



GOVERNANÇA DE TECNOLOGIA DA INFORMAÇÃO
ARTEFATO PROCESSO DE DESENVOLVIMENTO DE SOFTWARE (GERENCIAMENTO
DA QUALIDADE E CONFIGURAÇÃO)

MINISTÉRIO DO PLANEJAMENTO, DESENVOLVIMENTO E GESTÃO
SECRETARIA DE COORDENAÇÃO E GOVERNANÇA DAS EMPRESAS ESTATAIS
DIRETORIA DE ORÇAMENTO DE ESTATAIS
COORDENAÇÃO-GERAL DE GESTÃO DA INFORMAÇÃO DE ESTATAIS

BRASÍLIA - 2018

**MINISTÉRIO DO PLANEJAMENTO, DESENVOLVIMENTO
E GESTÃO**
**SECRETARIA DE COORDENAÇÃO E GOVERNANÇA
DAS EMPRESAS ESTATAIS**

UNIVERSIDADE DE BRASÍLIA

Fernando Antonio Ribeiro Soares

Secretário

Márcia Abrahão Moura

Reitora

André Nunes

Diretor do Departamento de Orçamento de Estatais

Sanderson Cesar Macedo Barbalho

Diretor do Centro de Apoio ao Desenvolvimento
Tecnológico – CDT

Gerson Batista Pereira

Coordenador-Geral de Gestão da Informação de Estatais

Rafael Timóteo de Sousa Júnior

Coordenador do Laboratório de Tecnologias
da Tomada de Decisão – LATITUDE

EQUIPE TÉCNICA

Natal Henrique Troz Guglihermi – SEST

Otávio Porto Barbosa – SEST

EQUIPE TÉCNICA

Georges Daniel Amvame Nze

(Pesquisador Sênior)

Claudia Jacy Barrenco Abbas

(Pesquisador Sênior)

Edna Dias Canedo

(Pesquisador Sênior)

Rodrigo de Souza Goncalves

(Pesquisador Sênior)

Adyr Andrade de Menezes

Amanda Aline Figueiredo Carvalho

Bruno Justino Garcia Praciano

Demétrio Antônio da Silva Filho

Fabricio de Oliveira Taguatinga

Glauber Luiz Lopes da Silva

Jean Victor Ribeiro Vieira

João Batista Alves Diniz

Jorge Guilherme Silva dos Santos

José Maria dos Reis Lisboa

Leomar Camargo de Souza

Marcus Vinicius Bomfim Guimaraes Barbalho

Moramay Coutinho Guimarães Coelho

Pedro Thiago Rocha de Alcântara

Priscilla Gonçalves da Silva e Souza

Rafaella Aparecida Rosa Lima

Rosa Cristina Portela Dias Jácome

Ruyther Parente da Costa

Victor Matheus da Silva

B823g

Brasil. Ministério do Planejamento, Desenvolvimento e Gestão.

Governança de tecnologia da informação : artefato processo de desenvolvimento de software (gerenciamento da qualidade e configuração) / Ministério do Planejamento, Desenvolvimento e Gestão, Secretaria de Coordenação e Governança das Empresas Estatais, Coordenação-Geral de Gestão da Informação de Estatais; Universidade de Brasília. -- Brasília : MP, 2018.

70 p.

1. Governança Digital 2. Tecnologia da Informação 3. Empresas Estatais
4. Administração Pública I. Título II. Universidade de Brasília.

CDU 658.115:004

HISTÓRICO DE VERSÕES

09/03/2018 | Versão 1.0

Descrição: inclusão dos artefatos, definição do processo, adequação do passo-a-passo, objetivos e capa ao processo.

Autor: Edna Dias Canedo e Priscilla Gonçalves da Silva e Souza.

Revisor: Natal Henrique Troz Guglilhermi e Otávio Porto Barbosa.

SUMÁRIO

1. INTRODUÇÃO.....	5
2. VISÃO GERAL	5
2.1. Objetivo	5
2.2. Justificativa.....	5
3. PROCESSO DE DESENVOLVIMENTO DE SOFTWARE (GERENCIAMENTO DA QUALIDADE E CONFIGURAÇÃO)	6
3.1. Definição	6
3.2. Passo a passo.....	6
4. ARTEFATOS	8
4.1. Documentos.....	8
4.1.1 Processo de Desenvolvimento de Software	9
4.1.2 Plano de Gerenciamento de Qualidade	14
4.1.3 Plano de Testes	17
4.1.4 Rastreabilidade de Requisitos	62
4.1.5 Plano de Manutenção de Soluções.....	68
5. REFERÊNCIAS BIBLIOGRÁFICAS.....	70
5.1. Documentos.....	70

INTRODUÇÃO

Em observância às normas e diretrizes de Tecnologia da Informação (TIC) do Poder Executivo Federal, disseminadas pela Secretaria de Tecnologia da Informação e Comunicação do Ministério do Planejamento, Desenvolvimento e Gestão (SETIC/MP), na condição de Órgão Central do Sistema de Administração dos Recursos de Tecnologia da Informação (SISP) e, conforme preconiza o Decreto Presidencial nº 7.579, de 11 de outubro de 2011, o Ministério do Planejamento, Desenvolvimento e Gestão (MP), como Órgão Setorial integrante do SISP, vincula-se aos preceitos definidos pelo Sistema relativamente à governança e gestão de tecnologia da informação.

Diante do tema e também em decorrência de orientação do TCU, conforme Acórdão 3051/2014 a SEST deve atuar no desenvolvimento de ações que promovam a disseminação da cultura de Governança de TIC nas Empresas Estatais, para facilitar o cumprimento dos objetivos definidos e exigidos no planejamento estratégico, como também na racionalização de recursos e retorno financeiro/operacional.

VISÃO GERAL

2.1. Objetivo

Identificar e apontar os passos necessários, de acordo com práticas listadas em literatura e conhecimento prático, Processo de Desenvolvimento de Software (Gerenciamento da Qualidade e Configuração).

2.2. Justificativa

A SEST, institucionalmente, como órgão de Coordenação e Governança das Empresas Estatais, deve promover e orientar a Governança de TIC dessas entidades. As iniciativas nesse sentido devem ser planejadas e priorizadas a partir do alinhamento dos investimentos de TIC aos objetivos estratégicos das organizações.

PROCESSO DE DESENVOLVIMENTO DE SOFTWARE (GERENCIAMENTO DA QUALIDADE E CONFIGURAÇÃO)

3.1. Definição

O Processo de Desenvolvimento de Software (Gerenciamento da Qualidade e Configuração) visa estabelecer e manter soluções identificadas em linha com os requisitos corporativos cobrindo concepção, desenvolvimento, transição e encerramento.

Nesse sentido, a principal tarefa nesta prática é executar a gestão de configuração de software e qualidade, preparação e execução de testes, gestão de requisitos e manutenção de software, aplicações, informações/dados, infraestrutura e serviços.

3.2. Passo a passo

Para a implantação do Processo de Desenvolvimento de Software (Gerenciamento da Qualidade e Configuração) é preciso executar as seguintes atividades:

1. Definir e documentar um Processo de Desenvolvimento de Software (PDS) Padrão a ser adotado pela Estatal, definindo a Metodologia de Desenvolvimento de Software, atores, artefatos e fases do PDS, bem como as atividades de desenvolvimento de software e de planejamento de recursos necessários para que o software tenha o ambiente necessário para seu funcionamento.
2. Desenhar/projetar soluções de alto nível, a partir da lista de requisitos definidos, especificados, classificados e priorizados com o solicitante (cliente) nas entrevistas de levantamento de requisitos/necessidades.
3. Desenhar e detalhar os componentes da solução (arquitetura da aplicação) de alto nível, utilizando ferramentas padronizadas, tais como: como histórias de usuários, modelo de dados, diagramas de classes e diagrama de casos de uso.
4. Desenvolver os componentes da solução proposta detalhadamente, segundo projeto realizado, documentando todo o processo de desenvolvimento.
5. Adquirir os componentes da solução desejada, avaliando sua viabilidade e necessidade na estatal.
6. Construir soluções utilizando os componentes desenvolvidos e adquiridos para atender as necessidades da estatal.
7. Estabelecer e manter um sistema de gerenciamento de configurações e de gerenciamento de mudanças para controlar todos os produtos de trabalho necessários ao funcionamento da estatal.

8. Executar auditorias de configuração para manter a integridade das linhas de base de configuração, para entrega ao cliente com a devida documentação e registro.
9. Definir mecanismos de medição e análise, especificando métricas para abordar os objetivos de medição. As metas devem ser coletadas e registradas durante todo o processo de desenvolvimento de software.
10. Especificar como as informações da medição serão obtidas e armazenadas na base de dados da estatal.
11. Analisar e interpretar os dados/resultados da medição, gerando relatórios de análises e discutindo com a equipe os resultados obtidos.
12. Gerenciar e armazenar as informações da medição, especificações da medição e análise dos resultados para servirem de base histórica no desenvolvimento de outros softwares que a estatal venha a desenvolver posteriormente.
13. Comunicar os resultados das atividades de medição e análise a todos os envolvidos no processo de desenvolvimento de software e discutindo possíveis melhorias.
14. Definir o processo de Garantia de Qualidade de Produto e do Processo, definindo processos para gerenciar o projeto e construir os entregáveis, bem como critérios de precisão para avaliar a qualidade dos entregáveis a estatal.
15. Avaliar se os produtos de trabalho estão de acordo com as descrições dos processos, padrões e procedimentos adotados pela estatal.
16. Estabelecer e manter os registros das atividades de garantia de qualidade.
17. Estabelecer um Sistema de Gerenciamento de Qualidade (QMS) de acordo com os padrões internacionais e com um profissional da área de qualidade.
18. Definir e gerenciar os padrões de qualidade, práticas e procedimentos adotados pela estatal.
19. Centralizar o gerenciamento da qualidade em torno dos clientes, de maneira que sua satisfação seja atendida.
20. Integrar o gerenciamento da qualidade nas soluções para desenvolvimento e entrega de serviços.
21. Preparar os testes da solução, documentando no Plano de Testes as técnicas de teste a serem executadas, os responsáveis e os conhecimentos necessários para a realização dos testes da solução pela equipe responsável.
22. Realizar os testes de solução e documentar nos planos de testes todos os resultados obtidos. É importante apresentar e discutir os resultados dos testes com a equipe de desenvolvimento de software da estatal.

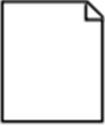
23. Gerenciar mudanças nos requisitos da solução, atualizando as especificações de acordo com as observações do requisitante ou de acordo com as demais estatais e ou departamentos envolvidos, e mantendo o acompanhamento de requisitos individuais e suas ligações entre os requisitos dependentes. A rastreabilidade dos requisitos é fundamental para o gerenciamento de mudanças.
24. Manter a melhoria contínua dos serviços e produtos, definindo atividades de manutenção do sistema, diretrizes sobre os processos de backup de dados, de suporte continuado aos usuários e de atendimento de novos requisitos que surgem do próprio uso e mudanças de processos no ambiente de trabalho da estatal.

ARTEFATOS

4.1. Documentos

Modelos de documentos para a criação e operacionalização do Processo de Desenvolvimento de Software (Gerenciamento da Qualidade, Configuração):

Estabelecer o Processo de Desenvolvimento de Software (PDS)	
 Processo de Desenvolvimento de Software (PDS)	Nome: Processo de Desenvolvimento de Software
	Objetivo: Definir a Metodologia de Desenvolvimento de Software, atores, artefatos, fases, atividades de desenvolvimento de software e de planejamento dos recursos necessários para que o software tenha o ambiente necessário para seu funcionamento.
Gerenciar a Qualidade dos Entregáveis	
 Plano de Gerenciamento de Qualidade	Nome: Plano de Gerenciamento de Qualidade
	Objetivo: Estabelecer processos que serão usados para gerenciar a qualidade do projeto e construir os entregáveis.
Planejar os Testes dos Entregáveis	
 Plano de Testes	Nome: Plano de Testes
	Objetivo: Planejar os recursos referentes ao planejamento dos testes a serem realizados no produto (técnicas, responsáveis, artefatos, etc.).

Realizar a Gestão dos Requisitos	
 Rastreabilidade de Requisitos	Nome: Rastreabilidade de Requisitos
	Objetivo: Registrar e detalhar os requisitos para facilitar a gestão dos mesmos em caso de mudança/atualização e/ou nos requisitos que possuem dependência entre eles.
Estabelecer Atividades de Manutenção do Sistema	
 Plano de Manutenção de Soluções	Nome: Plano de Manutenção de Soluções
	Objetivo: Estabelecer as atividades e diretrizes de manutenção de sistema, tais como diretrizes sobre os processos de backup de dados, de suporte continuado aos usuários e de atendimento de novos requisitos que surgem do próprio uso e mudanças nos processos do ambiente de trabalho.

