[RF-13] Cadastrar Cliente

O sistema deve permitir ao ator incluir cadastros de clientes no banco de dados do sistema, a partir dos seguintes dados: nome do cliente, RG, CPF, sexo, saldo gasto, saldo devedor e data de nascimento.

[RF-14] Remover Cliente

O sistema deve permitir a exclusão de um cliente do banco de dados do sistema.

[RF-15] Editar Cliente

O sistema deve permitir alteração nos dados do cliente.

[RF-16] Buscar Cliente

O sistema deve permitir a consulta do cliente pelo nome.

[RF-17] Controle de Pagamentos

O sistema deve aceitar pagamento em cheque ou em dinheiro.

[RF-18] Pagamento em Dinheiro

O sistema deve aceitar pagamento em dinheiro somente à vista.

[RF-19] Pagamento em Cheque

O sistema deve permitir que clientes comprem a prazo utilizando cheque, desde que tenham se cadastrado previamente na Farmácia.

[RF-20] Verificar Situação do Cliente

O sistema deve realizar uma verificação para achar possíveis pendências do cliente na Farmácia.

[RF-21] Realizar Vendas

O sistema deve permitir que um usuário escolha os produtos e quantidade dos mesmos e assim realize vendas.

[RF-22] Atualizar Estoque

O sistema deve controlar a saída de produtos que são vendidos, a fim de notificar uma possível falta dos mesmos.

5.2 – Requisitos Não-Funcionais

Os Requisitos Não-Funcionais são aqueles que expressam como deve ser feito. Em geral se relacionam com padrões de qualidade como: confiabilidade, performance, robustez, etc. São muito importantes, pois definem se o sistema será eficiente para a tarefa que se propõe a fazer ou não. Um sistema ineficiente certamente não será usado. Neles também são apresentados restrições e especificações de uso para os requisitos funcionais. Segue abaixo a relação e descrição dos requisitos não-funcionais do sistema, conforme o padrão a seguir:

[RNF-'id'] 'Nome do Requisito Não-Funcional'

Operacionalização: 'Nome da Operacionalização'

'Descrição da Operacionalização'

5.2.1 - Requisitos do Processo

Requisitos referentes ao modo de desenvolvimento

5.2.1.1 - Quanto a Implementação

Tecnologia JAVA: O sistema deverá ser implementado na linguagem JAVA.

5.2.2 - Requisitos do Produto

Requisitos referentes às características e qualidades.

5.2.2.1 - Quanto à Segurança dos Dados:

[RNF-01] Confidencialidade

Operacionalização: Política de Login e Senha. [Aceita]

O sistema deverá garantir a confidencialidade dos dados a partir da implementação de um sistema de login e senha, em que cada funcionário terá acesso às funcionalidades e aos dados conforme seu cargo.

➔ Prejudica a Usabilidade.

Operacionalização: Uso de Criptografia. [Recusada]

O sistema utilizará um esquema de criptografia para a proteção dos dados, a fim de garantir o acesso exclusivo a quem tenha permissão apropriada.

➔ Prejudica a Performance.

[RNF-02] Integridade

Operacionalização: Política de Backup. [Aceita]

O sistema deverá garantir a integridade dos dados a partir de uma política de backup.

➔ Aumenta o Custo.

5.2.2.2 - Quanto à Usabilidade:

[RNF-01] Usabilidade

Operacionalização: Interface Amigável. [Aceita]

O sistema possuirá uma interface gráfica amigável, com informações e funcionalidades objetivas, facilitando a vida do usuário.

➔ Prejudica a Performance.

Operacionalização: Teclas de Atalho. [Aceita]

O sistema possuirá teclas de atalho para acesso mais rápido aos dados e as funcionalidades mais utilizadas.

Operacionalização: Livre Acesso às Informações. [Recusada]

O sistema disponibilizará acesso às informações sem qualquer tipo de restrições hierárquicas para as mesmas.

➔ Prejudica a Confidencialidade.

[RNF-02] Ajuda

Operacionalização: Manual de Utilização. [Aceita]

O sistema possuirá um manual de utilização do software, seja ele impresso, ou em um sub-menu do próprio software.

Operacionalização: Ajuda Online. [Recusada]

O sistema disponibilizará um site para suprir qualquer dúvida em relação à usabilidade do software.

➔ Aumenta o Custo.

5.2.2.3 - Quanto à Performance:

[RNF-01] Performance

Operacionalização: Hardware Recomendado. [Indecisa]

A rapidez de execução do sistema dependerá dos recursos do computador utilizado. Recomenda-se:

* Processador de 1000 megahertz ou superior;
* 512 megabytes de memória RAM ou superior;

* Disco rígido de 20 ou mais gigabytes;

* Placa de vídeo Super VGA (800 × 600) ou de maior resolução, e monitor;

* Unidade de CD-ROM ou DVD;

* Teclado e mouse.

- ➔ Aumenta o Custo.
- ➔ Melhora a Integridade.

5.2.2.4 - Quanto à Portabilidade:

[RNF-01] Portabilidade

Operacionalização: Máquina Virtual JAVA. [Aceita]

O sistema deverá executar em qualquer plataforma, desde que exista uma Máquina Virtual JAVA para a plataforma pretendida.

- ➔ Diminui o Custo.

5.2.2.5 – Quanto ao Custo:

[RNF-01] Custo

Operacionalização: Tecnologia JAVA. [Aceita]

O sistema será desenvolvido utilizando ferramentas gratuitas, o que resultará numa economia significativa, não sendo necessárias aquisições de softwares para a implementação.

- ➔ Favorece a Portabilidade.
- ➔ Diminui a Performance.

6 – MODELAGEM ORGANIZACIONAL USANDO A TÉCNICA I*

A notação I* foi desenvolvida como um método de modelagem orientado a agentes, centrado na característica intencional dos mesmos. A técnica de modelagem I* foca nos relacionamentos entre os atores e suas dependências. Os atores são vistos como tendo propriedades intencionais, tais como objetivos, opiniões, habilidades e compromissos. Os atores dependem uns dos outros para cumprir objetivos, realizar tarefas e fornecer recursos. Através da cooperação de outros, um ator pode alcançar objetivos que serão difíceis de serem alcançados sozinho. A técnica I* consiste de dois modelos: o modelo de Dependências Estratégicas (SD - Strategic Dependency), e o modelo de Razões Estratégicas (SR – Strategic Rationale).

6.1 - Modelo de Dependências Estratégicas (SD)

O modelo de Dependências Estratégicas é uma rede de relacionamentos de dependência entre atores. O modelo SD foca na captura na estrutura intencional de um processo, ou seja, na captura das motivações e intenções por trás das atividades e fluxos em um processo.

O modelo consiste numa série de nós e links. Cada nó representa um ator. Um ator é uma entidade ativa que realiza ações para atingir determinados objetivos. Cada link entre dois atores indica que um ator depende do outro para, de alguma forma, cumprir algum objetivo. O ator que depende de outro é chamado de ‘depender’ e o ator responsável por cumprir a dependência é chamado de ‘dependee’. O objetivo da dependência é denominado ‘dependum’.

Através da dependência, o ‘depender’ é capaz de alcançar objetivos que não seria capaz de alcançar sozinho. Existem quatro tipos de dependências: dependência de objetivo (goal), dependência de tarefa (task), dependência de recurso (resource) e dependência de objetivo soft (soft goal).

