**Documentação de um**

**Software**

**Versão 1.0**

**Autora:** **Ryan Junio Lacerda Calisto De Oliveira**

**Revisor: “”**

**2022**

# ÍNDICE DETALHADO

1. **INTRODUÇÃO AO DOCUMENTO................................................................................................. 6** 
   1. TEMA............................................................................................................................................ 6
   2. OBJETIVO DO PROJETO .................................................................................................................. 6
   3. DELIMITAÇÃO DO PROBLEMA ........................................................................................................ 6
   4. JUSTIFICATIVA DA ESCOLHA DO TEMA ........................................................................................... 6
   5. MÉTODO DE TRABALHO................................................................................................................. 6
   6. ORGANIZACÃO DO TRABALHO ....................................................................................................... 7
   7. GLOSSÁRIO ................................................................................................................................... 7
2. **DESCRICÃO GERAL DO SISTEMA............................................................................................... 8** 2.1. DESCRIÇÃO DO PROBLEMA ............................................................................................................ 8
   1. PRINCIPAIS ENVOLVIDOS E SUAS CARACTERÍSTICAS ....................................................................... 8
   2. REGRAS DE NEG CIOG................................................................................................................... 8
3. **REQUISITOS G DO SISTEMA.......................................................................................................... 9** 
   1. REQUISITOS FUNCIONAIS............................................................................................................... 9
   2. REQUISITOS NÃO-FUNCIONAIS......................................................................................................10
   3. PROTÍTIPO...................................................................................................................................10
   4. MÉTRICAS E CRONOGRAMA..........................................................................................................11
4. **ANÁLISE E *DESIGN*........................................................................................................................12** 
   1. ARQUITETURA DO SISTEMA ..........................................................................................................12
   2. MODELO DO DOMÍNIO ..................................................................................................................12
   3. DIAGRAMAS DE INTERAÇÃO..........................................................................................................13
   4. DIAGRAMA DE CLASSES ...............................................................................................................14
   5. DIAGRAMA DE ATIVIDADES ..........................................................................................................14
   6. DIAGRAMA DE ESTADOS...............................................................................................................14
   7. DIAGRAMA DE COMPONENTES ......................................................................................................15
   8. MODELO DE DADOS......................................................................................................................16

*4.8.1. Modelo Lógico da Base de Dados.........................................................................................16 4.8.2. Criação Física do Modelo de Dados.....................................................................................16*

*4.8.3. Dicionário de Dados............................................................................................................16*

* 1. AMBIENTE DE DESENVOLVIMENTO ...............................................................................................16
  2. SISTEMAS E COMPONENTES EXTERNOS UTILIZADOS....................................................................16

1. **IMPLEMENTAÇÃO ........................................................................................................................17**
2. **TESTES.............................................................................................................................................18** 6.1. PLANO DE TESTES ........................................................................................................................18

6.2. EXECUÇÃO DO PLANO DE TESTES..................................................................................................18

1. **IMPLANTAÇÃO ..............................................................................................................................19** 7.1. DIAGRAMA DE IMPLANTAÇÃO.......................................................................................................19

7.2. MANUAL DE IMPLANTAÇÃO..........................................................................................................19

1. **MANUAL DO USUÁRIO .................................................................................................................20**
2. **CONCLUSÕES E CONSIDERAÇÕESES FINAIS..............................................................................21**

**BIBLIOGRAFIA ......................................................................................................................................22**

**COMENT`RIOS SOBRE A DOCUMENTA˙ˆO..................................................................................24**

1. **COMO UTILIZAR O PRODUTO DE DOCUMENTA˙ˆO NO TG?............................................24**
2. **NO MEU TG IREI UTILIZAR MODELAGEM ESTRUTURADA, QUAL A DIFEREN˙A? ..24**
3. **NO MEU TG HAVER` UM ESTUDO TE RICO AL M DO SISTEMA, QUAL A**

**DIFEREN˙A?...........................................................................................................................................25**

1. **QUAL O MATERIAL QUE POSSO CONSULTAR CASO TENHA D VIDAS?......................25**
2. **QUAIS SˆO AS FERRAMENTAS CASE QUE PODEM SER UTILIZADAS?.............................25**

**GLOSS`RIO ............................................................................................................................................28**

## 1. Introdução ao documento

**1.1. Tema**

O Tema do projeto é a criação de um crud, contendo as operações de CREATE, READ, UNREAD, DELETE, comuns em vários processos seletivos de empresas que querem recrutar novos programadores. Para este desafio, será implementado um sistema de cadastro de produtos que permite realizar essas quatro operações básicas.

