



**MINISTÉRIO DOS TRANSPORTES**

DEPARTAMENTO NACIONAL DE INFRA-ESTRUTURA DE TRANSPORTES

DIRETORIA DE ADMINISTRAÇÃO E FINANÇAS - DAF

COORDENAÇÃO-GERAL DE MODERNIZAÇÃO E INFORMÁTICA – DAF

# INSTRUÇÃO DE SERVIÇO PARA DOCUMENTAÇÃO DE PLANO DE DESENVOLVIMENTO DE APLICATIVOS

**IS-CGMI-01/2005**

**Aprovada pela Portaria nº 1493 de 22/11/2005**

## Histórico de Versões

<b>Data</b>	<b>Versão</b>	<b>Descrição</b>	<b>Autor</b>	<b>Revisor</b>	<b>Aprovado por</b>
<dd/mm/aaaa>	<x.x>	<detalhes>	<nome>	<nome>	<nome>

**Setembro-2005**

# SUMÁRIO

1. Prefácio.....	4
2. Objetivo.....	5
3. Definições.....	6
4. Visão do Projeto.....	8
4.1. Objetivo.....	10
4.2. Posicionamento.....	10
4.2.1. Cenário.....	10
4.2.2. Oportunidade de Negócio.....	10
4.2.3. Descrição do Projeto.....	10
4.3. Envolvimento.....	10
4.3.1. Abrangência.....	10
4.3.2. Papel das Partes Interessadas.....	10
4.3.3. <nome da parte interessada>.....	11
4.4. Papel dos Atores (<opcional, caso já estejam descritos na visão do sistema>).....	11
4.4.1. <nome do ator>.....	11
4.5. Necessidades e Funcionalidades.....	11
4.6. Restrições.....	12
4.7. Expectativa de Entrega do Produto.....	12
4.7.1. Proposta de Solução Tecnológica Escolhida.....	13
5. Cronograma.....	14
5.1. Objetivo.....	15
5.2. Relação de Cronogramas.....	15
5.2.1. Acompanhamento do projeto.....	15
5.2.2. Fases de acompanhamento.....	15
5.2.3. Detalhamento do Projeto – Fase de Iniciação.....	16
5.2.4. Detalhamento do Projeto – Fase de Elaboração.....	16
5.2.5. Detalhamento do Projeto – Fase de Construção.....	16
5.2.6. Detalhamento do Projeto – Fase de Transição.....	16
6. Documento de requisitos de software.....	18
6.1. Objetivo.....	19
6.2. Descrição Geral.....	19
7. Especificação de Requisitos Não Funcionais.....	20
7.1. Objetivo.....	21
7.2. Requisitos Não Funcionais.....	21
7.2.1. Usabilidade.....	21
7.2.2. Confiabilidade.....	22
7.2.3. Performance.....	22
7.2.4. Suportabilidade.....	22
7.2.5. Requisitos Legais.....	23
7.2.6. Segurança.....	23
7.2.7. Outros Requisitos Não Funcionais.....	23
7.2.8. Infra-estrutura Tecnológica.....	24
8. Especificação de Regras de Negócio.....	25
8.1. Objetivo.....	26
8.2. <Nome do Grupo de Regras>.....	26
8.2.1. <Nome da Regras>.....	26
8.3. <Nome de outro Grupo de Regras>.....	26

8.3.1. .n.1<Nome da Regras>.....	26
9. Diagrama de classes.....	27
9.1. Objetivo.....	28
9.2. Uso.....	28
10. Modelo de dados.....	29
10.1. Objetivo.....	30
10.2. Finalidade.....	30
10.3. Propriedades A serem entregues.....	30
11. Relação de casos de uso.....	32
11.1. Diagrama de Casos de Uso.....	33
11.2. Casos de Uso.....	33
11.2.1. <nome do caso de uso>.....	33
12. Especificação de caso de uso.....	34
12.1. Nome do Caso de Uso.....	35
12.2. Objetivo.....	35
12.3. Tipo de Caso de Uso <opcional>.....	35
12.4. Atores.....	35
12.5. Pré-Condições <opcional>.....	35
12.6. Fluxo Principal.....	36
12.7. Fluxos Alternativos <opcionais>.....	36
12.8. Fluxos de Exceção <opcional>.....	36
12.9. Pós-condições <opcional>.....	36
12.10. Requisitos Não Funcionais <opcional>.....	36
12.11. Ponto de Extensão <opcional>.....	37
12.12. Critérios de Aceite.....	37
12.13. Frequência de Utilização.....	37
12.14. Observações <opcionais>.....	37
12.15. Referências <opcionais>.....	37
12.16. Interface do usuário <opcional>.....	37
13. Orçamento.....	38
13.1. Objetivo.....	39
14. Glossário.....	40
14.2. <Definição dos termos>.....	41
15. Termo de aprovação de especificação de sistema.....	42
.....	42
15.1. <Definição dos termos>.....	43

## 1. PREFÁCIO

A presente Instrução de Serviço foi elaborada pela CGMI/DNIT para servir como documento básico na elaboração de Documentação de Plano de Desenvolvimento de Aplicativos.

Está baseada na metodologia **RUP (Rational Unified Process)**, customizada para atender as necessidades do DNIT, com uma abordagem voltada para utilização das “melhores práticas” do RUP.

As atividades foram grupadas em disciplinas para distribuição de tarefas e responsabilidades, visando um melhor entendimento do processo desenvolvimento de aplicativos no DNIT.

Recursos de Qualidade de Sistemas, **CMM (Capability Maturity Model) nível 2**, foram utilizados, com a adoção de metodologia baseada em Análise por Pontos de Função para cálculo de custos e prazos, sendo adotado padrões definidos pelo **IFPUG – International Function Point Users Group**.

## 2. OBJETIVO

Esta Instrução de Serviço fixa conceitos e estabelece procedimentos no âmbito do DNIT, para preparação, apresentação, e alteração de Documentação de Plano de Desenvolvimento de Aplicativos.

Sua meta é garantir a produção de software de alta qualidade, atendendo às necessidades dos usuários dentro de um cronograma e de um orçamento previsível e apoiar ao Gerenciamento de Projetos, compartilhando os mesmos padrões e princípios estabelecidos pelo **PMI (Project Management Institute)**.

Está previsto uma evolução contínua na documentação, e ela pode ser apresentada para atender dois tipos de demandas:

- a) **Documentação de Aplicativos Novos** – Classificada como o processo iniciado com a aprovação da especificação de um projeto novo, sua evolução de desenvolvimento com as iterações nas diversas fases propostas, até sua homologação como processo concluído.
- b) **Documentação de Aplicativos Existentes** – Classificada como o processo iniciado com a aprovação da especificação de um projeto de documentação do aplicativo, sua execução, até sua homologação como processo concluído.