Descrição Textual:

Em nosso modelo, focamos nossa atenção em quatro atores:

- Gerente, que necessita de acesso seguro, e que tem prioridade exclusiva no cadastro de funcionários e ainda herda as prioridades do Caixa e do Atendente.
- Atendente, que necessita de usabilidade do sistema para um atendimento ágil aos clientes, podendo adicionar produtos e criar uma lista de compras do cliente.
- Caixa, que controlará os cadastros de clientes, as realizações de vendas, os pagamentos e os

cheques.

- Sistema terá o objetivo de viabilizar as operações acima.

➔ **Figura 1.**

6.2 - Modelo de Razões Estratégicas (SR)

O modelo de dependência estratégica mostra somente os relacionamentos externos entre os atores. No modelo de razão estratégica, temos uma representação e uma racionalização mais explícita sobre os interesses dos atores, e de como estes interesses podem ser atendidos ou afetados pelos diversos sistemas. No modelo SR temos uma visão interna dos atores, proporcionando um nível de detalhamento maior do modelo. Os elementos intencionais (objetivos, tarefas, recursos e objetivos soft) aparecem no modelo SR como elementos internos ligados por relacionamentos meio-fim e decomposição de tarefas. Estes relacionamentos proporcionam uma representação explicita das razões por trás das dependências entre os atores e quais são alternativas por trás dos processos.

No modelo SR são introduzidas duas classes de ligações: ligações meio-fim e decomposição de tarefas. Uma ligação meio-fim indica um relacionamento entre um fim e um meio para alcançar este fim. O meio é geralmente expresso na forma de uma tarefa, que incorpora a noção de como fazer alguma coisa. Uma ligação de decomposição de tarefa é utilizada para descreve os componentes de uma tarefa.

Descrição Textual:

O ator expandido foi o Sistema, pois ele é a parte central do esquema, e provê todas as funcionalidades para os outros atores. O principal objetivo do sistema consiste em controlar os cadastros de uma Farmácia. Esta tarefa será composta de várias partes:

- Rapidez nos cadastros realizados pelo funcionário para um maior aproveitamento do tempo.
- Interface agradável, aumentando a usabilidade por parte dos funcionários.
- Segurança para o gerente, que possui funcionalidades especiais no sistema.
- Cadastro, exclusão, consulta e alteração de dados do banco de dados. Todas as opções passaram por uma verificação antes para que não haja dados duplicados no sistema, e para que dados sejam excluídos erroneamente.
- Controle de pagamento com opções de pagamento em dinheiro e cheque.

➔ **Figura 2.**

7 – MODELAGEM NFR FRAMEWORK

O modelo foi proposto por (Chung et al. 2000), Universidade de Toronto.

Ele fornece uma representação sistemática e global dos requisitos não-funcionais, tratando-os de forma explícita como metas a serem atingidas.

- Baseado na análise qualitativa dos requisitos não-funcionais.

7.1 – SIG

O SIG (Softgoal Interdependency Graph) representa o inter-relacionamento entre os requisitos não-funcionais. O uso do NFR Framework é feito através da construção incremental e interativa de grafos SIG. Eles descrevem a interdependência entre softgoals e como eles são decompostos.

- Inspirado nas estruturas de árvore e/ou para solução de problemas.

Descrição Textual:

Em nosso grafo SIG, consideramos os seguintes requisitos não-funcionais e suas respectivas operacionalizações, necessários para a empresa obter um padrão de qualidade considerável:

- Usabilidade – Interface Amigável, Teclas de Atalho, Livre Acesso às Informações, Ajuda (Manual ou Online);
- Performance – Hardware Recomendado;
- Custo – Tecnologia JAVA;
- Portabilidade – Máquina Virtual JAVA;
- Segurança – Integridade: Política de Backup; Confidencialidade: Política de Login e Senha, Uso de Criptografia;

Para maiores detalhes, consulte a documentação da análise de requisitos em "*5.2 - Requisitos Não-Funcionais"*, localizada na página 11 desse mesmo documento.

➔ **Figura 3.**

8 – DIAGRAMA DE CASOS DE USO

O Diagrama de Casos de Uso descreve a funcionalidade proposta para o sistema. Segundo Ivar Jacobson [Object-Oriented Software Engineering: A Use Case Driven Approach, 1992], podemos dizer que um Caso de Uso é um "documento narrativo que descreve a seqüência de eventos de um ator que usa um sistema para completar um processo". Um Caso de Uso representa uma unidade discreta da interação entre um usuário (humano ou máquina) e o sistema. Um Caso de Uso é uma unidade de um trabalho significante. Cada Caso de Uso tem uma descrição da funcionalidade que irá ser construída no sistema proposto. Um Caso de Uso pode "usar" outra funcionalidade de Caso de Uso ou "estender" outro Caso de Uso com seu próprio comportamento.

Casos de Uso são tipicamente relacionados a "atores". Um ator é um humano ou entidade máquina que interage com o sistema para executar um significante trabalho.

➔ **Figura 4.**

8.1 – Descrições Textuais dos Casos de Uso

Caso de Uso 01: Controle de Funcionários

Objetivo: *Controlar os cadastros dos funcionários.*
Ator: *Gerente.*
Pré-condição: *O Ator precisa estar logado e ter privilégios de gerente.*
Cenário Primário:
1. Gerente clica na aba "Controle de Funcionários".
2. Na aba "Controle de Funcionários" o Gerente escolhe entre "Cadastrar Funcionário", "Editar Funcionário", "Remover Funcionário" ou "Buscar Funcionário".
3. Gerente faz as alterações que deseja.
4. Gerente salva o sistema com os novos dados.
Extensões:
2a. Adicionar funcionário. Caso de uso "Cadastrar funcionário" <<extend>>.
2b. Alterar funcionário. Caso de uso "Editar funcionário" <<extend>>.
2c. Consultar funcionário. Caso de uso "Buscar funcionário" <<extend>>.
2d. Excluir funcionário. Caso de uso "Remover funcionário" <<extend>>.
Pós-condição: *Lista de funcionários atualizada.*

Caso de Uso 02: Cadastrar Funcionário

Objetivo: *Adicionar novo funcionário.*
Ator: *Gerente.*
Pré-condição: *O Ator precisa estar logado no sistema e ter privilégios de gerente.*
Cenário Primário:
1. Gerente entra com os dados do funcionário. (Caso de Uso "Verificação de Cadastros" <<include>>).
2. Gerente salva os dados do novo funcionário.
Pós-condição: *Lista de funcionários atualizada.*

Caso de Uso 03: Remover Funcionário

Objetivo: *Excluir dados do funcionário.*
Ator: *Gerente.*
Pré-condição: *O Ator precisa estar logado no sistema e ter privilégios de gerente.*
Cenário Primário:
1. Gerente informa o nome do funcionário a ser excluído.
2. Sistema localiza o funcionário (Caso de Uso "Buscar funcionário" <<include>>).
3. Funcionário é excluído e sistema é salvo com a nova configuração.
Extensões:
1a. Cadastro inexistente.
Pós-condição: *Lista de funcionários atualizada.*

Caso de Uso 04: Editar Funcionário

Objetivo: *Alterar dados de um funcionário.*
Ator: *Gerente.*
Pré-condição: *O Ator precisa estar logado no sistema e ter privilégios de gerente.*
Cenário Primário:
1. Gerente informa o nome do funcionário a ser alterado.
2. Sistema localiza o funcionário (Caso de Uso "Buscar funcionário" <<include>>).
3. Funcionário é editado e o sistema é salvo com a nova configuração.
Extensões:
1a. Cadastro inexistente.
Pós-condição: *Dados do funcionário atualizados.*

Caso de Uso 05: Buscar Funcionário

Objetivo: *Consultar dados de um funcionário.*
Ator: *Gerente.*
Pré-condição: *O Ator precisa estar logado no sistema e ter privilégios de gerente.*
Cenário Primário:
1. Gerente informa o nome do funcionário a ser consultado.
2. Sistema localiza funcionário e exibe seus dados na tela.
3. A consulta é finalizada.
Extensões:
1a. Cadastro inexistente.