### 1.2. Objetivo do Projeto

Testar os meus conhecimentos em programação, e possivelmente conseguir uma vaga como programador na área.

### 1.3. Delimitação do Problema

O objetivo do projeto é criar um sistema capaz de cadastrar vários produtos de uma loja em um banco de dados e posteriormente, criar funções que permitem a operações básicas para gerenciamento desses produtos como por exemplo, deletar, criar, editar e listar todos os produtos.

### 1.4. Justificativa da Escolha do Tema

A escolha do tema em questão de deve ao fato de que esse sistema faz parte de um processo seletivo elaborado por uma empresa de tecnologia

### 1.5. Método de Trabalho

A arquitetura utilizada: MVC, (model, view, controller)

metodologia de desenvolvimento de software: Kanban

modelagem de dados: Orientada a objetos

**1.6. Organização do Trabalho**

Documentação, levantamento de requisitos, regras de negócio, protótipo, arquitetura do sistema, diagramas, modelagem de dados, design, implementação, testes, publicação.

## 2. Descrição Geral do Sistema

Este capítulo tem como objetivo descrever de forma geral o sistema, o escopo e as principais funções. A descrição geral do sistema deve abrange os itens a seguir. Como referência pode-se consultar e utilizar o modelo rup\_vision\_sp.dot artefato do RUP.

### 2.1. Descri ªo do Problema

Neste item deve ser descrito o problema que ser resolvido com o desenvolvimento do sistema. As questões a seguir devem ser respondidas.

* Quem Ø afetado pelo sistema?
* Qual Ø o impacto do sistema?
* Qual seria uma boa solução para o problema?

### 2.2. Principais Envolvidos e suas Características

**2.1.1. Usuários do Sistema**

O software se destina a lojas de produtos que querem acabar com todos os seus processos via papel, planilhas eletrônicas. O sistema será utilizado principalmente pelos vendedores que irão cadastrar um produto, verificar se há determinado produto em estoque, deletar este produto e editar.

**2.1.2. Desenvolvedores do Sistema.**

Ryan Junio Lacerda Calisto de Oliveira

### 2.3. Regras de Negócio

Neste item devem ser descritas as regras de negócio relevantes para o sistema, como por exemplo, restrições de negócio, restrições de desempenho, tolerância falhas, volume de informação a ser armazenada, estimativa de crescimento de volume, ferramentas de apoio, etc.

## 3. Requisitosdo Sistema

Este capítulo tem como objetivo descrever os requisitos do sistema. No caso de sistemas que possuam usuários / solicitantes reais para o levantamento de requisitos, pode-se utilizar o modelo de documento de entrevista com usuários do RUP de Solicitações dos Principais Envolvidos (rup\_stkreq.dot).

### 3.1. Requisitos Funcionais

Neste item devem ser apresentados os requisitos funcionais que especificam a ıes que um sistema deve ser capaz de executar, ou seja, as funções do sistema. Os requisitos funcionais geralmente sªo melhor descritos em diagramas de caso de uso, juntamente com o detalhamento dos atores e de cada caso de uso.

A seguir Ø apresentada a notação básica de um diagrama de caso de uso.

czzzncz

Caso de Uso 1

ATOR 1

Caso de Uso N

ATOR 2

Caso de Uso 2

*Notação básica do diagrama de caso de uso.*

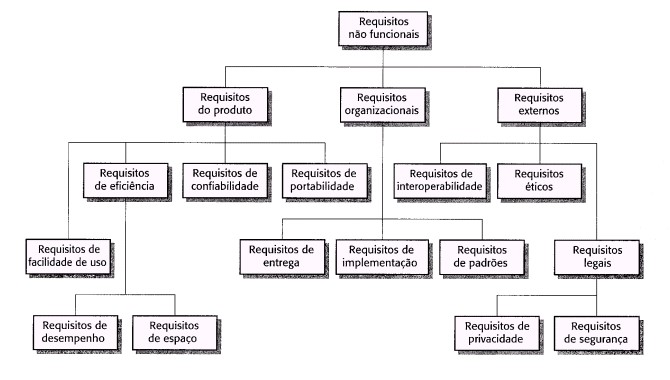
Cada ator do diagrama de caso de uso deve ser descrito de forma sucinta (2 linhas) e cada caso de uso deve ser especificado. A seguir são apresentados itens básicos para a especificação dos casos de uso do diagrama.