4.1.1 Processo de Desenvolvimento de Software

Processo de Desenvolvimento de Software para o <Estatal>

Código e Sigla do Projeto:

Nome do Projeto:

Versão do Documento: <Nº da versão para a estatal>

<Sigla da estatal>
 <Nome do Projeto>
 <Nome da Aplicação>

Histórico da Revisão

Data	Versão	Descrição	Autor
<dd/mm/aaaa>	<x.x>	<detalhes>	<nome>

Sumário

1. Introdução Erro! Indicador não definido.

*<Nesta seção devem ser incluídas informações que facilitam a compreensão do documento, tais como a sua finalidade, seu conteúdo, os padrões utilizados na escrita do documento, etc. O **Documento de Processo de Desenvolvimento de Software** fornece uma base de alto nível para a documentação do processo de desenvolvimento de software adotado pela estatal, seja ele ágil ou tradicional, definindo a Metodologia de Desenvolvimento de Software, a qual deve se tornar instrumento de formalização e estruturação de processos e práticas de desenvolvimento de software da estatal. Ele fornece também informações úteis para o processo de aprovação do projeto, sendo um documento importante para a construção dos softwares/produtos da estatal.>*

2. Processo de Desenvolvimento de Software para o <estatal>

2.1 Visão Geral do PDS_<estatal>

<Forneça uma descrição geral do processo de desenvolvimento de Software, apresentando o processo com os respectivos conceitos e estrutura do PDS >

2.2 Fases do Processo do PDS_<estatal>

<Forneça uma descrição geral das fases do processo adotado>

2.3 Eixos de Trabalho do PDS_<estatal>

<Forneça uma descrição geral dos eixos de trabalho e seus objetivos do PDS>

2.4 Atores do PDS_<estatal>

<Forneça uma descrição geral dos atores envolvidos e suas atribuições no PDS>

2.5 Artefatos do PDS_<estatal>

<Forneça uma descrição geral dos artefatos, objetivo e atores responsáveis pela sua elaboração>

<Nome completo do Responsável >

<Cargo >

Observações:

Processo de Desenvolvimento de Software:

O processo de software aborda não só as atividades ligadas ao desenvolvimento de software como também as atividades ligadas ao planejamento dos recursos necessários para que o software tenha o ambiente necessário para o seu funcionamento.

O processo de software tem seis fases (concepção e alinhamento estratégico, especificação e dimensionamento, estratégia de desenvolvimento, desenvolvimento, implantação e estabilização, e sustentação e evolução) e oito eixos de trabalho (alinhamento estratégico, gestão de projetos, produção colaborativa, gestão de segurança, engenharia de software, gestão da contratação, gestão de infraestrutura e gestão de sustentação). Essas fases e eixos podem ser adequadas a cada Estatal. Não sendo obrigatório ter todas estas fases.

Os eixos de trabalho são áreas de conhecimento que são essenciais ao pleno desenvolvimento, manutenção (evolutiva, adaptativa e corretiva) e uso do software.

Alinhamento Estratégico: Visa promover o alinhamento da necessidade do software com as necessidades de negócio da estatal descritas nos seus instrumentos estratégicos, como por exemplo: Plano Diretor de Tecnologia da Informação (PDTI), Estratégia Geral de Tecnologia da Informação (EGTI), Planejamento Estratégico Institucional (PEI) da estatal e outros.

Gestão de Projetos: Visa promover uma adequada gestão dos projetos. Os processos de gestão de projetos serão mapeados tendo como referência a Metodologia de Gerenciamento de Projetos

Produção Colaborativa: Visa o desenvolvimento conjunto de software, ou seja, processos que promovam o levantamento de requisitos comuns a mais de uma estatal para que possam desenvolver ou contratar um software colaborativamente. Aqui também busca-se identificar se uma solução já foi desenvolvida e pode ser adotada ou adequada.

Gestão de Contratação: Promover o conjunto de boas práticas para contratações de soluções de TI. Os processos da gestão de contratação serão baseados e alinhados com a lei 13303 e no Manual de Contratações de Soluções de Tecnologia da Informação.

Engenharia de Software: Desenvolvimento e manutenção de sistemas baseado nas melhores práticas difundidas no mercado e na literatura, e em metodologias utilizadas pela

Engenharia de Software, como, por exemplo, o processo unificado, normas NBR ISO/IEC 12207, NBR ISO/IEC 15504 e outros.

Gestão de Segurança: Desenvolvimento seguro de software que envolve tanto a segurança do ambiente de desenvolvimento quanto da aplicação desenvolvida. Além disso, visa preservar a confidencialidade, integridade e disponibilidade das informações, através do estabelecimento de políticas, práticas e processos.

Gestão de Infraestrutura: Construir um ambiente que tenha a capacidade necessária para prover serviços e uma estrutura adequada ao desenvolvimento de software.

Gestão de Sustentação: Planejamento das condições necessárias para que o software desenvolvido seja mantido, operado e evoluído de forma sustentável e viável.

Atores do PDS: Alguns dos participantes, podendo haver outros identificados pelas estatais.

Ator

Atribuição

Área de Tecnologia da Informação

Unidade setorial ou seccional do SISP, bem como área correlata, responsável por gerir a Tecnologia da Informação da estatal ou entidade.

Área Requisitante

Pessoa ou grupo responsável pela solicitação do produto, serviço ou resultado do projeto. Deverão informar as necessidades, expectativas e requisitos, e aprovar as entregas.

Analista de Métricas

Responsável pela realização da estimativa inicial de tamanho funcional do software, com base nos requisitos iniciais, e pela contagem detalhada.

Analista de Negócio

Mediador da comunicação entre a área requisitante e a área de TI. Traduz as necessidades de negócio para uma visão mais organizada e técnica, própria da área de TI.

Analista de Requisitos

Responsável pelo levantamento, análise e especificação de requisitos. Levanta as necessidades do usuário e as formaliza em documentos técnicos que nortearão o desenvolvimento ou manutenção de um *software*.

Analista de Suporte

Profissional responsável pelas atividades de suporte ao usuário, abrangendo

	<p><i>software</i> e <i>hardware</i>, instalação de aplicativos e configuração de sistemas.</p>
Analista de Infraestrutura Tecnológica	<p>Define as necessidades de infraestrutura (<i>software/hardware</i>) para execução do <i>software</i> sendo desenvolvido ou mantido. É responsável pela preparação ou verificação do ambiente onde o <i>software</i> será executado.</p>
Analista de Segurança da Informação	<p>Apoia na definição das regras de segurança, no planejamento e verificação dos riscos referente à execução de um <i>software</i>.</p>
Analista de Sustentação	<p>Responsável pela elaboração do Plano de Sustentação de um <i>software</i>, levantando riscos e recursos envolvidos na manutenção do uso de um <i>software</i> após a sua entrega.</p>
Analista de Tecnologia da Informação	<p>Responsável por definir a estratégia de desenvolvimento e a metodologia mais adequada para atendimento à necessidade de <i>software</i>.</p>
Analista de Teste	<p>Responsável pela criação do projeto dos testes, utilizando técnicas, critérios e tipos de teste adequados ao projeto de <i>software</i> a ser testado. Gera os casos de teste por meio da identificação e priorização dos cenários de teste. Também elabora os procedimentos para a execução dos testes.</p>
Arquiteto de Software	<p>Define plataforma, componentes, <i>frameworks</i> e demais informações necessárias à especificação da arquitetura de um <i>software</i>.</p>
Comitê de Tecnologia da Informação	<p>É formado por representantes das áreas de negócio e da TI de uma instituição e tem a função e o poder de priorizar as ações e dirigir o alinhamento dessas e dos investimentos com os objetivos estratégicos da organização, além de monitorar os resultados do desempenho da TI.</p>

4.1.2 Plano de Gerenciamento de Qualidade

Plano de Gerenciamento da Qualidade de Software

Sigla da Estatal>

<Nome do Projeto>

Histórico da Revisão

Data	Versão	Descrição	Autor	Área/Setor
<dd/mm/aaaa>	<x.x>	<detalhes>	<nome>	

Visão Geral do Projeto

Descreva a razão da realização do projeto bem como o histórico e seu contexto. Escreva como o trabalho será realizado focando nos valores do negocio. Esta sessão pode ser copiada da Definição do Projeto ou de um documento similar. (Remover esta sessão de comentários do documento final)

Padrão Qualidade

Listar padrão de qualidade (ou padrões) que a empresa ou organização definiu previamente como critério para ser seguido pelo projeto em questão. (Remover esta sessão de comentários do documento final)

Ferramentas de Qualidade

Listar ferramenta (ou ferramentas) relacionada à qualidade que o projeto utilizará. (Remover esta sessão de comentários do documento final)

Critérios de Precisão e Perfeição

O propósito do critério de perfeição é estabelecer juntamente com o cliente antecipadamente os critérios para que a deliverable seja considerada correta e completa. Desta forma, quando estes termos estiverem definidos, você poderá esperar que o cliente de fato estará satisfeito. Se você definir estes critérios e expectativas previamente, você

estará muito mais apto a alcançar às expectativas do cliente. Em outras palavras, não deverá surgir qualquer surpresa.

Garantia da Qualidade

A Garantia da Qualidade não se refere diretamente as deliverables específicas. Esta refere-se ao processo usado na criação das deliverables. Em geral a atividade de garantia da qualidade esta focada no processo de gerenciamento e de entrega da solução, e poderá ser realizada por um gestor, por um cliente ou por uma terceira parte revisora. Por exemplo, um revisor independente de projeto talvez não esteja habilitado em dizer se o conteúdo de uma deliverable em particular é aceitável ou não. Entretanto, eles deverão estar habilitados a dizer, baseado no processo utilizado para criação da deliverable, se está parece ser aceitável ou não. (Remover esta sessão de comentários do documento final).

Procedimentos de Garantia da Qualidade

Listar as principais atividades de garantia da qualidade que este projeto realizará.

Garantia da Qualidade: Funções e Responsabilidades

Função	Responsabilidades da Garantia de Qualidade do Projeto
Gerente do Projeto	
Revisor do Projeto de Qualidade	
Patrocinador do Projeto	
Membros da Equipe de Projeto	
(Etc.)	

Observações:

Abordagem do gerenciamento da Qualidade

A abordagem usada no gerenciamento da qualidade inclui, o Plano de Qualidade e três principais processos: O Plano de Qualidade (este documento) é concretizado durante a fase de planejamento para ratificar que as principais deliverables sejam completadas com um nível de qualidade que satisfaçam as necessidades do cliente.

Crítérios de Perfeição e Precisão: A equipe do projeto e os principais stakeholders devem previamente concordar e estabelecer os critérios a serem usados para a definição de quando cada uma das principais deliverables estará completa e correta. Neste momento, antes das deliverables serem formalmente aprovadas, estas deverão ser confrontadas com os critérios previamente estabelecidos. Desta forma as deliverables são avaliadas de acordo com os critérios estabelecidos antes destes serem formalmente aprovados.

Garantia de Qualidade: As atividades de garantia de qualidade estão focadas na utilização de processos, para gerenciar, produzir e entregar a solução, podendo ser realizadas pelo Sistema de Informação Gerencial, pelo patrocinador do projeto ou por uma terceira pessoa que revise estas atividades.

Controle de qualidade: As atividades de controle de qualidade são realizadas continuamente no andamento do projeto para verificar se o gerenciamento do projeto e as deliverables são de alta qualidade.

Se você utilizar os serviços de uma empresa terceirizada para executar um projeto, você também transfere as responsabilidades do dia a dia de gerenciamento de projetos. Entretanto, sua empresa necessita ainda ter um nível de envolvimento para validar que o projeto está tendo um bom andamento, e que o terceirizado entregará dentro de suas expectativas. Isto requer do terceirizado fornecer as informações significativas e proativas sobre o status do projeto. Também, você deve estar confortável com os processos que o terceirizado está usando para gerenciar o projeto.

A garantia de qualidade refere-se aos processos que estão sendo usados para gerenciar o projeto e construir os deliverables. (Isto está no contraste do controle de qualidade, que se refere às atividades para criar os deliverables.) Desde que você não seja o responsável pela criação dos deliverables (controle de qualidade) você necessita estar confortável de que o terceirizado está usando processos sadios (garantia de qualidade) para gerenciar o projeto e construir os deliverables.

Esta lista de verificação da garantia de qualidade para projetos terceirizados contém os critérios necessários para verificar:

- Que o acordo contém a informação necessária para gerenciar subsequentemente o relacionamento. Por exemplo, se você quiser os relatórios periódicos, você necessita certificar-se que está incluído na definição ou no contrato do projeto. Se você quiser validar que os deliverables ínterim são aceitáveis a sua empresa, você necessita concordar com os Critérios de Precisão e Conclusão antes do tempo.
- Que o projeto está prosseguindo na trilha e que você sabe sobre todos os desvios do plano. Estas perguntas contínuas podem ser validadas em datas programadas, ou no acordado previamente das entregas do projeto.
- Que o projeto será concluído dentro de suas expectativas baseado na definição e no contrato do projeto.

