Documentação

Projeto de Software

Professor: Rohit Gheyi

Equipe:

* Felipe Lindemberg
* Fagner Gomes
* Felipe José
* Igleson Freire
* Wallison Fernando

Conteúdo - Sistema Novo

[1 Introdução 5](#_Toc269035784)

[1.1 Motivação 5](#_Toc269035785)

[1.2 Visão da Solução 5](#_Toc269035786)

[1.3 Visão Geral do Documento 5](#_Toc269035787)

[2. Planejamento 6](#_Toc269035788)

[2.1. Introdução 6](#_Toc269035789)

[2.2. Gerência do Tempo 6](#_Toc269035790)

[2.2.1. Descrição das Estimativas 7](#_Toc269035791)

[2.2.2. Análise das Estimativas 7](#_Toc269035792)

[2.3. Gerência do Custo 8](#_Toc269035793)

[2.4. Gerência de Riscos 8](#_Toc269035794)

[2.5. Recursos do Projeto 9](#_Toc269035795)

[3. Processo 10](#_Toc269035796)

[4. Requisitos 10](#_Toc269035797)

[4.1. Requisitos Não-Funcionais 10](#_Toc269035798)

[4.1.1. Requisitos de Processo 10](#_Toc269035799)

[4.1.2. Requisitos de Produto 11](#_Toc269035800)

[4.1.2.1. Segurança 11](#_Toc269035801)

[4.1.2.2. Performance 11](#_Toc269035802)

[4.1.3. Requisitos Externos 11](#_Toc269035803)

[4.1.3.1. Requisitos Legais 11](#_Toc269035804)

[4.1.3.2. Restrições Econômicas 11](#_Toc269035805)

[4.2. Requisitos Funcionais 12](#_Toc269035806)

[4.2.1. Especificação dos Requisitos 12](#_Toc269035807)

[4.3. Diagrama de Casos de Uso 13](#_Toc269035808)

[4.4. Prototipação de Telas 14](#_Toc269035809)

[5. Projeto Arquitetural 14](#_Toc269035810)

[6. Modelagem 14](#_Toc269035811)

[6.1. Modelo de Domínio 15](#_Toc269035812)

[6.2. Análise dos Casos de Uso 15](#_Toc269035813)

[6.3. Caso de Uso: Criar Nova Venda 16](#_Toc269035814)

[6.3.1. Diagrama de Sequência do Sistema (DSS) 16](#_Toc269035815)

[6.3.2. Contrato 16](#_Toc269035816)

[6.4. Diagrama de Classes do Projeto 17](#_Toc269035817)

[7. Especificação Formal 17](#_Toc269035818)

[8. Implementação 18](#_Toc269035819)

[9. Testes 18](#_Toc269035820)

**Conteúdo - Sistema Real**

[10. Introdução 19](#_Toc269035821)

[11. Análise de Qualidade 19](#_Toc269035822)

[12.  Testes 19](#_Toc269035823)

[13. Evolução 20](#_Toc269035824)

**Sistema Novo**

# Introdução

## Motivação

Cada vez mais as TV’s oferecem aplicativos, e com a disponibilidade de internet nas TV’s uma gama de funcionalidades foram adicionadas. Porém, a principal forma de interação entre usuário e TV continua sendo primitiva, utilizando o controle remoto.

Com a popularização dos dispositivos móveis, um aplicativo para controlar a TV será muito útil, pois irá facilitar e agilizar a comunicação do usuário final com as funcionalidades da TV.

|  |  |
| --- | --- |
| O problema é... | Diversificar e facilitar os modos de interação entre usuário final e TV. |
| Que afeta... | A empresa que produz a TV e o usuário final. |
| O impacto disto é... | Irá facilitar a forma de interação com as funcionalidades da TV e irá agregar valor a mesma. |
| A solução seria... | Desenvolver um aplicativo para dispositivos ANDROID que permita a conexão por meio de rede (mesma rede) entre o Gadget e a TV. Esse aplicativo irá fornecer as principais funções que um controle remoto tradicional oferece e mais alguns diferencias, como um teclado que irá facilitar muito a digitação de textos e atalhos específicos para aplicativos instalados na TV. |

## Visão da Solução

Melhor interação em um mesmo ambiente (dispositivo móvel), agilidade de comunicação com a TV e comodidade do usuário.

