

ANEXO T DO TERMO DE REFERÊNCIA

**PROCESSO DE DESENVOLVIMENTO DE
SOFTWARE DO IBAMA**

PDS-IBAMA

Versão 2.0



**Ministério do Meio Ambiente
Instituto Brasileiro do Meio Ambiente e dos Recursos Naturais Renováveis
Diretoria de Planejamento, Administração e Logística
Centro Nacional de Telemática**

PROCESSO DE DESENVOLVIMENTO DE SOFTWARE DO IBAMA

Versão 2.0

- 2016 -



Ministério do Meio Ambiente
Instituto Brasileiro do Meio Ambiente e dos Recursos Naturais Renováveis
Diretoria de Planejamento, Administração e Logística
Centro Nacional de Telemática

Presidenta da República

Dilma Viana Rousseff

Ministra do Meio Ambiente

Izabella Mônica Vieira Teixeira

Presidenta do Instituto Brasileiro do Meio Ambiente e dos Recursos Naturais Renováveis

Marilene de Oliveira Ramos Murias dos Santos

Diretor Substituto de Planejamento, Administração e Logística

Anna Flávia de Senna Franco

Chefe do Centro Nacional de Telemática

Rosana de Souza Ribeiro Freitas

Grupo de Trabalho - Ordem de Serviço/DIPLAN nº 10 de 17 de novembro de 2014

Alcione Jandir Candéas Ramos

Felipe Ventorim Rodrigues Paulin

Marcelo Lima Bittencourt

Márcio Pereira Lima

Maurício Pires Martins

Rosana de Souza Ribeiro Freitas



SUMÁRIO

1. Objetivos	05
2. Aplicação	05
3. Organização do PDS-Ibama	06
3.1. Processo de Desenvolvimento de Software do Ibama	06
3.2. Subprocessos do PDS-Ibama	06
3.3. Fases do PDS-Ibama	07
3.4. Atividades do PDS-Ibama	08
4. Organização por temas	08
5. Papéis e Atribuições	10
6. Subprocessos do PDS-Ibama	13
6.1. Concepção	13
6.2. Desenvolvimento	19
6.2.1. Requisitos	21
6.2.2. Designer e Arquitetura	28
6.2.3. Implementação	35
6.2.4. Testes	38
6.2.5. Homologação	43
6.3. Transição	52
7. Monitoramento com indicadores	59
8. Artefato do PDS-Ibama	61
9. Glossário	63
9.1. Glossário de artefatos	63
9.2. Glossário geral	67
10. Guias técnicos	71
11. Referências bibliográficas	72



INTRODUÇÃO

A norma de Processo de Desenvolvimento de Software do Ibama (PDS-Ibama) é resultado da experiência adquirida pelo órgão durante a vigência da norma de Metodologia de Desenvolvimento de Sistemas (MDS) publicada em maio de 2011. O Processo de Desenvolvimento de Software abrange a Metodologia de Desenvolvimento de Sistemas e, ainda, amplia seu conteúdo, marcando assim mais um avanço na maturidade em governança de tecnologia de informação do Ibama.

O PDS-Ibama foi escrito visando a utilização de metodologia e a padronização para desenvolvimento e manutenção de software, visando garantir a escalabilidade, portabilidade, usabilidade, manutenibilidade e acessibilidade dos sistemas. A obediência às normas e regulamentos é uma necessidade intrínseca à Administração Pública e, como não poderia deixar de ser, os sistemas corporativos do Ibama devem garantir que todas as ações estejam de acordo com essas normas (incluindo aquelas citadas na Seção de Guias Técnicos).

Nesse sentido, o PDS-Ibama constitui-se em uma norma de utilização obrigatória para o desenvolvimento e a manutenção de projetos de sistemas no âmbito do Ibama, descrevendo um conjunto de regras, padrões e tarefas imprescindíveis para a execução de projetos com qualidade, produtividade e segurança no âmbito do Ibama.



1. OBJETIVO

O objetivo desta norma é prover uma sistemática simples e eficiente para o desenvolvimento e manutenção de software no Ibama. Ao longo de sua utilização poderá sofrer adequações de acordo com a maturidade do processo e das necessidades vindouras.

A aplicação deste Processo de Desenvolvimento de Sistemas (PDS) tem como principais objetivos padronizar o processo de desenvolvimento de software, primar pelo atendimento dos requisitos de negócio estabelecidos para um projeto com eficácia e qualidade, possibilitar um maior controle quanto à entrega do sistema no prazo definido, agregar qualidade ao produto, com flexibilidade para receber evoluções exigidas pelo dinamismo dos requisitos de negócio e possibilitar o acompanhamento e o gerenciamento dos processos.

2. APLICAÇÃO

Esta norma aplica-se a todos os funcionários, estagiários, consultores, prestadores de serviços e demais colaboradores do CNT que necessitem solicitar, especificar e desenvolver sistemas ao Ibama.



3. ORGANIZAÇÃO DO PDS-IBAMA

3.1 PROCESSO DE DESENVOLVIMENTO DE SOFTWARE DO IBAMA

Conforme orienta a MMP-Ibama, um processo é “um conjunto de atividades que toma uma entrada, adiciona valor e fornece uma saída, gerando um produto ou serviço valorado”¹, e é representado de forma gráfica através de uma *pool*².

O PDS-Ibama é um processo que é iniciado com a formalização de uma demanda e finalizado com a entrega de um desenvolvimento ou manutenção de um software.

3.2 SUBPROCESSOS DO PDS-IBAMA

A MMP-Ibama define que um subprocesso é um “conjunto de atividades”³ que permite apresentar este conjunto de modo contraído.

O PDS-Ibama é um processo formado por três subprocessos: Concepção, Desenvolvimento e Transição do software.

A Figura 1: “Relação do PDS-Ibama com outras normas” apresenta os subprocessos do PDS-Ibama e, também, os relaciona com os subprocessos de outras metodologias: Gestão de Demandas de TI; Sustentação de Software e MGP-Ibama (subprocessos: Planejamento, Monitoramento e Controle, Execução e Encerramento), de forma a oferecer uma visão ampla do desenvolvimento de software no Ibama. O PDS-Ibama, faz, em seu texto, algumas referências a estes outros subprocessos, atividades e artefatos destas outras normas.

¹ MMP-Ibama, versão 1.0, 2014, página 19.

² MMP-Ibama, versão 1.0, 2014, página 20.

³ MMP-Ibama, versão 1.0, 2014, página 19.

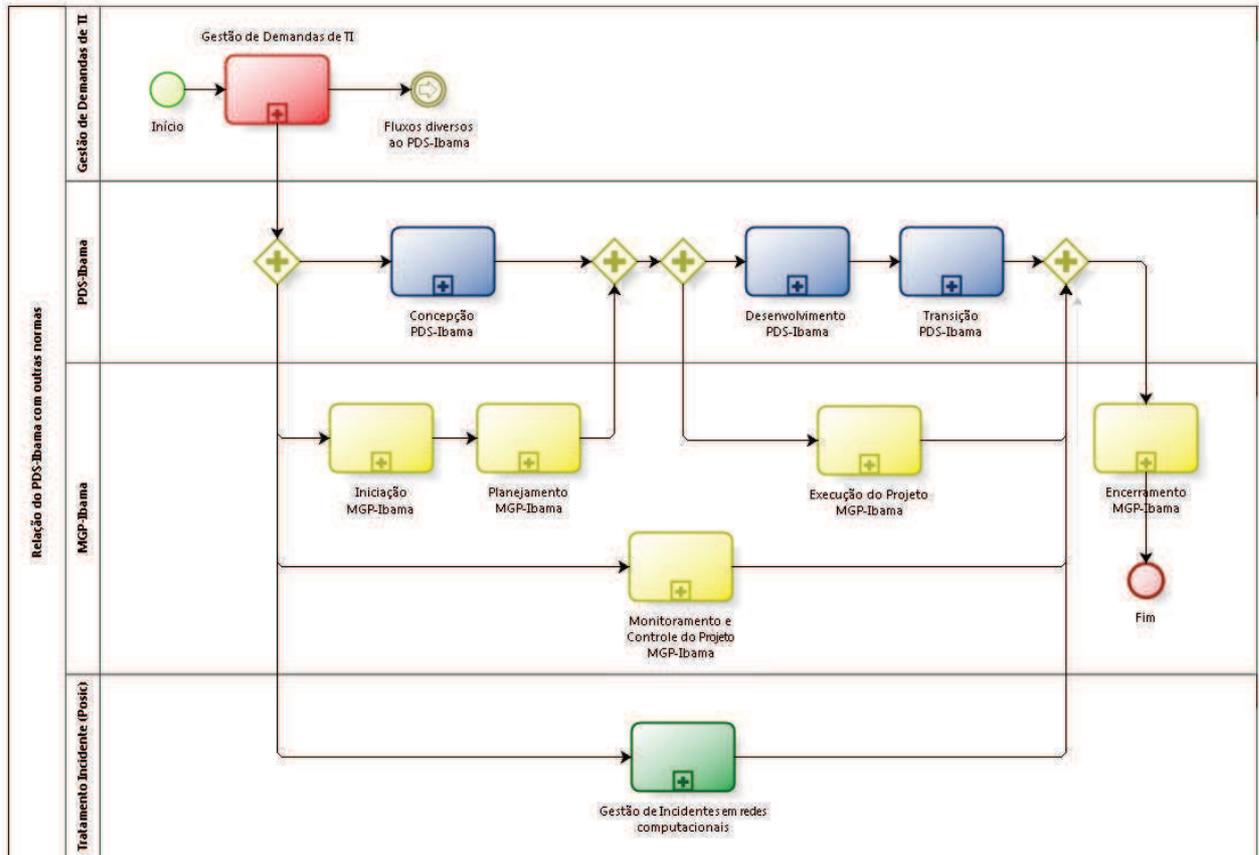


Figura 1: Relação do PDS-Ibama com outras normas

3.3 FASES DO PDS-IBAMA

As atividades de alguns processos podem ser particionadas de forma a organizar e categorizar um processo. Nas representações gráficas dos processos, estas partições são representadas através de raias (ou *lanes*)⁴ que dividem as atividades de um processo.