Esta demanda pode prever documentação do **Refatoramento**, ou seja, pequenos ajustes no código do sistema, que não altere o seu comportamento funcional, mas que melhore algumas qualidades não-funcionais, tais como: simplicidade, flexibilidade, desempenho, clareza do código e melhor integração com a metodologia adotada pelo DNIT.

### 3. DEFINIÇÕES

Para efeito desta Instrução de Serviço, aplicam-se as seguintes definições:

[1] Análise de riscos - Tarefa que permite, depois de identificados, analisar os riscos e buscar uma solução para os mesmos, assim como alocar responsáveis para solucioná-los.

[2] Arquitetura - Estrutura e organização lógica de funcionamento de um sistema computacional. A Arquitetura do sistema é a sua organização em grandes módulos.

[3] Artefato - Documento gerado durante o processo, que pode ser um arquivo de código-fonte, uma parte da documentação do sistema, um projeto arquitetural, etc. Alguns artefatos são mantidos durante todo o projeto, enquanto outros apenas durante uma certa etapa do projeto.

[4] Escopo - Abrangência do domínio do problema.

[5] Glossário - Artefato gerado no intuito de esclarecer termos não familiares à equipe de desenvolvimento ou ao próprio cliente. Possibilita uma melhor comunicação entre os envolvidos no projeto, evitando dúvidas no entendimento de aspectos do problema.

[6] Iteração - Período de tempo para implementação de tarefas. Várias iterações compõem um release.

[7] Modelo Lógico de Dados - Modelo que representa a estrutura lógica, relativa ao banco de dados de um sistema.

[8] Mudanças de requisitos - São mudanças em características do sistema que afetam diretamente o desenvolvimento do mesmo. Podem acontecer devido a modificações de negócio ou por decisão do cliente. É muito importante que a equipe saiba como lidar com esse tipo de mudanças, uma vez que elas podem representar um grande risco para o sucesso do projeto.

[9] Padrões de projeto - Descrição das melhores práticas e bons desenvolvedores para prover soluções mais robustas a problemas já conhecidos.

[10] Papel - Um papel constitui um conjunto de responsabilidades. Determinam quais serão os comportamentos dos membros da equipe.

[11] Processo de negócio - Compreende um conjunto de atividades realizadas na empresa, associadas às informações que manipula, utilizando os recursos (técnicas, métodos, ferramentas, sistemas de informação, recursos financeiros, etc) e a organização da empresa. Normalmente está direcionado a um determinado mercado/cliente, com

fornecedores bem definidos.

[12] Produto - Objeto concreto obtido ao final do processo de desenvolvimento, representando o que foi proposto pela equipe inicialmente. Inclui aspectos tais como manual de instruções, help on-line e programa de instalação (ou guia de instalação). Deve satisfazer os requisitos propostos pelo cliente e usuário.

[13] Projeto Arquitetural - Artefato cujo objetivo é descrever o funcionamento do sistema a ser desenvolvido num nível alto de abstração. Deve explicitar como as partes do sistema interagem entre si.

[14] Refatoramento - Prática de reestruturar um código que já está funcionando sem alterar funcionalidade.

[15] Release - Período de tempo para implementação de uma versão funcional do sistema. É dividido em iterações.

[16] Reunião de acompanhamento - Reunião semanal que visa recolher e analisar métricas e riscos, bem como o andamento geral do projeto.

[17] Revisão de código - Consiste da revisão interna por parte de um testador, do código produzido por um desenvolvedor, desde que o testador e o desenvolvedor que gerou o código sejam pessoas diferentes. Prima pela qualidade, capacidade e clareza do código.

[18] Status - Situação ou estado de alguma tarefa em uma determinada iteração.

[19] Testes - Verificação automática de código, abordando se a funcionalidade atende ao especificado anteriormente. É uma análise de pré-condições e pós-condições dentro de um determinado ambiente de configuração.

[20] Testes de aceitação - Testes de aceitação são testes funcionais executáveis definidos pelo cliente a partir de uma conversa. É testada toda uma funcionalidade definida pelo cliente.

[21] Testes de Unidade - São testes para classes e métodos individuais. São sempre automatizados. Em um projeto devem sempre estar rodando 100% para todas as classes testadas.

[22] Termo de aprovação de especificação de sistema – Documento assinado pelo DNIT e empresa contratada, concordando com os termos de conteúdo, prazo, preço e condições para execução dos serviços propostos.

#### 4. VISÃO DO PROJETO

##### Histórico de Versões

<b>Data</b>	<b>Versão</b>	<b>Descrição</b>	<b>Autor</b>	<b>Revisor</b>	<b>Aprovado por</b>
<dd/mm/aaaa>	<x.x>	<detalhes>	<nome>	<nome>	<nome>

# Responsáveis

<Incluir na ficha técnica abaixo os níveis organizacionais de acordo com o projeto. Os itens desta ficha técnica podem ser alterados para se adequarem à estrutura da unidade de gestão.>

## 1. DNIT

### **Diretor de Administração e Finanças - DAF**

<informar o nome>

### **Coordenador-Geral de Modernização e Informática**

<informar o nome>

### **Gestor**

<informar o nome>

### **Coordenação-Geral Usuária**

<informar o nome>

### **Gestor Usuário**

<informar o nome>

## 2. Empresa Prestadora de Serviço

<informar o nome>

### **Presidente**

<informar o nome>

### **Diretor Superintendente**

<informar o nome>

### **Superintendência de Negócio**

<informar o nome>

### **Coordenador de Negócio**

<informar o nome>

### **Gerente de Projeto**

<informar o nome>

### **Líder de Projeto**

<informar o nome>

### **Desenvolvimento**

<informar o nome>

## 4.1.OBJETIVO

O propósito deste documento é coletar, analisar e definir as necessidades de alto-nível e características do projeto de software <nome do projeto de software e sigla>, focando nas potencialidades requeridas pelos afetados e usuários-alvo, e como estes requisitos serão abordados no projeto de software.

A visão do projeto documenta o ambiente geral de processos a ser desenvolvido para o sistema durante o projeto, fornecendo a todos os envolvidos uma descrição compreensível deste e suas macro-funcionalidades.

O Documento de Visão do Projeto documenta apenas as necessidades e funcionalidades do sistema que estarão sendo atendidas no projeto de software.