Caso de Uso 06: Verificação de Cadastros

Objetivo: *Verificar se registro já existe no banco de dados.*
Ator: *Gerente, Caixa, Atendente.*
Pré-condição: *O Ator precisa estar logado no sistema.*
Cenário Primário:
1. O sistema verifica se o registro já está cadastrado.
2. Sistema envia a resposta para o ator.

Caso de Uso 07: Controle de Produtos

Objetivo: *Controlar o estoque de produtos.*
Atores: *Atendente, Gerente.*

***Pré-condição**: O Ator precisa estar logado.*

Cenário Primário:

1. O Ator clica na aba "Controle de Produtos"

2. Na aba "Controle de Produtos" aparecerão as opções: "Cadastrar Produto", "Editar Produto", "Remover Produto" e "Buscar Produto".

3. O ator faz as alterações que deseja.

4. O ator salva o sistema com os novos dados.

Extensões:

2a. Cadastrar produto. Caso de Uso "Cadastrar Produto" <<extend>>

2b. Editar produto. Caso de Uso "Editar Produto" <<extend>>

2c. Buscar produto. Caso de uso "Buscar Produto" <<extend>>

2d. Remover produto. Caso de uso "Remover Produto" <<extend>>

***Pós-Condição:** Lista de Produtos atualizada.*

Caso de Uso 08: Cadastrar Produto

***Objetivo:** Adicionar novo produto.*

***Ator:** Gerente, Atendente.*

***Pré-condição:** O Ator precisa estar logado no sistema.*

Cenário Primário:

1. O ator entra com os dados do produto. (Caso de uso "Verificação de Cadastros" <<include>>).

2. O ator salva os dados do produto.

***Pós-condição:** Lista de produtos atualizada.*

Caso de Uso 09: Remover Produto

***Objetivo:** Excluir um produto do banco de dados do sistema.*

***Ator:** Gerente, Atendente.*

***Pré-condição:** O Ator precisa estar logado no sistema.*

Cenário Primário:

1. O ator informa o nome do produto a ser excluído.

2. Sistema localiza o produto (Caso de Uso "Buscar produto" <<include>>).

3. Produto é excluído e sistema é salvo com a nova configuração.

Extensões:

1a. Cadastro inexistente.

***Pós-condição:** Lista de produtos atualizada.*

Caso de Uso 10: Editar Produto

***Objetivo:** Alterar dados do produto.*

***Ator:** Gerente, Atendente, Caixa.*

***Pré-condição:** O Ator precisa estar logado no sistema.*

Cenário Primário:

1. O ator informa o nome do produto a ser alterado.

2. Sistema localiza o produto (Caso de Uso "Buscar produto" <<include>>).

3. Produto é editado e sistema é salvo com a nova configuração.

Extensões:

1a. Cadastro inexistente.

***Pós-condição:** Dados do produto atualizados.*

Caso de Uso 11: Buscar Produto

Objetivo: *Consultar dados de um produto.*
Ator: *Gerente, Atendente.*
Pré-condição: *O Ator precisa estar logado no sistema.*
Cenário Primário:
1. O ator informa o nome do produto a ser consultado.
2. Sistema localiza produto e exibe seus dados na tela.
3. A consulta é finalizada.
Extensões:
1a. Cadastro inexistente.

Caso de Uso 12: Controle de Clientes

Objetivo: *Controlar os cadastros dos clientes.*
Ator: *Gerente, Caixa.*
Pré-condição: *O Ator precisa estar logado.*
Cenário Primário:
1. Ator clica na aba "Controle de Clientes".
2. Na aba "Controle de Clientes" o Gerente escolhe entre "Cadastrar Cliente", "Editar Cliente", "Remover Cliente" ou "Buscar Cliente".
3. Ator faz as alterações que deseja.
4. Ator salva o sistema com os novos dados.
Extensões:
2a. Adicionar cliente. Caso de uso "Cadastrar Cliente" <<extend>>.
2b. Alterar cliente. Caso de uso "Editar Cliente" <<extend>>.
2c. Consultar cliente. Caso de uso "Buscar Cliente" <<extend>>.
2d. Excluir cliente. Caso de uso "Remover Cliente" <<extend>>.
Pós-condição: *Lista de clientes atualizada.*

Caso de Uso 13: Cadastrar Cliente

Objetivo: *Adicionar novo cliente.*
Ator: *Gerente, Caixa.*
Pré-condição: *O Ator precisa estar logado no sistema.*
Cenário Primário:
1. Ator entra com os dados do cliente. (Caso de uso "Verificação de Cadastro" <<include>>).
2. Gerente salva os dados do novo cliente.
Pós-condição: *Lista de clientes atualizada.*

Caso de Uso 14: Remover Cliente

Objetivo: *Excluir dados do cliente.*
Ator: *Gerente, Caixa.*
Pré-condição: *O Ator precisa estar logado no sistema.*
Cenário Primário:
1. Ator informa o nome do cliente a ser excluído.
2. Sistema localiza o cliente (Caso de Uso "Buscar cliente" <<include>>).
3. Cliente é excluído e sistema é salvo com a nova configuração.

Extensões:
1a. Cadastro inexistente.
Pós-condição: *Lista de clientes atualizada.*

Caso de Uso 15: Editar Cliente

Objetivo: *Alterar dados de um cliente.*
Ator: *Gerente, Caixa.*
Pré-condição: *O Ator precisa estar logado no sistema.*
Cenário Primário:
1. Ator informa o nome do cliente a ser alterado.
2. Sistema localiza o cliente (Caso de Uso "Buscar cliente" <<include>>).
3. Cliente é editado e sistema é salvo com a nova configuração.
Extensões:
1a. Cadastro inexistente.
Pós-condição: *Dados do cliente atualizados.*

Caso de Uso 16: Buscar Cliente

Objetivo: *Consultar dados de um cliente.*
Ator: *Gerente, Caixa.*
Pré-condição: *O Ator precisa estar logado no sistema.*
Cenário Primário:
1. Ator informa o nome do cliente a ser consultado.
2. Sistema localiza funcionário e exibe seus dados na tela.
3. A consulta é finalizada.
Extensões:
1a. Cadastro inexistente.