Esta lista de verificação contém as perguntas básicas. Esta lista deve ser suplementada com as perguntas que são específicas para seu projeto e para seu contrato com o terceirizado. Se qualquer parte do contrato fizer acordos ou assumir compromissos fora do contrato, estes devem ser adicionados na lista de verificação de modo que possa ser monitorado durante todo o projeto.

4.1.3 Plano de Testes

[TEMPLATE DE PLANO DE TESTES]

<Sigla da Estatal>

<Nome do Projeto>

Histórico da Revisão

Data	Versão	Descrição	Autor	Área/Setor
<dd/mm/aaaa>	<x.x>	<detalhes>	<nome>	

Identificação do Plano de Testes: <inserir o Código do Plano de Testes.>

<inserir o Nome do Plano de Testes>

1. Introdução

1.1. Objeto

<Descrição do software a ser testado>

1.2. Objetivo

<Descrição do propósito do teste realizado>

2. Escopo

<Descrição dos diversos tipos de testes que fazem parte deste plano e a lista prioritizada dos itens de software a serem testados. Também faz parte do escopo a lista de itens de software que não serão testados. Para cada item devem ser especificadas as funcionalidades e características que serão e as que não serão testadas>

3. Abordagem

<Especifica a forma de realização dos testes. Abrange, entre outros aspectos, as técnicas, ferramentas e restrições, além disso, são definidos critérios para iniciação, aprovação e encerramento dos testes. também são definidas as condições para a suspensão e retomada dos testes.>

4. Missão de Avaliação e Motivação dos Testes

[Forneça uma visão geral da missão e da motivação dos testes que serão conduzidos nesta iteração]

4.1 Fundamentos

[Forneça uma breve descrição dos fundamentos que justificam o esforço de teste definido neste Plano de Teste. Inclua informações como, por exemplo, o problema principal que está sendo resolvido, os principais benefícios da solução, a arquitetura planejada da solução e um breve histórico do projeto. Quando essas informações estiverem definidas em outros documentos, você poderá incluir referências a esses documentos mais detalhados caso seja apropriado. Esta seção só deverá conter de três a cinco parágrafos]

4.2 Missão de Avaliação

[Forneça uma breve sentença que defina a missão do esforço de avaliação na iteração atual. Essa sentença poderá incorporar uma ou mais preocupações incluindo:

- localizar o maior número de erros possível*
- localizar problemas importantes*
- avaliar os riscos da qualidade perceptível*
- informar sobre os riscos perceptíveis do projeto*
- certificar segundo um padrão*
- verificar uma especificação (requisitos, design ou alegações)*
- informar sobre a qualidade do produto*
- satisfazer os envolvidos*
- informar sobre os testes*
- cumprir as determinações do processo*
- e assim por diante*

Cada missão fornece um contexto diferente para o esforço de teste e altera a maneira como o teste deverá ser abordado.]

4.3 Motivadores dos Testes

[Forneça um resumo dos principais elementos que motivarão o esforço de teste nesta iteração. Os testes poderão ser motivados por uma série de fatores como, por exemplo, riscos de qualidade, riscos técnicos, riscos do projeto, casos de uso, requisitos funcionais, requisitos não funcionais, elementos de design, falhas ou erros suspeitos, solicitações de mudança etc.]

5. Itens de Teste-Alvo

A listagem abaixo identifica os itens de software, de hardware e elementos de suporte do produto que foram identificados como objetivos dos testes. Esta lista representa os itens que serão testados.

[Forneça uma lista de nível superior dos principais itens que estarão sujeitos a teste. Essa lista deve incluir itens produzidos diretamente pela equipe de desenvolvimento do projeto e itens de que dependem esses produtos. Por exemplo, o hardware de processamento básico, dispositivos periféricos, sistemas operacionais, produtos ou componentes de terceiros etc. É recomendável agrupar a lista por categoria e atribuir importância relativa a cada motivador.]

6. Resumo dos Testes Planejados

[Esta seção apresenta os recursos recomendados para o projeto <Nome do Projeto>, suas principais responsabilidades e seu conjunto de conhecimentos ou de habilidade.]

6.1 Resumo das Inclusões dos Testes

[Esta seção fornece um resumo de nível superior dos testes que serão executados. O resumo fornecido aqui representa uma visão geral de nível superior dos testes que serão e dos que não serão executados]

6.2 Resumo dos Outros Candidatos a Possível Inclusão

[Descreva separadamente as áreas de teste cuja avaliação e investigação você supõe que poderão ser úteis, mas que ainda não foram suficientemente pesquisadas para justificar com certeza a importância de examiná-las.]

6.3 Resumo das Exclusões dos Testes

[Forneça um resumo de nível superior dos possíveis testes que poderiam ter sido conduzidos, mas que foram explicitamente excluídos deste plano. Se você não for implementar ou executar um tipo de teste, informe claramente que o

teste não será executado ou implementado e justifique por que. A seguir, há exemplos de justificativas que poderão ser usadas:

- *"Esses testes não contribuem para alcançar a missão de avaliação."*
- *"Não há recursos suficientes para executar esses testes."*
- *"Esses testes são desnecessários devido aos testes executados por xxxx."*

Segundo um prisma heurístico, se você achar que é perfeitamente concebível que um dos membros de seu público espere que um determinado aspecto de teste seja incluído e se você não pretender ou não puder incluí-lo, justifique sua exclusão. Se a equipe concordar que a exclusão é óbvia, você provavelmente não precisará listá-la.]

7. Abordagem dos Testes

[Esta seção apresenta a estratégia recomendada para criar e implementar os testes necessários. As seções 3, Itens de Teste-Alvo, e 4, Resumo dos Testes Planejados, identificaram que itens serão testados e que tipos de testes serão executados. Esta seção descreve como esses testes serão realizados.

Um aspecto a ser considerado na abordagem dos testes é as técnicas que serão usadas. Deverá ser incluído um resumo de como cada técnica poderá ser implementada, de uma perspectiva manual e/ou automatizada, e os critérios para comprovar que a técnica é útil e eficaz. Para cada técnica, forneça uma descrição a seu respeito e defina por que é uma parte importante da abordagem dos testes resumindo brevemente como ela ajuda a alcançar a Missão de Avaliação ou como aborda os Motivadores dos Testes.

Outro aspecto a ser discutido nesta seção é os modelos de Erro ou Falha que são aplicáveis e as maneiras de abordar como avaliá-los.

À medida que definir cada aspecto da abordagem, você deverá atualizar a seção 10, Responsabilidades, Perfil da Equipe e Necessidades de Treinamento, para documentar a configuração do ambiente de teste e outros recursos que serão necessários para implementar cada aspecto.]

7.1 Catálogos Iniciais de Ideias de Teste e Outras Fontes de Referência

[Forneça uma listagem dos recursos existentes que serão consultados para estimular a identificação e a seleção de testes específicos a serem conduzidos. É fornecido um Catálogo de Ideias de Teste de exemplo na seção de exemplos do RUP.]

7.2 Tipos e Técnicas de Teste

7.2.1 Teste de Integridade de Dados e de Banco de Dados

[Os bancos de dados e os processos de banco de dados deverão ser testados como um subsistema independente. Esse teste deve testar os subsistemas sem que a Interface do Usuário do objetivo do teste faça interface com os dados. É necessário efetuar pesquisas adicionais referentes ao Sistema de Gerenciamento de Banco de Dados (DBMS) a fim de identificar as ferramentas e técnicas que poderão existir para suportar os testes identificados na tabela a seguir.]

<p>Objetivo Técnico:</p>	<p>da</p> <p><i>[Experimentar processos e métodos de acesso a banco de dados independentes da UI para que você possa observar e registrar comportamentos-alvo incorretos ou a existência de dados corrompidos.]</i></p>
<p>Técnica:</p>	<p><i>[Dispense cada processo e método de acesso a banco de dados, propagando dados válidos e inválidos ou solicitações de dados em cada um deles.</i></p> <p><i>Inspeccione o banco de dados para assegurar que os dados foram distribuídos conforme o planejado e que todos os eventos de banco de dados ocorreram de forma adequada, ou revise os dados retornados para assegurar que os dados corretos foram recuperados pelas razões corretas.]</i></p>
<p>Estratégias:</p>	<p><i>[Descreva uma ou mais estratégias que podem ser usadas pela técnica para observar, de forma precisa, os resultados do teste. A estratégia combina elementos do método através do qual a observação pode ser feita e das características dos resultados específicos que indicam um provável êxito ou falha do teste. O ideal é que as estratégias sejam auto verificadas, permitindo que os testes automatizados façam uma avaliação inicial do êxito ou falha do teste. No entanto, tenha atenção para reduzir os riscos inerentes à determinação automática dos resultados.]</i></p>

<p>Ferramentas Necessárias:</p>	<p><i>[A técnica exige as seguintes ferramentas: Ferramenta de Automação de Scripts de Teste restaurador e reprodutor de imagem da configuração básica ferramentas de backup e de recuperação ferramentas de monitoramento de instalação (registro, disco rígido, CPU, memória etc.) ferramentas e utilitários SQL de banco de dados ferramentas de geração de dados]</i></p>
<p>CrITÉrios de Êxito:</p>	<p><i>[A técnica suporta o teste de todos os principais processos e métodos de acesso a banco de dados]</i></p>
<p>Considerações Especiais:</p>	<p><i>[Os testes poderão necessitar de drivers ou de um ambiente de desenvolvimento DBMS para inserir ou modificar dados diretamente no banco de dados. Os processos deverão ser disparados manualmente. Deverão ser usados bancos de dados pequenos ou de tamanho mínimo (com um número limitado de registros) para aumentar a visibilidade de quaisquer eventos não aceitáveis.]</i></p>

7.2.2 Teste de Funcionamento

[O teste de funcionamento do objetivo do teste deve concentrar-se em todos os requisitos de teste que possam ser diretamente associados a casos de uso ou funções e regras de negócios. Esse teste tem por fim verificar a adequada aceitação, processamento e recuperação dos dados, e a implementação apropriada das regras de negócios. Esse tipo de teste baseia-se em técnicas de caixa preta; ou seja, verificar o aplicativo e seus processos internos interagindo com o aplicativo através da Interface Gráfica do Usuário (GUI) e analisar a saída ou os resultados. A tabela a seguir identifica um resumo do teste recomendado para cada aplicativo]

<p>Objetivo Técnico:</p>	<p>da</p> <p><i>[Experimentar a funcionalidade do objetivo do teste, incluindo a navegação, a entrada, o processamento e a recuperação de dados a fim de observar e registrar o comportamento-alvo.]</i></p>
<p>Técnica:</p>	<p><i>[Experimentar os recursos e fluxos ou funções de cada um dos cenários de caso de uso, utilizando dados válidos e inválidos para verificar se:</i></p> <p><i>os resultados esperados ocorrerão quando forem usados dados válidos</i></p> <p><i>as mensagens de erro ou de aviso apropriadas serão exibidas quando forem usados dados inválidos</i></p> <p><i>cada regra de negócio será aplicada de forma adequada]</i></p>
<p>Estratégias:</p>	<p><i>[Descreva uma ou mais estratégias que podem ser usadas pela técnica para observar, de forma precisa, os resultados do teste. A estratégia combina elementos do método através do qual a observação pode ser feita e das características dos resultados específicos que indicam um provável êxito ou falha do teste. O ideal é que as estratégias sejam auto verificadas, permitindo que os testes automatizados façam uma avaliação inicial do êxito ou falha do teste. No entanto, tenha atenção para reduzir os riscos inerentes à determinação automática dos resultados.]</i></p>

<p>Ferramentas Necessárias:</p>	<p><i>[A técnica exige as seguintes ferramentas: Ferramenta de Automação de Scripts de Teste restaurador e reprodutor de imagem da configuração básica ferramentas de backup e de recuperação ferramentas de monitoramento de instalação (registro, disco rígido, CPU, memória etc.) ferramentas de geração de dados]</i></p>
<p>CrITÉrios de Êxito:</p>	<p><i>[A técnica suporta o teste de: todos os principais cenários de caso de uso todos os principais recursos]</i></p>
<p>Considerações Especiais:</p>	<p><i>[Identifique ou descreva os itens ou problemas (internos ou externos) que exercem influência sobre a implementação e a execução do teste de funcionamento.]</i></p>

7.2.3 Teste de Ciclos de Negócios

[O Teste de Ciclos de Negócios deverá emular as atividades executadas no <Nome do Projeto> ao longo do tempo. Deverá ser identificado um período como, por exemplo, um ano, e deverão ser executadas as transações e atividades que ocorreriam durante esse período de um ano. Isso incluirá todos os ciclos diários, semanais e mensais, assim como os eventos que mudam com as datas como, por exemplo, lembretes.]