## 4.2. POSICIONAMENTO

### 4.2.1.Cenário

<Descrição sucinta do cenário atual que motiva a construção da ferramenta.>

### 4.2.2. Oportunidade de Negócio

<Descrição sucinta de oportunidades de negócio que o projeto proporciona, em virtude do cenário atual.>

### 4.2.3. Descrição do Projeto

<Descrever as principais funcionalidades disponibilizadas pelo produto para o projeto, dando uma visão geral dos processos que permeiam seu funcionamento.>

## 4.3. ENVOLVIMENTO

### 4.3.1. Abrangência

<Descrever o alcance do projeto no contexto dos seus usuários e demais envolvidos.>

### 4.3.2. Papel das Partes Interessadas

<Qualquer pessoa afetada ou significativamente interessada no produto de um projeto de software, ou em algum aspecto específico deste. As partes interessadas constituem o grupo de indivíduos diretamente envolvido com as atividades de definição, desenvolvimento e validação do sistema, e podem contribuir de inúmeras maneiras para o projeto, tais como fornecendo recursos, provendo requisitos e/ou feedback no uso do sistema. Clientes, Tomadores de Decisão, Donos do Negócio, Equipes de Desenvolvimento, dentre outros, são exemplos comuns de partes interessadas.

Algumas questões podem ser úteis na identificação das partes interessadas:

- Quem é o cliente (financiador do projeto)?

- Quem define os requisitos do projeto de software?
- Quem influencia o desenvolvimento do projeto de software?
- Quem irá avaliar e homologar o sistema quando este estiver pronto?>

#### 4.3.3. <nome da parte interessada>

<Descrever todas as partes interessadas. >

Descrição	<breve descrição da parte interessada.>
Papel no desenvolvimento	<descrever brevemente a importância e o papel desempenhado pela parte interessada no projeto de software.>
Insumos ao projeto de software	<listar os produtos a serem entregues para servir de insumo ao desenvolvimento do projeto de software, caso existam.>
Representante	<identificação da(s) pessoa(s) e respectivo(s) órgão(s) que representa(m) a parte interessada.>

#### 4.4. PAPEL DOS ATORES (<opcional, caso já estejam descritos na visão do sistema>)

<Uma entidade humana ou não-humana que interage diretamente com o sistema. Atores não-humanos pode ser hardware ou outros sistemas. Um ator participa em um cenário enviando ou recebendo mensagens do sistema.

Algumas questões podem ser úteis na identificação de atores de um sistema:

- Quem fornecerá, usará ou recuperará informações do sistema?
- Quem irá operar o sistema?
- De onde o sistema obtém as informações necessárias à sua operação?
- Quais sistemas externos interagem com o sistema?>

##### 4.4.1.<nome do ator>

<Descrever todos os atores.>

Descrição	<breve descrição do ator.>
Papel	<descrever brevemente a importância e o papel desempenhado pelo ator no sistema.>
Insumos ao sistema	<listar os produtos e/ou informações a serem entregues pelo ator como insumo ao funcionamento do sistema, caso existam.>
Representante	< identificação da(s) pessoa(s) e respectivo(s) órgão(s) que representa(m) o ator (ou grupo de atores).>

#### 4.5. NECESSIDADES E FUNCIONALIDADES

<Descrever, no quadro abaixo, as principais necessidades e funcionalidades requeridas pelas partes interessadas para o projeto, relacionando a cada uma delas os atores envolvidos.>

Necessidade <n>		Benefício
<Descrição da necessidade.>		<Crítico, Importante ou Útil>
Id Func.	Descrição das Funcionalidades/atores envolvidos	
F<n.m>	<Descrever a funcionalidade que dá suporte à consecução da necessidade.>	
	<Nome do ator envolvido com a funcionalidade. O nome do ator deve estar em negrito e definido no Documento de Visão do Sistema.>	
F<n. m+1>	<nova funcionalidade>	
	<novo(s) ator(es)>	

#### 4.6. RESTRIÇÕES

<Relacionar os fatores que afetam a perfeita implementação das características propostas para o sistema, descrevendo premissas assumidas; restrições à modelagem, implantação e operacionalização do sistema; e dependências que estes fatores geram em relação ao desenvolvimento do aplicativo, explicitando os riscos e impactos envolvidos.>

Exemplos de Restrições:

- Funcionais/Negócio - Relacionar as restrições funcionais do sistema. (Ex.: “O sistema não deve permitir a visualização dos dados cadastrais do contribuinte, caso a declaração esteja cancelada, ou o usuário não pertença à UA de arquivamento da declaração.”)
- Tecnológicas - Relacionar as restrições tecnológicas do sistema. (Ex.: “O aplicativo somente poderá ser executado no browser Internet Explorer versão 4.0 ou superior.”)
- Operacionais - Relacionar as restrições operacionais do sistema; mapeamento de Perfil por Funcionalidade. (Ex.: “As bases de assunto e as tabelas de domínio devem estar devidamente criadas no ambiente SIEF para permitir a correta execução do sistema; a funcionalidade Baixa de Contribuinte só é permitida para o perfil X.”)>

#### 4.7. EXPECTATIVA DE ENTREGA DO PRODUTO

<Relacionar os grandes marcos de entrega de cada fase do desenvolvimento do produto, identificando os subprodutos a serem gerados. Cada item deverá conter pelo menos a descrição do marco/produto e a data acordada para a entrega.>

#### **4.7.1.Proposta de Solução Tecnológica Escolhida**

<Descrever a proposta de solução tecnológica escolhida. Este item será opcional para o caso de projetos de manutenção onde não houver mudança na solução tecnológica escolhida e já estiver sido documentada anteriormente.>

## 5. CRONOGRAMA

### Histórico de Versões

<b>Data</b>	<b>Versão</b>	<b>Descrição</b>	<b>Autor</b>	<b>Revisor</b>	<b>Aprovado por</b>
<dd/mm/aaaa>	<x.x>	<detalhes>	<nome>	<nome>	<nome>

## 5.1.OBJETIVO

<Apresentar os cronogramas do projeto elaborados no software Microsoft Project, impressos e anexados no documento, e entregues também em mídia magnética/óptica.

Os cronogramas de acompanhamento do projeto devem ser apresentados com a visão “Gráfico de Gantt” do software, sendo informadas as colunas:

- Nome da Tarefa,
- Duração (dias úteis),
- Data início da tarefa,
- Data fim da tarefa e
- Representação gráfica da tarefa.>

## 5.2.RELAÇÃO DE CRONOGRAMAS

<Devem ser entregues os seguintes cronogramas:>

### 5.2.1.Acompanhamento do projeto

<Informar:

- Projeto, duração (dias úteis), data início do projeto, data fim do projeto e representação gráfica do projeto.
- Alocação de cada profissional, sua categoria, duração (dias úteis), data início da alocação, data fim da alocação e representação gráfica da alocação.>

### 5.2.2.Fases de acompanhamento

<Informar:

- Projeto, duração (dias úteis), data início do projeto, data fim do projeto e representação gráfica do projeto.
- Fases de Iteração (Iniciação, Elaboração, Construção e Transição), duração (dias úteis), data início da fase, data fim da fase e Representação gráfica da fase.
- Alocação de cada profissional na fase, sua categoria, duração (dias úteis), data início da alocação na fase, data fim da alocação na fase e representação gráfica da alocação na fase.>