* Nome do Caso de Uso
* Breve descrição
* Atores envolvidos
* Pré-condições
* Sequência de Eventos ou Fluxo Principal de Eventos
* Pós-condições
* Exceções ou Fluxo Secundário de Eventos
* Observações

Para maiores detalhes de especificação de casos de uso consultar e utilizar o modelo rup\_ucspec.dot artefato do RUP.

### 3.2. Requisitos Não-Funcionais

Neste item devem ser apresentados os requisitos não funcionais, que especificam restrições sobre os servi os ou funções providas pelo sistema. A seguir são apresentados alguns tipos de requisitos não funcionais. Para maiores detalhes de requisitos não-funcionais consultar e utilizar o modelo de documento rup\_ucspec.dot artefato do RUP.



*Sommerville, Ian. Engenharia de Software, 6“edição.*

### 3.3. Protótipo

Neste item deve ser apresentado o protótipo do sistema que consiste na interface preliminar contendo um subconjunto de funcionalidades e telas. O protótipo deve ser incrementalmente evoluindo até a concordância completa dos requisitos previstos para o sistema, de comum acordo com o usuário. O protótipo Ø um recurso que deve ser adotado como estratégia para levantamento, detalhamento, valida ªo de requisitos e modelagem de interface com o usuário (usabilidade).

As telas do sistema podem ser criadas na própria linguagem de desenvolvimento ou em qualquer outra ferramenta de desenho. Cada tela deve possuir uma descri ªo detalhada do seu funcionamento. Alguns itens importantes na descri ªo sªo:

* Objetivo da tela;
* De onde Ø chamada e que outras telas pode chamar; - Regras:
  + Dom nio (tamanho de campo, tipo de dados que aceita valor default);
  + Tipo de usuÆrios que podem acessar;
  + L gica de neg cio (campos obrigat rios, validade entre datas, preenchimento anterior de um campo para efetuar uma opera ªo, etc).

A descri ªo detalhada das telas deve registrar informa ıes que possam ser consultadas na implementa ªo do sistema, facilitando, agilizando e minimizando erros de implementa ªo e na execu ªo de testes.

**3.3.1. Diagrama de Navega ªo**

Neste item deve ser apresentada a seq Œncia de navega ªo das telas.

### 3.4. MØtricas e Cronograma

Neste item devem ser estimados os esfor os necessÆrios em termos de recursos alocados G e tempo para a obten ªo do sistema. Para realizar a estimativa, indicam-se o uso de alguma tØcnica de mØtrica, como Pontos de Fun ªo ou Pontos de Caso de Uso.

Ap s os cÆlculos de mØtricas deve-se elaborar o cronograma detalhado do sistema, que contempla todas as tarefas descritas e os recursos alocados para cada tarefa, com datas para in cio e tØrmino de cada atividade. A seq Œncia das tarefas e a divisªo entre os recursos devem ser realizadas de acordo com o processo de desenvolvimento de software escolhido para o desenvolvimento do sistema, descrito no item 1.5.

Para elabora ªo do cronograma pode-se utilizar uma ferramenta como o Microsoft Project.

## 4. AnÆlise e *Design*

Este cap tulo tem como objetivo analisar e detalhar a solu ªo do sistema de acordo com os requisitos levantados e validados no cap tulo 3. Para isso, deve-se ter uma visªo geral da arquitetura do sistema e a modelagem da solu ªo do sistema atravØs de diagramas. Para maiores detalhes pode-se consultar artefatos do RUP da fase de anÆlise e *design*.