<p>Objetivo da Técnica:</p>	<p><i>[Experimentar processos de segundo plano e do objetivo do teste de acordo com as programações e os modelos de negócios necessários, a fim de observar e registrar o comportamento-alvo.]</i></p>
------------------------------------	--

Técnica:	<p><i>[O teste simulará vários ciclos de negócios executando o seguinte:</i></p> <p><i>Os testes destinados a inspecionar o funcionamento do objetivo do teste serão modificados ou melhorados para aumentar o número de vezes que cada função é executada, a fim de simular vários usuários diferentes ao longo de um período de tempo especificado.</i></p> <p><i>Todas as funções que mudam com as datas ou o tempo serão executadas usando datas ou períodos de tempo válidos e inválidos.</i></p> <p><i>Todas as funções que ocorrerem segundo uma programação periódica serão executadas ou iniciadas no momento adequado.</i></p> <p><i>O teste incluirá o uso de casos válidos e inválidos para verificar se:</i></p> <ul style="list-style-type: none"><i>• Os resultados esperados ocorrerão quando forem usados dados válidos.</i><i>• As mensagens de erro ou de aviso apropriadas serão exibidas quando forem usados dados inválidos.</i><i>• Cada regra de negócio será aplicada de forma adequada.]</i>
Estratégias:	<p><i>[Descreva uma ou mais estratégias que podem ser usadas pela técnica para observar, de forma precisa, os resultados do teste. A estratégia combina elementos do método através do qual a observação pode ser feita e das características dos resultados específicos que indicam um provável êxito ou falha do teste. O ideal é que as estratégias sejam auto verificadas, permitindo que os testes automatizados façam uma avaliação inicial do êxito ou falha do teste. No entanto, tenha atenção para reduzir os riscos inerentes à determinação automática dos resultados.]</i></p>

<p>Ferramentas Necessárias:</p>	<p><i>[A técnica exige as seguintes ferramentas: Ferramenta de Automação de Scripts de Teste restaurador e reproduzidor de imagem da configuração básica ferramentas de backup e de recuperação ferramentas de geração de dados]</i></p>
<p>CrITÉrios de Êxito:</p>	<p><i>[A técnica suporta o teste de todos os ciclos de negócios essenciais.]</i></p>
<p>Considerações Especiais:</p>	<p><i>[Os eventos e as datas do sistema poderão exigir atividades de suporte especiais. É necessário um modelo de negócios para identificar requisitos e procedimentos de teste adequados.]</i></p>

7.2.4 Teste de Interface do Usuário

[O Teste de Interface do Usuário (UI) verifica a interação do usuário com o software. O teste de UI tem por fim assegurar que a UI forneça ao usuário o acesso e a navegação adequados através das funções do objetivo do teste. Além disso, o teste de UI assegura que os objetos contidos na UI funcionem conforme o esperado e estejam em conformidade com padrões corporativos ou da indústria.]

<p>Objetivo Técnico:</p>	<p><i>[Experimentar o seguinte para observar e registrar a conformidade com padrões e o comportamento-alvo: A navegação pelo objetivo do teste para verificar se reflete os requisitos e funções de negócios, incluindo a navegação janela a janela e campo a campo, e o uso de métodos de acesso (teclas de tabulação, movimentos do mouse e teclas aceleradoras). As características e os objetos das</i></p>
---------------------------------	---

	<p><i>janelas poderão ser experimentados como, por exemplo, menus, tamanho, posição, estado e foco.]</i></p>
Técnica:	<p><i>[Crie ou modifique testes para cada janela a fim de verificar a navegação adequada e os estados de objeto apropriados para cada janela e objeto do aplicativo.]</i></p>
Estratégias:	<p><i>[Descreva uma ou mais estratégias que podem ser usadas pela técnica para observar, de forma precisa, os resultados do teste. A estratégia combina elementos do método através do qual a observação pode ser feita e das características dos resultados específicos que indicam um provável êxito ou falha do teste. O ideal é que as estratégias sejam auto</i></p> <p><i>verificadas, permitindo que os testes automatizados façam uma avaliação inicial do êxito ou falha do teste. No entanto, tenha atenção para reduzir os riscos inerentes à determinação automática dos resultados.]</i></p>
Ferramentas Necessárias:	<p><i>[A técnica necessita da Ferramenta de Automação de Scripts de Teste.]</i></p>
Crítérios de Êxito:	<p><i>[A técnica suporta o teste de cada tela ou janela principal que será muito usada pelo usuário final.]</i></p>

Considerações Especiais:	<i>[Nem todas as propriedades referentes a objetos personalizados e de terceiros poderão ser acessadas.]</i>
---------------------------------	--

7.2.5 *Determinação do Perfil de Desempenho*

[Este é um teste de desempenho em que os tempos de resposta, as taxas de transação e outros requisitos que mudam com o tempo são medidos e avaliados. Este teste tem por fim verificar se os requisitos de desempenho foram alcançados. Ele é implementado e executado para determinar o perfil dos comportamentos de desempenho do objetivo do teste e ajustá-los em função de condições como, por exemplo, configurações de hardware ou de carga de trabalho.]

Observação: As transações da tabela a seguir são "transações de negócios lógicas". Essas transações são definidas como casos de uso específicos que se espera que um ator do sistema execute utilizando o objetivo do teste como, por exemplo, adicionar ou modificar um determinado contrato.]

Objetivo Técnico:	<p><i>da</i></p> <p><i>[Experimentar comportamentos referentes a funções de negócios ou transações funcionais designadas nas condições abaixo, a fim de observar e registrar o comportamento-alvo e os dados de desempenho do aplicativo:</i></p> <p><i>carga de trabalho antecipada normal</i></p> <p><i>carga de trabalho antecipada no pior caso]</i></p>
Técnica:	<p><i>[Use os Procedimentos de Teste desenvolvidos para o Teste de Ciclos de Negócios ou de Funcionamento.</i></p> <p><i>Modifique os arquivos de dados a fim de aumentar o número de transações ou modifique os scripts a fim de aumentar o número de iterações que ocorrem em cada transação.</i></p> <p><i>Os scripts deverão ser executados em uma máquina (o melhor é avaliar o desempenho de um único usuário, uma única transação) e deverão ser repetidos para vários clientes (virtuais ou reais, consulte Considerações Especiais abaixo).]</i></p>

<p>Estratégias:</p>	<p><i>[Descreva uma ou mais estratégias que podem ser usadas pela técnica para observar, de forma precisa, os resultados do teste. A estratégia combina elementos do método através do qual a observação pode ser feita e das características dos resultados específicos que indicam um provável êxito ou falha do teste. O ideal é que as estratégias sejam auto verificadas, permitindo que os testes automatizados façam uma avaliação inicial do êxito ou falha do teste. No entanto, tenha atenção para reduzir os riscos inerentes à determinação automática dos resultados.]</i></p>
<p>Ferramentas Necessárias:</p>	<p><i>[A técnica exige as seguintes ferramentas:</i></p> <p><i>Ferramenta de Automação de Scripts de Teste</i></p> <p><i>uma ferramenta para a determinação do perfil de desempenho do aplicativo como, por exemplo, o Rational Quantify</i></p> <p><i>ferramentas de monitoramento de instalação (registro, disco rígido, CPU, memória etc.)</i></p> <p><i>ferramentas de restrição de recursos como, por exemplo, enlatados]</i></p>
<p>Critérios de Êxito:</p>	<p><i>[A técnica suporta o teste de:</i></p> <p><i>Uma única transação ou um único usuário: uma emulação bem-sucedida dos scripts de transação sem que ocorra nenhuma falha devido a problemas de implementação do teste.</i></p> <p><i>Várias transações ou vários usuários: uma emulação bem-sucedida da carga de trabalho sem que ocorra nenhuma falha devido a problemas de implementação do teste.]</i></p>

<p>Considerações Especiais:</p>	<p><i>[O teste abrangente do desempenho inclui ter uma carga de trabalho em segundo plano no servidor.</i></p> <p><i>Há vários métodos que podem ser usados para executar esse teste, incluindo:</i></p> <p><i>"Encaminhar as transações" diretamente para o servidor, geralmente como chamadas de Linguagem de Consulta Estruturada (SQL).</i></p> <p><i>Criar carga de usuário "virtual" para simular muitos clientes, geralmente algumas centenas deles. Para se obter essa carga, geralmente são usadas ferramentas de Emulação de Terminal Remoto. Essa técnica também pode ser usada para que a rede fique repleta de "tráfego".</i></p> <p><i>Usar vários clientes físicos, cada qual executando scripts de teste, para inserir carga no sistema.</i></p> <p><i>O teste de desempenho deverá ser executados em uma máquina dedicada ou em um período de tempo dedicado. Isso permitirá o controle total e a medição exata.</i></p> <p><i>Os bancos de dados usados para o Teste de Determinação de Perfil de Desempenho deverá ter um tamanho real ou deverão ser dimensionados igualmente em escala.]</i></p>
--	---

7.2.6 Teste de Carga

[O teste de carga é um teste de desempenho que sujeita o objetivo do teste a diferentes cargas de trabalho para medir e avaliar as habilidades e os comportamentos de desempenho dele, a fim de verificar se este continua a funcionar adequadamente com essas diferentes cargas de trabalho. O teste de carga tem por fim determinar e assegurar que o sistema funcione adequadamente com uma carga de trabalho superior à carga máxima esperada. Além disso, esse teste avalia as características do desempenho como, por exemplo, tempos de resposta, taxas de transação e outros aspectos que mudam com o tempo.

[Observação: As transações da tabela a seguir são "transações de negócios lógicas". Essas transações são definidas como funções específicas que se

espera que um usuário final do sistema execute utilizando o aplicativo como, por exemplo, adicionar ou modificar um determinado contrato.]

<p>Objetivo Técnico:</p>	<p><i>da</i></p> <p><i>[Experimentar casos de negócio ou transações designadas em várias condições de carga de trabalho, a fim de observar e registrar o comportamento-alvo e os dados de desempenho do sistema.]</i></p>
<p>Técnica:</p>	<p><i>[Use os Scripts de Teste de Transação desenvolvidos para os Testes de Ciclos de Negócios ou de Funcionamento como uma base, mas lembre-se de remover as iterações e os atrasos desnecessários.</i></p> <p><i>Modifique os arquivos de dados a fim de aumentar o número de transações ou modifique os testes a fim de aumentar o número de vezes que cada transação ocorre.</i></p> <p><i>As cargas de trabalho devem incluir cargas de pico — por exemplo, diárias, semanais e mensais.</i></p> <p><i>As cargas de trabalho devem representar cargas médias assim como cargas de pico.</i></p> <p><i>As cargas de trabalho devem representar picos instantâneos e picos sustentados.</i></p> <p><i>As cargas de trabalho devem ser executadas com diferentes Configurações do Ambiente de Teste.]</i></p>
<p>Estratégias:</p>	<p><i>[Descreva uma ou mais estratégias que podem ser usadas pela técnica para observar, de forma precisa, os resultados do teste. A estratégia combina elementos do método através do qual a observação pode ser feita e das características dos resultados específicos que indicam um provável êxito ou falha do teste. O ideal é que as estratégias sejam auto verificadas, permitindo que os testes automatizados façam uma avaliação inicial do êxito ou falha do teste. No entanto, tenha atenção para reduzir os riscos inerentes à determinação automática dos resultados.]</i></p>

<p>Ferramentas Necessárias:</p>	<p><i>[A técnica exige as seguintes ferramentas:</i></p> <p><i>Ferramenta de Automação de Scripts de Teste</i></p> <p><i>Ferramenta de controle e de programação de carga de transações</i></p> <p><i>ferramentas de monitoramento de instalação (registro, disco rígido, CPU, memória etc.)</i></p> <p><i>ferramentas de restrição de recursos como, por exemplo, enlatados</i></p> <p><i>ferramentas de geração de dados]</i></p>
<p>CrITÉrios de Êxito:</p>	<p><i>[A técnica suporta o teste de Emulação de Carga de Trabalho, que é a emulação bem-sucedida da carga de trabalho sem que nenhuma falha ocorra devido a problemas de implementação do teste.]</i></p>
<p>Considerações Especiais:</p>	<p><i>[O teste de carga deverá ser executado em uma máquina dedicada ou em um período de tempo dedicado. Isso permitirá o controle total e a medição exata.</i></p> <p><i>Os bancos de dados usados para o teste de carga deverão ter um tamanho real ou deverão ser dimensionados igualmente em escala.]</i></p>

7.2.7 Teste de Stress

[O teste de stress é um tipo de teste de desempenho implementado e executado para compreender como ocorrem falhas no sistema devido a condições que estão no limite ou fora do limite das tolerâncias esperadas. Normalmente isso envolve poucos recursos ou a concorrência por recursos. As condições de poucos recursos revelam como ocorrem falhas no objetivo do teste que não estão aparentes em condições normais. Outros defeitos poderão resultar de uma concorrência por recursos compartilhados como, por exemplo, bloqueios de banco de dados ou largura de banda de rede, embora alguns desses testes sejam geralmente abordados nos testes funcionais e de carga.

Observação: As transações mencionadas na tabela a seguir são transações de negócios lógicas.]