### **5.2.3.Detalhamento do Projeto – Fase de Iniciação**

<Informar:

- Projeto, duração (dias úteis), data início do projeto, data fim do projeto e representação gráfica do projeto.
- Fases de Iniciação, duração (dias úteis), data início da fase, data fim da fase e Representação gráfica da fase.
- Grupo de atividades, duração (dias úteis), data início das tarefas do grupo, data fim das tarefas do grupo e representação gráfica das tarefas do grupo.>
- Atividades, duração (dias úteis), data início das tarefas, data fim das tarefas e representação gráfica das tarefas.>

### **5.2.4.Detalhamento do Projeto – Fase de Elaboração**

<Informar:

- Projeto, duração (dias úteis), data início do projeto, data fim do projeto e representação gráfica do projeto.
- Fases de Iniciação, duração (dias úteis), data início da fase, data fim da fase e Representação gráfica da fase.
- Grupo de atividades, duração (dias úteis), data início das tarefas do grupo, data fim das tarefas do grupo e representação gráfica das tarefas do grupo.>
- Atividades, duração (dias úteis), data início das tarefas, data fim das tarefas e representação gráfica das tarefas.>

### **5.2.5.Detalhamento do Projeto – Fase de Construção**

<Informar:

- Projeto, duração (dias úteis), data início do projeto, data fim do projeto e representação gráfica do projeto.
- Fases de Iniciação, duração (dias úteis), data início da fase, data fim da fase e Representação gráfica da fase.
- Grupo de atividades, duração (dias úteis), data início das tarefas do grupo, data fim das tarefas do grupo e representação gráfica das tarefas do grupo.>
- Atividades, duração (dias úteis), data início das tarefas, data fim das tarefas e representação gráfica das tarefas.>

### **5.2.6.Detalhamento do Projeto – Fase de Transição**

<Informar:

- Projeto, duração (dias úteis), data início do projeto, data fim do projeto e representação gráfica do projeto.

- Fases de Iniciação, duração (dias úteis), data início da fase, data fim da fase e Representação gráfica da fase.

- Grupo de atividades, duração (dias úteis), data início das tarefas do grupo, data fim das tarefas do grupo e representação gráfica das tarefas do grupo.>

- Atividades, duração (dias úteis), data início das tarefas, data fim das tarefas e representação gráfica das tarefas.>

## 6. DOCUMENTO DE REQUISITOS DE SOFTWARE

### Histórico de Versões

<b>Data</b>	<b>Versão</b>	<b>Descrição</b>	<b>Autor</b>	<b>Revisor</b>	<b>Aprovado por</b>
<dd/mm/aaaa>	<x.x>	<detalhes>	<nome>	<nome>	<nome>

## 6.1.OBJETIVO

<Este documento tem por objetivo se concentrar na coleta e na organização de todos os requisitos envolvendo o projeto, visando determinar a localização e a organização correta dos requisitos de software para cada característica de um determinado release do produto.

Para tal, podem ser incluídos vários itens do modelo de casos de uso do sistema para descrever os requisitos funcionais dessa característica, juntamente com o conjunto relevante de requisitos detalhados em Especificações Suplementares.

Como você pode se deparar com diferentes ferramentas para coletar esses requisitos é importante entender que a coleta dos requisitos pode ser feita com vários e diferentes artefatos e ferramentas.

Por exemplo, talvez você ache adequado coletar requisitos textuais, como requisitos não funcionais, Restrições de Design etc., com uma ferramenta de documentação de texto nas Especificações Suplementares. Por outro lado, talvez você ache útil coletar alguns (ou todos os) requisitos funcionais nos casos de uso e ache prático usar uma ferramenta adequada às necessidades de definição do modelo de casos de uso.

O que precisamos é de um conjunto (ou seja, um pacote) de artefatos que descreva o comportamento externo completo do sistema, isto é, um pacote que diga especificamente: "Aqui está o que o sistema deve fazer para oferecer essas características".>

## 6.2.DESCRICÃO GERAL

<Esta seção do DRS deve descrever os fatores gerais que afetam o produto e seus requisitos. Ela não deve especificar requisitos específicos. Em vez disso, deve fornecer uma base para esses requisitos, que serão definidos detalhadamente na Seção 3, e facilitar sua compreensão. Inclue itens como:

- perspectiva do produto
- funções do produto
- características do usuário
- restrições
- suposições e dependências
- subconjuntos de requisitos>

## 7. ESPECIFICAÇÃO DE REQUISITOS NÃO FUNCIONAIS

### Histórico de Versões

<b>Data</b>	<b>Versão</b>	<b>Descrição</b>	<b>Autor</b>	<b>Revisor</b>	<b>Aprovado por</b>
<dd/mm/aaaa>	<x.x>	<detalhes>	<nome>	<nome>	<nome>

## 7.1.OBJETIVO

<O objetivo da especificação de requisitos não funcionais é apresentar os requisitos não contempláveis no modelo de Use Case que importem no desenvolvimento do sistema, tais como:

- a) Requisitos legais e normativos
- b) Atributos de qualidade do sistema: usabilidade, performance, suportabilidade, confiabilidade.
- c) Outros requisitos de ambiente, compatibilidade e restrições do projeto.

A relação de requisitos aqui apresentada tem o intuito de servir de auxílio no preenchimento deste documento. No entanto, ela não esgota (nem se pretende a isso) a imensa variedade de requisitos não funcionais (RNF) percebidos e mapeados na literatura. Cabe à equipe de desenvolvimento estabelecer quais os RNF mais importantes dentro do contexto de negócio do sistema-alvo, documentando-os nas seções em que melhor se encaixarem. Os requisitos podem ser inseridos como frases simples e objetivas. **A representação de identificadores em negrito, e suas descrições, é apenas uma sugestão.>**

## 7.2.REQUISITOS NÃO FUNCIONAIS

<Enumerar e descrever os requisitos não funcionais, observando os critérios abaixo relacionados:>

### 7.2.1. Usabilidade

<Relacionar os requisitos que melhor definem as facilidades de uso do software, o nível de consistência dos dados apresentados, e de documentação. Requisitos mais comuns:

- **Conformidade com padrões** – adequação do sistema a padrões internos ou externos à empresa (padrões de interface, negócio, mercado, e outros).
- **Nível de habilidade do usuário** – indica requisitos de utilização especiais de acordo com o nível dos usuários finais: sofisticados ou não; acostumados à utilização intensiva de sistemas ou iniciantes na automação de sistemas.
- **Presença de ferramentas de auxílio** – helps on-line, menus, tool tips, documentação em geral.
- **Treinamento** – necessidades de tempo e meios para a capacitação plena dos usuários na ferramenta.
- **Dicas para o usuário** – aplicação irá fornecer dicas a respeito do significado de cada campo em tela.>