Caso de Uso 17: Controle de Pagamentos

Objetivo: *Controlar as formas de pagamento.*
Ator: *Caixa, Gerente.*
Pré-condição: *O Ator precisa estar logado.*
Cenário Primário:
1. O Ator clica na aba "Controle de Pagamento".
2. O Ator escolherá entre as seguintes opções: "Cheque" ou "Dinheiro".
Extensões:
2a. Pagamento em cheque. Caso de uso Pagamento em cheque <<extend>>
2b. Pagamento em dinheiro. Caso de uso Pagamento em dinheiro <<extend>>
Pós-condição: *Pagamento efetuado.*

Caso de Uso 18: Pagamento em Dinheiro

Objetivo: *Receber do cliente em dinheiro.*
Ator: *Caixa, Gerente.*
Pré-Condição: *O ator precisa estar logado no sistema.*
Cenário Primário:
1. Ator informa o valor da compra ao sistema.
2. Ator informa valor dado pelo cliente.
3. Sistema informa se há necessidade de troco.

Caso de Uso 19: Pagamento em Cheque

Objetivo: *Receber a compra em cheque.*

Ator: *Caixa, Gerente.*

Pré-Condição: *O ator precisa estar logado no sistema.*

Cenário Primário:

1. Ator informa o valor da compra ao sistema.

2. O ator verifica o cadastro do cliente. (Caso de uso "Verificar situação do cliente" <<include>>).

3. Ator informa valor dado pelo cliente.

4. Sistema informa se há necessidade de troco.

Caso de Uso 20: Verificar Situação do Cliente

Objetivo: *Verificar se cliente tem alguma pendência no estabelecimento.*

Ator: *Caixa, Gerente.*

Pré-Condição: *O ator precisa estar logado no sistema.*

Cenário Primário:

1. Ator verifica se cliente tem cadastro na farmácia (Caso de Uso "Buscar cliente" <<include>>).

2. Ator verifica condição do cliente.

3 Ator finaliza compra.

Extensões:

1a. Caso o cliente não seja cadastrado é feito um novo cadastro. Caso de uso "Cadastrar cliente" <<extend>>.

Caso de Uso 21: Realizar Vendas

Objetivo: *Controlar a saída de produtos vendidos.*

Atores: *Gerente, Caixa.*

Pré-Condição: *O ator precisa estar logado no sistema.*

Cenário Primário:

1. Ator escolhe os produtos.

2. Ator define a quantidade de produtos.

3. O sistema realiza a venda. (Caso de Uso "Controle de Pagamentos" <<include>>).

3. Conclui venda. (Caso de uso "Atualizar estoque" <<include>>).

Cenário Secundário:

2a. Se houver uma quantidade insuficiente de produtos o sistema alertará o ator.

Pós-Condição:

1. *O estoque deverá estar atualizado.*
2. *A lista de vendas deverá estar atualizada.*

Caso de Uso 22: Atualizar Estoque

Objetivo: *Controlar a saída de produtos vendidos.*

Atores: *Gerente, Caixa.*

Pré-Condição: *O ator precisa estar logado no sistema.*

Cenário Primário:

1. É descontado do estoque atual os produtos que foram vendidos.

Pós-Condição: *O estoque deverá estar atualizado.*

9 – DIAGRAMA DE CLASSES

É uma modelagem muito útil para o sistema, pois visa permitir a visualização das classes que irão compor o sistema junto com os respectivos atributos e métodos, bem como mostrar como as classes se relacionam, complementam e transmitem informações entre si.

➔ **Figura 5.**

9.1 – Descrições Textuais das Classes

Classe Funcionario

Essa classe representa o funcionário da Farmácia com os seus atributos do mundo real. Possui os métodos para cadastrar, editar, remover e buscar clientes. O atributo permissao é fundamental para essa classe, através dele sabemos quais operações o funcionário pode executar.

Classe Caixa

É uma especialização da classe Funcionario, adicionando-se os métodos para adicionar, remover e modificar clientes, realizar vendas e os métodos de pagamentos. Um caixa está associado a 0 ou mais clientes(classe Cliente). Um caixa pode fazer 0 ou mais vendas, o que reflete sua relação com a classe Venda.

Classe Gerente

É uma especialização da classe Caixa. Adiciona-se a classe os métodos adicionar, remover e modificar funcionários e produtos. O gerente tem o controle dos produtos, ou seja, ele se relaciona com a classe Produto com cardinalidade de 1 a N.

Classe Cliente

É uma classe onde seus atributos dizem respeito aos dados comuns a todos os indivíduos. Inclui também métodos para consultar saldo e contas do indivíduo. Um cliente efetua 0 ou mais compras, logo o mesmo esta associado à classe Venda. A associação do cliente com a classe Caixa se deve ao fato dO cliente ter que efetuar alguma compra, pois a mesma(compra) é feita atravéz de um caixa. Um cliente tem 0 ou mais cheque o que representa a relação do mesmo com a classe Cheque.

Classe Venda

Classe que contém atributos e métodos referentes à operação de venda. A mesma se relaciona com um ou mais itens de venda(classe ItensVenda), outra relação importante é de um para um com a classe Pagamento. A classe Venda se relaciona a um cliente e a um caixa.

Classe Produto

Esta classe contém atributos e funções referentes aos produtos, tais como codigo, quantidade, tarja, preço, etc. Um produto está associado com a classe ItemVenda com a cardinalidade um para um. Os produtos estão associados com a classe do Gerente, que é a classe encarregada de controlar os produtos.

Classe ItemVenda

Esta classe contém os atributos quantidade e produto, os quais indicam quantos produtos de um determinado tipo. Está relacionado de um para um com a classe Produto e uma outra relação é com a classe Venda podendo ter um ou mais objetos do tipo ItemVenda.

Classe Cheque

Esta classe contém as caracteristicas para identificar um cheque de um cliente, seus atributos são o número do cheque, valor, número da conta, e data de validade. Esta classe se relaciona com apenas um cliente.

Classe Pagamento

A classe Pagamento contém os atributos de um pagamento, tipoPagamento (cheque, dinheiro), status do pagamento (efetuado,pendente), e o valor. Esta classe se associa com um objeto da classe Venda.

10 – FIGURAS

Figura 1: Modelo de Dependências Estratégicas (SD)