## Visão Geral do Documento

Este documento está organizado da seguinte forma:

* Na Seção 1, apresentou-se a motivação e o problema que levou à construção desse trabalho;
* Na Seção 2 são apresentados aos requisitos organizacionais do produto;
* Na Seção 3...

# 2. Planejamento

# 2.1. Introdução

No plano de projeto, apresentamos a descrição do Sistema de Informatização para Clínicas Médicas (daqui por diante abreviado como SICLIM) que visa informatizar ao máximo as operações que ocorrem diariamente em uma clinica médica de pequeno/médio porte e que em alguns casos, ainda são manuais. Operações essas como cadastro de pacientes, marcação de consultas, histórico de ficha médica de um paciente, e calculo de ordenados a partir da quantidade de consultas realizadas pelos médicos, adicionando outros fatores necessários como taxa de cada plano de saúde...

## Gerência do Tempo

**Artefatos:**

1. Estimar o tempo de cada atividade no início da disciplina, baseando-se na experiência do grupo.
   1. Estimar o tempo de cada atividade no início e ao final do 1o mês, utilizando os formulários que serão enviados para o grupo. *(Milestone 1)*
2. Realizar a Estimativa de Esforço (casos de uso). *(Milestone 1)*
   1. Adicionar [template](http://groups.google.com/group/es-ufcg/web/planilha-estimativa_esforco.xls) da estimativa
3. Definir cronograma de atividades para cada membro *(Milestone 1)*
4. Descrever as estimativas no relatório. Lembrar de explicar qual a estimativa escolhida para a criação do cronograma de atividades *(Milestone 1)*
5. Atualizar cronograma de acordo com a evolução do projeto e expor resultado no documento final *(Documento final)*
6. Identificar o tempo real gasto em cada atividade através do Thymer, adicionar printscreen.
   1. Preencher o 3º formulário que será enviado para o grupo com o valor real *(Documento Final)*
7. Analisar e comparar os valores dados em cada estimativa e no configuração do cronograma de atividades ao término do projeto *(Documento final)*
   1. Acertou? Errou? Por que?
   2. Subestimou alguma atividade? Qual?
   3. O que vocês podem fazer para acertar na estimativa de tempo para um próximo projeto? Analise cada atividade separadamente.

**Ferramentas**:

1. **1º Formulário de Estimativa do Tempo - (1)**
2. **2º Formulário de Estimativa do Tempo - (1)**
3. **3º Formulário de Estimativa do Tempo - (6)**
4. **Planilha do UCP (ver** [template](http://groups.google.com/group/es-ufcg/web/planilha-estimativa.xls)**) - (2)**

**Thymer e MS project - (3 e 5)**

1. **Relatório do Word (4 e 7)**

### Descrição das Estimativas

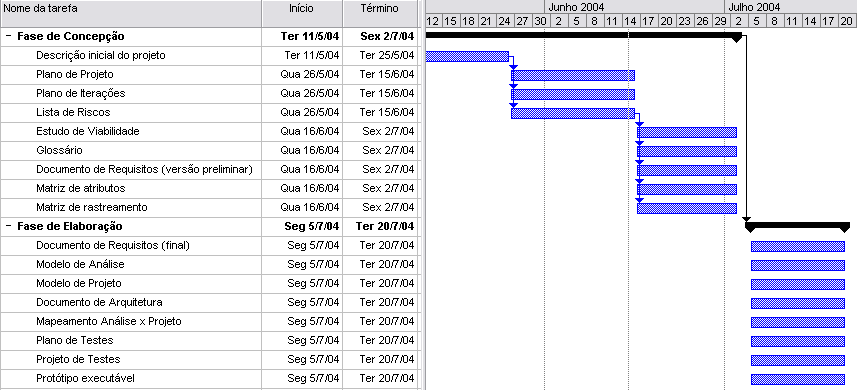


Figura 1 – Cronograma do projeto inicial

Façam o detalhamento das tarefas e quem foi responsável por cada tarefa

# 

Figura 2 – Alocação de Pessoas e Tarefas

### Análise das Estimativas

Mostre e compare as duas estimativas realizadas durante o projeto com o valor real. Qual foi o percentual de acerto? Subestimou alguma atividade? Faça um gráfico no Excel comparando as 9 atividades. Justifique a mudança. O que vocês farão nos próximos projetos para evitar alguns dos problemas encontrados?