### 7.2.2. Confiabilidade

<Relacionar os requisitos que melhor definem a confiabilidade no software. Engloba aspectos como previsibilidade, acurácia de resultados, resistência a falhas, recuperabilidade, entre outros. Requisitos mais comuns:

- a. **Integridade dos dados** – medidas necessárias à preservação da integridade dos dados no banco como um todo.
- b. **Disponibilidade** – especificar o percentual de tempo disponível do sistema, em horas de uso, manutenção.
- c. **Medidas de tempo entre falhas** – especificar em horas, dias ou mês qual o tempo que pode existir entre uma falha e outra.
- d. **Medidas de tempo de reparo** – tempo que o sistema pode permanecer indisponível após uma falha.
- e. **Máximo de defeitos ou taxa de defeitos** – pode ser expresso em termos de defeitos / linhas de código ou defeitos/ponto de função, estes defeitos devem ser categorizados em termos de defeitos secundários, significantes ou críticos.
- f. **Rotinas operacionais** – definir necessidade de backup, janela de tempo, carga, etc. >

### 7.2.3. Performance

<Relacionar os requisitos que melhor definem a performance do software. Engloba considerações sobre tempo de resposta, velocidade de processamento, eficiência, consumo de recursos e volume de produção. Requisitos mais comuns:

- a. **Tempo de resposta para uma transação.**
- b. **Numero de usuários do sistema, distribuído ao longo do tempo.**
- c. **Estimativas de transações com o banco de dados.**
- d. **Quantidade de acessos simultâneos.**
- e. **Quantidade de transações por segundo.**
- f. **Capacidade** – quantidade de usuários ou transações que o sistema pode acomodar.
- g. **Área de Armazenamento** – quantidade de área de armazenamento dos dados do sistema.>

### 7.2.4. Suportabilidade

<Relacionar os requisitos que melhor definem a capacidade do sistema de suportar mudanças, evoluções e reparos. Definem a testabilidade, extensibilidade, adaptabilidade, manutenibilidade, compatibilidade, entre outros. Requisitos mais comuns:

- a. **Padrões de programação** – observar as orientações técnicas.
- b. **Padrões gerais** – quaisquer padrões ou leiautes, internos ou externos à organização, que influenciem no desenvolvimento do aplicativo, exceto

aqueles que regulem requisitos de interface – já referenciados em usabilidade.

- c. **Ferramentas de avaliação de impacto**
- d. **Características de extensibilidade de linguagem adotada**
- e. **Facilidades de instalação**
- f. **Elaboração e Distribuição de novas versões**
- g. **Interoperabilidade** – estabelece regras de inter-relacionamento entre sistemas afins
- h. **Mídia de Armazenamento** – é obrigatório à descrição da mídia de armazenamento.>

### 7.2.5. Requisitos Legais

<Relacionar os requisitos não funcionais derivados da legislação que regula a construção do sistema, que restringem ou controlam de alguma maneira o seu desenvolvimento. >

### 7.2.6. Segurança

- ✓ <Relacionar os requisitos que definem a política de segurança adotada para a aplicação.>
- ✓ **Mecanismos ou sistemas de controle de acesso** – relacionar os mecanismos ou sistemas que serão utilizados, por exemplo, Senha-Rede, Senha-SIEF, etc.
- ✓ **Transações e perfis de acesso** – relacionar as transações e perfis definidos, e o grau de restrição ao acesso a funcionalidades e dados controlados pela aplicação.
- ✓ **Sigilo** – definir critérios e definir os procedimentos que garantam o sigilo do sistema a ser desenvolvido.
- ✓ **Auditoria** – definir critérios e procedimentos necessários à auditoria, indicando transações e dados do sistema demandados a serem auditados. Em relação aos dados de auditoria definir a forma, sistema e/ou ferramentas de acesso, o tempo de retenção, o meio físico de guarda, os mecanismos ou sistemas de controle de acesso e os perfis e restrições de acesso.

### 7.2.7. Outros Requisitos Não Funcionais

- a. <Relacionar outros requisitos identificados para a aplicação, que não estejam mapeados dentro da classificação adotada no corpo do documento. Alguns possíveis exemplos são:
- b. **Condições de entrega** – restrições e acordos estabelecidos para que a entrega do sistema possa ser considerada válida.
- c. **Econômicos** – restrições ao projeto de ordem econômica.

- d. **Distribuição** – descrição sucinta da forma de distribuição, por exemplo, base centralizada ou descentralizada; se vai haver servidor de aplicação; distribuição de camadas.
- e. **Éticos** – restrições de ordem ética.

#### **7.2.8. Infra-estrutura Tecnológica**

<Relacionar em linhas gerais a infra-estrutura tecnológica necessária para o desenvolvimento e a utilização do sistema.>

## 8. ESPECIFICAÇÃO DE REGRAS DE NEGÓCIO

### Histórico de Versões

<b>Data</b>	<b>Versão</b>	<b>Descrição</b>	<b>Autor</b>	<b>Revisor</b>	<b>Aprovado por</b>
<dd/mm/aaaa>	<x.x>	<detalhes>	<nome>	<nome>	<nome>

## **8.1.OBJETIVO**

<O objetivo da Especificação de Regras de Negócio é documentar as regras que são aplicáveis ao negócio, e que direcionam em maior ou menor grau o funcionamento dos casos de uso. Em geral, regras de negócio constituem declarações de políticas ou condições que devem ser satisfeitas pelo processamento da aplicação. Recomenda-se, para fins de um melhor entendimento, que as regras sejam organizadas em grupos de assunto. Adicionalmente, a depender do volume de regras a serem documentadas, documentos distintos podem ser criados para cada tipo de assunto, e assim conter as regras específicas deste assunto.>

## **8.2. <NOME DO GRUPO DE REGRAS>**

### **8.2.1. <Nome da Regras>**

<descrição da regra>

## **8.3. <NOME DE OUTRO GRUPO DE REGRAS>**

### **8.3.1..n.1<Nome da Regras>**

**9. DIAGRAMA DE CLASSES**

<b>Data</b>	<b>Versão</b>	<b>Descrição</b>	<b>Autor</b>	<b>Revisor</b>	<b>Aprovado por</b>
<dd/mm/aaaa>	<x.x>	<detalhes>	<nome>	<nome>	<nome>

## 9.1.OBJETIVO

<Os diagramas de classes devem mostrar a estrutura estática do modelo, principalmente os elementos existentes, como classes, sua estrutura interna e seus relacionamentos com outras classes. Eles não devem mostrar informações temporárias.

Um diagrama de classes é apresentado como um conjunto de elementos do modelo declarativos (estáticos) - como classes, pacotes e seus relacionamentos - que são conectados entre si e a seu conteúdo como um gráfico.