De forma a facilitar o acompanhamento dos projetos e a fiscalização de contratos de desenvolvimento de software, o subprocesso de Desenvolvimento foi dividido em cinco fases: Requisitos, Design e Arquitetura, Implementação, Testes e Homologação.

O subprocesso de Transição possui apenas uma fase, denominada de Implantação em Produção, definida desta forma com o intuito de utilizar esta nomenclatura que já faz parte da cultura do

⁴ MMP-Ibama, versão 1.0, 2014, página 21.



Ibama no acompanhamento dos projetos.

As fases do PDS-Ibama atuam como marcos da execução do processo, facilitando o acompanhamento do projeto pelos diversos atores e a mensuração de esforço depreendido no projeto.

3.4 ATIVIDADES DO PDS-IBAMA

O PDS-Ibama é composto por atividades, que são apresentadas de forma gráfica no formato de caixas com a descrição de seus objetivos, suas entradas, suas tarefas, suas saídas, seus responsáveis e participantes.

4. ORGANIZAÇÃO POR TEMAS

O PDS-Ibama aborda diversos temas, que são campos de conhecimento essenciais ao pleno desenvolvimento, manutenção e uso do software. Para diferenciar os temas, foram utilizadas as cores:

- Vermelho: Gestão da Demanda;
- Amarelo: Gestão de Projetos;
- Azul: Desenvolvimento de Software;
- Roxo: Gestão de Infraestrutura;
- Verde: Gestão de Tratamento de Incidentes em redes computacionais.

Temas	Objetivos
 Gestão de Demandas de TI	<p>Apresentar o fluxo de trabalho que antecede o processo de desenvolvimento de software em si, encaminhando os diversos tipos de demandas de software para seus destinos apropriados e liberando apenas as demandas de desenvolvimento e manutenção de sistemas para o processo de desenvolvimento de software.</p> <p>Verificar o alinhamento da demanda de software com as necessidades de negócio do órgão descrita no Plano Diretor de Tecnologia da Informação (PDTI).</p>
 Gestão de Projetos	<p>Visa promover uma adequada gestão dos projetos. Os processos de gestão de projetos serão mapeados tendo como referência a Metodologia de Gerenciamento de Projetos do Ibama (MGP-Ibama).</p>
 Desenvolvimento de Software	<p>Desenvolvimento e manutenção de sistemas baseado nas melhores práticas difundidas no mercado e na literatura, e em metodologias utilizadas por órgãos e entidades da Administração Pública Federal, como, por exemplo, o processo unificado, normas NBR ISO/IEC 12207, NBR ISO/IEC 15504 e outros.</p>
 Gestão de Infraestrutura	<p>Construir um ambiente que tenha a capacidade necessária para prover serviços e uma estrutura adequada ao desenvolvimento de software.</p>
 Gestão de Tratamento de Incidentes	<p>Visa planejar como os incidentes serão tratados, indicando qual ação será tomada e quem será responsável por tratar o incidente. Norma ISO/IEC18044:2004 e Norma Complementar nº 10 da POSIC/IBAMA.</p>

Tabela 1 - Descrição das Áreas Temáticas Integradas do PDS-Ibama



5. PAPÉIS E ATRIBUIÇÕES

Papéis	Atribuições
Administrador de Banco de Dados	Responsável pela modelagem e implementação física do banco de dados, gerando todas as estruturas e procedimentos necessários ao armazenamento, recuperação e exclusão de dados. Define e realiza atividades de operação, segurança, backup e recuperação de dados dos servidores de banco de dados, garantido eficiência na utilização das bases de dados. Também realiza tratamento do Custo de <i>Queries</i> e propõe soluções de melhoria de performance em Banco de Dados.
Administrador de Dados	Responsável pelo gerenciamento dos dados e informações a partir da perspectiva de seus significados e valores para a organização como um todo, alinhado aos processos de negócio. Desenvolve e administra as estratégias, procedimentos e práticas para o processo de gerência dos recursos de dados e aplicativos, promovendo-lhes os valores de autenticidade, autoridade, precisão, acessibilidade, segurança e inteligibilidade.
Arquiteto de Banco de Dados	Profissional responsável em validar o modelo de dados dos sistemas em observância aos padrões estabelecidos e visão corporativa dos sistemas.
Analista de Infraestrutura Tecnológica	Profissional responsável em definir e garantir o atendimento das necessidades de infraestrutura de software/hardware para execução do software sendo desenvolvido ou mantido. É responsável pela preparação, verificação e configuração do ambiente onde o software será executado.
Analista de Métricas	Responsável por realizar estimativas, medições ou validações de tamanho funcional e não funcional de um projeto (desenvolvimento ou manutenção) ou de um software instalado, seguindo os padrões estabelecidos pelos organismos internacionais, guias e diretrizes contratuais da organização, auxiliando na geração dos indicadores e respectivas análises e sugestões de ações. Deve apoiar análises de viabilidade e impacto, gestão de requisitos e controle de qualidade de artefatos de acordo com as melhores práticas. Apoiar o gerente na derivação do tamanho do projeto em esforço, prazo e custo, ponderação das iterações (ou ordens de serviço) e definição da melhor estratégia para o desenvolvimento ou manutenção de um software, conforme as regras estabelecidas na empresa. É responsável, ainda, por manter a base histórica de medições dos projetos e sistemas, realizar análises estatísticas, apoiar a calibração contínua dos modelos de estimativas e análises de produtividade, tendências e desvios e evoluir os modelos, normas, padrões e procedimentos de medição e análise na empresa, além de apoiar a melhoria dos demais processos (ex.: MGP-Ibama, gestão de demandas).
Analista de Requisitos	Responsável pela elicitação, análise e especificação de requisitos. Identifica as necessidades do usuário e as formaliza em documentos técnicos que nortearão o desenvolvimento ou manutenção de um software.
Analista de Segurança da Informação	Apoia na definição das regras de segurança, no planejamento e verificação dos riscos envolvidos no desenvolvimento e implantação do software.
Analista de Teste	Responsável pela iniciação, planejamento, execução e finalização do processo de teste. Cria os artefatos de testes, utilizando técnicas, critérios e tipos de teste adequados ao projeto de software a ser testado. Também elabora os procedimentos para a execução dos testes.



Ministério do Meio Ambiente
Instituto Brasileiro do Meio Ambiente e dos Recursos Naturais Renováveis
Diretoria de Planejamento, Administração e Logística
Centro Nacional de Telemática

	Faz a intermediação entre as atividades de teste e as demais atividades do projeto.
Arquiteto de Software	Define plataforma, componentes, frameworks e demais informações necessárias à especificação da arquitetura de um software. Em se tratando de contratação de fábrica de software, a empresa deverá ter ao menos um arquiteto de software, que terá seus produtos avaliados e aprovados pelo arquiteto de software do Ibama.
Centro Nacional de Telemática - CNT	É a unidade de TI do Ibama, vinculada, tecnicamente, à Diretoria de Planejamento, Administração e Logística – DIPLAN. Compete ao CNT, entre outras atribuições: - planejar, desenvolver, implantar, atualizar e gerenciar a infraestrutura de informática, comunicação e transmissão de dados do IBAMA; - propor normas e padronizar a especificações relacionadas à TI; - normatizar, planejar, coordenar, orientar, executar e avaliar o desenvolvimento de planos e projetos referentes às ações de TI; - orientar a alocação de recursos e gerenciar a aquisição de hardware, software e serviços de TI; - coordenar os serviços de atendimento a usuários e de suporte às redes de comunicação de dados e bancos de dados; - identificar demandas e necessidades de inovações tecnológicas, e propor soluções sistematizadas com base no uso de modernos recursos metodológicos e tecnológicos.
Comitê de Gestão de Mudanças	Grupo composto por representantes das áreas requisitantes e de TI, que avaliarão e apoiarão nas análises de impacto do estudo de viabilidade e na implantação das demandas de software.
Comitê de Tecnologia da Informação - CTI	Grupo formado por titulares das áreas finalísticas e da área de tecnologia da informação para assegurar que seus membros estejam envolvidos nas questões e decisões relevantes de Tecnologia da Informação, como aprovar as atividades referentes ao desenvolvimento de sistemas existentes no PDTI (Plano Diretor de Tecnologia da Informação) e definição das prioridades referentes às necessidades de tecnologia da informação para o Instituto.
Equipe de Apoio a Governança de TI	Responsável pelo apoio a gestão da governança de TI, triagem das demandas, verificação do alinhamento com o PDTI e análise da necessidade e planejamento da contratação de soluções de TI.
Equipe de Desenvolvimento	É composta pelos envolvidos na execução das atividades técnicas do subprocesso de Desenvolvimento.
Fiscal Técnico	Servidor representante da área de tecnologia da informação, indicado pela autoridade competente dessa área para fiscalizar tecnicamente o contrato.
Gerente de Projetos	Responsável pelo ciclo de vida do projeto e também do sistema, participando desde o gerenciamento das demandas, análise de viabilidade, estratégias de definição dos projetos e respectivas liberações externas. Participa também da escolha das estratégias de desenvolvimento e definição das respectivas iterações e liberações internas, até a implantação da solução e do acompanhamento do ciclo de sustentação. 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, analisando tendências e propondo novas soluções e melhorias. Também realiza o mapeamento dos processos de negócio, além de realizar ou facilitar os treinamentos e repasse de conhecimento. Deverá ter conhecimento da área de negócio e dos processos da área de TI.
Líder de Projetos	Representante de empresa contratada para desenvolvimento e manutenção de sistemas.