## Gerência do Custo

**Artefatos:**

1. Utilizar as técnicas de estimativas de custo apresentadas em sala de aula (pelo menos a baseada em caso de uso)
   1. Adicionar [template](http://groups.google.com/group/es-ufcg/web/planilha-estimativa_esforco.xls) da estimativa
2. **Ferramenta**:
   1. Relatório do Word
   2. Planilha do Excel

Façam a análise de custo com a planilha no site (técnica baseada em casos de uso – justifiquem as escolhas). Estimem também o custo de hardware e lucro. Ver slide da aula.

* Custo
  + Pessoal
    - (2000\*3 pessoas + 5000 gerente)\*10 meses = 110000 reais
  + Equipamento (software+hardware)
    - 4 computadores (2000) + software (10000) = 18000 reais
  + Diversos (luz, água, aluguel, viagens)
    - 1000 reais/mês \* 10 = 10000 reais
* Total = 110000 + 18000 + 10000 = 138000 reais
* Erro da estimativa = 20% em cima do total (inflação)
* Lucro = 30% em cima do erro
* Custo total = 215280 reais

## Gerência de Riscos

**Artefatos:**

1. Listar os riscos no início do projeto *(Milestone 1)*
   1. Métrica mínima: 1 risco por pessoa
2. Monitorá-los durante as fases do projeto e, no documento final, analisar cada risco, indicar algum que porventura surgiu e qual foi a reação da equipe, plano de conteção, etc. *(Documento Final)*
3. **Ferramenta**: Relatório no Word

Citem os riscos iniciais e analisem-os ao longo do projeto. Como cada risco inicial foi gerenciado? Surgiu algum risco fora do planejado inicialmente? Se sim, qual? O que vocês fizeram para lidar com isso (estratégia de gerenciamento dos riscos)? Descrevam como monitoram os riscos ao longo do projeto.

Os riscos do projeto serão avaliados pelo menos uma vez em cada iteração e documentados nesta tabela. Os riscos de maior importância são listados primeiro na tabela. Como o projeto não terá muitas dificuldades de desenvolvimento, como mostra o Estudo de Viabilidade, os riscos detectados foram poucos.

|  |  |  |
| --- | --- | --- |
| **Classificação do Risco** | **Impacto e Descrição do Risco** | **Estratégia de Diminuição e/ou Plano de Contingência** |
| Alto | Integrante indisponível para realização de alguma tarefa | Procurar planejar ao máximo de antecedência para que isso não ocorra e fazer com que no mínimo dois integrantes da equipe dominem o desenvolvimento de uma determinada tarefa |
| Médio | Pouca experiência em Banco de Dados | Treinar durante a disciplina de Gerenciamento de Dados e Informação |
| Médio | Tempo de desenvolvimento não ideal | Gerenciamento de cronograma e pontualidade |
| Médio | Dificuldade na manutenção de consistência | Gerenciar Banco de Dados e realizar próxima tarefa quando anterior concluída |
| Médio | Complexidade de Interface gráfica | Definir padrão e dividir para que cada integrante implemente uma parte da interface gráfica |
| Baixo | Desenvolvimento em 3 níveis | Modularização do sistema, para que cada parte funcione independente |

Tabela 1. Xxx

## Recursos do Projeto

Abaixo estão listados os recursos de hardware e software a serem utilizados no projeto.

**Pessoas**

* ...

**Recursos de Hardware:**

* Computadores do DSC – configuração variável dependendo de cada laboratório;
* Computadores pessoais de cada integrante.

**Recursos de Software:**

* Eclipse 3.1 – Utilizado para programar o projeto em Java
* Microsoft Word 2003 e plugin pdf: elaboração e melhor leitura dos relatórios
* Rational Rose – modelar o projeto em UML
* JUnit – framework para testes de casos de uso, em Java
* Oracle 10g – Versão do SGBD a ser utilizado para guardar os dados do SICLIM.
* CVS: controle de versão para melhor gerenciamento do projeto

Houve custo na aquisição de algum software e hardware? Comentem.