<p>Objetivo Técnico:</p>	<p>da</p> <p><i>[Experimentar as funções do objetivo do teste nas seguintes condições de stress a fim de observar e registrar o comportamento-alvo que identifica e documenta as condições que fazem com que o sistema deixe de funcionar adequadamente:</i></p> <p><i>pouca ou nenhuma memória disponível no servidor (memória RAM e espaço de armazenamento persistente)</i></p> <p><i>número máximo real ou fisicamente capaz de clientes conectados ou simulados</i></p> <p><i>vários usuários executando as mesmas transações nos mesmos dados ou contas</i></p> <p><i>conjunto ou volume de transações que geram "sobrecarga" (consulte Determinação do Perfil de Desempenho acima)]</i></p>
<p>Técnica:</p>	<p><i>[Use os testes de Carga ou de Determinação do Perfil de Desempenho.</i></p> <p><i>Para testar recursos limitados, os testes deverão ser executados em uma única máquina, e a memória RAM e o espaço de armazenamento persistente no servidor deverão ser reduzidos ou limitados.</i></p> <p><i>Para os testes de stress restantes, deverão ser usados vários clientes, executando-se os mesmos testes ou testes complementares a fim de produzir o conjunto ou volume de transações no pior caso.]</i></p>
<p>Estratégias:</p>	<p><i>[Descreva uma ou mais estratégias que podem ser usadas pela técnica para observar, de forma precisa, os resultados do teste. A estratégia combina elementos do método através do qual a observação pode ser feita e das características dos resultados específicos que indicam um provável êxito ou falha do teste. O ideal é que as estratégias sejam auto verificadas, permitindo que os testes automatizados façam uma avaliação inicial do êxito ou falha do teste. No entanto, tenha atenção para reduzir os riscos</i></p>

	<p><i>inerentes à determinação automática dos resultados.]</i></p>
Ferramentas Necessárias:	<p><i>[A técnica exige as seguintes ferramentas:</i></p> <p><i>Ferramenta de Automação de Scripts de Teste</i></p> <p><i>Ferramenta de controle e de programação de carga de transações</i></p> <p><i>ferramentas de monitoramento de instalação (registro, disco rígido, CPU, memória etc.)</i></p> <p><i>ferramentas de restrição de recursos como, por exemplo, enlatados</i></p> <p><i>ferramentas de geração de dados]</i></p>
Critérios de Êxito:	<p><i>[A técnica suporta o teste de Emulação de Pressão. O sistema poderá ser emulado, de maneira eficaz, em uma ou mais condições definidas como condições de stress, e poderá ser capturada uma observação do estado resultante do sistema durante e depois de a condição ter sido emulada.]</i></p>

<p>Considerações Especiais:</p>	<p><i>[Para gerar stress na rede talvez seja necessário que as ferramentas da rede a sobrecarreguem com mensagens ou pacotes.</i></p> <p><i>O armazenamento persistente usado para o sistema deverá ser reduzido temporariamente a fim de restringir o espaço disponível para que o banco de dados se desenvolva.</i></p> <p><i>Sincronize o acesso simultâneo dos clientes aos mesmos registros ou contas de dados.]</i></p>
--	---

7.2.8 Teste de Volume

[O teste de volume sujeita o objetivo do teste a grandes volumes de dados a fim de determinar se serão atingidos limites que farão com que o software deixe de funcionar. Esse teste também identifica o volume ou carga máxima contínua que o objetivo do teste pode suportar durante um determinado período de tempo. Por exemplo, se o objetivo do teste estiver processando um conjunto de registros de banco de dados para gerar um relatório, um Teste de Volume usará um grande banco de dados de testes e verificará se o software se comportou normalmente e gerou o relatório correto.]

<p>Objetivo da Técnica:</p>	<p><i>[Experimentar as funções do objetivo do teste nos seguintes cenários de elevado volume para observar e registrar o comportamento-alvo:</i></p> <p><i>O número máximo (real ou fisicamente capaz) de clientes conectados, ou simulados, todos executando a mesma função de negócios (desempenho), no pior caso, durante um longo período de tempo.</i></p> <p><i>Foi atingido o tamanho máximo do banco de dados (real ou em escala) e várias consultas ou transações de relatório são executadas simultaneamente.]</i></p>
------------------------------------	--

<p>Técnica:</p>	<p><i>[Use os testes de Carga ou de Determinação do Perfil de Desempenho.</i></p> <p><i>Deverão ser usados vários clientes, executando-se os mesmos testes ou testes complementares a fim de produzir o conjunto ou volume de transações no pior caso (consulte Teste de Stress) durante um longo período de tempo.</i></p> <p><i>Será criado o tamanho máximo do banco de dados (real, em escala ou preenchido com dados representativos) e serão usados vários clientes para executar consultas e transações de relatório simultaneamente durante longos períodos de tempo.]</i></p>
<p>Estratégias:</p>	<p><i>[Descreva uma ou mais estratégias que podem ser usadas pela técnica para observar, de forma precisa, os resultados do teste. A estratégia combina elementos do método através do qual a observação pode ser feita e das características dos resultados específicos que indicam um provável êxito ou falha do teste. O ideal é que as estratégias sejam auto verificadas, permitindo que os testes automatizados façam uma avaliação inicial do êxito ou falha do teste. No entanto, tenha atenção para reduzir os riscos inerentes à determinação automática dos resultados.]</i></p>
<p>Ferramentas Necessárias:</p>	<p><i>[A técnica exige as seguintes ferramentas:</i></p> <p><i>Ferramenta de Automação de Scripts de Teste</i></p> <p><i>Ferramenta de controle e de programação de carga de transações</i></p> <p><i>ferramentas de monitoramento de instalação (registro, disco rígido, CPU, memória etc.)</i></p> <p><i>ferramentas de restrição de recursos como, por exemplo, enlatados</i></p> <p><i>ferramentas de geração de dados]</i></p>

<p>Crítérios de Êxito:</p>	<p><i>[A técnica suporta o teste de Emulação de Volume. É possível emular, de forma eficaz, grandes quantidades de usuários, dados, transações ou outros aspectos do sistema utilizados em volume e poderá ser capturada uma observação sobre as mudanças de estado do sistema durante o teste de volume.]</i></p>
<p>Considerações Especiais:</p>	<p><i>[Que período de tempo seria considerado aceitável para as condições de elevado volume, conforme observado acima?]</i></p>

7.2.9 Teste de Segurança e de Controle de Acesso

[O Teste de Segurança e de Controle de Acesso concentra-se em duas áreas de segurança principais:

Segurança no nível do aplicativo, incluindo o acesso aos Dados ou às Funções de Negócios

Segurança no nível do sistema, incluindo efetuar login ou acessar remotamente o sistema

Com base no nível de segurança desejado, a segurança no nível do aplicativo assegura que os atores estejam restritos a funções ou casos de uso específicos, ou que tenham acesso limitado aos dados disponíveis. Por exemplo, todos têm permissão para inserir dados e criar novas contas, mas apenas os gerentes poderão excluí-los. Se houver segurança no nível dos dados, o teste assegurará que o "tipo de usuário um" possa ver todas as informações de um cliente, incluindo dados financeiros. No entanto, o "tipo de usuário dois" somente verá os dados demográficos referentes ao mesmo cliente.

A segurança no nível do sistema assegura que somente os usuários a que tenha sido concedido acesso ao sistema serão capazes de acessar os aplicativos e somente através dos gateways apropriados.]

<p>Objetivo Técnico:</p>	<p>da</p> <p><i>[Experimentar o objetivo do teste nas seguintes condições para observar e registrar o comportamento-alvo:</i></p> <p><i>Segurança no nível do aplicativo: um ator poderá acessar somente as funções ou os dados para o quais seu tipo de usuário tenha recebido permissão.</i></p> <p><i>Segurança no nível do sistema: somente os atores com acesso ao sistema e aos aplicativos têm permissão para acessá-los].</i></p>
<p>Técnica:</p>	<p><i>[Segurança no nível do aplicativo: Identifique e liste cada tipo de usuário e as funções ou os dados para os quais cada tipo tem permissão de acesso.</i></p> <ul style="list-style-type: none"> ● <i>Crie testes para cada tipo de usuário e verifique cada permissão criando transações específicas para cada tipo de usuário.</i> ● <i>Modifique o tipo de usuário e execute novamente os testes para os mesmos usuários. Em cada caso, verifique se as funções ou dados adicionais estão corretamente disponíveis ou se têm seu acesso negado.</i> <p><i>Acesso no nível do sistema: Consulte Considerações Especiais abaixo.]</i></p>
<p>Estratégias:</p>	<p><i>[Descreva uma ou mais estratégias que podem ser usadas pela técnica para observar, de forma precisa, os resultados do teste. A estratégia combina elementos do método através do qual a observação pode ser feita e das características dos resultados específicos que indicam um provável êxito ou falha do teste. O ideal é que as estratégias sejam auto verificadas, permitindo que os testes automatizados façam uma avaliação inicial do êxito ou falha do teste. No entanto, tenha atenção para reduzir os riscos inerentes à determinação automática dos resultados.]</i></p>

<p>Ferramentas Necessárias:</p>	<p><i>[A técnica exige as seguintes ferramentas: Ferramenta de Automação de Scripts de Teste Ferramentas de investigação e contra a violação da segurança por "hackers" Ferramentas de Administração da Segurança do Sistema Operacional]</i></p>
<p>Critérios de Êxito:</p>	<p><i>[A técnica suporta o teste das funções apropriadas. É possível também que os dados afetados pelas configurações de segurança sejam testados para cada tipo de ator conhecido.]</i></p>
<p>Considerações Especiais:</p>	<p><i>[O acesso ao sistema deverá ser revisto ou discutido com o administrador de sistemas ou de rede adequado. Talvez esse teste não seja necessário, já que poderá ser uma das funções da administração de sistemas ou de rede.]</i></p>

7.2.10 Teste de Tolerância a Falhas e de Recuperação

[O teste de tolerância a falhas e de recuperação assegura que o objetivo do teste possa tolerar e se recuperar de uma série de falhas de hardware, software ou de rede com perda indevida de dados ou da integridade dos dados.

Para os sistemas que devem ser mantidos em execução, o teste de tolerância a falhas assegura que, ao ocorrer uma condição de tolerância a falhas, os sistemas alternativos ou de backup "assumirão" adequadamente o papel do sistema danificado sem qualquer perda de dados ou transações.

O teste de recuperação é um processo de teste antagonista em que o aplicativo ou o sistema é exposto a condições extremas, ou condições simuladas, para gerar falhas como, por exemplo, falhas de Entrada/Saída (E/S) de Dispositivo, ou ponteiros e chaves de banco de dados inválidos. Os processos de recuperação são disparados e o aplicativo ou o sistema é monitorado e inspecionado para verificar se foi efetuada a recuperação adequada do aplicativo ou do sistema e de dados.]

<p>Objetivo Técnico:</p>	<p>da</p> <p><i>[Simular as condições de falha e experimentar os processos de recuperação (manuais e automatizados) para restaurar o estado conhecido e desejado do banco de dados, dos aplicativos e do sistema. Os seguintes tipos de condições estão incluídos no teste para observar e registrar o comportamento após a recuperação:</i></p> <ul style="list-style-type: none"><i>interrupção da energia para o cliente</i><i>interrupção da energia para o servidor</i><i>interrupção da comunicação através dos servidores de rede</i><i>perda da comunicação ou interrupção da energia para os DASD (Dynamic Access Storage Devices, Dispositivos de Armazenamento de Acesso Dinâmico) e os controladores DASD</i><i>ciclos incompletos (processos de filtragem de dados interrompidos, processos de sincronização de dados interrompidos)</i><i>ponteiros ou chaves de banco de dados inválidos</i><i>elementos de dados inválidos ou corrompidos no banco de dados]</i>
---------------------------------	---

<p>Técnica:</p>	<p><i>[Os testes de Funcionamento e de Ciclos de Negócios poderão ser usados como uma base para criar uma série de transações para suportar os testes de tolerância a falhas e de recuperação e principalmente para definir os testes que serão executados para verificar se a recuperação teve êxito.</i></p> <p><i>Interrupção da energia para o cliente: desligue o PC.</i></p> <p><i>Interrupção da energia para o servidor: simule ou inicie procedimentos de desligamento do servidor.</i></p> <p><i>Interrupção através de servidores de rede: simule ou inicie uma perda de comunicação com a rede (desconecte fisicamente os cabos de comunicação ou desligue os servidores ou roteadores de rede).</i></p> <p><i>Perda da comunicação ou interrupção da energia para os DASD e os controladores DASD: simule ou elimine fisicamente a comunicação com um ou mais DASDs ou controladores DASD.</i></p> <p><i>Depois que as condições acima ou as condições simuladas tiverem sido alcançadas, as transações adicionais deverão ser executadas e, quando o estado desse segundo ponto do teste for atingido, os procedimentos de recuperação deverão ser disparados.</i></p> <p><i>O teste de ciclos incompletos utiliza a mesma técnica descrita acima, exceto pelos processos de banco de dados propriamente ditos, que deverão ser anulados ou prematuramente encerrados.</i></p> <p><i>O teste das condições a seguir exige que seja atingido um estado conhecido do banco de dados. Vários campos, ponteiros e chaves de banco de dados deverão ser corrompidos manualmente e diretamente no banco de dados (através das ferramentas de banco de dados). As transações adicionais deverão ser executadas usando os Testes de Ciclos de Negócios e de Funcionamento do Aplicativo e deverão ser</i></p>
------------------------	---