Ministério do Meio Ambiente
Instituto Brasileiro do Meio Ambiente e dos Recursos Naturais Renováveis
Diretoria de Planejamento, Administração e Logística
Centro Nacional de Telemática

	Atua acompanhando a execução dos projetos de forma gerencial e técnica, gerenciando sua equipe e as atividades dos projetos que lhe forem passados por Ordens de Serviço. Responsável por prestar informações precisas e tempestivas sobre o andamento do processo de desenvolvimento, zelando pela qualidade do processo e do software entregue, sempre observando as normas do Instituto, como de Segurança da Informação, de codificação, de interface, de Banco de Dados, de Ambientes, entre outras.
Multiplicadores	Responsáveis por participar dos treinamentos de utilização dos sistemas e disseminar o conhecimento adquirido aos usuários.
Programador	Responsável por realizar todas as atividades relacionadas à codificação e testes unitários, conforme as normas, padrões e diretrizes do Ibama, implementando os casos de uso, integrando os componentes e módulos do sistema, além de gerar e manter as rotinas de implantação.
Requisitante	Pessoa ou grupo responsável por uma demanda de software que será atendida por meio de um produto ou serviço. Deverão informar as necessidades, expectativas e requisitos, além de testar, homologar as entregas e realizar análises de impacto do estudo de viabilidade e na implantação das demandas de software em conjunto com o Gerente de Projetos.



6. SUBPROCESSOS DO PDS-IBAMA

Após a seleção de um escopo específico pelo subprocesso de Gestão de Demandas, inicia o ciclo de vida do Projeto de desenvolvimento/melhoria de um software, que corresponderá a um ciclo do PDS-Ibama, envolvendo a **Concepção, Desenvolvimento e a Transição**.

A Concepção é marcada pelo início do desenvolvimento/melhoria do software, definição do escopo e estratégia de execução.

O Desenvolvimento é marcado pela execução de uma interação que contempla um pacote de casos de uso, formalizado por uma Ordem de Serviço, que passa pelas fases de desenvolvimento do software. O projeto evoluirá por iterações que repetirão o subprocesso de Desenvolvimento com liberações internas de pacotes de casos de uso, em ambiente de homologação, até concluir uma versão do software a ser entregue, gerando a liberação externa no subprocesso de Transição.

A Transição é marcada pela liberação externa da interação em ambiente de produção para utilização dos usuários finais de acordo com as expectativas do requisitante, incluindo treinamento, migrações de dados, procedimentos de instalação e encerramento do projeto.

6.1 CONCEPÇÃO

6.1.1 Introdução

Este subprocesso marca o início do projeto de desenvolvimento ou melhoria de software. Por meio de levantamentos junto ao usuário, devem ser definidas e priorizadas as necessidades do software, delineando o escopo do projeto que define a solução de tecnologia da informação a ser desenvolvida para atender a necessidade de automatização sinalizada na demanda do requisitante.

O projeto evoluirá por iterações que repetirão o subprocesso de Desenvolvimento com liberações internas de pacotes de casos de uso, em ambiente de homologação, até concluir uma versão do software a ser entregue, gerando a liberação externa no subprocesso de Transição, quando, enfim, ocorrerá o encerramento do projeto.

Neste momento do projeto, a priorização de requisitos possui sensível importância devido ao seu impacto no esforço e prazo do projeto. Devem ser priorizados os Casos de Uso mais críticos que auxiliem na criação e validação da arquitetura.

As estimativas de prazo e esforço dependerão da estratégia escolhida, pois poderá ser explorado o paralelismo entre as iterações ou entre as atividades de uma iteração, como também atividades adicionais de validação de cada pacote representarão um esforço fixo a ser adicionado no prazo originalmente previsto. Recomenda-se que cada iteração (pacotes de casos de uso) sejam autocontidos e funcionais.

Os principais produtos desse subprocesso são o Documento de visão do sistema e o Plano de Desenvolvimento, que compreende, principalmente, a visão geral do sistema e a lista de requisitos mapeados em Casos de Uso e sua distribuição em pacotes, que serão executados nas iterações via Ordens de Serviço, constando, ainda, as estimativas e o cronograma de liberações.

Critérios de aceite definidos em termos de parâmetros objetivos e mensuráveis devem ser estabelecidos para verificar se a solução ou serviço recebido para atender a demanda está em conformidade com os requisitos especificados.

Durante todo o processo, os riscos devem ser continuamente monitorados e atenuados, bem como, a gestão de mudança, devendo, se necessário, rever-se a estratégia dos pacotes de casos de uso, sempre em concordância com o requisitante.

6.1.2 Objetivos

- Iniciar formalmente o projeto.
- Compreender a necessidade do requisitante e registrar o escopo do projeto.
- Realizar o planejamento do projeto segundo a MGP-Ibama.
- Criar o Plano de Desenvolvimento, identificando e priorizando os requisitos do projeto, definindo a estratégia de desenvolvimento e gerando estimativas gerenciais de custo e prazo.

6.1.3 Fluxo de Trabalho

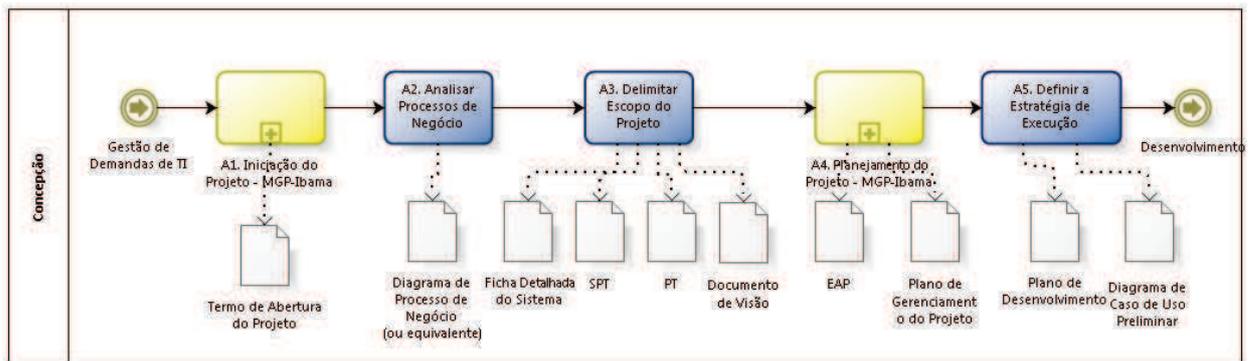


Figura 2: PDS-Ibama: Concepção



A1. Iniciação do Projeto

Objetivo: Formalizar o novo projeto, apresentando as informações básicas para iniciar o planejamento.

- Esse subprocesso corresponde ao grupo de processos de Iniciação da Metodologia de Gestão de Projetos do Ibama (MGP-Ibama)

Ver Metodologia de Gerenciamento de Projetos - MGP-Ibama.

A2. Analisar Processos de Negócio

Objetivo: Entender o negócio e a necessidade da área requisitante por meio da identificação, mapeamento, análise de processos de negócio e definição das fronteiras do sistema.

Entradas:

- MGP-Ibama - Termo de Abertura do Projeto;
- MGP-Ibama - Análise de Viabilidade do Projeto;
- PDS-Ibama - Documento de Visão do Sistema, se existente;
- Mapeamento de Processos de Negócio, ou equivalente.

Descrição da Atividade:

- Entender os processos da Área Requisitante na qual a solução será implantada, considerando as fronteiras do negócio a ser modelado e identificando os problemas atuais e as possibilidades de melhoria;
- Identificar e analisar os processos de negócio: Diagrama, Mapa ou Modelo de Processos, Fluxogramas, Mapa Mentais, entre outros;
- Se necessário, mapear os processos de negócio do projeto, utilizando representações compatíveis com a complexidade da demanda: Diagrama, Mapa ou Modelo de Processos, Fluxogramas, Mapa Mentais, entre outros;
- Definir as fronteiras do sistema, identificando as interações que acontecem entre ele e o mundo externo, bem como os atores externos envolvidos nessas interações;
- Validar, junto aos envolvidos, as informações levantadas e os documentos gerados.

Saídas:

- Diagrama de Processos de Negócio, ou equivalente;



Responsável:

- Gerente de Projetos.

Participantes:

- Líder de projeto;
- Requisitante;
- Administrador de Dados.

A3. Delimitar Escopo do Projeto

Objetivo: Analisar a solicitação recebida, identificar os requisitos básicos e definir o escopo do projeto.

Entradas:

- MGP-Ibama - Termo de Abertura do Projeto.
- MGP-Ibama - Análise de Viabilidade do Projeto;
- PDS-Ibama - Glossário;
- Diagrama de Processos de Negócio, ou equivalente.

Descrição da Atividade:

- Avaliar a demanda recebida e realizar entrevista com a Área Requisitante, a fim de levantar informações sobre as principais necessidades dos envolvidos no projeto;
- Definir o escopo e o não escopo do projeto, identificando o que faz e o que não faz parte do projeto, com relação a características e funcionalidades. O escopo deve ser inteligível pelas áreas gerenciais e técnica;
- Identificar os principais requisitos (funcionais e não funcionais) e os riscos para o projeto. Os requisitos podem ser: de negócio, tecnológicos, recursos humanos, legais, desempenho, segurança, sociais, ambientais e culturais, entre outros;
- Listar as funcionalidades do sistema que serão criadas ou alteradas no projeto, sinalizando a priorização desejada pelo requisitante;
- Validar, junto aos envolvidos, as informações levantadas e os documentos gerados.
- Elaborar proposta técnica conforme as funcionalidades identificadas com estimativa de prazo, volume e custo.

Saídas:

- PDS-Ibama - Documento de Visão do Sistema;
- Solicitação de Proposta Técnica (SPT);
- Proposta Técnica (PT);



- PDS-Ibama - Glossário atualizado.

Responsável:

- Líder de Projeto ou Analista de Requisitos.

Participantes:

- Gerente de Projetos;
- Líder de Projetos;
- Analista de Requisitos;
- Administrador de Dados;
- Requisitante.

A4. Planejamento do Projeto - MGP-Ibama

Objetivo: Planejar as ações do projeto a fim de alcançar os objetivos para os quais o projeto foi criado.

- Esse subprocesso corresponde ao grupo de processos de Planejamento da Metodologia de Gestão de Projetos do Ibama (MGP-Ibama)

Ver Metodologia de Gerenciamento de Projetos - MGP-Ibama.

A5. Definir a Estratégia de Execução

Objetivo: Definir a estratégia de execução do projeto, por meio dos pacotes de casos de uso.