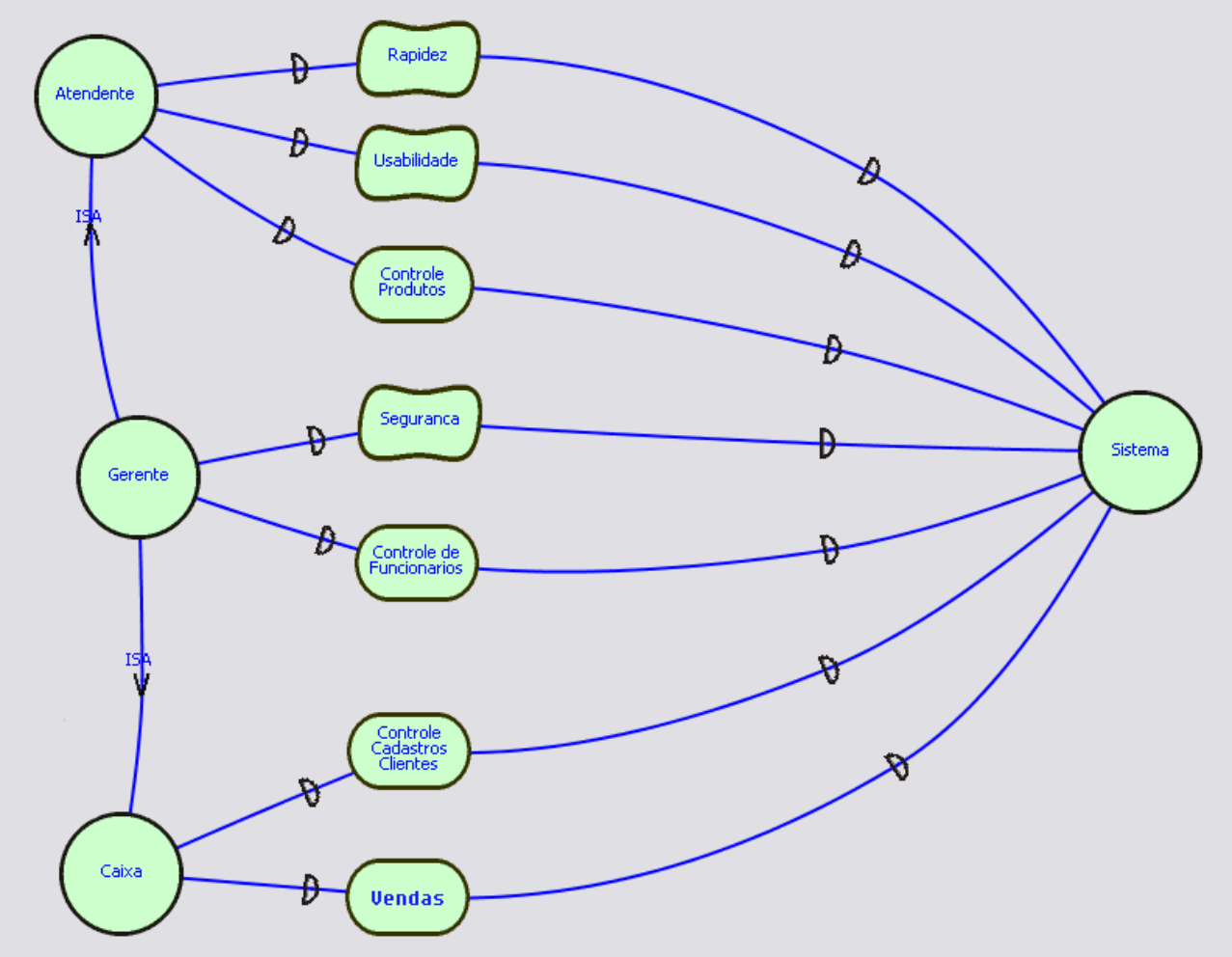


Figura 2: Modelo de Razões Estratégicas (SR)