# 3. Processo

**Artefatos:**

1. Discorrer sobre o processo utilizado no projeto (*Milestone 1*)
   1. Métrica: no máximo uma página
2. **Ferramenta:** Relatório no Word

# 4. Requisitos

**Artefatos:**

1. Descrever os requisitos funcionais (RF) e não-funcionais (RNF) do sistema em um documento *(Milestone 1)*
   1. Métrica mínima: 3 RFs por pessoa  e 1 RNF por grupo
2. Criar Diagrama de Casos de Uso do Sistema *(Milestone 1)*
   1. Métrica Mínima: Correspondência com os RFs
3. Prototipar as telas do sistema e expô-las no documento *(Milestone 1)*
   1. Métrica mínima: 1 por pessoa
4. **Ferramenta:**
   1. Relatório no word - (1)

Vejam o documento da Imagine Cup para definições dos termos usados aqui e mais algumas dicas! (tópico 4)

## Requisitos Não-Funcionais

Alguns requisitos não-funcionais (NFRs) foram identificados para o projeto do TIG. Abaixo, são apresentadas as descrições para os três tipos de NFR (de processo, de produto e externos), destacando-se o grupo de requisitos de produto, cujo grafo SIG foi construído e comentado [6].

### Requisitos de Processo

|  |  |
| --- | --- |
| **Ident.** | Descrição |
| **RNF/PROC-01** | O sistema deve ser implementação na linguagem JAVA. Logo ele poderá ser executado em diferentes sistemas operacionais, como o Linux, Windows, etc. |
| **RNF/PROC-02** | Toda atualização nas informações tratadas pelo sistema deverá ser feita no servidor. |
| **RNF/PROC-03** | Deverá ser feita uma documentação que contenha o diagrama de classes, já que a linguagem utilizada será orientada a objetos, e informações sobre o código-fonte do projeto. |
| **RNF/PROC-04** | Deverão ser utilizadas ferramentas CASE, e é imprescindível a criação da modelagem usando a linguagem UML. |

### Requisitos de Produto

### Segurança

|  |  |
| --- | --- |
| **Ident.** | Descrição |
| **RNF/SEG-05** | Apenas quando o turista for consultar seu extrato no hotel será necessário fazer o Logon. |
| **RNF/SEG-06** | A atualização do sistema será feita apenas a partir da rede interna do hotel. |
| **RNF/SEG-07** | As informações sobre o turista permanecerão armazenadas apenas no banco de dados do hotel, não sendo armazenadas no banco de dados do sistema. |

### Performance

|  |  |
| --- | --- |
| **Ident.** | Descrição |
| **RNF/PER-08** | O tempo de resposta para operações que envolvam dados multimídia (imagens, vídeos, etc.) não deve exceder 30 segundos. Para qualquer outra operação do sistema, esse não deve exceder 5 segundos. |
| **RNF/PER-09** | O espaço disponível em disco para informações deve ser capaz de armazenar todos os dados/atualizações que forem cadastrados a partir do servidor. |

### Requisitos Externos

### Requisitos Legais

|  |  |
| --- | --- |
| **Ident.** | Descrição |
| **RNF/LEG-20** | Caberá ao hotel garantir a veracidade das informações sobre os locais cadastrados. |

### Restrições Econômicas

|  |  |
| --- | --- |
| **Ident.** | Descrição |
| **RNF/ECO-21** | O custo total para a implantação do sistema (não incluindo os custos com manutenção) não deve extrapolar em mais de 10% o valor inicialmente previsto em seu Estudo de Viabilidade (R$ 33.031,50). |

## Requisitos Funcionais

Os requisitos funcionais descrevem o que o sistema faz, isto é, as funções necessárias para se obter os objetivos do sistema. No TIG, apenas o requisito [RF-16] foi classificado como importante, visto que o sistema poderia ocasionalmente funcionar sem ele,  e somente o RF-17 foi classificado como desejável, tendo-se em vista que o mesmo seria apenas um adicional para o sistema. Segue abaixo uma lista resumida dos requisitos funcionais do TIG.

|  |  |  |
| --- | --- | --- |
| **Cód.** | **Nome** | **Prioridade** |
| **RF-01** | Cadastro de Hóspede | Essencial |
| **RF-02** | Remoção de Hóspede | Essencial |
| **RF-03** | Cadastro de Categorias de Locais | Essencial |
| **RF-04** | Remoção de Categorias de Locais | Essencial |
| **RF-05** | Cadastro de Local | Essencial |
| **RF-06** | Remoção de Local | Essencial |
| **RF-07** | Atualização de Local | Essencial |
| **RF-08** | Cadastro de Informação do Hotel | Essencial |
| **RF-09** | Remoção de Informação do Hotel | Essencial |
| **RF-10** | Atualização de Informação do Hotel | Essencial |
| **RF-11** | Efetuar Login (Administrador) | Essencial |
| **RF-12** | Consultar Extrato | Essencial |
| **RF-13** | Consultar Informações do Hotel | Essencial |
| **RF-14** | Efetuar Login (Hóspede) | Essencial |
| **RF-15** | Consultar Informações de Locais | Essencial |
| **RF-16** | Consultar Informações Úteis | Importante |
| **RF-17** | Consultar Ajuda On Line | Desejável |