### 4.1. Arquitetura do Sistema

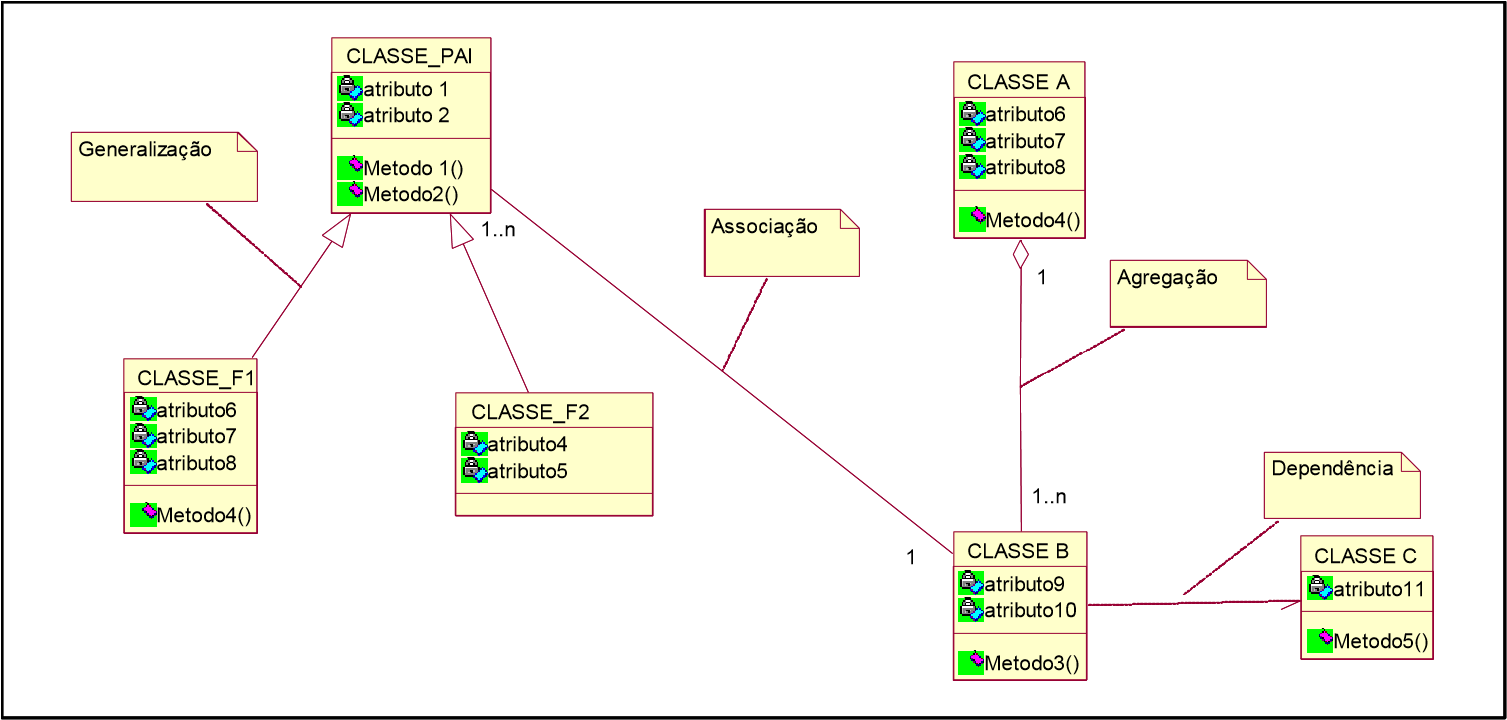
Neste item deve ser apresentada a arquitetura de infra-estrutura do sistema, demonstrando o tipo de arquitetura que serÆ utilizada (por exemplo, cliente/servidor de n-camadas), a configura ªo de hardware, de rede e de software a serem utilizados, bem como o dimensionamento m nimo de conexıes.

### 4.2. Modelo do Dom nio

Neste item deve ser apresentado o modelo do dom nio, que representa um primeiro modelo conceitual do diagrama de classes. Posteriormente, esse diagrama deve ser validado e complementado para compor o diagrama de classes final.

O diagrama de classes deve possuir todas as classes identificadas do sistema, deve conter os atributos e mØtodos de cada classe, e os relacionamento entre elas.

A seguir Ø apresentada a nota ªo bÆsica de um diagrama de classes



*Nota ªo bÆsica do diagrama de classes.*

### 4.3. Diagramas de Intera ªo

O diagrama de intera ªo Ø composto pelos diagramas de seq Œncia e colabora ªo (comunica ªo, versªo 2.0 UML) e modela os aspectos din micos do sistema, mostrando a intera ªo formada por um conjunto de objetos permitindo identificar mensagens que poderªo ser enviadas entre eles.

**4.3.1. Diagrama de Seq Œncia**

Neste item devem ser apresentados os diagramas de seq Œncia essenciais ao sistema. Um diagrama de seq Œncia representa intera ıes de objetos organizadas em uma seq Œncia temporal, apresentando os objetos que participam da intera ªo e a seq Œncia das mensagens trocadas. O diagrama de seq Œncia deve validar o diagrama de classes e vice-versa. A seguir Ø apresentada a nota ªo bÆsica de um diagrama de seq Œncia.