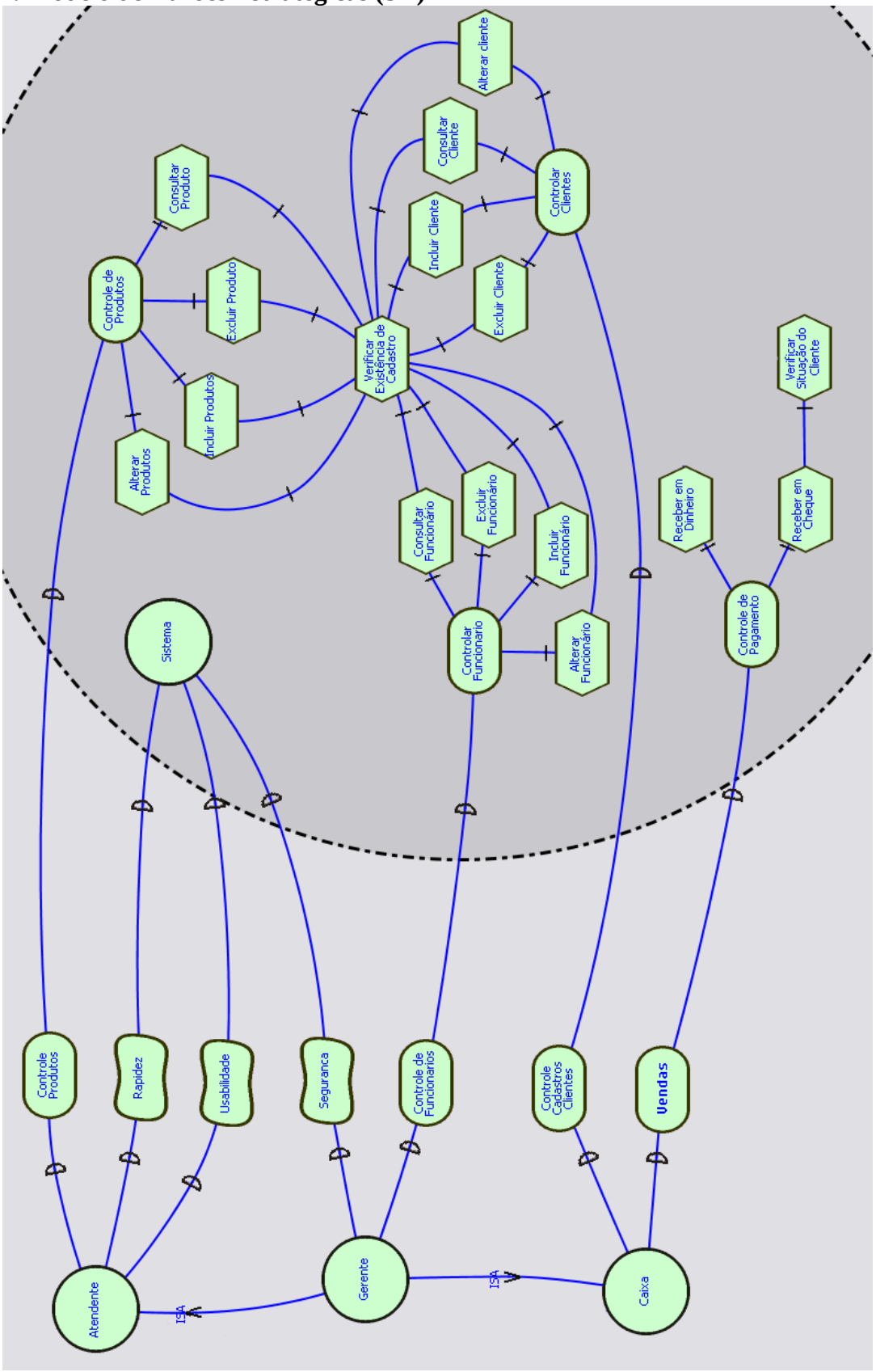


Figura 3: SIG - Softgoal Interdependency Graph

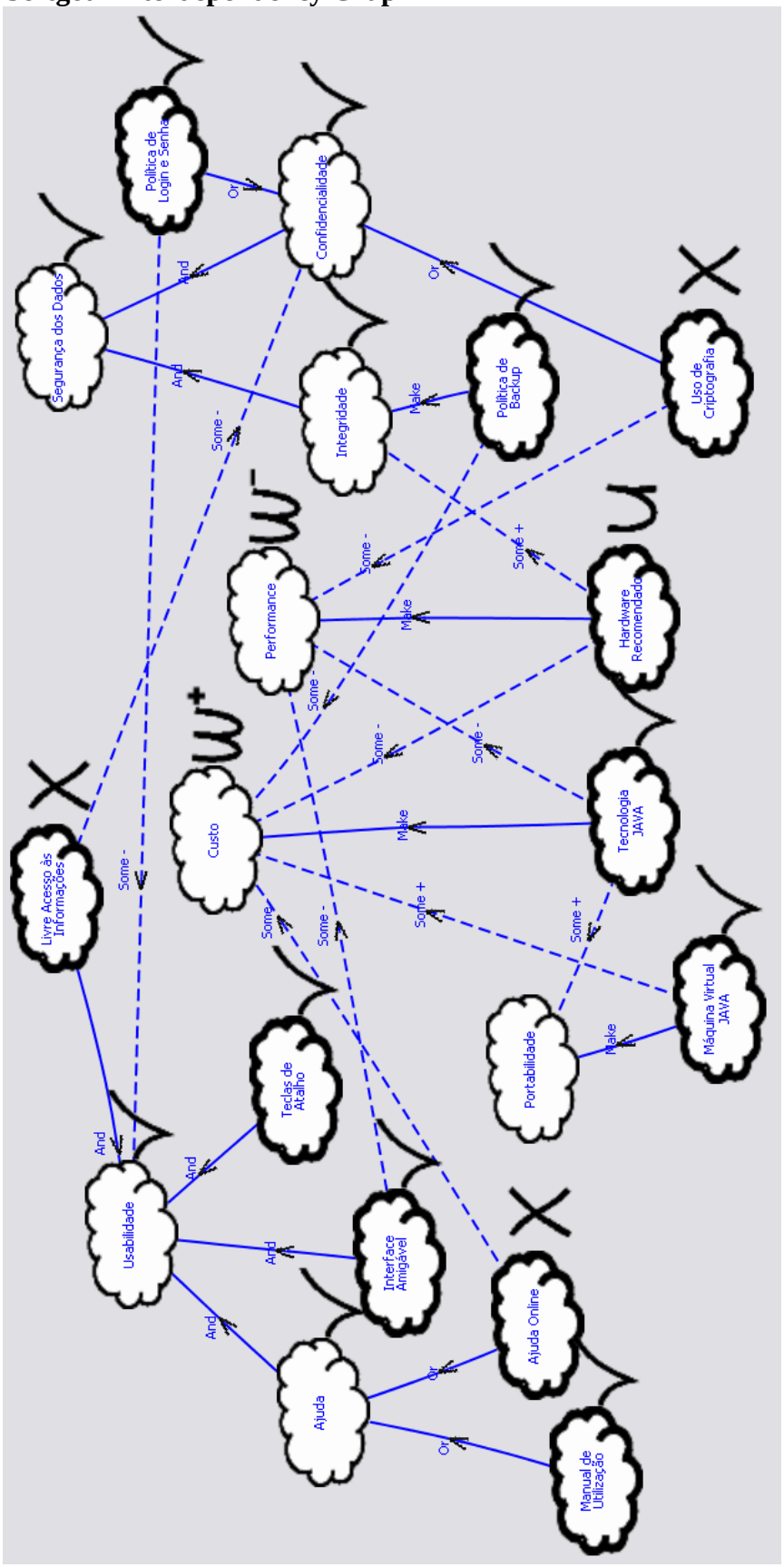


Figura 4: Diagrama de Casos de Uso.
Figura 4.1: Controle de Clientes.

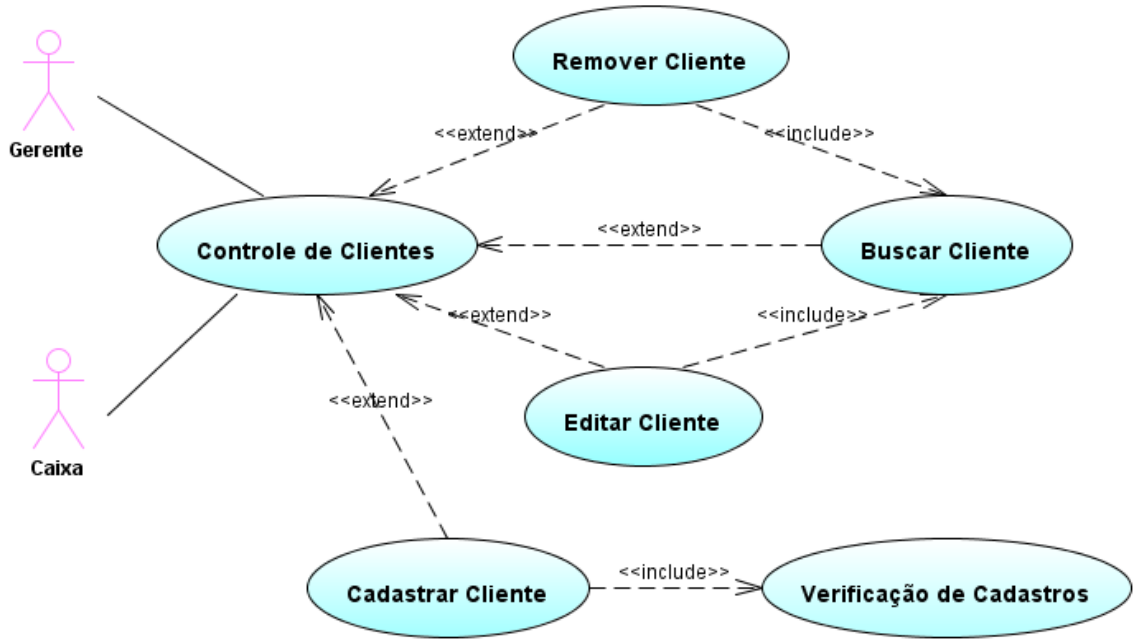


Figura 4.2: Controle de Funcionários.

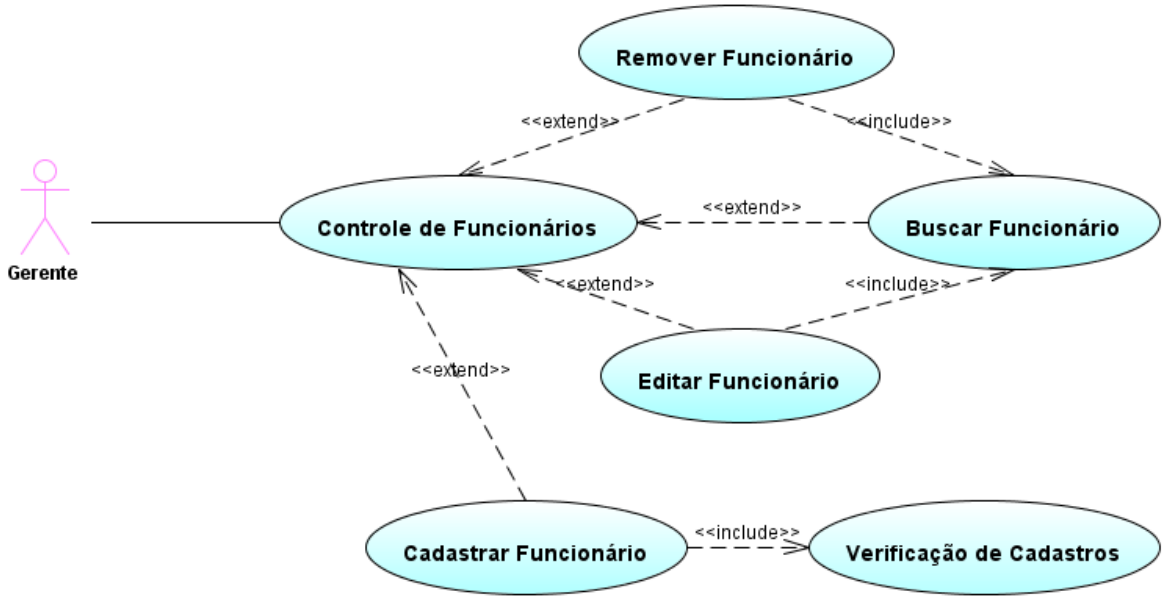


Figura 4.3: Controle de Pagamento.