### Especificação dos Requisitos

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| RF-01 | | | | | |
| **Nome:** | Cadastro de Hóspede | | | | |
| **Descrição:** | O sistema deve permitir que o administrador do sistema cadastre um novo hóspede em seu banco de dados. Para efetuar esse cadastro serão necessários apenas os números do quarto e a senha do hóspede. Uma mensagem deve ser exibida caso o usuário já exista. | | | | |
| Atores: | Administrador do Sistema | | | | |
| Prioridade: | Essencial | | | **Anexo:** | Anexo 01 |
| Requisitos Não Funcionais Associados: | | | NFR/USA-12 e NFR/PER-08 | | |
| Entradas e pré-condições: | | | Administrador ter efetuado o *login* no sistema | | |
| Saídas e pós-condições: | | | O usuário cadastrado no banco de dados do sistema. | | |
| Fluxos de eventos | | | | | |
| Fluxo principal: | | 1.   O administrador do sistema informa os dados do hóspede necessários para a realização do cadastro:   * Número do quarto em que está hospedado; * Senha; * Confirmação da senha.   2.   O sistema verifica se a senha é igual à confirmação da senha;  3.   O sistema verifica se o hóspede já está cadastrado;  4.   O sistema armazena os dados do usuário no banco de dados e informa que o cadastro foi realizado com sucesso | | | |
| Fluxo secundário 1: | | No fluxo principal 2, se a senha e a sua confirmação forem diferentes, o sistema exibe uma mensagem solicitando que as senhas sejam digitadas novamente, voltando ao passo 1. | | | |
| Fluxo secundário 2: | | No fluxo principal 3, se o usuário já estiver cadastrado, o sistema exibe uma mensagem informando o ocorrido e retorna ao passo 1. | | | |

## Diagrama de Casos de Uso

 O diagrama de casos de uso, expresso em UML (*Unified Modeling Language*), expressa os requisitos não funcionais do sistema na forma de casos de uso. Segundo o RUP (*Rational Unified Process*), para cada requisito funcional tem-se um caso de uso [7]. Utilizando-se do diagrama de casos de uso do TIG (Figura 6), poder-se-á então construir o diagrama de classes do sistema, executando se as etapas de análise e projeto do RUP [7] (Vide **Seção 6** / **Apêndice F**). A descrição textual detalhada dos requisitos funcionais, seus fluxos de atividades e requisitos não funcionais associados pode ser encontrada no **Apêndice E**. Na Figura 1, tem-se a representação gráfica, em UML, dos casos de uso do TIG.

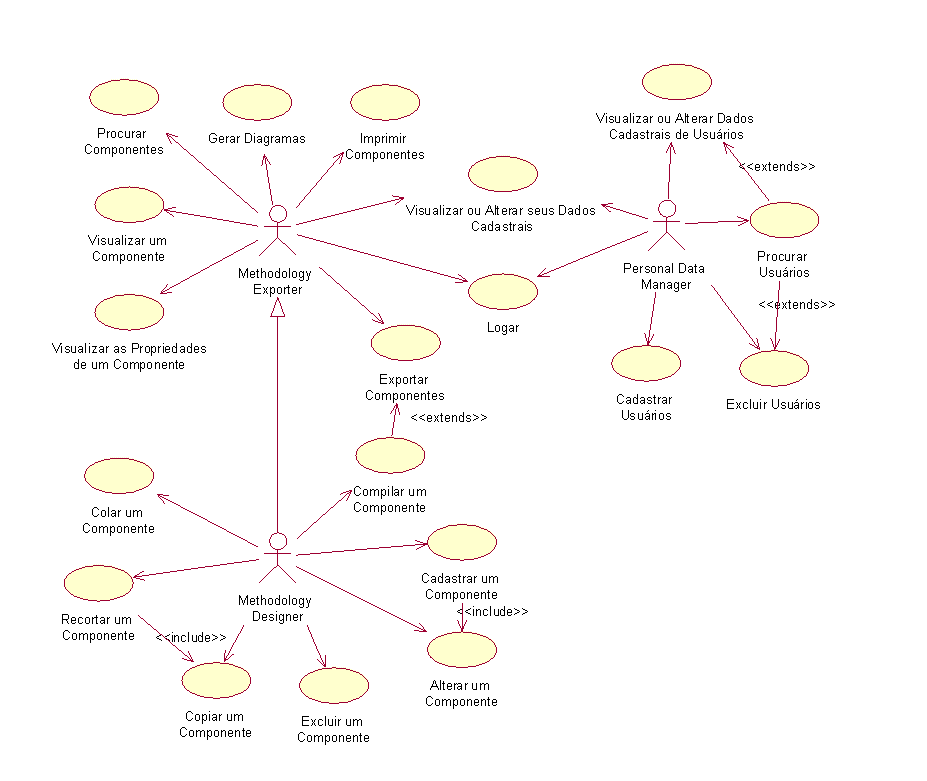
Vejam o documento da Imagine Cup para definições dos termos usados aqui e mais algumas dicas! (tópico 10)