Os diagramas de classe podem ser organizados em (e pertencentes a) pacotes, mostrando apenas o que é relevante em um determinado pacote; e devem:

- a) Garantir que a classe informe o comportamento necessário às realizações dos casos de uso.
- b) Garantir que sejam fornecidas informações suficientes para implementar a classe sem ambigüidades.
- c) Tratar dos requisitos não-funcionais relativos à classe.
- d) Incorporar os mecanismos de design usados pela classe.>

## 9.2.USO

<Apresente cada classe em pelo menos um diagrama. Algumas vezes, é possível entender melhor o modelo se uma classe aparecer várias vezes na mesma visão; por exemplo, caso você deseje fazer a diferenciação entre diferentes objetos da classe.

- a) As classes mais importantes e seus relacionamentos. Diagramas deste tipo podem funcionar como um resumo de modelo de objetos e são de grande ajuda para revisar o modelo. Esses diagramas provavelmente serão incluídos na visão lógica da arquitetura.
- b) Classes coerentes ou relacionadas funcionalmente.
- c) Classes que pertencem ao mesmo pacote.
- d) Hierarquias de agregação importantes.
- e) Estruturas importantes de objetos de entidade, como estruturas de classe com relacionamentos de associação, de agregação e de generalização. Se for possível, crie um diagrama de classes que contenha todas as classes dos objetos de longa duração e seus relacionamentos. Esse tipo de diagrama é especialmente útil para revisar o que está armazenado no sistema e as estruturas de armazenamento.
- f) Os pacotes e suas dependências, possivelmente ilustrando a divisão em camadas.
- g) As classes que participam de uma determinada realização de casos de uso.
- h) Uma única classe, seus atributos, operações e relacionamentos com outras classes.>

**10.MODELO DE DADOS**

<b>Data</b>	<b>Versão</b>	<b>Descrição</b>	<b>Autor</b>	<b>Revisor</b>	<b>Aprovado por</b>
<dd/mm/aaaa>	<x.x>	<detalhes>	<nome>	<nome>	<nome>

## 10.1.OBJETIVO

<O modelo de dados é um subconjunto do modelo de implementação que descreve a representação lógica e física dos dados persistentes no sistema. Também abrange qualquer comportamento definido no banco de dados, como procedimentos armazenados, triggers, restrições, etc.>

## 10.2.FINALIDADE

<O modelo de dados é usado para descrever a estrutura lógica e possivelmente física das informações persistentes gerenciadas pelo sistema.

O modelo de dados é especificamente necessário quando a estrutura de dados persistentes não pode ser obtida mecânica e automaticamente da estrutura de classes persistentes no modelo de design.

Além de definir estruturas de dados persistentes, ele é usado para definir o mapeamento entre classes de design persistentes e estruturas de dados persistentes.

Ele é necessário com mais freqüência quando o modelo de design é um modelo de objetos e o mecanismo de armazenamento persistente é baseado em um banco de dados relacional, embora geralmente seja necessário sempre que esse mecanismo é baseado em uma tecnologia não orientada a objetos.>

## 10.3.PROPRIEDADES A SEREM ENTREGUES

Nome da Propriedade	<Breve Descrição>	<Representação em UML>
Introdução	<É uma descrição textual que funciona como uma rápida introdução do modelo.>	<Valor rotulado, do tipo "texto curto".>
Pacotes	<Os pacotes usados para fins de agrupamento organizacional.>	<Incluídos por meio da associação "representa" ou recursivamente através da agregação "possui".>
Tabelas	<As tabelas do modelo de dados, pertencentes aos pacotes.>	<Componentes, estereotipados como "tabela".>
Relacionamentos	<Os relacionamentos entre as tabelas do modelo.>	<Associações, estereotipadas como "chave estrangeira".>
Colunas	<Os valores de dados das tabelas.>	<Atributos, estereotipados como "coluna".>

Diagramas	<Os diagramas do modelo, pertencentes aos pacotes.>	<Atributos, estereotipados como "coluna".>
Índices	<As estruturas de acesso a dados usadas para acelerar o acesso em caminhos especificados.>	<Componentes, estereotipados como "índice".>
Triggers	<Comportamento ativado por evento e associado a tabelas.>	<Operação, estereotipada como "trigger".>
Procedimentos	<Comportamento chamado de forma explícita, associado a tabelas ou ao modelo como um todo.>	<Componente, estereotipado como "procedimento".>

## 11.RELAÇÃO DE CASOS DE USO

### Histórico de Versões

<b>Data</b>	<b>Versão</b>	<b>Descrição</b>	<b>Autor</b>	<b>Revisor</b>	<b>Aprovado por</b>
<dd/mm/aaaa>	<x.x>	<detalhes>	<nome>	<nome>	<nome>

<As informações contidas neste artefato podem ser obtidas automaticamente pelas ferramentas Rose (Diagrama de Interação) e RequisitePro – Matriz de Atributos (Casos de Uso). Desta forma, para que não haja o risco de inconsistência devido à cópia de informações, será opcional para os que estiverem utilizando as ferramentas acima citadas. Pode também ser inserido um *link* para o diagrama no Rose. >

## **11.1.DIAGRAMA DE CASOS DE USO**

<Representar graficamente os principais casos de uso que compõem o sistema e suas interações com atores/partes interessadas, interna e externamente. Em sistemas de alta complexidade, com um diagrama de casos de uso extenso, pode ser construído *link* para o documento na ferramenta Rose, caso este não caiba dentro do espaço reservado no documento.>

## **11.2.CASOS DE USO**

### **11.2.1.<nome do caso de uso>**

<Iniciar com verbo no Infinitivo. Utilizar nome completo, onde as primeiras letras devem ser maiúsculas, exceto preposições. Este nome deve retratar claramente a ação a ser realizada. >

<**Descrição do caso de uso.** Descrever a finalidade do caso de uso, oferecendo uma idéia geral e sucinta do propósito do caso de uso.>

## 12. ESPECIFICAÇÃO DE CASO DE USO

### Histórico de Versões

<b>Data</b>	<b>Versão</b>	<b>Descrição</b>	<b>Autor</b>	<b>Revisor</b>	<b>Aprovado por</b>
<dd/mm/aaaa>	<x.x>	<detalhes>	<nome>	<nome>	<nome>

## 12.1. NOME DO CASO DE USO

<Iniciar com verbo no Infinitivo. Utilizar nome completo, onde as primeiras letras devem ser maiúsculas, exceto preposições. Este nome deve retratar claramente a ação a ser realizada. >

## 12.2. OBJETIVO

<Descrição da finalidade do caso de uso. Deve oferecer uma idéia geral do propósito do caso de uso.>

## 12.3. TIPO DE CASO DE USO <OPCIONAL>

<Classificar o Caso de Uso em questão em Concreto ou Abstrato. Um caso de uso concreto é iniciado por um ator e constitui um fluxo completo de eventos. "Completo" significa que a instância do caso de uso executa o fluxo inteiro invocado pelo ator.