Entradas:

- PDS-Ibama - Glossário;
- MGP - Estrutura Analítica de Projeto (EAP);
- Proposta Técnica da Iteração (PT);
- PDS-Ibama - Documento de Visão do Sistema;
- Diagrama de Processos de Negócio, ou equivalente.



Descrição da Atividade:

- Mapear os requisitos constantes no Diagrama de Casos de Uso (Conjunto de Casos de Uso).
- Priorizar os Casos de Uso mais críticos que auxiliem na criação e validação da arquitetura.
- Distribuir os Casos de Uso em pacotes, que serão desenvolvidos nas iterações por meio de Ordens de Serviços no Plano de Desenvolvimento. É recomendado que não exista dependência funcional entre os pacotes de casos de uso, caso seja necessário, que fique explicitado esta dependência no cronograma de execução do projeto.
- O Plano de Desenvolvimento deve planejar a estrutura do projeto de forma a facilitar sua gestão.
- Atualizar o Documento de Visão do Sistema, caso necessário.

Saídas:

- PDS-Ibama - Documento de Visão do Sistema atualizado;
- PDS-Ibama - Plano de Desenvolvimento; (Gerente de projetos)
- PDS-Ibama - Diagrama de Caso de Uso preliminar;
- PDS-Ibama - Glossário;

Responsável:

- Líder de Projeto ou Analista de Requisitos.

Participantes:

- Gerente de Projetos;
- Arquiteto de Software;
- Analista de Infraestrutura Tecnológica;
- Requisitante.



6.2 DESENVOLVIMENTO

No PDS-Ibama, o projeto evolui por iterações que repetem o subprocesso de Desenvolvimento. Cada iteração contempla um pacote de casos de uso, formalizado por uma Ordem de Serviço e realiza uma passagem completa por todas as fases do subprocesso de Desenvolvimento, como se fosse um pequeno projeto em cascata. As Ordens de Serviço de um projeto podem ser desenvolvidas em paralelo, desde que não ofereçam riscos ao seu gerenciamento.

Na primeira iteração os requisitos arquiteturalmente significativos são descritos com as respectivas interfaces e a arquitetura candidata é definida.

Em cada iteração os casos de uso são especificados, realizados, implementados e testados, a arquitetura é aprimorada e as interfaces são projetadas e implementadas, com uma liberação interna em ambiente de homologação para validação pelo cliente.

Ao término de cada iteração deverá haver uma revisão do tamanho do projeto e das estimativas das iterações, visando adequar o planejamento ao aumento de conhecimento dos processos de negócio envolvidos e soluções na automatização.



Ministério do Meio Ambiente
Instituto Brasileiro do Meio Ambiente e dos Recursos Naturais Renováveis
Diretoria de Planejamento, Administração e Logística
Centro Nacional de Telemática

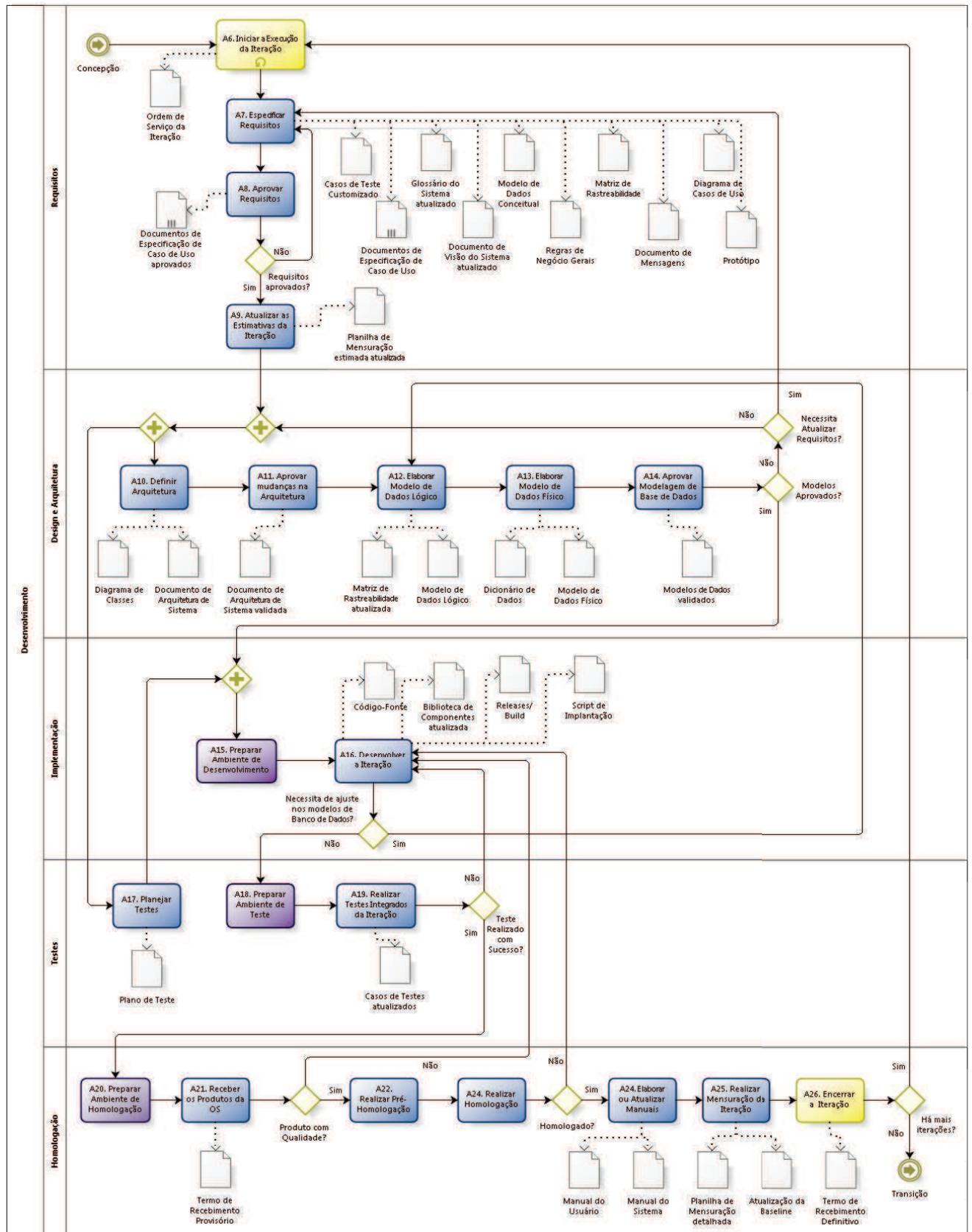


Figura 3: PDS-Ibama: Desenvolvimento



6.2.1 REQUISITOS

6.2.1.1 INTRODUÇÃO

Para cada pacote de casos de uso estabelecido na Concepção, uma iteração é planejada, realizando os Casos de Uso agrupados para o pacote. Nessa fase planeja-se a iteração e faz-se o detalhamento dos requisitos do pacote identificados pelos processos e atividades anteriores, garantindo a visibilidade do escopo para os programadores e clientes e viabilizando, adicionalmente, um adequado controle das mudanças ao longo do ciclo de vida. Caso seja imprescindível absorver uma mudança que altere o escopo de um projeto, ao ser aprovada pelo processo de gestão de mudança, deve-se analisar e medir o impacto utilizando a metodologia apropriada para cálculo do esforço adicional e revisão do tamanho, prazo e custo do projeto, considerando o momento em que ocorreu no ciclo do projeto.

Pode-se fazer uso de protótipos (telas/relatórios) como um mecanismo para a identificação dos requisitos funcionais do software, auxiliando o analista de requisitos e o cliente a entender melhor o que deve ser construído quando os requisitos estiverem confusos.

É fator crítico de sucesso o uso de ferramenta de apoio à modelagem e gestão efetiva dos requisitos.

6.2.1.2 OBJETIVOS

- Realizar uma especificação detalhada dos requisitos identificados para o pacote de casos de uso da respectiva iteração, com base nas documentações aprovadas nos processos anteriores e reuniões complementares com o requisitante, a partir da qual serão construídos modelos na fase subsequente (Design e Arquitetura), a fim de representar o software a ser desenvolvido;
- Identificar atributos suficientes para o projeto, retirando informações que não possuam fins claros;
- Possibilitar a análise dos impactos de mudanças;
- Equilibrar qualidade de informação com custo operacional;
- Facilitar o acompanhamento gerencial dos requisitos e sua rastreabilidade, permitindo o controle de pendências;
- Estabelecer e manter concordância com os envolvidos sobre o que o software deve fazer;
- Oferecer aos programadores uma compreensão das necessidades do software.



6.2.1.3 ATIVIDADES

A6. Iniciar a Execução da Iteração

Objetivo:

Definir os recursos necessários (tempo, custo, equipe, artefatos) de acordo com o escopo, os riscos e a estratégia para desenvolvimento da iteração, estabelecendo as diretrizes a serem seguidas pela equipe de desenvolvimento e formalizando a iteração.

Entradas:

- Termo de Abertura do Projeto;
- Declaração de Escopo do Projeto;
- Documento de Visão do Sistema;
- Plano de Desenvolvimento;
- Proposta Técnica (PT).

Descrição das Atividades:

- Analisar os requisitos funcionais, não funcionais e restrições estabelecidas para a Iteração. Caso seja uma manutenção, identificar os artefatos que serão impactados;
- Analisar as lições aprendidas;
- Identificar os envolvidos;
- Definir os recursos necessários (humanos, financeiros, infraestrutura, entre outros);
- Verificar a necessidade de adaptação do processo (atividades, artefatos, papéis e responsabilidades);
- Definir os marcos intermediários da iteração;
- Atribuir os recursos às atividades ao longo do tempo, buscando a melhor aproximação da expectativa de prazo estabelecida;
- Definir o subconjunto de artefatos que devem ser gerados para essa iteração dentre os previstos para a fase (Lista de artefatos, a constar na Ordem de Serviço);
- Definir os critérios de avaliação/aceite da iteração em termos de parâmetros objetivos e mensuráveis para verificar a conformidade com as necessidades estabelecidas;
- Acompanhar a formalização da Ordem de Serviço da Iteração;
- Elaborar ou atualizar os artefatos pertinentes, utilizando o recurso de controle de alterações, caso já existam, para que fique evidenciado o que foi alterado;
- Rever planejamento inicial e sinalizar necessidade de mudança;
- Formalizar o início da iteração.