<p>Estratégias:</p>	<p><i>[Descreva uma ou mais estratégias que podem ser usadas pela técnica para observar, de forma precisa, os resultados do teste. A estratégia combina elementos do método através do qual a observação pode ser feita e das características dos resultados específicos que indicam um provável êxito ou falha do teste. O ideal é que as estratégias sejam auto verificadas, permitindo que os testes automatizados façam uma avaliação inicial do êxito ou falha do teste. No entanto, tenha atenção para reduzir os riscos inerentes à determinação automática dos resultados.]</i></p>
<p>Ferramentas Necessárias:</p>	<p><i>[A técnica exige as seguintes ferramentas: restaurador e reprodutor de imagem da configuração básica ferramentas de monitoramento de instalação (registro, disco rígido, CPU, memória etc.) ferramentas de backup e de recuperação]</i></p>
<p>Critérios de Êxito:</p>	<p><i>[A técnica suporta o teste de: Um dos desastres simulados envolvendo uma ou mais combinações do aplicativo, banco de dados e do sistema. Uma ou mais recuperações simuladas envolvendo uma ou mais combinações do aplicativo, banco de dados e do sistema em um estado conhecido desejado.</i></p>

<p>Considerações Especiais:</p>	<p><i>[O teste de recuperação é altamente invasivo. Os procedimentos para desconectar cabos (simular perda de energia ou de comunicação) talvez não sejam desejáveis ou viáveis. Poderão ser necessários métodos alternativos como, por exemplo, ferramentas de software de diagnóstico.</i></p> <p><i>Serão necessários Recursos dos Sistemas (ou Operações de Computador), Bancos de Dados e Grupos de Redes.</i></p> <p><i>Esses testes deverão ser executados após o expediente de trabalho ou em uma máquina isolada.]</i></p>
--	---

7.2.11 Teste de Configuração

[O teste de configuração verifica o funcionamento do objetivo do teste em diferentes configurações de software e de hardware. Na maior parte dos ambientes de produção, as especificações de hardware específicas para as estações de trabalho cliente, as conexões de rede e os servidores de banco de dados variam. Nas estações de trabalho cliente, poderão ser carregados diferentes softwares (por exemplo, aplicativos, drivers etc) e, a qualquer momento, muitas combinações diferentes poderão ficar ativas utilizando diferentes recursos.]

<p>Objetivo Técnica:</p>	<p><i>[Experimentar o objetivo do teste nas configurações de hardware e de software necessárias, a fim de observar e registrar o comportamento-alvo em diferentes configurações e identificar mudanças no estado da configuração.]</i></p>
---------------------------------	--

<p>Técnica:</p>	<p><i>[Use Scripts de Teste de Funcionamento.</i></p> <p><i>Abra e feche vários softwares relacionados que não sejam o objetivo do teste como, por exemplo, os aplicativos Microsoft Excel e Word, como parte do teste ou antes do início do teste.</i></p> <p><i>Execute as transações selecionadas para simular atores interagindo com softwares que sejam o objetivo do teste e com os que não sejam o objetivo do teste.</i></p> <p><i>Repita o processo acima, minimizando a memória convencional disponível na estação de trabalho cliente.]</i></p>
<p>Estratégias:</p>	<p><i>[Descreva uma ou mais estratégias que podem ser usadas pela técnica para observar, de forma precisa, os resultados do teste. A estratégia combina elementos do método através do qual a observação pode ser feita e das características dos resultados específicos que indicam um provável êxito ou falha do teste. O ideal é que as estratégias sejam auto verificadas, permitindo que os testes automatizados façam uma avaliação inicial do êxito ou falha do teste. No entanto, tenha atenção para reduzir os riscos inerentes à determinação automática dos resultados.]</i></p>
<p>Ferramentas Necessárias:</p>	<p><i>[A técnica exige as seguintes ferramentas:</i></p> <p><i>restaurador e reproduzidor de imagem da configuração básica</i></p> <p><i>ferramentas de monitoramento de instalação (registro, disco rígido, CPU, memória etc)</i></p>

<p>Critérios de Êxito:</p>	<p><i>[A técnica suporta o teste de uma ou mais combinações dos itens de teste-alvo executadas em ambientes de implantação suportados esperados.]</i></p>
<p>Considerações Especiais:</p>	<p><i>[Que software que não seja o objetivo do teste é necessário, está disponível e acessível na área de trabalho?</i></p> <p><i>Que aplicativos são normalmente usados?</i></p> <p><i>Que dados estão em execução nos aplicativos; por exemplo, uma grande planilha aberta no Excel ou um documento de 100 páginas no Word?</i></p> <p><i>O NetWare, os servidores de rede, os banco de dados, entre outros, de todos os sistemas também precisam ser documentados como parte desse teste.]</i></p>

7.2.12 Teste de Instalação

[O teste de instalação tem duas finalidades. A primeira é assegurar que o software possa ser instalado em diferentes circunstâncias (como uma nova instalação, uma atualização e uma instalação completa ou personalizada) em condições normais e anormais. Entre as condições anormais estão o espaço insuficiente no disco, a falta de privilégios para criar diretórios e assim por diante. A segunda finalidade é verificar se, depois de instalado, o software funcionará corretamente. Isso geralmente implica executar uma série de testes que foram desenvolvidos como parte dos Testes de Funcionamento.]

<p>Objetivo Técnico:</p>	<p><i>[Experimentar a instalação do objetivo do teste em cada configuração de hardware exigida nas condições a seguir para observar e registrar o comportamento da instalação e as mudanças no estado da configuração:</i></p> <p><i>nova instalação: uma nova máquina, em que nunca foi instalado anteriormente o <Nome do Projeto></i></p> <p><i>atualização: uma máquina em que foi instalado anteriormente o <Nome do Projeto>, na mesma versão</i></p>
---------------------------------	---

	<p>atualização: uma máquina em que foi instalado anteriormente o <Nome do Projeto>, na versão antiga]</p>
<p>Técnica:</p>	<p><i>[Desenvolva scripts automatizados ou manuais para validar a condição da máquina-alvo.</i></p> <ul style="list-style-type: none"> ● <i>novo: nunca instalado</i> ● <i>mesma versão ou versão mais antiga instalada</i> <p><i>Inicie ou execute a instalação.</i></p> <p><i>Utilizando um subconjunto predeterminado de scripts de Teste de Funcionamento, execute as transações.]</i></p>
<p>Estratégias:</p>	<p><i>[Descreva uma ou mais estratégias que podem ser usadas pela técnica para observar, de forma precisa, os resultados do teste. A estratégia combina elementos do método através do qual a observação pode ser feita e das características dos resultados específicos que indicam um provável êxito ou falha do teste. O ideal é que as estratégias sejam auto verificadas, permitindo que os testes automatizados façam uma avaliação inicial do êxito ou falha do teste. No entanto, tenha atenção para reduzir os riscos inerentes à determinação automática dos resultados.]</i></p>

<p>Ferramentas Necessárias:</p>	<p><i>[A técnica exige as seguintes ferramentas: restaurador e reprodutor de imagem da configuração básica ferramentas de monitoramento de instalação (registro, disco rígido, CPU, memória etc.)]</i></p>
<p>CrITÉrios de Êxito:</p>	<p><i>[A técnica suporta o teste da instalação do produto desenvolvido em uma ou mais configurações de instalação.]</i></p>
<p>Considerações Especiais:</p>	<p><i>[Que transações de devem ser selecionadas para constituir um teste que comprova que o aplicativo de foi instalado com êxito e que não está faltando nenhum componente de software principal?]</i></p>

8. CritÉrios de Entrada e de Saída

8.1 Plano de Teste

8.1.1 CritÉrios de Entrada de Plano de Teste

[Especifique os critÉrios que serão usados para determinar se a execução do Plano de Teste poderá ser iniciada.]

8.1.2 CritÉrios de Saída de Plano de Teste

[Especifique os critÉrios que serão usados para determinar se a execução do Plano de Teste foi concluída ou se a continuação da execução não será vantajosa.]

8.1.3 CritÉrios de Suspensão e de Reinício

[Especifique os critÉrios que serão usados para determinar se os testes deverão ser prematuramente suspensos ou concluídos antes que o plano tenha sido totalmente executado. Especifique também segundo que critÉrios os testes poderão ser reiniciados.]

8.2 Ciclos de Teste

8.2.1 CritÉrios de Entrada de Ciclo de Teste

[Especifique os critÉrios que serão usados para determinar se o esforço do próximo Ciclo de Teste deste Plano de Teste poderá ser iniciado.]

8.2.2 Critérios de Saída de Ciclo de Teste

[Especifique os critérios que serão usados para determinar se o esforço de teste do Ciclo de Teste atual deste Plano de Teste é considerado suficiente.]

8.2.3 Término Anormal do Ciclo de Teste

[Especifique os critérios que serão usados para determinar se os testes deverão ser prematuramente suspensos ou concluídos para o ciclo de teste atual, ou se o futuro build a ser testado deverá ser alterado.]

9. Produtos Liberados

[Nesta seção, liste os vários artefatos que serão criados pelo esforço de teste e que serão produtos liberados úteis aos vários envolvidos do esforço de teste. Não liste todos os produtos do trabalho; liste apenas os que propiciam benefícios diretos tangíveis aos envolvidos e os que permitem medir o êxito do esforço de teste.]

9.1 Sumários de Avaliação de Testes

[Forneça um breve resumo da forma e do conteúdo dos sumários de avaliação de testes e indique com que frequência eles serão gerados.]

9.2 Geração de Relatórios sobre Cobertura de Teste

[Forneça um breve resumo da forma e do conteúdo dos relatórios usados para medir a extensão do teste e indique com que frequência eles serão gerados. Forneça uma indicação referente ao método e às ferramentas usados para registrar, medir e reportar a extensão do teste.]

9.3 Relatórios da Qualidade Perceptível

Forneça um breve resumo da forma e do conteúdo dos relatórios usados para medir a qualidade perceptível do produto e indique com que frequência eles serão gerados. Forneça uma indicação referente ao método e às ferramentas usados para registrar, medir e reportar a qualidade perceptível do produto. Você poderá incluir análises dos Incidentes e Solicitações de Mudança ao longo da Cobertura de Teste.]

9.4 Registros de Incidentes e Solicitações de Mudança

[Forneça um breve resumo do método e das ferramentas usados para registrar, rastrear e gerenciar incidentes dos testes, as solicitações de mudança associadas e seus status.]

9.5 Conjunto de Testes de Regressão e Scripts de Teste de Suporte

[Forneça um breve resumo dos recursos dos testes que serão liberados para permitir testes de regressão contínuos dos builds subsequentes do produto, a fim de ajudar a detectar as regressões na qualidade do produto.]

9.6 Produtos de Trabalho Adicionais

[Nesta seção, identifique os produtos de trabalho que são opcionais ou os que não deverão ser usados para medir ou avaliar a execução bem-sucedida do Plano de Teste.]

9.6.1 Resultados Detalhados dos Testes

[Trata-se de um conjunto de planilhas do Microsoft Excel relacionando os resultados determinados para cada caso de teste ou refere-se ao repositório dos registros de testes e dos resultados determinados mantidos por um produto de teste especializado.]

9.6.2 Scripts de Teste Funcionais Automatizados Adicionais

Estes scripts consistem em um conjunto dos arquivos de código-fonte dos scripts de teste automatizados ou no repositório do código-fonte e dos executáveis compilados referentes aos scripts de teste mantidos pelo produto de automação de testes.]

9.6.3 Guia de Teste

[O Guia de Teste abrange um amplo conjunto de categorias incluindo Catálogos de Idéias de Testes, Orientações de Práticas Adequadas, Padrões de Teste, Modelos de Erros e de Falhas, Padrões de Design de Automação etc.]

9.6.4 Matrizes de Rastreabilidade

[Utilizando uma ferramenta como o Rational RequisitePro ou o Microsoft Excel, forneça uma ou mais matrizes de relacionamentos de rastreabilidade entre os itens rastreados.]

10. Fluxo de Trabalho de Teste

[Forneça um resumo do fluxo de trabalho a ser seguido pela equipe de teste no desenvolvimento e na execução deste Plano de Teste.

O fluxo de trabalho de teste específico que você usará deve ser documentado separadamente no Caso de Desenvolvimento do projeto. Ele deve explicar como o projeto personalizou o fluxo de trabalho de teste básico do RUP (normalmente fase a fase). Na maior parte dos casos, é recomendável que, nesta seção do Plano de Teste, você insira uma referência à seção relevante do Caso de Desenvolvimento. Poderá ser útil e suficiente simplesmente incluir um diagrama ou uma imagem ilustrando o fluxo de trabalho de teste.