Figura 1 - XXXX

## Prototipação de Telas

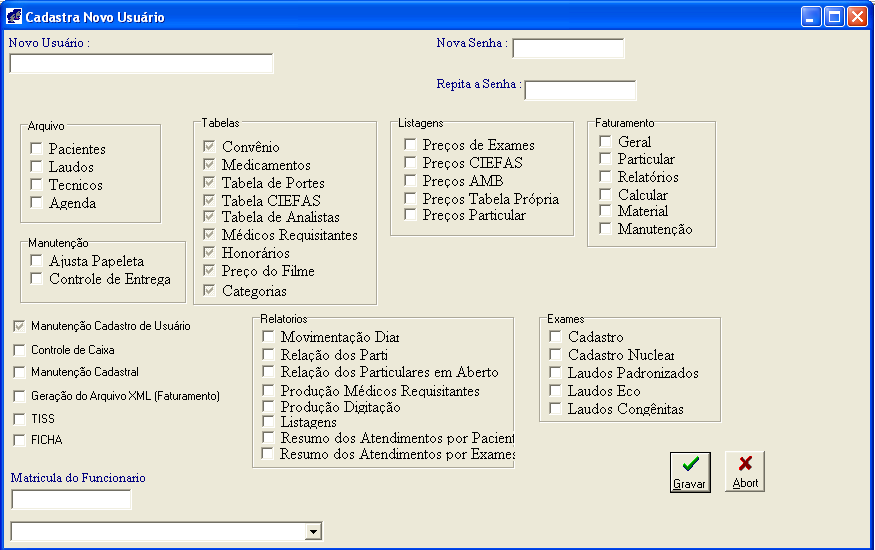


Figura 2 - XYYYYY

Vejam o documento da Imagine Cup para definições dos termos usados aqui e mais algumas dicas! (tópicos 16 e 17)

# 5. Projeto Arquitetural

1. Descrevam a arquitetura do sistema desenvolvido *(Milestone 2)*
   1. Métricas mínimas: 1 ou 2 parágrafos
2. **Ferramenta:** Relatório no Word

# 6. Modelagem

**Artefatos:**

1. Modelar o sistema novo: *(Milestone 2)*
   1. Métricas mínimas:
      1. Um *modelo de domínio*
      2. 1 *diagrama de seqüência do sistema* por pessoa
      3. 1 *contrato de uma operação* por pessoa
2. Criar o diagrama de classes detalhado do sistema *(Milestone 2)*
3. Adicionar os diagramas no relatório
4. **Ferramenta:**
   1. Ferramenta para criar os diagramas (qualquer) - (1 e 2)
   2. Relatório no word - (3)

## 6.1. Modelo de Domínio

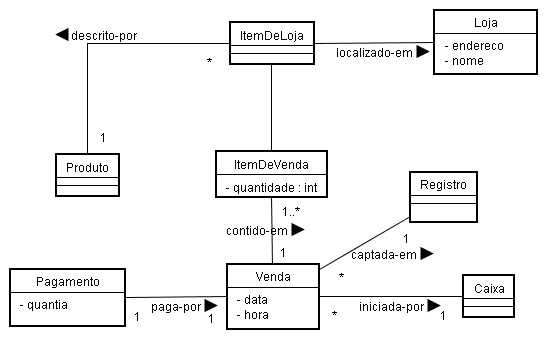
Incluir aqui uma figura com o diagrama de classes UML referente ao modelo de domínio do sistema, levando em conta apenas os casos de uso detalhados na especificação dos casos de uso.