: ATOR 1

Objeto : CLASSE B

Mensagem

Retorno Mensagem

*Nota ªo bÆsica do diagrama de seq Œncia.*

**4.3.2. Diagrama de Colabora ªo / Comunica ªo**

Esse diagrama Ø uma alternativa para o diagrama de seq Œncia (item 4.3.1). Neste item devem ser apresentados os diagramas de colabora ªo/comunica ªo essenciais ao sistema. Um diagrama de colabora ªo descreve um padrªo de intera ªo entre objetos, apresentando os objetos que participam da intera ªo bem como os seus links e mensagens trocadas.

Geralmente as ferramentas CASE geram automaticamente o diagrama de colabora ªo/comunica ªo a partir do diagrama de seq Œncia.

A seguir Ø apresentada a nota ªo bÆsica de um diagrama de colabora ªo/comunica ªo.

: ATOR 1

Objeto :

CLASSE B

1:

Mensagem

2:

Retorno Mensagem

*Nota ªo bÆsica do diagrama de colabora ªo.*

**4.4. Diagrama de Classes**

Neste item deve ser apresentado o diagrama de classes completo e validado.

### 4.5. Diagrama de Atividades

Neste item deve ser apresentado o diagrama de atividades, que representa o detalhamento de tarefas e o fluxo de uma atividade para outra de um sistema.

Nem todos os sistemas necessitam da elabora ªo do diagrama de atividades, pois nem todas as tarefas do sistema necessitam de um detalhamento. Com isso, deve-se analisar a real necessidade e no que este diagrama irÆ auxiliar na implementa ªo do sistema, como: detalhamento de *workflow*, de mØtodos, entre outros.

A seguir Ø apresentada a nota ªo bÆsica de um diagrama de atividades.

Atividade 2

Atividade 1

Evento 1

Incio

Evento 2

Tomada de decisªo 1

Fim

Atividade 4

Evento 6

Atividade 3

Tomada de decisªo 2

Evento 3

Evento 4

*Nota ªo bÆsica do diagrama de atividades.*

### 4.6. Diagrama de Estados

Neste item deve ser apresentado o diagrama de estados, que especifica as seq Œncias de estados pelas quais o objeto pode passar durante seu ciclo de vida em resposta a eventos.

Nem todos as classes necessitam da elabora ªo do diagrama de estados, pois nem todas as classes mudam muito de estado no seu ciclo de vida. Com isso, deve-se analisar a real necessidade desse diagrama para o desenvolvimento do sistema.

A seguir Ø apresentada a nota ªo bÆsica de um diagrama de estados.

Estado 1

Estado 2

evento a

evento b

evento c

evento d

Estado Inicial

do Objeto

Estado do

Objeto

Evento

Ocorrido que

faz o objeto

mudar de

estado

Estado Final

do Objeto

*Nota ªo bÆsica do diagrama de estados.*

### 4.7. Diagrama de Componentes

Neste item deve ser apresentado o diagrama de componentes que apresenta a organiza ªo e as dependŒncias entre os componentes **G**.

A seguir Ø apresentada a nota ªo bÆsica de um diagrama de componentes.

Componente 1

Componente 2

*Nota ªo bÆsica do diagrama de componentes.*

### 4.8. Modelo de Dados

**4.8.1. Modelo L gico da Base de Dados**

Neste item deve ser apresentado o modelo l gico da base de dados, que pode ser o modelo entidade-relacionamento ou objeto da base de dados. No caso do modelo entidade-relacionamento o modelo l gico deve passar por todas as regras de normaliza ªo.

Como base para gera ªo do modelo l gico pode-se utilizar o diagrama de classes. Geralmente ferramentas CASE geram automaticamente o modelo l gico da base de dados a partir do diagrama de classes.

**4.8.2. Cria ªo F sica do Modelo de Dados**

Neste item deve ser realizada a cria ªo f sica do banco de dados, ou seja, a cria ªo de *scripts*.

**4.8.3. DicionÆrio de Dados**

Neste item deve ser criado o dicionÆrio de dados do banco de dados, com o obejtivo de documentar todas as tabelas, atributos, *stored procedures* G.

### 4.9. Ambiente de Desenvolvimento

Neste item devem ser apresentados os softwares de desenvolvimento (linguagem de programa ªo, banco de dados, ferramentas, etc.), equipamentos de hardware e redes que sejam essenciais para o desenvolvimento do sistema.