Observação: Caso, ao longo da Iteração, um requisito revele-se inviável (ex.: muito caro para ser implementado ou requeira uma arquitetura que não pode ser mantida), uma decisão de negócio pode ser tomada para alteração do escopo e redirecionamento estratégico da iteração e do projeto,



Ministério do Meio Ambiente
Instituto Brasileiro do Meio Ambiente e dos Recursos Naturais Renováveis
Diretoria de Planejamento, Administração e Logística
Centro Nacional de Telemática

podendo haver a repriorização dos Casos de Uso. Nesses casos, deve-se acionar os processos de controle de mudança e revisar os planejamentos do projeto. Em caso de contratação, deve-se ter mecanismos para valorar o retrabalho envolvido.

Saídas:

- Ordem de Serviço da Iteração;
- Planilha de Mensuração (atualizada);
- Atas de Reunião;

Responsável:

- Gerente do Projeto e Requisitante.

Participantes:

- Analista de Métricas;
- Requisitante;
- Analista de Infraestrutura Tecnológica.

A7. Especificar Requisitos

Objetivo: Apropriar-se do domínio da aplicação e detalhar, junto ao requisitante e áreas envolvidas, os requisitos funcionais e não funcionais a serem oferecidos pelo sistema (novos ou impacto sobre funções, dados, características técnicas e de qualidade já existentes) ou na implantação do projeto associados ao escopo estabelecido para a iteração.

Caso a estratégia de desenvolvimento tenha sido utilizar pacotes de casos de uso, na primeira iteração deve-se explorar o detalhamento dos requisitos com foco na eliminação dos principais riscos e o estabelecimento de uma arquitetura de hardware e software estável que suporte, de forma adequada, os requisitos (funcionais e não-funcionais), a partir da qual o sistema poderá ser construído ou evoluído.

Entradas:

- Termo de Abertura do Projeto;
- Diagrama de Processo de Negócio;
- Diagrama de caso de uso preliminar;
- Documento de Visão do Sistema;
- Plano de Desenvolvimento;
- Ordem de Serviço da Iteração;
- Documento de Mensagens;
- Casos de Uso relacionados ao escopo da iteração;
- Regras de Negócio Gerais;
- Glossário;
- Biblioteca de componentes.

Descrição das Atividades:

- Analisar documentação;



Ministério do Meio Ambiente
Instituto Brasileiro do Meio Ambiente e dos Recursos Naturais Renováveis
Diretoria de Planejamento, Administração e Logística
Centro Nacional de Telemática

- Buscar conhecimento sobre o negócio;
- Analisar detalhadamente as necessidades e problemas gerais do projeto, considerando os requisitos, limitações, impactos em outros sistemas e restrições identificadas durante a Concepção;
- Organizar Workshop de Requisitos com todos os envolvidos em potencial (áreas: requisitante, Administração de Dados, Infraestrutura, Gerentes de Projetos, Fiscais dos Contratos). Promover quantos forem necessários e complementar o levantamento com entrevistas individuais;
- Identificar, detalhadamente, os Requisitos Funcionais para o escopo da iteração, com as respectivas regras de negócio, mensagens e os diagramas que forem pertinentes para uma melhor representação e entendimento pelos envolvidos, observando as melhores práticas e organizando nos Casos de Uso, em observância ao Guia para Elaboração de Casos de Uso, caso haja;
- Verificar a existência de informações requeridas nas bases corporativas ou em bases de outros sistemas;
- Especificar as interfaces com o usuário ou com outros sistemas (ou identificar necessidades de adequação, se já existentes), detalhando todas as ações, controles, regras e exceções relacionadas às funcionalidades representadas pelas interfaces;
- Analisar a fronteira estabelecida para o sistema no Documento de Visão e ajustar, caso necessário;
- Identificar, detalhadamente, os Requisitos Não Funcionais (específicos ou gerais) relacionados aos Casos de Uso da iteração, buscando os padrões do governo e do órgão (requisitos técnicos e de qualidade – orientados pela ISO25010, requisitos legais e regulamentares, padrões de aplicativos, sistemas operacionais e ambiente, requisitos de compatibilidade e restrições do projeto, entre outros);
- Verificar a possibilidade de reutilização de componentes;
- Propor soluções alternativas potencialmente mais efetivas ou quando o requisito inicial do usuário não for viável de acordo com as expectativas;
- Identificar os relacionamentos entre os requisitos elicitados para montar ou atualizar as matrizes de rastreabilidade, viabilizando o gerenciamento adequado dos requisitos;
- Elaborar os Casos de Teste e Roteiro de Testes para validação dos requisitos da iteração, com a finalidade de verificar se o produto gerado atende aos requisitos de forma adequada;
- Identificar termos e definições que devem ser contemplados no Glossário do Sistema;
- Elaborar protótipos;
- Elaborar o modelo de dados conceitual;
- Elaborar ou atualizar os artefatos previstos para a fase, com recurso de controle de alteração para que fique evidenciado o que foi impactado a partir das informações obtidas ao longo da atividade, caso já existam, seguindo as regras para elaboração e controle de versão/configuração no armazenamento;
- Publicar os artefatos no repositório de acordo com os padrões de controle de versão/gerência de configuração;



Ministério do Meio Ambiente
Instituto Brasileiro do Meio Ambiente e dos Recursos Naturais Renováveis
Diretoria de Planejamento, Administração e Logística
Centro Nacional de Telemática

- Promover reuniões ou pontos de controle para validar com os envolvidos o entendimento durante a elaboração dos requisitos;
- Aferir a qualidade dos artefatos produzidos;
- Registrar em ata o local, data, participantes, assuntos, desenvolvimento e encaminhamento das reuniões realizadas;
- Sinalizar necessidade de mudança (novas características, evoluções, correções e alterações).

Saídas:

- Documento de Visão do Sistema (atualizado);
- Glossário (atualizado);
- Diagrama de Casos de Uso (atualizado);
- Documentos de Especificação de Casos de Uso da Iteração (atualizado);
- Documento de Mensagens (atualizados, se for manutenção);
- Modelo de dados conceitual;
- Casos de Teste automatizado;
- Roteiro de Testes;
- Matriz de Rastreabilidade (atualizada);
- Regras de Negócio Gerais (atualizada);
- Atas de Reunião;
- Protótipos;
- Outros documentos previstos na OS.

Responsável:

- Analista de Requisitos.

Participantes:

- Requisitante;
- Líder de Projeto;
- Gerente do Projeto;
- Analista de Métricas;
- Gerentes de outros Sistemas;
- Áreas de apoio (Administração de Dados, de Banco de Dados, Infraestrutura, fiscalização de contratos de software);



A8. Aprovar Requisitos

Objetivo: Validar os artefatos de requisitos elaborados na atividade anterior conforme dos critérios de aceite definidos para verificar a conformidade com as necessidades estabelecidas.

Entradas:

- Ordem de Serviço da Iteração;
- Todos os artefatos previstos na Ordem de Serviço da Iteração;
- Demais artefatos relacionados com a demanda (Ex. Documento de Visão do Sistema, Glossário, Casos de Uso, Regras de Negócio Gerais etc.)

Descrição das Atividades:

- Verificar se os artefatos previstos para a iteração foram gerados, entregues e armazenados conforme as normas de gerência de configuração/control de versão;
- Analisar artefatos a partir dos requisitos, com o apoio de especialistas e com o requisitante, quando necessário.
- Avaliar a aderência às normas e padrões adotados pelo Ibama.
- Submeter os artefatos para ajuste quando não atender ao especificado.
- Registrar as não conformidades das funcionalidades através de Termo de Desvio de Qualidade, estabelecendo-se um prazo para as correções.
- Havendo não conformidades, conforme os critérios de avaliação/aceite da iteração, submeter os artefatos para ajuste, estabelecendo o prazo conforme definido na Ordem de Serviço da Iteração;
- O Requisitante e o Gerente de Projetos devem aprovar os Casos de Uso;
- O Requisitante, o Líder de Projeto, o Analista de Requisitos e o Gerente de Projetos devem assinar os Casos de Uso.

Saídas:

- Especificação de Casos de Uso aprovados e assinados.

Responsável:

- Gerente de Projetos.

Participantes:

- Requisitante;
- Líder de Projetos;
- Analista de Requisitos;
- Áreas de apoio (Administração de Dados, Infraestrutura, fiscalização de contratos de software, Equipe de Desenvolvimento).

A9. Atualizar as Estimativas da Iteração



Objetivo: A partir do refinamento dos requisitos, obter novas estimativas do tamanho do projeto e respectivas derivações de esforço, prazo e custo, revendo as ponderações das iterações para validação/ajustes no planejamento.

Entradas:

- Ordem de Serviço da Iteração;
- Todos os artefatos previstos na Ordem de Serviço da Iteração;
- Demais artefatos relacionados com a demanda (Ex. Documento de Visão do Sistema, Glossário, Casos de Uso, Regras de Negócio Gerais etc.);
- Declaração de Escopo do Projeto;
- Diagrama de Processos de Negócio;
- Protótipos da iteração;
- Planilha de Mensuração estimada (anterior).

Descrição das Atividades:

- Reunir artefatos da iteração necessários para a identificação dos elementos que subsidiarão a aplicação da técnica de estimativa de tamanho;
- Definir e aplicar o método de medição a ser utilizado, de acordo com as normas do Ibama e baseado no grau de informações recebido;
- Sinalizar não conformidades ao identificar inconsistências nos artefatos ou necessidades adicionais de informação. Dependendo da gravidade, retornar para o gerente solicitar ao responsável, estabelecendo prazo para ajuste, ou assumir premissas para viabilizar um resultado, quando necessário, registrando-as para apoio à tomada de decisão;
- Consolidar a estimativa do tamanho funcional do projeto e derivação em esforço, prazo e custo para o projeto e ponderação dos pacotes de casos de uso (iterações);
- Analisar os requisitos não funcionais para verificar a necessidade de ajuste das estimativas;
- Publicar e comunicar o resultado da atualização das estimativas;
- Atualizar a base histórica de medições do projeto e do sistema;
- Propor melhorias no processo de aprovação dos requisitos;
- Rever planejamento inicial, atualizando-o quando necessário.