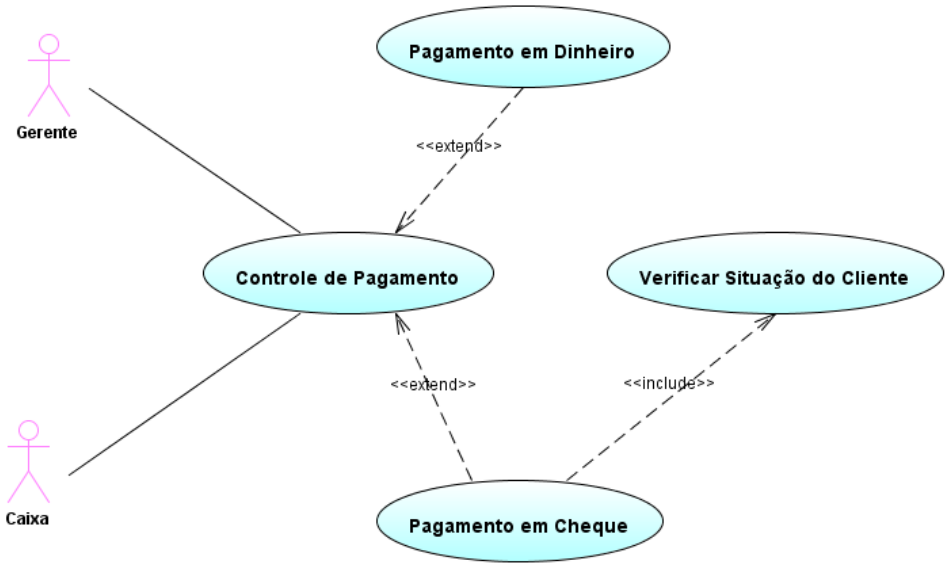


Figura 4.4: Controle de Produtos.

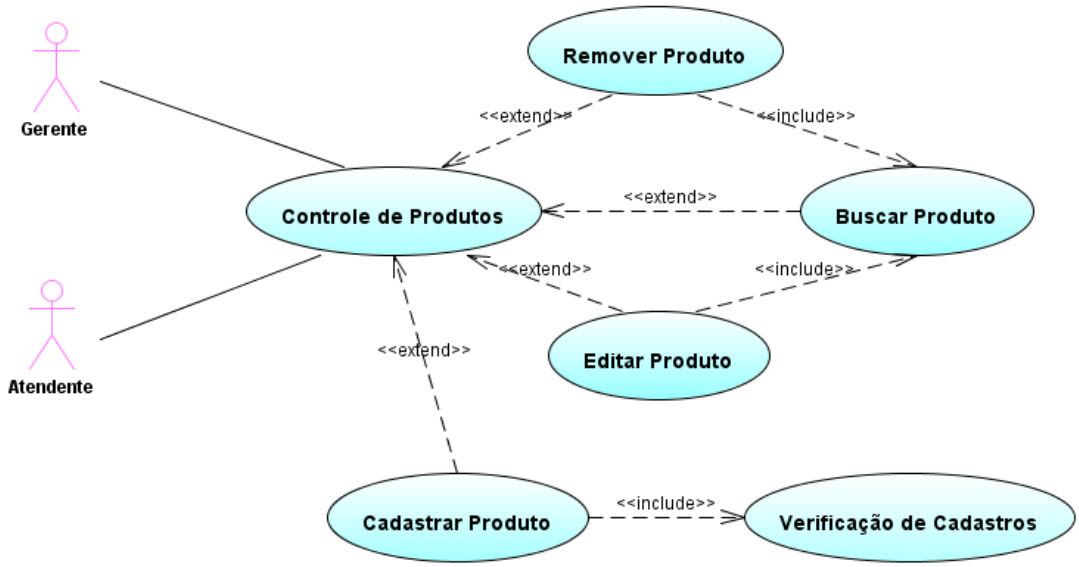


Figura 4.5: Controle de Vendas.