### 4.10. Sistemas e componentes externos utilizados

Neste item devem ser descritos os sistemas e componentes externos que serªo utilizados no sistema. Por exemplo, sistemas que serªo integrados ao sistema desenvolvido, componentes comprados ou livre que estªo sendo utilizados para facilitar ou complementar o desenvolvimento do sistema.

## 5. Implementa ªo

Este cap tulo tem como objetivo a implementa ªo das classes em termos de componentes, ou seja, toda a implementa ªo deve ser realizada de acordo com as defini ıes das fases anteriores e todos os recursos da programa ªo orientada a objetos que a linguagem escolhida oferece.

Geralmente ferramentas CASE geram automaticamente pseudoc digos fontes (dependendo da linguagem utilizada) baseados no diagrama de classes.

Algumas boas prÆticas de programa ªo devem ser seguidas para um maior entendimento do c digo. Algumas delas sªo:

* Cabe alho de fun ıes contendo campos como descri ªo, data de cria ªo, autor, etc;
* ComentÆrios no c digo;
* Padroniza ªo de nomes de variÆveis, par metros, fun ıes, tabelas, *stored procedures*, etc;
* Verifica ªo de declara ªo das variÆveis;
* Tratamento de erros;
* Cria ªo de *stored procedures* para acesso aos dados da base;
* Utiliza ªo de padrıes de projeto G;
* Otimiza ªo do c digo, utilizando os melhores algoritmos e fun ıes de recursividade.

## 6. Testes

Este cap tulo tem como objetivo identificar defeitos no sistema, validar as fun ıes do sistema, verificar se os requisitos foram implementados de forma adequada e avaliar a qualidade do software. Para maiores detalhes pode-se consultar artefatos do RUP da fase de Testes.

### 6.1. Plano de Testes

Neste item deve ser criado o plano de testes do sistema, permitindo a valida ªo do sistema por parte do desenvolvedor, atravØs da verifica ªo dos requisitos do sistema desenvolvido. Inicialmente, identificam-se os requisitos tØcnicos e funcionais do sistema, e listam-se todas as situa ıes que podem ocorrer com o sistema (essas situa ıes podem ser elaboradas atravØs do diagrama de caso de uso e dos diagramas de seq Œncia). Deve-se realizar testes de consistŒncia de campos, funcionalidades, desempenho, etc. O Plano de Testes do Sistema deverÆ conter, no m nimo.

* N” do Teste;
* Descri ªo do Teste; ! Resultado Esperado**.**

Por conter todos os testes do sistema, este plano poderÆ ser um anexo na documenta ªo do sistema. Alguns tipos de testes a serem realizados sªo: teste de funcionalidades, teste de usabilidade, teste de desempenho, teste de carga, teste de *stress*, teste de volume, teste de seguran a e controle de acesso, teste de toler ncia a falhas e recupera ªo, teste de configura ªo, teste de instala ªo, etc.. Para maiores detalhes consultar o modelo de documento de plano de testes do RUP rup\_tstpln.dot.

### 6.2. Execu ªo do Plano de Testes

Neste item devem ser registrados os testes realizados no sistema tendo como base o Plano de Testes do Sistema.

O registro dos testes deve conter a identifica ªo do sistema, o nome do realizador dos testes e a configura ªo do ambiente onde foi realizado o teste. AlØm disso, para cada teste, deve-se ter os seguintes dados:

* N” do teste;
* Resultado Obtido; e ! ComentÆrios (se necessÆrio).

## 7. Implanta ªo

Este cap tulo tem como objetivo apresentar informa ıes relevantes para a implanta ªo e funcionamento do sistema.

### 7.1. Diagrama de Implanta ªo

Neste item deve ser apresentado o diagrama de implanta ªo que representa a parte f sica do sistema, exibindo os dispositivos, as mÆquinas de processamento em tempo de execu ªo e os componentes que nelas serªo instalados.

A seguir Ø apresentada a nota ªo bÆsica de um diagrama de implanta ªo.

Dispositivo

Processador

Processador

Processador

*Nota ªo bÆsica do diagrama de implanta ªo.*

### 7.2. Manual de Implanta ªo

Neste item deve ser elaborado o manual de instala ªo. Este manual deve conter a descri ªo passo a passo de como deve ser realizada a instala ªo do sistema.

Para maiores detalhes pode-se consultar artefatos do RUP da fase de Instala ªo.