Saídas:

- Planilha de Mensuração estimada atualizada;
- *Baseline* de medição da Aplicação atualizada.

Responsável:

- Analista de Métricas.

Participantes:

- Gerente de Projetos;
- Requisitante;
- Líder de Projetos;
- Analista de Requisitos;



- Equipe de desenvolvimento;
- Áreas de apoio (Administração de Dados, Infraestrutura, fiscalização de contratos de software).

6.2.2 DESIGN E ARQUITETURA

6.2.2.1. INTRODUÇÃO

O projeto técnico de uma solução de software, ou *design*, compreende atividades durante as quais decisões táticas e estratégicas são tomadas para atender os requisitos funcionais e não funcionais de um sistema, as restrições do projeto, os padrões corporativos estabelecidos, as tecnologias a serem utilizadas na solução e os riscos envolvidos, elaborando modelos para descrever a realização dos Casos de Uso a serem utilizados como abstrações para a implementação.

A arquitetura de um sistema de software, por sua vez, é a sua organização, ou estrutura (em determinado momento), que pode ser repetidamente decomposta em partes que interagem por meio de interfaces, relações que conectam partes, e restrições para associar partes. Tais partes compreendem classes, componentes e subsistemas que mapeiam as funcionalidades do produto e compreendendo, ainda, considerações acerca de desempenho, flexibilidade, reutilização, abrangência, restrições tecnológicas e econômicas, portabilidade e estética.

Como já mencionado neste documento, a primeira iteração deve priorizar a realização dos Casos de Uso mais críticos e arquiteturalmente significativos, buscando definir uma arquitetura executável, candidata para a solução do software. Uma arquitetura executável é uma implementação parcial do sistema, criada para demonstrar propriedades e funções selecionadas, especialmente aquelas que satisfazem requisitos não funcionais, a fim de que a capacidade funcional completa do sistema possa ser adicionada na fase de implementação de uma forma segura.

6.2.2.2. OBJETIVOS

- Transformar os requisitos de negócio em modelos (lógico e físico) do sistema a ser implementado;
- Desenvolver uma arquitetura para o sistema, abrangendo as decisões significativas sobre sua estrutura, projetando-a para oferecer alto desempenho e atendimento das necessidades das áreas de negócio do Ibama;
- Antecipar e mitigar os riscos técnicos do projeto relacionados a desempenho, rendimento, capacidades, confiabilidade etc.;
- Selecionar os elementos estruturantes e as interfaces por meio das quais o sistema é



composto, em conjunto com o comportamento, conforme especificado pelos requisitos;

- Garantir o alinhamento técnico do modelo e arquitetura às diretrizes e estratégias tecnológicas definidas pelo CNT, mantendo a estabilidade, a consistência e a atualização das informações nos artefatos e ferramentas envolvidas.

6.2.2.3. ATIVIDADES

A10. Definir Arquitetura

Objetivo: Definir os subsistemas e componentes de um sistema de software e dos relacionamentos entre eles. Esta atividade especifica o projeto arquitetural do sistema contemplando todos os elementos para atender os requisitos de desenvolvimento, manutenção e produção do sistema.

O projeto arquitetural contempla a estrutura dos componentes de dados e programas necessários ao sistema e serve como um meio de comunicação entre o arquiteto e outros membros da equipe do projeto, visando estabelecer a estrutura colaborativa de elementos do sistema e do ambiente corporativo do Ibama. Está também relacionado aos estilos e padrões arquiteturais adotados.

Entradas:

- Ordem de Serviço da Iteração;
- Todos os artefatos previstos na Ordem de Serviço da Iteração;
- Demais artefatos relacionados com a demanda (Ex. Documento de Visão do Sistema, Glossário, Casos de Uso, Regras de Negócio Gerais etc.);
- Diagrama de Processos de Negócio;
- Biblioteca de componentes.

Descrição da Atividade:

- Analisar as características do projeto e avaliar se a arquitetura já existente no ambiente corporativo atende ou se são necessárias mudanças, adaptações e/ou novas definições;
- Sendo necessárias mudanças, adaptações e/ou novas definições da arquitetura, estas deverão ser submetidas a grupo de trabalho multidisciplinar do CNT que analisará a solicitação;
- Transformar os requisitos em um design do sistema a ser criado, respeitando os padrões de arquitetura do Ibama;
- Projetar componentes a fim de fornecer comportamento apropriado aos requisitos do sistema;
- Avaliar os componentes quanto a sua possível aplicabilidade na arquitetura do software;
- Propor padrões de projetos, estrutura de componentes, pacotes e interfaces;
- Definir aspectos de arquitetura segura;
- Definir aspectos mínimos de arquitetura de hardware: capacidade de processamento e



memória, armazenamento, backup, *throughput* (proporcional à quantidade de acessos simultâneos), disponibilidade (*Cluster*, *Load-Balance* ou Sistemas Distribuídos) e parâmetros de monitoração;

- Definir os critérios e requisitos (ferramentas, *plug-in*, entre outros) para instalação do sistema nos ambientes, quando aplicável.

Saídas:

- Diagrama de Classes (Quando solicitado na OS);
- Biblioteca de componentes (atualizada);
- Documento de Arquitetura do Sistema.

Responsável:

- Arquiteto de Software.

Participante:

- Líder de Projetos;
- Analista de Requisitos.

A11. Aprovar mudanças na Arquitetura

Objetivo: Verificar e aprovar mudanças, adaptações e/ou novas definições da arquitetura de sistema.

Entradas:

- Documento de Arquitetura do sistema.

Descrição da Atividade:

- Avaliar risco da arquitetura, verificando os pontos de entrada (interfaces, serviços, protocolos, códigos, etc.) e saída (mensagens, respostas, relatórios, etc.) do sistema que podem ser utilizados para fins maliciosos no software;
- Identificar mecanismos de segurança na aplicação (autenticação, controle de acesso, uso de criptografia, *logs*);
- Modelar as ameaças com a finalidade de avaliar e documentar os pontos fracos ou riscos de segurança de uma aplicação;

Saídas:

- Documento de Arquitetura do sistema validado.

Responsável:

- Grupo de Trabalho multidisciplinar do CNT, composto por representantes dos Áreas de Projetos, Infraestrutura de Redes, Banco de Dados, Segurança da Informação e Chefia do CNT.



Participantes:

- Analista de Segurança da Informação;
- Líder de Projetos;
- Arquiteto de Software;
- Equipe de Tratamento de Incidentes em Redes Computacionais (ETIR);
- Gerente do Projeto.

A12. Elaborar Modelo de Dados Lógico

Objetivo: Modelar o relacionamento entre as entidades de negócio, especificando o projeto do banco de dados do sistema sob o ponto de vista lógico, e não físico.

Entradas:

- Ordem de Serviço da Iteração;
- Todos os artefatos previstos na Ordem de Serviço da Iteração;
- Demais artefatos relacionados com a demanda (Ex. Documento de Visão do Sistema, Glossário, Casos de Uso, Regras de Negócio Gerais etc.);
- Diagrama de Processos de Negócio;
- Biblioteca de componentes;
- Modelos de dados de sistemas envolvidos;
- Esquemas das bases corporativas.

Descrição da Atividade:

- Mapear (ou atualizar, se for manutenção ou não for a primeira iteração), em tabelas, as entidades de negócio do escopo do projeto (se primeira iteração) ou da iteração para dar suporte ao armazenamento e à recuperação de dados;
- Definir os relacionamentos entre as tabelas;
- Identificar e documentar a rastreabilidade entre as tabelas, os Casos de Uso e as Funções de Dados.
- Normalizar o modelo até a terceira forma normal;
- Aplicar padrões de nomenclatura e relacionamentos de objetos de banco de dados do Ibama.
- Submeter ao Administrador de Banco de Dados o modelo de dados lógico para verificação e validação de conformidade com os padrões de nomenclatura, relacionamentos, normalização e rastreabilidade.

Saídas:

- Modelo de Dados Lógico;
- Matriz de Rastreabilidade (atualizada).

Responsável:

- Administrador de Dados.

Participante:



- Líder de Projetos;
- Analista de Requisitos;
- Arquiteto de Banco de Dados;
- Administrador de Banco de Dados.

A13. Elaborar Modelo de Dados Físico

Objetivo: Especificar o projeto do banco de dados, definindo e mantendo o modelo de dados físico do sistema, bem como, os requisitos e componentes arquiteturais do banco de dados, levando em conta as limitações do Sistema Gerenciador de Banco de Dados (SGBD) utilizado e os requisitos não funcionais, de forma a garantir a persistência dos dados com consistência, integridade e eficiência.

Entradas:

- Ordem de Serviço da Iteração;
- Todos os artefatos previstos na Ordem de Serviço da Iteração;
- Demais artefatos relacionados com a demanda (Ex. Documento de Visão do Sistema, Glossário, Casos de Uso, Regras de Negócio Gerais etc.)
- Modelo de Dados Lógico;
- Biblioteca de componentes;
- Diagrama de Classes.