Os detalhes mais específicos das tarefas de teste individuais poderão ser definidos de várias maneiras diferentes, dependendo da cultura do projeto. Veja os exemplos a seguir:

poderão ser definidos como uma lista de tarefas nesta seção do Plano de Teste ou em um apêndice complementar

poderão ser definidos em uma programação central do projeto (frequentemente em uma ferramenta de programação como o Microsoft Project)

poderão ser documentados em listas de tarefas "dinâmicas" individuais para cada membro da equipe, que geralmente são muito detalhadas para serem inseridas no Plano de Teste

poderão ser documentados em um quadro branco localizado em um local central e atualizado dinamicamente

poderão simplesmente não serem documentados formalmente

Com base na cultura de seu projeto, você deverá listar suas tarefas de teste específicas aqui ou fornecer um texto descritivo explicando o processo utilizado por sua equipe para efetuar o planejamento detalhado de tarefas. Você também poderá fazer referência ao local em que os detalhes serão armazenados, se for adequado.

Para os Planos de Teste Mestre, é recomendável evitar o planejamento detalhado de tarefas, que freqüentemente será um esforço improdutivo se efetuado como uma atividade antecipada no início do projeto. Um Plano de Teste Mestre poderá descrever, de maneira útil, as fases e o número de iterações, e fornecer uma indicação dos tipos de teste que geralmente são planejados para cada Fase ou Iteração.

Observação: Nos casos em que as informações referentes a processos e ao planejamento detalhado forem registradas em um local central e separadamente deste Plano de Teste, você terá que gerenciar os problemas originados pelo fato de existirem cópias duplicadas das mesmas informações. Para evitar que os membros da equipe consultem informações desatualizadas, é recomendável, nesse caso, inserir o mínimo possível de informações sobre processos e planejamento no Plano de Teste para facilitar a constante manutenção das informações e, portanto, simplesmente fazer referência ao material que se encontra no "Plano Mestre".]

11. Necessidades Ambientais

[Esta seção apresenta os recursos não humanos necessários ao Plano de Teste.]

11.1 Hardware Básico do Sistema

Os conjuntos de tabelas a seguir apresentam os recursos do sistema necessários ao esforço de teste descrito neste *Plano de Teste*.

[É possível que os elementos específicos do sistema de teste não sejam totalmente compreendidos nas iterações iniciais, sendo assim, espera-se que esta seção seja preenchida ao logo do tempo. É recomendável que o sistema simule o ambiente de produção, reduzindo o acesso concorrente e o tamanho do banco de dados, se e quando for necessário.]

[Observação: Adicione ou exclua itens conforme o necessário.]

Recurso	Quantidade	Nome e Tipo
----------------	-------------------	--------------------

Servidor de Banco de Dados Rede ou Sub rede Nome do Servidor Nome do Banco de Dados		
		A ser definido
		A ser definido
		A ser definido
PCs de Teste Cliente Inclua requisitos de configuração especiais		
		A ser definido
Repositório de Teste Rede ou Sub-rede Nome do Servidor		
		A ser definido
		A ser definido
PCs de Desenvolvimento de Teste		A ser definido

11.2 Elementos de Software Básicos do Ambiente de Teste

São necessários os seguintes elementos de software básicos no ambiente de teste deste *Plano de Teste*.

[Observação: Adicione ou exclua itens conforme o necessário.]

Nome do Elemento de Software	Versão	Tipo e Outras Observações
NT Workstation		Sistema Operacional
Windows 2000		Sistema Operacional
Internet Explorer		Navegador da Internet
Netscape Navigator		Navegador da Internet
Microsoft Outlook		Software Cliente de E-Mail
Network Associates McAfee Virus Checker		Software de Detecção e Recuperação de Vírus

11.3 Ferramentas de Produtividade e de Suporte

Serão utilizadas as seguintes ferramentas para suportar o processo de teste deste *Plano de Teste*.

[Observação: Adicione ou exclua itens conforme o necessário.]

Categoria ou Tipo de Ferramenta	Nome da Marca da Ferramenta	Fornecedor ou Desenvolvida Internamente	Versão
Gerenciamento de Teste			
Controle de Defeitos			
Ferramenta ASQ para teste funcional			
Ferramenta ASQ para teste de desempenho			
Gerador de Perfil ou Monitor de Cobertura de Teste			

Gerenciamento de Projeto			
Ferramentas DBMS			

11.4 Configurações do Ambiente de Teste

Devem ser fornecidas e suportadas as seguintes Configurações de Ambiente de Teste para este projeto.

Nome da Configuração	Descrição	Implementada na Configuração Física
Configuração do usuário comum		
Mínima configuração suportada		
Motivada por funções visuais e motoras		
Sistema Operacional Internacional de Dois Bytes		
Instalação de Rede (não cliente)		

12. Responsabilidades, Perfil da Equipe e Necessidades de Treinamento

[Esta seção apresenta os recursos necessários para abordar o esforço de teste no Plano de Teste; as responsabilidades principais e os conjuntos de conhecimentos ou de habilidades exigidos desses recursos.]

12.1 Pessoas e Papéis

Esta tabela mostra as suposições referentes ao perfil da equipe do esforço de teste.

[Observação: Adicione ou exclua itens conforme o necessário.]

Recursos Humanos		
Papel	Recursos Mínimos Recomendáveis (número de papéis alocados em tempo integral)	Responsabilidades ou Comentários Específicos

<p>Gerente de Testes</p>		<p>Supervisiona o gerenciamento. Entre as responsabilidades estão incluídas:</p> <ul style="list-style-type: none"> ● planejamento e logística ● combinar missão ● identificar motivadores ● adquirir recursos apropriados ● apresentar relatórios de gerenciamento ● defender os interesses do teste ● avaliar a eficiência do esforço de teste
<p>Analista de Teste</p>		<p>Identifica e define os teste específicos a serem conduzidos. Entre as responsabilidades estão incluídas:</p> <ul style="list-style-type: none"> ● identificar ideias de teste ● definir detalhes dos testes ● determinar os resultados dos testes ● documentar solicitações de mudança ● avaliar a qualidade do

		produto
Designer de Teste		<p>Define a abordagem técnica referente à implementação do esforço de teste.</p> <p>Entre as responsabilidades estão incluídas:</p> <ul style="list-style-type: none">● definir a abordagem dos testes● definir a arquitetura de automação de teste● verificar as técnicas de teste● definir os elementos de testabilidade● estruturar a implementação dos testes

<p>Testador</p>		<p>Implementa e executa os testes.</p> <p>Entre as responsabilidades estão incluídas:</p> <ul style="list-style-type: none"> ● implementar os testes e os conjuntos de testes ● executar os conjuntos de testes ● registrar os resultados ● analisar as falhas dos testes e possibilitar a recuperação posterior ● documentar incidentes
<p>Administrador do Sistema de Teste</p>		<p>Assegura a manutenção e o gerenciamento dos recursos e do ambiente do teste.</p> <p>Entre as responsabilidades estão incluídas:</p> <ul style="list-style-type: none"> ● administrar o sistema de gerenciamento de teste ● instalar e suportar o acesso às configurações do ambiente de teste e aos laboratórios de teste, bem

		<p>como a recuperação deles</p>
<p>Administrador do Banco de Dados, Gerente do Banco de Dados</p>		<p>Assegurar o gerenciamento e a manutenção dos recursos e do ambiente dos dados de teste (banco de dados).</p> <p>Entre as responsabilidades estão incluídas:</p> <ul style="list-style-type: none"> • suportar a administração dos dados de teste e das plataformas de teste (banco de dados)

<p>Designer</p>		<p>Identifica e define as operações, os atributos e as associações das classes de teste.</p> <p>Entre as responsabilidades estão incluídas:</p> <ul style="list-style-type: none"> • define as classes de teste necessárias para suportar os requisitos de estabilidade conforme definido pela equipe de teste
<p>Implementador</p>		<p>Implementa as classes de teste e os pacotes de teste e efetua testes unitários nos mesmos.</p> <p>Entre as responsabilidades estão incluídas:</p> <ul style="list-style-type: none"> • cria os componentes de teste necessários para suportar os requisitos de estabilidade conforme definido pelo designer

12.2 Perfil da Equipe e Necessidades de Treinamento

Esta seção resume como abordar o perfil da equipe e o treinamento dos profissionais que ocuparão os papéis de teste no projeto.

[O modo como abordar o perfil da equipe e o treinamento dos profissionais varia de projeto para projeto. Se esta seção integrar um Plano de Teste Mestre, indique em que pontos do ciclo de vida do projeto serão necessárias diferentes habilidades e um número diferente de integrantes da equipe. Se for um Plano de Teste de Iteração, você deverá concentrar-se principalmente em que momento, durante a Iteração, poderá ocorrer um treinamento e de que tipo ele será.

Refleta sobre suas necessidades de treinamento e planeje uma programação de treinamento com base em uma abordagem que sustente que o treinamento só deverá ser realizado no momento certo. Há sempre a tentação de realizar o treinamento muito antes de quando ele será realmente necessário, em um período em que a equipe de teste esteja aparentemente ociosa. Quando isso é feito, corre-se o risco de os ensinamentos do treinamento já terem sido esquecidos justamente no momento em que forem necessários.

Procure por oportunidades de combinar a compra de ferramentas de produtividade com o treinamento dessas ferramentas e retarde o treinamento, de comum acordo com o fornecedor, apenas para o momento em que for realmente necessário. Se tiver um número de pessoas suficiente, é recomendável realizar um treinamento personalizado, possivelmente no próprio local de sua organização.

Frequentemente, a equipe de teste necessita do suporte e das habilidades dos membros de outras equipes, que não a integram de forma direta. Certifique-se de programar, no seu plano, a participação adequada de Administradores de Sistema, Administradores de Banco de Dados e Desenvolvedores, que são profissionais necessários para viabilizar o esforço de teste.]

Observações:

O plano de teste é um dos documentos produzidos na condução de um projeto. Ele funciona como:

- Um “integrador” entre diversas atividades de testes no projeto;
- Mecanismo de comunicação para os *stakeholders* (isto é a equipe de testes e outros interessados);
- Guia para execução e controle das atividades de testes.

O plano de teste, que pode ser elaborado pelo gerente de projeto ou gerente de testes, visa planejar as atividades a serem realizadas, definir os métodos a serem empregados, planejar a capacidade necessária, estabelecer métricas e formas de acompanhamento do processo.

Estas são as tarefas relacionadas a teste:

1.Planejar Teste
2.Identificar os Requisitos de Teste
3.Avaliar o Risco
4.Desenvolver a Estratégia de Teste
5.Identificar os Recursos de Teste
6.Criar Programação
7.Gerar Plano de Teste
8.Projetar Teste
9.Análise da Carga de Trabalho
10.Identificar e Descrever Casos de Teste
11.Identificar e Estruturar Procedimentos de Teste
12.Revisar e Acessar a Cobertura de Teste
13.Implementar Teste
14.Registrar ou Programar Scripts de Teste

15. Identificar a funcionalidade específica de Teste no modelo de design e de implementação
16. Estabelecer Conjuntos de Dados Externos
17. Executar Teste
18. Executar os Procedimentos de Teste
19. Avaliar a Execução do Teste
20. Recuperar-se de uma Interrupção de Teste
21. Verificar os Resultados
22. Investigar os Resultados Inesperados
23. Registrar Defeitos
24. Avaliar Teste
25. Avaliar Cobertura de Caso de Teste
26. Avaliar Cobertura de Código
27. Analisar os Defeitos
28. Determinar se os Critérios de Conclusão e os Critérios de Sucesso do Teste foram obedecidos

4.1.4 Rastreabilidade de Requisitos

[TEMPLATE DE RASTRABILIDADE DE REQUISITOS]

<Sigla da Estatal>

<Nome do Projeto>

histórico da Revisão

Data	Versão	Descrição	Autor	Área/Setor
<dd/mm/aaaa>	<x.x>	<detalhes>	<nome>	

LISTA DE REQUISITOS

1. Requisitos

1.1 Nome

<Forneça o nome utilizado para identificar o requisito.>

1.2 Descrição

<Forneça uma descrição do requisito.>

1.3 Tipo

<Forneça o tipo de requisito.>

1.4 Prioridade

<Forneça a prioridade prévia identificada.>

1.5 Observação

<Descreva observações adicionais..>

Observações:

Tipos de Requisitos:

Requisitos Funcionais. São declarações de funções que o sistema deve fornecer, como o sistema deve reagir a entradas específicas e como deve se comportar em determinadas situações. Em alguns casos, os requisitos funcionais podem também explicitamente declarar o que o sistema não deve fazer.

Requisitos Não-funcionais. São restrições sobre os serviços ou as funções oferecidas pelo sistema. Entre eles destacam-se restrições de tempo, restrições sobre o processo de desenvolvimento, padrões, entre outros.

Requisitos de Domínio. São requisitos que se originam do domínio de aplicação do sistema e que refletem características desse domínio. Podem ser requisitos funcionais ou não funcionais.