## 8. Manual do UsuÆrio

Este cap tulo tem como objetivo a elabora ªo de um manual do usuÆrio. Este manual deve conter a descri ªo passo a passo de como utilizar o sistema. Para maiores detalhes pode-se consultar artefatos do RUP da fase de Instala ªo.

## 9. Conclusıes e Considera ıes Finais

Este cap tulo tem como objetivo apresentar e demonstrar a aplicabilidade dos resultados obtidos, suas limita ıes, inova ªo, poss veis integra ıes com outros projetos e continua ªo do sistema em trabalhos futuros.

## Bibliografia

Neste item devem-se apresentar todas as obras (livros, artigos, Internet, revistas, etc...) utilizadas na elabora ªo da documenta ªo e na implementa ªo do projeto.

# ComentÆrios sobre a Documenta ªo

Esta Ø uma parte que responde a algumas dœvidas freq entes referente documenta ªo. Caso o leitor esteja utilizando esse documento para elabora ªo do TG, leia com a aten ªo o item 1, 2 e 3 desta se ªo.

## ComentÆrios sobre a documenta ªo

1. **Como utilizar o produto de documenta ªo no TG?**

Esse documento tem como objetivo fornecer um roteiro que auxilie os alunos no desenvolvimento de software orientado a objetos, utilizando nota ªo UML. Com isso, o roteiro deve ser algo flex vel e adaptÆvel a cada projeto. Cada grupo deve analisar e decidir em conjunto com o seu professor orientador os itens do documento que se aplicam para o projeto escolhido e se nªo hÆ necessidade de criar ou substituir itens, informa ıes e diagramas complementares ao roteiro.

AlØm da aplicabilidade dos itens do documento, cada grupo deve decidir com o seu orientador o processo a ser utilizado ao longo do desenvolvimento, as etapas que devem ser cumpridas em cada reuniªo de orienta ªo e o tipo de modelagem e implementa ªo a ser utilizada. O desenvolvimento orientado a objetos Ø sugerido pela tendŒncia de mercado, caso a escolha seja por um desenvolvimento estruturado ver item 2 desta se ªo.

1. **No meu TG irei utilizar Modelagem Estruturada, qual Ø a diferen a?** 
   1. diferen a bÆsica estÆ nos diagramas e na implementa ªo. A seguir sªo apresentados os itens que podem ser substitu dos no roteiro e qual seria um item correspondente para a modelagem estruturada.
      * **Item 3.1. Requisitos Funcionais:** apesar do diagrama de caso de uso ser da modelagem orientada a objetos, este pode ser utilizado na modelagem estruturada para identificar os usuÆrios, os sistemas externos e as fun ıes do sistema. Caso o diagrama de caso de uso nªo seja utilizado deve-se realizar uma descri ªo textual e detalhada de cada requisito;
      * **Item 4.2 e 4.4. Diagrama de Classes:** deve-se substituir o diagrama de classes pelo diagrama de fluxo de dados (DFD) n vel 0 (diagrama de contexto), n vel 1, n vel 2,...n vel n se necessÆrio.
      * **Item 4.3. Diagrama de Intera ªo:** devem-se substituir os diagramas de intera ªo pelo dicionÆrio de dados dos diagramas de fluxos de dados;
      * **Item 4.5. Diagrama de Atividades:** apesar do diagrama de atividades ser da modelagem orientada a objetos, este pode ser utilizado na modelagem estruturada para o detalhamento de fun ıes complexas do sistema. Isso tambØm pode ser realizado atravØs de fluxogramas, diagramas de Nassi-Schneiderman, portugol, etc;
      * **Item 4.7. Diagrama de Componentes:** deve-se substituir o diagrama de componentes pelo diagrama hierÆrquico de fun ıes (DHF)
      * **Item 5. Implementa ªo:** deve-se utilizar toda a modelagem criada para implementa ªo do sistema utilizando os recursos da programa ªo estruturada;
      * **Item 6.1. Plano de Teste:** o plano de teste deve seguir o item 6.1, mas as situa ıes de teste podem ser extra das dos diagramas de fluxo de dados (DFD).
      * **Item 7.1. Diagrama de Implanta ªo:** apesar do diagrama de implanta ªo ser da modelagem orientada a objetos, este pode ser utilizado na modelagem estruturada para representar a parte f sica do sistema.