Figura 3 - QWWWW

## 6.2. Análise dos Casos de Uso

A seguir, apresento os casos de uso que serão mapeados para diagramas de seqüência do sistema.

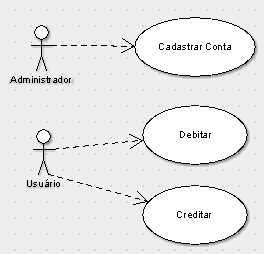


Figura 4 - TTTTT

Para cada caso de uso analisado, inclua um diagrama de seqüência do sistema com as operações do caso de uso, seguido pelos contratos das operações.

## Caso de Uso: Criar Nova Venda

### Diagrama de Sequência do Sistema (DSS)

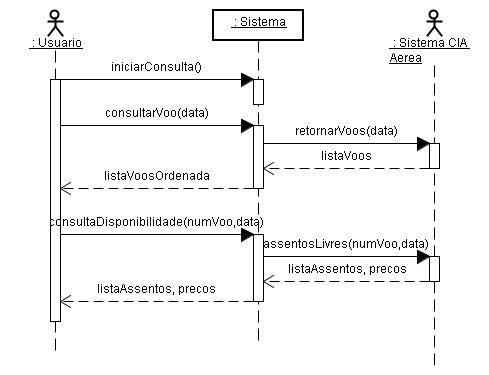
****

Figura 5 - UUUUU

### Contrato

**Operações**

* criarNovaVenda()

**Pré-condições**

* nenhuma.

**Pós-condições**

* Foi criado um objeto *ve* de Venda;
* *ve* foi associada ao Registro *reg*;
* *ve.data* e *ve.hora* foram inicializados.

## Diagrama de Classes do Projeto

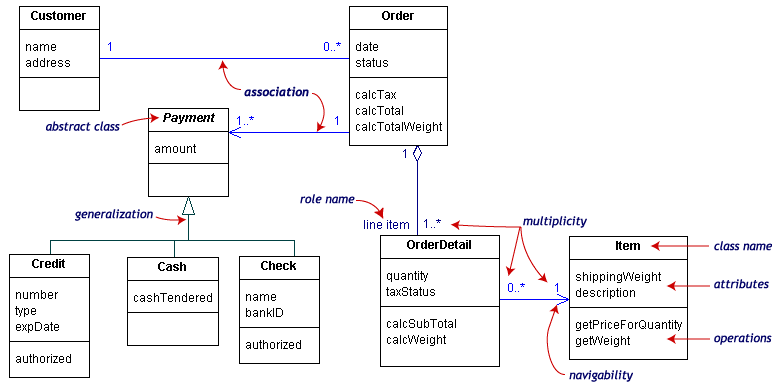
Incluir aqui um diagrama de classes UML de projeto referente às classes projetadas nas realizações de casos de uso (incluindo atributos e métodos). É melhor dividir em vários diagramas para que um diagrama apenas não seja de difícil visualização.

Figura 6 - OOOO

# 7. Especificação Formal

**Artefatos:**

1. Façam uma especificação formal da estrutura do sistema usando Alloy. Especificar o modelo de domínio usando Alloy. *(Milestone 2)*
   1. Métrica Média: 20-30 LOC
2. Especificar alguns fatos e realizar análises. Descrever as análises feitas no relatório. *(Milestone 2)*
3. **Ferramenta:** 
   1. Alloy Analizer - (1)
   2. Relatório no Word - (2)

# 8. Implementação

**Artefatos:**

1. Implementem o sistema usando Grails *(Milestone 2 e Documento Final)*
   1. Métrica mínima: 1 RNF de produto por equipe (ex.: Segurança, Usabilidade)
      1. Usem os plugins de Grails
      2. Não é necessário implementar o requisito em todo o sistema
   2. Métrica mínima: 3 RF por pessoa
      1. Não abusar do gerador de grails, é importante que o sistema possua funcionalidades que não sejam geradas automaticamente
2. Anexar imagens das telas do sistema final e suas respectivas descrições *(Documento final)*
3. Apresentar o sistema em alguma máquina que permita a avaliação, notebook, computador de algum laboratório, ou em algum servidor que permita o acesso via URL (sendo essa última a mais interessante)
4. **Ferramenta:**
   1. Grails e Netbeans - (1)
   2. Relatório no Word - (2)