Descrição da Atividade:

- Validar o Modelo de Dados Lógico, onde a partir dele será elaborado o modelo de dados físico;
- Mapear classes ou entidades para a visão física, com os respectivos atributos, garantindo os relacionamentos estabelecidos entre as tabelas;
- Avaliar o desempenho do modelo de dados e o acesso aos dados;
- Adequar a normalização do modelo para atender às necessidades de desempenho;
- Otimizar as estruturas de dados;
- Otimizar acesso aos dados por meio de indexação;
- Definir tabelas de referência;
- Definir valores *default* (padrão) para os atributos de dados;
- Definir regras para a integridade referencial e de dados;
- Definir, se necessário, rotinas de automação para o monitoramento dos dados;
- Definir parâmetros de armazenamento ou *tablespace*;
- Definir os parâmetros para criação do Banco de Dados para os ambientes de Desenvolvimento, Testes, Homologação, Treinamento e Produção;
- Gerar e/ou atualizar o *script* de criação/atualização do banco de dados;
- Aplicar os padrões de nomenclatura de objetos de banco de dados do Ibama, tipos de



dados, integridade entre entidades/objetos, especificação contendo a descrição dos atributos das tabelas dentre outros itens, inclusive propor melhorias;

- Criar, definir e refinar o modelo de dados físico para dar suporte ao armazenamento e à recuperação de classes persistentes;
- Submeter ao Arquiteto de Banco de Dados os modelos de dados lógico e físico para verificação e validação de conformidade com os padrões de nomenclatura, tipos de dados, relacionamentos, normalização, integridade referencial e de dados, parâmetros de armazenamento e *tablespace*, rastreabilidade, desempenho do modelo e acesso aos dados.

Saídas:

- Dicionário de Dados (Descrições, Índices, *Scripts*, *Procedures*, *Triggers*, *Views*, *Grants* e Parâmetros para a criação do Banco de Dados);
- Modelo de Dados Físico.

Responsável:

- Administrador de Banco de Dados.

Participante:

- Líder de Projetos;
- Arquiteto de Banco de Dados;
- Analista de Requisitos;
- Administrador de Dados;
- Arquiteto de Software.

A14. Aprovar Modelagem de Banco de Dados

Objetivo: Verificar e aprovar a modelagem de banco de dados lógico e físico.

Entradas:

- Diagrama de Classes;
- Documento de Arquitetura do sistema;
- Dicionário de Dados;
- Modelo de Dados Físico;
- Modelo de Dados Lógico.

Descrição da Atividade:

- Verificar se os artefatos previstos para a iteração foram gerados, entregues e armazenados conforme esta norma e as normas de Padrão de Objetos e Estruturas de Banco de Dados e Padrão dos Ambientes de Construção de Sistemas de Informação.
- Avaliar a aderência às normas e padrões adotados pelo Ibama.
- Submeter os artefatos para ajuste quando não atender ao especificado.
- Registrar as não conformidades das funcionalidades através de Termo de Desvio de Qualidade, estabelecendo-se um prazo para as correções.



Ministério do Meio Ambiente
Instituto Brasileiro do Meio Ambiente e dos Recursos Naturais Renováveis
Diretoria de Planejamento, Administração e Logística
Centro Nacional de Telemática

- Verificar a adequação das nomenclaturas, ações de objetos de banco de dados, *scripts* de criação e manipulação de banco de dados, normalização e critérios de desempenho.
- Identificar mecanismos de segurança na aplicação (autenticação, controle de acesso, uso de criptografia, *logs*).

Saídas:

- Modelos de Dados Lógico e Físico validados.

Responsável:

- Arquiteto de Banco de Dados.

Participantes:

- Líder de Projetos;
- Arquiteto de Software;
- Administrador de Banco de Dados;
- Administrador de Dados;
- Gerente do Projeto.



6.2.3. IMPLEMENTAÇÃO

6.2.3.1. INTRODUÇÃO

Nesta fase é realizada a transformação dos requisitos de uma iteração em código-fonte, na linguagem estabelecida, para fornecer uma nova funcionalidade ou evoluir uma funcionalidade existente, construindo as classes de dados que serão armazenadas pelo sistema e o comportamento associado a elas em métodos que viabilizarão as operações, ou regras de negócio.

Recomenda-se que seja definida uma estratégia de desenvolvimento visando reutilização.

6.2.3.2 OBJETIVOS

- Preparar a infraestrutura necessária para o desenvolvimento de um determinado grupo de requisitos;
- Produzir (ou manter) o código-fonte conforme o design e a arquitetura estabelecida, aplicando recursos reutilizáveis existentes ou identificando oportunidades para futura reutilização;
- Planejar e executar os testes unitários das partes do software, realizando as devidas correções, quando for o caso;
- Realizar a integração do código-fonte;
- Gerar uma liberação interna do projeto;
- Gerar o *Script* de Implantação.



6.2.3.3 ATIVIDADES

A15. Preparar Ambiente de Desenvolvimento

Objetivo: Esta atividade tem o propósito de preparar a infraestrutura necessária para o desenvolvimento de um determinado grupo de requisitos. O ambiente de desenvolvimento deverá reproduzir o futuro ambiente de produção e observar as restrições do projeto.

Entradas:

- Biblioteca de Componentes;
- Documento de Arquitetura de sistema;
- Documento de Casos de Uso;
- Documento de Visão do Sistema.

Descrição da Atividade:

- Definir e configurar os servidores necessários para suportar o ambiente de desenvolvimento, conforme o Padrão dos Ambientes de Construção de Sistemas de Informação do Ibama;
- Instalar e configurar os sistemas, ferramentas, *plug-ins* e acessórios necessários para o desenvolvimento.

Saídas:

- Endereço do ambiente de desenvolvimento.

Responsável:

- Analista de Infraestrutura Tecnológica;

Participante:

- Gerente de Projetos;
- Líder de Projetos;
- Administrador de Banco de Dados.



A16. Desenvolver a Iteração

Objetivo: Esta atividade tem o propósito de implementar, testar e integrar a solução da iteração.

Entradas:

- Ordem de Serviço da Iteração;
- Todos os artefatos previstos na Ordem de Serviço da Iteração;
- Demais artefatos relacionados com a demanda (Ex. Documento de Visão do Sistema, Glossário, Casos de Uso, Regras de Negócio Gerais etc.)
- Biblioteca de Componentes;
- Diagramas de Classes;
- Diagramas de Pacotes;
- Dicionário de Dados;
- Documento de Arquitetura de sistema;
- Casos de Teste;
- Roteiro de Testes automatizado;
- Documento de Casos de Uso;
- Documento de Visão do Sistema;
- Documento de Mensagens;
- Glossário;
- Matriz de Rastreabilidade;
- Modelos de Dados;
- Protótipos.

Descrição da Atividade:

- Produzir o código-fonte conforme a padronização de codificação e o *framework* adotado;
- Controlar linha de base (*baseline*, *label*, *build*, *release*) dos códigos fontes/itens de configuração e verificar a correta definição da linha de base conforme os itens de configuração do escopo da Ordem de Serviço da Iteração;
- Para novos sistemas, desenvolver utilizando o padrão de projeto Model-View-Controller (MVC);
- Revisar o código produzido de acordo com os padrões de codificação definido pela Ibama, realizando as devidas correções, quando for o caso;
- Planejar e executar os testes unitários, realizando as devidas correções, quando for o caso;
- Realizar a integração do código-fonte;
- Gerar uma liberação interna (*release*). Em caso de iteração única ou última iteração será gerada uma versão (*build*);
- Atualizar o repositório do projeto com o código-fonte e os documentos produzidos;
- Avaliar se os objetivos e os critérios estabelecidos foram alcançados e se foram concluídos todos os itens de trabalho atribuídos;



- Verificar se o código passou por todos os testes de programador antes de ser entregue a versão;
- Definir as atividades e recursos para a homologação/implantação do sistema;
- Gerar o *Script* de Implantação.

Saídas:

- Biblioteca de Componentes (atualizada);
- Código-fonte;
- *Release ou Build*;
- *Script* de Implantação.

Responsável:

- Programador;

Participante:

- Líder de Projetos;
- Analista de Requisitos;
- Arquiteto de Software;
- Analista de Métricas;
- Administrador de Banco de Dados;
- Administrador de Dados;
- Gerente de Projetos.

6.2.4 TESTES

6.2.4.1 INTRODUÇÃO

A fase de testes verifica as suposições, os riscos e as incertezas inerentes ao desenvolvimento e consiste na execução de atividades que permitam determinar se o que foi implementado na iteração atende às especificações no contexto do sistema como um todo (e não sob o ponto de vista individual, já contemplado pelo teste unitário na fase de implementação), fornecendo *feedback* sobre a maturidade do projeto para que os defeitos sejam eliminados o mais cedo possível no ciclo de vida do projeto.

Os Casos de Teste são elaborados na fase de Requisitos, baseados nos Casos de Uso, com condições específicas de satisfação que a Solução deve alcançar. A fase de Implementação, por sua vez, produz construções incrementais do sistema para serem validadas pela fase de teste que, em cada iteração, fornece um *feedback* objetivo a respeito do atendimento às necessidades do requisitante.

6.2.4.2 OBJETIVOS



O objetivo central desta fase é identificar falhas e não conformidades em relação à especificação, de forma que as causas possam ser reveladas e corrigidas pela equipe de desenvolvimento antes da entrega final do produto, garantindo que a liberação interna satisfaça os requisitos da iteração com qualidade.

São, ainda, objetivos específicos desta fase:

- Planejar a realização dos testes, considerando os Casos de Teste elaborados na fase de Requisitos;
- Preparar toda a infraestrutura necessária para a execução dos testes;
- Integrar, ao sistema, o que foi implementado na iteração;
- Executar os testes conforme planejado, avaliando, registrando e comunicando as não conformidades identificadas;
- Garantir que as mudanças no sistema não introduzam novos defeitos.

6.2.4.3 ATIVIDADES

A17. Planejar Testes

Objetivo: Esta atividade é responsável por definir o escopo, objetivos, estratégia, fases, recursos (ferramentas, hardware, entre outros), atividades e os tipos de testes necessários à execução, controle e acompanhamento do processo de desenvolvimento visando aumentar a confiabilidade do software. São executados de modo sistemático com o objetivo de encontrar falhas. O planejamento dos testes pode ser refinado conforme o melhor entendimento dos requisitos e a definição da arquitetura, sendo revisto a cada iteração e monitorado durante todo o processo de desenvolvimento de software.

Entradas:

- Ordem de Serviço da Iteração;
- Plano de Desenvolvimento;
- Documento de Visão do Sistema;
- Documento de Arquitetura;
- Casos de Teste automatizado;
- Biblioteca de Componentes;
- Documento de Casos de Uso;
- Glossário;
- Matriz de Rastreabilidade;
- Protótipos.