Para maiores detalhes sobre Modelagem Estruturada consultar o livro: AnÆlise Estruturada Moderna. Edward Yourdon. Editora Campus, 3“ edi ªo.

1. **No meu TG haverÆ um estudo te rico alØm do sistema, qual Ø a diferen a?** 
   1. principal diferen a Ø cria ªo de um cap tulo intermediÆrio entre o Cap tulo 1 e 2, que deve apresentar todo o estudo te rico da pesquisa. Esse cap tulo deve conter basicamente os itens a seguir, mas cabe ao professor orientador definir os itens a serem criados no estudo te rico.
      * Apresenta ªo do Trabalho Te rico;
      * Justificativa do Trabalho Te rico e a rela ªo com o projeto a ser desenvolvido;
      * Disserta ªo do Trabalho Te rico;
      * AnÆlise dos resultados/Conclusªo do Trabalho Te rico.

1. **Qual Ø o material que posso consultar caso tenha dœvidas?** 
   * + FOWLER, M.; SCOTT.K. **UML Essencial**. Editora Bookman, 3“ edi ªo.
     + BOOCH, G.; RUMBAUGH, J.; JACOBSON. I. **UML Guia do UsuÆrio**. Editora Campus, 2000.
     + BOGGS, W.; BOGGS, M. **Mastering UML com Rational Rose 2002**. Editora Alta Books, 2002.
     + QUATRANI. T. **Modelagem Visual com Rational Rose 2000 e UML**. Editora CiŒncia Moderna, 2001.
     + PRESSMAN, R. Software Engineering. A Practitioner·s Approach. 5“ edi ªo, 2003. McGrawHill.
     + SOMERVILLE, I. **Engenharia de Software**. Addison Wesley, 6“ edi ªo.
     + YOURDON, E **AnÆlise Estruturada Moderna**. Editora Campus, 3“ edi ªo.
     + Ferramenta *Rational Unified Process*.
     + Site: http://prof.usjt.br/anapaula (disciplinas de Metodologia de Desenvolvimento de Software e Engenharia de Software)
2. **Quais sªo as Ferramentas CASE que podem ser utilizadas?**

! *Rational Suite Enterprise*

* + - * *Rational Rose* para modelagem de sistemas orientados a objetos;
      * *Rational Unified Process* - RUP para utiliza ªo do processo de desenvolvimento e para consulta de artefatos, defini ıes, etc; # *Requisite Pro* para cria ªo e gerenciamento de requisitos.
      * AlØm de outras ferramentas da Su te...
    - *System Architect* para modelagem de sistemas orientados a objetos e estruturados.
    - *Microsoft Project* para elabora ªo de cronograma, planejamento e gerenciamento do projeto.
    - *Erwin* para elabora ªo do modelo l gico e f sico do banco de dados; ! *ArgoUML* para modelagem de sistemas orientados a objetos.
    - *Dome* para modelagem de sistemas orientados a objetos.
    - Outras....

# GlossÆrio

## GlossÆrio

* **Componente:** representa uma parte f sica da implementa ªo de um sistema, que inclui c digo de software, com o objetivo de criar c digo de software coeso para sua reutiliza ªo e facilidade de manuten ªo.

* **Ferramenta CASE (*Computer Aided Software Engineering*):** Ø uma ferramenta que auxilia no processo de desenvolvimento de software, ajudando a garantir a qualidade do projeto e facilitando a cria ªo de modelos, documentos.

* **Padrıes de Projeto (*design patterns*):** sªo solu ıes simples para problemas espec ficos no projeto de software orientado a objetos. Padrıes de projeto capturam solu ıes que foram desenvolvidas e aperfei oadas ao longo do tempo.

* **Recursos alocados:** pessoas que irªo trabalhar no projeto.

* **Regras de neg cio**: declara ıes e regras da pol tica ou condi ªo que deve ser satisfeita no mbito do neg cio.

* **Requisito:** um requisito descreve uma condi ªo ou capacidade qual um sistema deve se adaptar, sejam necessidades dos usuÆrios, um padrªo ou uma especifica ªo.

* ***Stored Procedures*:** Ø uma rotina escrita atravØs de comandos SQL, que tem como objetivo encapsular o processo de neg cio e sua reutiliza ªo. As *stored procedures* ficam armazenadas no gerenciador de banco de dados.

* **Usabilidade:** Ø a qualidade da interface homem-mÆquina, que permite que o usuÆrio realize com eficiŒncia e conforto as atividades a que o sistema se destina.