# 9. Testes

**Artefatos:**

1. Testar o sistema novo
   1. Criar Plano de Testes prévio *(Milestone 1)*
   2. Realizar os testes *(Milestone 2)*
      1. Métricas mínimas:
         1. testes de unidade (2 por pessoa)
         2. integração (2 por pessoa)
         3. funcional (1 por pessoa)
   3. Recomendação:
      1. Descrever bem o que o teste cobre, ou seja, o que o teste implementado avalia!
2. Ferramenta: Netbeans

**Sistema Real**

# 10. Introdução

Descrever as características do sistema selecionado.

# 11. Análise

**Artefatos**

1. Analisar a qualidade de um sistema real de acordo com: *(Milestone 1)*
   1. Algumas métricas já definidas no plugin Metrics
      1. Métrica mínima: Duas por pessoa
   2. Bad smells no código usando o Findbugs
      1. Métrica mínima: 3 por pessoa
   3. Analise a qualidade do software utilizando a pirâmide
   4. Adicionar resultados de ambas as ferramentas
2. Análise: *(Milestone 1)*
   1. Alguma métrica indica alguma boa/má qualidade?
   2. Alguma métrica pode indicar algum problema?
   3. O problema pode ser identificado visualmente?
   4. Quantidade, complexidade, ou seja, análise dos bad smells
3. **Ferramenta**:
   1. Metrics plugin para o Eclipse  - (1)
   2. FindBugs – (1)
   3. Relatório no Word – (2)
4. Utilize a ferramenta SLOCount para estimar o custo do software. Compare este resultado com outras técnicas de estimativa

# 12.  Testes

**Artefatos:**

1. Fazer uma análise da qualidade do software real através dos testes *(Milestone 2)*

* 1. Recomendações:
     1. Gerem testes, executem-os e tentem encontrar erros. Se encontrarem algum bug, confirmem e descrevam o problema
     2. Façam uma análise da cobertura dos testes
  2. Relacionar as métricas coletadas durante a fase de Gerência de Qualidade do Software Real com os resultados dos testes obtidos.
     1. Percebeu-se alguma relação entre elas?
     2. Alguma métrica que indicava problema resultou em algum bug?
  3. Adicionar imagens dos resultados gerados pelas ferramentas
  4. Dizer quantos testes foram gerados, qual a cobertura, o processo de realização dos testes.

1. **Ferramenta:** 
   1. Randoop, Tesful ou JAutoTest para gerar testes automaticamente
   2. Junit para executar os testes
   3. Cobertura, EclEmma ou Clover para analisar a cobertura
   4. MuClipse opcional para realizar testes de mutação
   5. Relatório no Word

# 13. Evolução

**Artefatos:**

1. Refatorar o sistema real para eliminar algum problema encontrado *(Milestone 2)*
   1. Métricas mínimas:
      1. Diferença entre o sistema refatorado e o inicial deve ser em torno de 100 LOC E Número de refacts diferentes = 5
   2. Não deve-se refatorar qualquer parte do sistema.
   3. Usem sintomas que o Findbugs ou o Metrics recomendou (motivação para o refatoramento).
   4. Executem o Metrics depois do refatoramento e avaliem se alguma métrica melhorou (qualidade).
   5. Comparem a pirâmide (mostrem as duas pirâmides antes e depois) antes e depois do modelo de qualidade. Melhorou segundo a pirâmide? Façam uma análise crítica.
2. Escrever um relatório reportando o que foi mudado, que refatoramentos foram aplicados, qual foi o tamanho da mudança *(Milestone 2)*
   1. Para cada modificação realizada, descrever brevemente o problema em termos de código, o que foi feito para resolvê-lo e expor um diff (sugestão: usar o diff de versões do SVN)
3. **Ferramenta:**
   1. Netbeans ou Eclipse - (1)
   2. Relatório no word - (2).

**Conclusões**

Dividir a conclusão por fases e expor as dificuldades encontradas e os aprendizados.

Como foi a interação com o cliente? Ele ficou satisfeito? Parte ou todo o sistema foi implantado? Que atividades vocês repetiriam em um projeto real?...