Um caso de uso abstrato nunca é instanciado diretamente por um ator. Casos de uso abstratos são incluídos em (relacionamento de *include*), estendidos por (relacionamento de *extend*), ou especializados em (relacionamento de generalização) outros casos de uso.

Quando um caso de uso concreto é iniciado, uma instância do caso de uso é criada. Esta instância também exibe o comportamento especificado pelo caso de uso abstrato associado. Assim, instâncias do caso de uso abstrato não são criadas em separado.

A diferença entre os dois tipos é importante, pois os atores irão "ver" e iniciar os casos de uso concretos no sistema.>

## 12.4. ATORES

Nome Ator	Tipo	
	Primário	Secundário
<Relacionar os atores que interagem com o caso de uso, começando pelo ator que inicia a interação. Utilizar nome completo, onde as primeiras letras devem ser maiúsculas, exceto preposições.>	<Marcar com um X caso o ator inicie o caso de uso.>	<Marcar com um X caso o ator participe de forma passiva, somente recebendo informações, por exemplo.>

## 12.5. PRÉ-CONDIÇÕES <OPCIONAL>

<Texto livre que identifica as pré-condições do caso de uso. Pré-condições são condições que devem ser satisfeitas para que o caso de uso possa iniciar. Na inexistência de pré-condições para o caso de uso, formatar a mensagem: "Nenhuma pré-condição identificada".>

## 12.6. FLUXO PRINCIPAL

< Descrever o cenário mais utilizado pelos atores. Este cenário apresenta o caminho percorrido pelo ator para atingir o objetivo com sucesso. A numeração dos passos deve ser seqüencial e iniciar em 1. A numeração dos subpassos deve preservar o índice do passo-pai, acrescido ao número identificador do subpasso, iniciando em 1.>

**P<n>. <Título do passo>**

**<descrição do passo.** sentença clara e objetiva descrevendo a função que o passo declara. Evitar a fragmentação de passos provocada pela decomposição funcional. *Opcional na existência de subpassos ou no caso do título e da descrição serem iguais.>*

**P<n.m>.** <sentença sucinta descrevendo a função que o subpasso declara. A adoção de subpassos é opcional, sendo recomendada apenas para passos complexos.>

## 12.7. FLUXOS ALTERNATIVOS <OPCIONAIS>

<Descrever os cenários alternativos utilizados pelos atores. A numeração dos procedimentos deve ser seqüencial e iniciar em 1. Os caminhos alternativos devem percorrer um fluxo completo, demonstrando:

- a. o passo a que estão associados.
- b. a condição que aciona a entrada no caminho alternativo
- c. as ações tomadas no caminho alternativo
- d. a ação de retorno. >

**A<n>.** <descrição do fluxo alternativo.>

## 12.8. FLUXOS DE EXCEÇÃO <OPCIONAL>

<Descrever os cenários de erros. Estes cenários apresentam os possíveis erros que podem ser observados quando da interação dos atores com a aplicação. A exemplo dos fluxos alternativos, descrever um fluxo completo, observando o disposto nos itens a, b, c, e d anteriores (seção 5).>

**E<n>.** <descrição do fluxo de exceção.>

## 12.9. PÓS-CONDIÇÕES <OPCIONAL>

<Texto livre que identifica as pós-condições. Pós-condições são condições que podem ser garantidas como verdadeiras ao final do caso de uso. Na inexistência de pós-condições para o caso de uso, formatar a mensagem: **“Nenhuma pós-condição identificada”**.>

## 12.10. REQUISITOS NÃO FUNCIONAIS <OPCIONAL>

<Enumerar os requisitos não funcionais identificados especificamente para este caso de uso. Descrever nesta seção requisitos relativos ao caso de uso que

não são cobertos pelo fluxo de eventos, ou por outros documentos (Visão, RNF), mas que podem influenciar o desenvolvimento do sistema. Na inexistência de requisitos não funcionais específicos do caso de uso, formatar a mensagem: **“Nenhum requisito não funcional identificado”**.>

#### **12.11. PONTO DE EXTENSÃO <OPCIONAL>**

<Fazer referência a pontos de extensão utilizado no detalhamento. Pontos de extensão são referências a outros casos de uso externos que complementam o fluxo de eventos do corpo do caso de uso chamador. Identificar o passo do fluxo básico ao qual a extensão está associada, descrevendo as condições e o momento para sua ativação. Na inexistência de pontos de extensão, formatar a mensagem: **“Nenhum ponto de extensão identificado”**.>

**PE<n>. <Título do ponto de extensão>**

<descrição do ponto de extensão.>

#### **12.12. CRITÉRIOS DE ACEITE**

<Documentar, de forma sucinta, os critérios de aceite ou casos de teste que possam auxiliar a completa validação do caso de uso para serem utilizados na fase de testes, onde devem ser mais bem detalhados, utilizando a ferramenta de testes.>

#### **12.13. FREQUÊNCIA DE UTILIZAÇÃO**

<Informar se é alta, média ou baixa e quais informações são mais acessadas. A frequência de utilização deve ser uma estimativa da quantidade de utilização do caso de uso pelos atores em um determinado período de tempo. Caso existam picos de utilização, esses devem ser descritos.>

#### **12.14. OBSERVAÇÕES <OPCIONAIS>**

<Espaço destinado para as anotações técnicas, informações adicionais a serem trabalhadas no futuro, ou lembretes.>

#### **12.15. REFERÊNCIAS <OPCIONAIS>**

<Item disponível para citar os modelos, diagramas, funcionalidades de outros projetos, e outros documentos que se relacionem ao caso de uso em questão.>

#### **12.16. INTERFACE DO USUÁRIO <OPCIONAL>**

<Informar o modelo da tela apresentada ao usuário.

"A interface do usuário é o que permite que as informações sejam transmitidas entre um usuário humano e os componentes de hardware ou software de um sistema de computador."

Item informado apenas para o caso de uso com ator usuário humano interagindo com o sistema.>

### 13. ORÇAMENTO

#### Histórico de Versões

<b>Data</b>	<b>Versão</b>	<b>Descrição</b>	<b>Autor</b>	<b>Revisor</b>	<b>Aprovado por</b>
<dd/mm/aaaa>	<x.x>	<detalhes>	<nome>	<nome>	<nome>

### 13.1.OBJETIVO

<Apresentar o custo financeiro do projeto adotando modelo de Qualidade de Software com suporte na implementação de recursos CMM (Capability Maturity Model) nível 2, em conformidade com a Gerência de Requisitos, Planejamento e Acompanhamento do Plano de Desenvolvimento de Aplicativos apresentado ao DNIT.