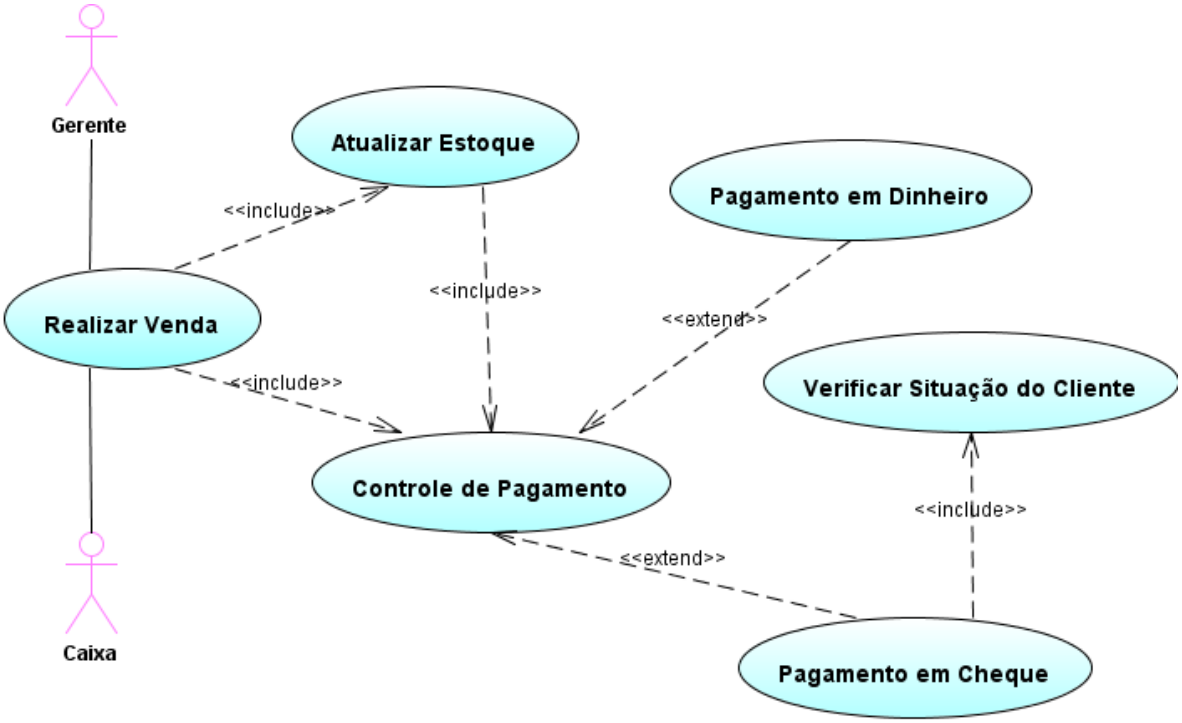
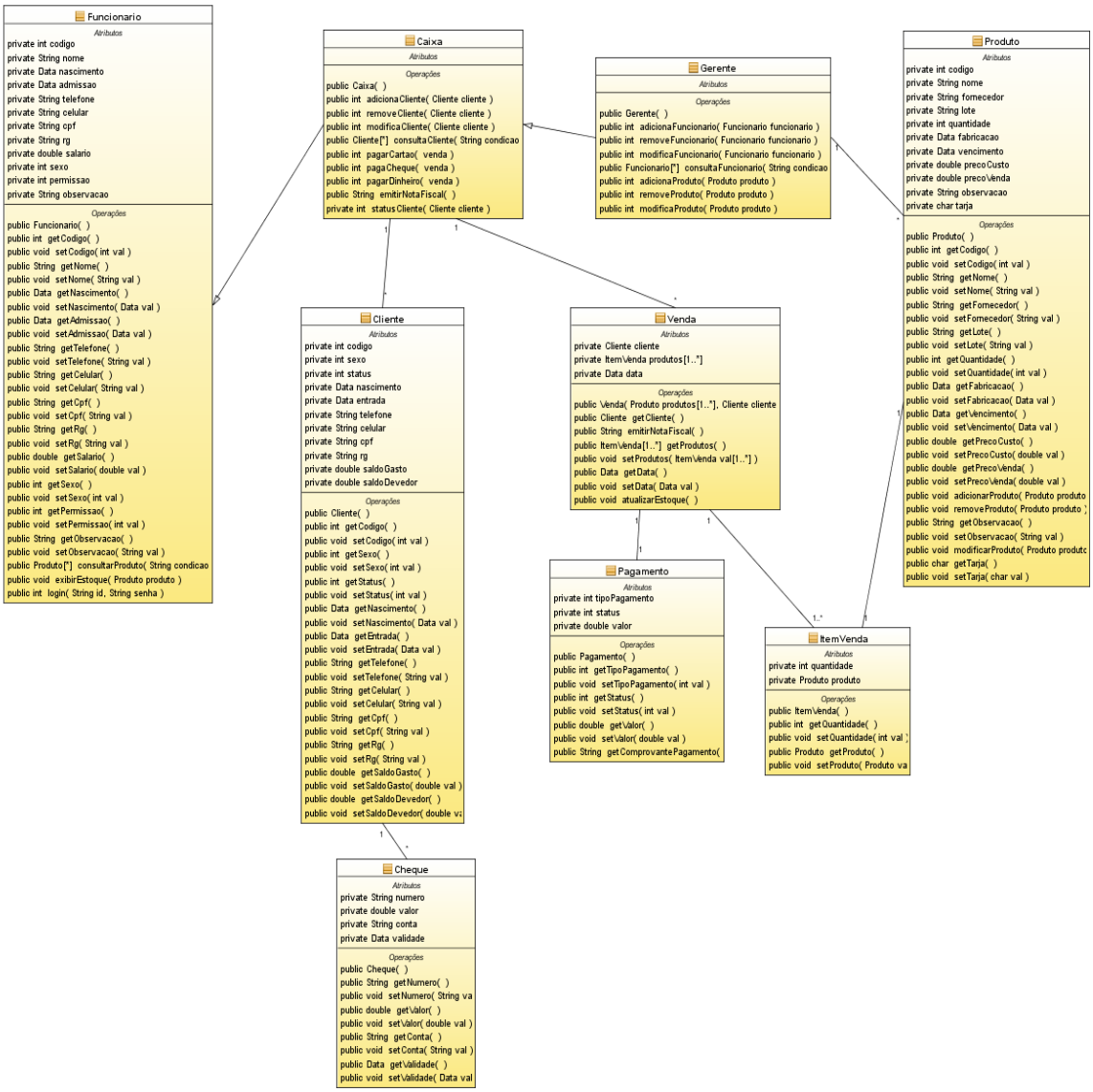


Figura 5: Diagrama de Classes



11 – CONCLUSÃO

Em busca de um fácil entendimento do funcionamento do sistema, utilizamos a Técnica I* para elaborar os Modelos de Dependências e de Razões Estratégicas. Para melhor esclarecimento de como satisfazer os requisitos não-funcionais, utilizamos a modelagem NFR Framework, com o grafo SIG. Usamos também ferramentas UML para a construção do modelo de Casos de Uso e Diagrama de Classes, a fim de visualizar as iterações entre os usuários e o sistema.

Procuramos, através deste documento, atender a todos os requisitos necessários para a satisfação das necessidades da empresa. Esta documentação nos ajudará e servirá de apoio para a implementação do nosso sistema.

12 – APÊNDICE A

COLETA DE INFORMAÇÕES

Baseamos nossa coleta de informações na entrevista com o proprietário da Farmácia São Lucas, o Sr. Valdir Antonio Stokmann. Ele explanou o funcionamento básico do ambiente organizacional. Questionado a respeito dos requisitos do sistema, ele expôs as necessidades primordiais para implementação. Também foi dado ênfase aos principais problemas encontrados no ambiente de trabalho e o que deve ser feito para que eles possam ser superados.

Resumidamente, as principais dúvidas foram:

1) Quais as atividades mais comuns no dia-a-dia de uma Farmácia.

"Vendemos medicamentos e produtos gerais, recebemos novos itens, pagamos faturas de fornecedores e atendemos nossos clientes da melhor forma possível."

2) Quais os problemas mais comuns encontrados na organização.

"Os preços dos produtos são marcados em apostilas que recebemos dos fornecedores. Os dados dos clientes são armazenados em fichas e essas são guardadas em um ficheiro. Localizar um determinado produto ou cliente requer tempo, coisa que não agrada o cliente. O comprovante de compra é feito a mão em um simples bloco com o nome e o endereço da Farmácia."

3) O que fazer para superá-los.

"Seria útil ter todas essas informações digitalizadas, economizando assim tempo e espaço, dando mais dinamicidade às operações da Farmácia. No caso da venda ao cliente, um simples comprovante de compra já seria o suficiente."

4) Qual o nível de capacitação dos funcionários com relação à Informática.

"Os funcionários possuem o nível básico de um usuário doméstico. Quanto mais simples for o sistema, melhor vai ser para que eles aprendam rápido."

5) Em quais aspectos a informatização do sistema ajudaria a empresa.

"Todos nossos registros ficam guardados em ficheiros, ou são apostilas com preços. Uma informatização resultaria em ganho de tempo e espaço."

6) O que se espera de um software para o gerenciamento de uma Farmácia.

"Nesse caso espera-se um controle eficaz de cadastro de dados no sistema, com funções de cadastro, alteração, exclusão e consulta de dados. Espera-se ainda que o sistema tenha uma usabilidade de fácil absorção por parte de quem o usa."

13 – APÊNDICE B

FORMULÁRIO DE RELATÓRIO DA EQUIPE

Descrição de papéis e contribuição de cada membro da equipe:

- Não houve uma divisão rígida entre os integrantes da equipe. Optamos por um desenvolvimento de projeto progressivo e igualitário.

NOME	% ESFORÇO NA EQUIPE
_____ Alessandro Rodrigo Franco	33,3%
_____ Fernando Luiz Grando	33,3%
_____ Fernando Martins	33,3%

14 – REFERÊNCIAS BIBLIOGRÁFICAS

http://www.inf.unioeste.br/~victor/processoII/ (acessado em Abril/2009)

http://www.inf.unioeste.br/~ivonei/PesI/ (acessado em Abril/2009)

PRESSMAN, R. S. **Engenharia de Software.** 6ª edição. McGraw-Hill, 2006.