Na realidade, a distinção entre esses diferentes tipos de requisitos não é tão clara como sugerem definições simples. Um requisito de usuário relacionado à proteção, digamos, parece ser um requisito não funcional. Contudo, quando desenvolvido com mais detalhes, pode levar aos outros requisitos que são claramente funcionais, como a necessidade de incluir recursos de autorização de usuários no sistema. Portanto, embora seja útil classificar os requisitos dessa maneira quando os discutimos, devemos lembrar que essa é, na verdade, uma distinção artificial.

Os requisitos funcionais para um sistema descrevem a funcionalidade ou os serviços que se espera que o sistema forneça. Eles dependem do tipo de software que está sendo desenvolvido, dos usuários de software que se espera verificar e do tipo de sistema que está sendo desenvolvido. Quando expressos como requisitos de usuário, eles são normalmente descritos de um modo bastante geral, mas os requisitos funcionais de sistema descrevem a função de sistema detalhadamente, suas entradas e saídas, exceções, etc.

Esses requisitos funcionais de usuário definem recursos específicos que devem ser fornecidos pelo sistema. Eles foram observados no documento de requisitos de usuário do sistema e ilustram que os requisitos funcionais podem ser escritos em diferentes níveis de detalhes

Requisitos Não Funcionais

Os requisitos não funcionais, como o nome sugere, são aqueles que não dizem respeito diretamente às funções específicas fornecidas pelo sistema. Eles podem estar relacionados a propriedades de sistema emergentes, como confiabilidade, tempo de resposta e espaço em disco.

Como alternativa, eles podem definir restrições para o sistema, como a capacidade dos dispositivos de E/S (entrada/saída) e as representações de dados utilizadas nas interfaces de sistema.

Muitos requisitos não funcionais dizem respeito ao sistema como um todo, e não a características individuais do sistema. Isso significa que eles são, frequentemente, mais importantes do que os requisitos funcionais individuais. Enquanto as falhas em cumprir com um requisito funcional individual podem degradar o sistema, a falha em cumprir um requisito não funcional de sistema pode tornar todo o sistema inútil. Por exemplo, se um sistema de aviação não atender a seus requisitos de confiabilidade, ele não será atestado como seguro para operação; se um sistema de controle em tempo real falhar em cumprir com seus requisitos de desempenho, as funções de controle não operarão corretamente.

Contudo, os requisitos não funcionais nem sempre dizem respeito ao sistema de software a ser desenvolvido. Alguns requisitos não funcionais podem restringir o processo que pode ser utilizado para desenvolver o sistema. São exemplos de requisitos de processo uma especificação dos padrões de qualidade, que deve ser utilizada no processo, uma especificação de que o projeto deve ser produzido com um conjunto especificado de ferramentas CASE e uma descrição de processo a ser seguido.

Classificação dos diferentes tipos de requisitos não funcionais:

1. **Requisitos de produtos** São os requisitos que especificam o comportamento do produto. Entre os exemplos estão os requisitos de desempenho sobre com que rapidez o sistema deve operar e quanta memória ele requer, os requisitos de confiabilidade, que estabelecem a taxa aceitável de falhas, os requisitos de portabilidade e os requisitos de facilidade de uso.

2. **Requisitos organizacionais** São procedentes de políticas e procedimentos nas organizações do cliente e do desenvolvedor. Entre os exemplos estão os padrões de processo que devem ser utilizados, os requisitos de implementação, como a linguagem de programação ou o método de projeto utilizado, e os requisitos de fornecimento, que especificam quando o produto e seus documentos devem ser entregues.

3. **Requisitos externos.** Esse amplo tópico abrange todos os requisitos procedentes de fatores externos ao sistema e a seu processo de desenvolvimento. Dentre eles destacam-se os requisitos de interoperabilidade, que definem como o sistema interage com sistemas em outras organizações, os requisitos legais, que devem ser seguidos para assegurar que o sistema opera de acordo com a lei, e os requisitos éticos. Os requisitos éticos são definidos em um sistema para garantir que este será aceitável para seus usuários e o público em geral.

Prioridade de Requisitos:

1. **Risco alto e prioridade alta:** requisitos nesta categoria são os mais críticos. Devem ser considerados o quanto antes.
2. **Risco alto e prioridade baixa:** embora os requisitos nesta categoria tenham risco alto, é necessário, antes de começar a considerá-los, negociar com o cliente em relação a sua verdadeira necessidade.
3. **Risco baixo e prioridade alta:** embora os requisitos tenham prioridade alta, é necessário ter em mente que os casos de uso de mais alto risco devem ser considerados primeiro.
4. **Risco baixo e prioridade baixa:** em situações em que o desenvolvimento do sistema está atrasado, estes requisitos são os primeiros a serem "cortados".

Para realizar a tarefa de elaboração de uma lista de requisitos, deve-se:

1. Realizar entrevista com o requisitante para identificar os requisitos funcionais e não funcionais ou criar fóruns específicos para a elicitación dos requisitos, no caso de desenvolvimento colaborativo;
2. Elaborar Lista de Requisitos;
3. Classificar e priorizar requisitos, detectar e resolver conflitos entre requisitos; confirmar e validar os requisitos com o cliente ou com as demais organizações envolvidas em caso de desenvolvimento colaborativo.
4. Enviar especificações para a Área Requisitante e, se for o caso, para outros envolvidos no processo;

5. Atualizar especificações de acordo com as observações do requisitante ou de acordo com as demais estatais e ou departamentos envolvidos;
6. Manter o acompanhamento de requisitos individuais e suas ligações entre os requisitos dependentes.

Técnicas de Levantamento de Requisitos

Entrevistas

A entrevista é um instrumento fundamental que pode ter uma grande variedade de objetivos, como no caso do jornalista, chefe de empresa, diretor de escola, professor, juiz, etc. Aqui nos interessa a entrevista, entendida como aquela na qual se busca a investigação dos requisitos.

A entrevista se baseia em uma técnica com seus procedimentos ou regras empíricas com os quais não só se amplia e se verifica como também, ao mesmo tempo, se aplica o conhecimento científico. A técnica é o ponto de interação entre a ciência e as necessidades práticas; é assim que a entrevista alcança a aplicação de conhecimentos científicos e, ao mesmo tempo, obtém ou possibilita levar a vida diária do ser humano ao nível do conhecimento e da elaboração científica. E tudo isso em um processo ininterrupto de interação.

A entrevista é um instrumento muito difundido e devemos delimitar o seu alcance em função de suas regras ou indicações práticas de sua execução.

A entrevista pode ser de dois tipos fundamentais: **aberta e fechada**.

- Na segunda as perguntas já estão previstas, assim como a ordem e a maneira de formulá-las, e o entrevistador não pode alterar nenhuma destas disposições.
- Na entrevista aberta, pelo contrário, o entrevistador tem uma liberdade para as perguntas ou para suas intervenções, permitindo-se toda a flexibilidade necessária em cada caso particular.

A entrevista fechada é, na realidade, um questionário que passa a ter uma relação estreita com a entrevista, na medida em que uma manipulação de certos princípios e regras facilita e possibilita a aplicação do questionário.

Contudo, a entrevista aberta não se caracteriza essencialmente pela liberdade de colocar perguntas, pois há uma flexibilidade suficiente para permitir, na medida do possível, que o entrevistado configure o campo da entrevista segundo sua estrutura psicológica particular, ou - dito de outra maneira – que o campo da entrevista se configure, o máximo possível, pelas variáveis que dependem da personalidade do entrevistado.

Podem-se diferenciar também as entrevistas segundo o beneficiário do resultado: assim, podemos distinguir:

- a. Entrevista que se realiza em benefício do entrevistado - que é o caso do levantamento de requisitos;
- b. Entrevista cujo objetivo é a pesquisa, na qual importam os resultados científicos;
- c. A entrevista que se realiza para um terceiro (uma instituição).

Workshops

Conduzir um workshop de requisitos implica reunir todos os envolvidos durante um período intensivo, concentrado. Um Analista de Sistemas atua como um facilitador da reunião. Todos os participantes deverão contribuir ativamente e os resultados da sessão deverão ser disponibilizados imediatamente para eles.

O workshop de requisitos fornece um framework para aplicar as outras técnicas de identificação como, por exemplo, brainstorming, encenação, interpretação de papéis e revisão dos requisitos existentes. Essas técnicas poderão ser usadas isoladamente ou combinadas. Todas poderão ser combinadas ao método de caso de uso. Por exemplo, você poderá criar uma ou algumas encenações para cada caso de uso previsto para o sistema. Você poderá usar a interpretação de papéis como uma maneira de compreender como os atores usarão o sistema e para ajudá-lo a definir os casos de uso.

Os resultados dos workshops de requisitos serão documentados em um ou mais artefatos de solicitações dos principais envolvidos. Desde que você tenha boas ferramentas de suporte, geralmente é aconselhável permitir que os envolvidos relatem informações desse tipo.

Brainstorm ou Tempestade Cerebral

Frequentemente a tempestade cerebral permite um desbloqueio, um aquecimento para a conversação entre o analista e o cliente. Tem como propostas o desenvolvimento da originalidade e da desinibição, bem como a produção de um grande número de ideias em prazo curto; numa palavra, à criatividade.

Seu funcionamento geral é o seguinte: dado um tema, o cliente (s) expressa oralmente, em uma palavra ou em frases bem curtas, tudo o que lhe vem à cabeça, sugerido por aquele tema, sem se preocupar em censurar essas ideias. O analista vai anotando tudo e a seguir faz a seleção das ideias, segundo algum critério prévio, seja agrupando-as por alguma semelhança, seja eliminando as que não podem ser postas em prática, etc.

4.1.5 Plano de Manutenção de Soluções

[TEMPLATE DE PLANO DE MANUTENÇÃO DAS SOLUÇÕES]

Sigla da Estatal>

<Nome do Projeto>

Histórico da Revisão

Data	Versão	Descrição	Autor	Área/setor
<dd/mm/aaaa>	<x.x>	<detalhes>	<nome>	

1. Atualização do Software

<esta seção deve ser revista periodicamente, em casos de implementação de funcionalidade nova, atualizações menores de performance, grandes atualizações que mudam versão ou até mesmo o produto, ou para corrigir uma falha de sistema ou de segurança>

2. Tratamento de Incidentes

<Insira uma lista ou fluxograma de como os incidentes serão tratados>

3. Requisitos em Manutenção

Nesta seção, deve-se preencher o formulário com os requisitos que irão ser mantidos.

IDENTIFICAÇÃO
NOME DO REQUISITO:
DESCRIÇÃO DO REQUISITO:
ARQUITETURA DO SISTEMA:
EQUIPE RESPONSÁVEL (informe telefone e e-mail):

ESCOPO DE PUBLICAÇÃO
ENDEREÇO WWW (URL) :
O SERVIÇO DE PUBLICAÇÃO SERÁ:
[<input type="checkbox"/>] INTERNET [<input type="checkbox"/>] INTRANET
TECNOLOGIAS ENVOLVIDAS
SISTEMA OPERACIONAL:
LINGUAGENS DE PROGRAMAÇÃO:
BANCO DE DADOS (informe o SGBD):
COMPONENTES DE TERCEIROS:
CONFIGURAÇÃO E SERVIÇOS AUTOMATIZADOS
PLANO DE BACKUP ESPECÍFICO:
TAREFAS AGENDADAS:
CONFIGURAÇÕES REGIONAIS:

<Nesta seção, devem ser informados os requisitos mínimos de segurança, que podem ser encontrados no documento “Lista de Requisitos” do eixo de Engenharia de Software>

6. Requisitos Mínimos de Arquitetura

<Nesta seção, devem ser informados os requisitos mínimos de Arquitetura, que podem ser encontrados no documento “Lista de Requisitos” do eixo de Engenharia de Software>

Observações:

É um documento de referência nas atividades de manutenção do sistema. Inclui, principalmente, diretrizes sobre os processos de backup de dados, de suporte continuado aos usuários e de atendimento de novos requisitos que surgem do próprio uso e mudanças de processos no ambiente de trabalho.

Descrição da Atividade:

- mapear tratamento de incidentes;
- identificar requisitos de Manutenibilidade;
- identificar requisitos de Atualização de Software;
- definir modalidade de contratação da Hospedagem, Suporte e Evolução do Software. Por exemplo, O software vai ser sustentado Internamente, pelo SERPRO ou será contratada uma empresa terceira;
- referenciar os requisitos mínimos de Segurança e Arquitetura;
- Alinhar o Plano de Sustentação e Suporte com o plano de implantação;
- Verificar Qualidade da Solução.

REFERÊNCIAS BIBLIOGRÁFICAS

5.1. Documentos

- Planejamento Estratégico da Secretaria 2015-2018.
- Guia de Comitê de TI do SISP (versão 2.0 – 2016).
- Guia de Gerenciamento de Projetos do SISP do SISP (Versão 1.0 MGP-SISP – 2011).
- Guia de Metodologia de Gerenciamento de Portfólio de Projetos do SISP (Versão 1.0 MGPP-SISP – 2013).
- Guia de Processo de Software para o SISP do SISP (Versão 1.0 PSW-SISP 2012).