Utilizar metodologia baseada em métricas de **Análise por Ponto de Função**, para definir custos e prazos, cuja referência adotada será definida pelo **IFPUG – International Function Point Users Group**.>

## 14.GLOSSÁRIO

### Histórico de Versões

<b>Data</b>	<b>Versão</b>	<b>Descrição</b>	<b>Autor</b>	<b>Revisor</b>	<b>Aprovado por</b>
<dd/mm/aaaa>	<x.x>	<detalhes>	<nome>	<nome>	<nome>

**14.1.**

**14.2.<DEFINIÇÃO DOS TERMOS>**

<Definir os termos importantes usados pelo projeto, de forma clara e objetiva, assegurando uma interpretação inequívoca dos documentos produzidos pelo plano de desenvolvimento de aplicativos>

## 15. TERMO DE APROVAÇÃO DE ESPECIFICAÇÃO DE SISTEMA Histórico de Versões

<b>Data</b>	<b>Versão</b>	<b>Descrição</b>	<b>Autor</b>	<b>Revisor</b>	<b>Aprovado por</b>
<dd/mm/aaaa>	<x.x>	<detalhes>	<nome>	<nome>	<nome>

**15.1.<DEFINIÇÃO DOS TERMOS>**

Código da demanda

Sistema

Módulo(s)

---

 Descrição da demanda

---

 Registros

Analista de Negócio Cliente	Local de trabalho	Rubrica

Participantes	Local de Trabalho	Rubrica

Atesto que a especificação do(s) módulo(s)/sistema está de acordo com as expectativas e necessidades do serviço solicitado, e que o conjunto de artefatos abaixo relacionados estabelecem o escopo, os requisitos e todas as definições necessárias para entendimento da demanda do(s) módulo(s)/sistema:

- |   |   |
|---|---|
| <input type="checkbox"/> Documento de Visão do Projeto              | <input type="checkbox"/> Modelo de Dados                                |
| <input type="checkbox"/> Cronograma                                 | <input type="checkbox"/> Relação de Casos de Uso                        |
| <input type="checkbox"/> Documento de Requisitos de Software        | <input type="checkbox"/> Especificação de Caso de Uso                   |
| <input type="checkbox"/> Especificação de Requisitos Não Funcionais | <input type="checkbox"/> Orçamento                                      |
| <input type="checkbox"/> Especificação de Regras de Negócio         | <input type="checkbox"/> Glossário                                      |
| <input type="checkbox"/> Diagrama de Classes                        | <input type="checkbox"/> Termo de Aprovação de Especificação de Sistema |
| <input type="checkbox"/> <Outros ( especificar)>                    |   |

Responsável

Unidade

Assinatura

---

 De Acordo:

Responsável

Unidade

Assinatura



1. Prefácio.....	4
2. Objetivo.....	5
3. Definições.....	6
4. Visão do Projeto.....	8
4.1. Objetivo.....	10
4.2. Posicionamento.....	10
4.2.1. Cenário.....	10
4.2.2. Oportunidade de Negócio.....	10
4.2.3. Descrição do Projeto.....	10
4.3. Envolvimento.....	10
4.3.1. Abrangência.....	10
4.3.2. Papel das Partes Interessadas.....	10
4.3.3. <nome da parte interessada>.....	11
4.4. Papel dos Atores (<opcional, caso já estejam descritos na visão do sistema>).....	11
4.4.1. <nome do ator>.....	11
4.5. Necessidades e Funcionalidades.....	11
4.6. Restrições.....	12
4.7. Expectativa de Entrega do Produto.....	12
4.7.1. Proposta de Solução Tecnológica Escolhida.....	13
5. Cronograma.....	14
5.1. Objetivo.....	15
5.2. Relação de Cronogramas.....	15
5.2.1. Acompanhamento do projeto.....	15
5.2.2. Fases de acompanhamento.....	15
5.2.3. Detalhamento do Projeto – Fase de Iniciação.....	16
5.2.4. Detalhamento do Projeto – Fase de Elaboração.....	16
5.2.5. Detalhamento do Projeto – Fase de Construção.....	16
5.2.6. Detalhamento do Projeto – Fase de Transição.....	16
6. Documento de requisitos de software.....	18
6.1. Objetivo.....	19
6.2. Descrição Geral.....	19
7. Especificação de Requisitos Não Funcionais.....	20
7.1. Objetivo.....	21
7.2. Requisitos Não Funcionais.....	21
7.2.1. Usabilidade.....	21
7.2.2. Confiabilidade.....	22
7.2.3. Performance.....	22
7.2.4. Suportabilidade.....	22
7.2.5. Requisitos Legais.....	23
7.2.6. Segurança.....	23
7.2.7. Outros Requisitos Não Funcionais.....	23
7.2.8. Infra-estrutura Tecnológica.....	24
8. Especificação de Regras de Negócio.....	25
8.1. Objetivo.....	26
8.2. <Nome do Grupo de Regras>.....	26
8.2.1. <Nome da Regras>.....	26
8.3. <Nome de outro Grupo de Regras>.....	26
8.3.1. .n.1<Nome da Regras>.....	26
9. Diagrama de classes.....	27

9.1. Objetivo.....	28
9.2. Uso.....	28
10. Modelo de dados.....	29
10.1. Objetivo.....	30
10.2. Finalidade.....	30
10.3. Propriedades A serem entregues.....	30
11. Relação de casos de uso.....	32
11.1. Diagrama de Casos de Uso.....	33
11.2. Casos de Uso.....	33
11.2.1. <nome do caso de uso>.....	33
12. Especificação de caso de uso.....	34
12.1. Nome do Caso de Uso.....	35
12.2. Objetivo.....	35
12.3. Tipo de Caso de Uso <opcional>.....	35
12.4. Atores.....	35
12.5. Pré-Condições <opcional>.....	35
12.6. Fluxo Principal.....	36
12.7. Fluxos Alternativos <opcionais>.....	36
12.8. Fluxos de Exceção <opcional>.....	36
12.9. Pós-condições <opcional>.....	36
12.10. Requisitos Não Funcionais <opcional>.....	36
12.11. Ponto de Extensão <opcional>.....	37
12.12. Critérios de Aceite.....	37
12.13. Frequência de Utilização.....	37
12.14. Observações <opcionais>.....	37
12.15. Referências <opcionais>.....	37
12.16. Interface do usuário <opcional>.....	37
13. Orçamento.....	38
13.1. Objetivo.....	39
14. Glossário.....	40
14.2. <Definição dos termos>.....	41
15. Termo de aprovação de especificação de sistema.....	42
.....	42
15.1. <Definição dos termos>.....	43

---