Descrição da Atividade:

- Estabelecer o objetivo do teste;
- Determinar riscos associados ao software;



- Estabelecer o escopo do teste;
- Estabelecer as tarefas de teste;
- Estabelecer as ferramentas de teste;
- Definir o ambiente do teste;
- Definir a equipe de teste;
- Definir os requisitos de arquitetura necessários para os testes caixa branca, caixa preta;
- Definir critérios para avaliar influência da arquitetura nos testes de requisitos não funcionais do sistema;
- Definir os tipos de teste como de integração, de validação, de carga, de desempenho, de funcionalidade, de segurança e de sistema;
- Construir o cronograma das atividades de teste;
- Definir critérios de aceitação;
- Definir critérios de suspensão;
- Definir como devem ser comunicados os resultados dos testes;
- Definir os eventos, pessoas responsáveis e dependências de evento necessárias para garantir o teste;
- Definir necessidade de operação assistida.

Saídas:

- Plano de Testes.

Responsável:

- Analista de Testes.

Participantes:

- Líder de Projetos;
- Gerente de projetos;
- Analista de requisitos;
- Arquiteto de software.

A18. Preparar Ambiente de Testes

Objetivo: Esta atividade tem o propósito de preparar toda a infraestrutura necessária para a execução do Plano de Teste, de acordo com o especificado no *Script* de Implantação.

A cada Ordem de Serviço, todo o código-fonte é integrado e disponibilizado para os testes por meio de uma liberação interna (*release*). Em caso de iteração única ou última iteração será



gerada uma versão (*build*).

Entradas:

- Plano de Teste;
- Documento de Visão do Sistema;
- *Release* ou *Build*;
- *Script* de Implantação.

Descrição da Atividade:

- Definir e configurar os servidores necessários para suportar o ambiente de testes, conforme o Padrão dos Ambientes de Construção de Sistemas de Informação do Ibama;
- Instalar e configurar os sistemas, ferramentas, *plug-ins*, e acessórios necessários para os testes;
- Executar o *Script* de Implantação no ambiente de testes;
- Divulgar o endereço do ambiente de testes, comunicando que o ambiente está preparado.

Saídas:

- Endereço do Ambiente de Testes.

Responsável:

- Analista de Infraestrutura Tecnológica;

Participante:

- Analista de Teste;
- Administrador de Banco de Dados;
- Gerente de Projetos;
- Líder de Projetos.



A19. Realizar Testes Integrados da Iteração

Objetivo: Executar os testes conforme planejamento e estratégia definidos na atividade “Planejar Testes” para validar a *release* ou *build*. Nesta fase deve-se procurar executar testes automatizados utilizando ferramenta específica.

Entradas:

- Plano de teste;
- Casos de Teste;
- Roteiro de Testes;
- Documento de Casos de Uso;
- Documento de Visão do Sistema;
- Glossário;
- Matriz de Rastreabilidade;
- Protótipos.

Descrição da Atividade:

- Executar os casos de testes e de integração de acordo com os procedimentos definidos no plano de teste;
- Avaliar e registrar os resultados obtidos no registro de testes;
- Revisar os casos de teste quando for identificada necessidade de aperfeiçoamento ou ajuste;
- Validar consistência entre os artefatos envolvidos no teste;
- Reportar a ocorrência e a gravidade das falhas;
- Demandar a atividade de revisão do software à área responsável pela correção ou à área responsável pelo encaminhamento dessa atividade, caso sejam identificados erros no software sendo testado.

Saídas:

- Casos de Teste atualizados com os registros;

Responsável:

- Analista de Testes.

Participantes:

- Líder de Projetos;
- Analista de Requisitos;
- Analista de Infraestrutura Tecnológica;
- Gerente de Projetos.



6.2.5 HOMOLOGAÇÃO

6.2.5.1 INTRODUÇÃO

Nesta fase os testes do cliente concentram-se em um ambiente independente do desenvolvimento e de produção, denominado homologação, para testes do cliente e da equipe técnica do órgão.

O requisitante valida a iteração realizada nesta fase. Serão avaliados se todos os requisitos solicitados no escopo da iteração foram atendidos pela liberação sob o ponto de vista do requisitante. Podem ser utilizados os Casos de Teste como apoio na abrangência da avaliação realizada.

6.2.5.2 OBJETIVOS

O objetivo central desta fase é verificar se todos os requisitos funcionais e não funcionais da iteração foram atendidos, de acordo com as expectativas do requisitante, e se os padrões de desenvolvimento do Ibama foram observados durante o processo.

São, ainda, objetivos específicos desta fase:

- Preparar toda a infraestrutura necessária para viabilizar os testes da iteração pelo requisitante;
- Homologar a demanda junto à área requisitante;
- Constituir um marco de aceitação, pelo cliente, de todos os artefatos gerados na iteração;
- Caso seja a última iteração, o resultado será o aceite do projeto do ponto de vista do requisitante, de maneira que uma liberação externa será gerada e entregue para o subprocesso de Transição.

6.2.5.3 FLUXO DE TRABALHO

A20. Preparar Ambiente de Homologação

Objetivo: Esta atividade tem o propósito de preparar toda a infraestrutura necessária para a validação, pelo cliente, de uma liberação interna, de acordo com o especificado no *Script* de Implantação. O ambiente de homologação deverá reproduzir o ambiente de produção. A cada Ordem de Serviço, todo o código-fonte é integrado e disponibilizado para a homologação por meio de uma liberação interna (*release*). Em caso de iteração única ou última iteração será gerada uma versão (*build*).



Entradas:

- Casos de Teste;
- Ordem de Serviço da Iteração;
- Documento de Arquitetura de sistema;
- Documento de Visão do Sistema;
- *Release* ou *Build*;
- *Script* de Implantação.

Descrição da Atividade:

- Definir e configurar os servidores necessários para suportar o ambiente de homologação, conforme o Padrão dos Ambientes de Construção de Sistemas de Informação do Ibama;
- Instalar e configurar os sistemas, ferramentas, *plug-ins*, e acessórios necessários para a homologação;
- Publicar o endereço do ambiente, comunicando que o ambiente está preparado ou sinalizar necessidades de adequações;
- Verificar se o ambiente de homologação é o mais próximo possível das condições de infraestrutura de produção;
- Preparar o banco de dados de homologação e, caso seja necessário, deve-se providenciar a migração da estrutura e/ou dos dados do banco de dados entre os ambientes;
- Criar, se for o caso, e atualizar a estrutura de diretório do projeto;
- Executar o *Script* de Implantação no ambiente de homologação;
- Verificar se todos os componentes estão no estado inicial correto para o início da homologação;
- Divulgar o endereço do ambiente de homologação, comunicando que o ambiente está preparado.

Saídas:

- Endereço do Ambiente de Homologação.

Responsável:

- Analista de Infraestrutura Tecnológica.

Participante:

- Administrador de Banco de Dados;
- Gerente de Projetos;
- Líder de Projetos.



A21. Receber e Avaliar os Produtos da OS

Objetivo: Receber os produtos da OS, incluindo o código-fonte e artefatos e avaliar a qualidade se estão em conformidade com os padrões definidos e com as normas estabelecidas.

Entradas:

- Todos os artefatos previstos na Ordem de Serviço da Iteração;
- Ordem de Serviço da Iteração;
- Plano de Desenvolvimento;
- MGP - Declaração de Escopo do Projeto;
- Padrões (de codificação, de interface etc.).

Descrição das Atividades:

- Verificar a lista de artefatos discriminada na OS. Caso não esteja completa, devolver a OS para que seja providenciada a entrega completa;
- Verificar se os artefatos previstos para a iteração foram gerados, entregues e armazenados conforme as normas de gerência de configuração/controle de versão;
- Caso seja a última iteração do projeto e tenha sido solicitado na OS, a empresa contratada deve entregar a Matriz de Rastreabilidade, o Manual do Sistema e o Manual do Usuário, quando aplicável;
- Emitir o Termo de Recebimento Provisório ao receber o objeto e artefatos da Ordem de Serviço;
- Avaliar a qualidade do código escrito e adequação aos padrões definidos conforme tecnologia adotada;
- Avaliar a aderência às normas e padrões adotados pelo Ibama;
- Registrar as não conformidades das funcionalidades através de Termo de Desvio de Qualidade, estabelecendo-se um prazo para as correções.

Saídas:

- Termo de Recebimento Provisório (TRP);
- Termo de Desvio de Qualidade (TDQ);
- Aceite de Liberação para Homologação.

Responsável:

- Fiscal Técnico.

Participantes:

- Gerente de Projetos;
- Líder de Projetos.
- Analista de Métricas;
- Administrador de Dados;
- Analista de Infraestrutura Tecnológica.



A22. Realizar Pré-Homologação

Objetivo: Verificar se a solução desenvolvida na iteração atende aos requisitos funcionais e não-funcionais estabelecidos.

Entradas:

- Ordem de Serviço da Iteração;
- Casos de Teste;
- Roteiro de Testes.

Descrição da Atividade:

- Testar as funcionalidades implementadas, podendo-se levar em consideração os casos de teste;
- Verificar o atendimento aos requisitos funcionais e não-funcionais;
- Registrar as não conformidades das funcionalidades através de Termo de Desvio de Qualidade, estabelecendo-se um prazo para as correções.
- Agendar reunião para apresentação das funcionalidades desenvolvidas para o requisitante, e demais interessados, no ambiente de homologação.

Saídas:

- Termo de Desvio de Qualidade (TDQ);
- Aceite da Homologação Técnica

Responsável:

- Gerente de Projetos.

Participante:

- Líder de Projetos;
- Analista de Requisitos;
- Analista de Testes.



A23. Realizar Homologação

Objetivo: Verificar se a solução desenvolvida na iteração atende aos requisitos estabelecidos.

Entradas:

- Ordem de Serviço da Iteração;
- Casos de Teste automatizados;
- Artefatos de Requisitos.

Descrição da Atividade:

- O líder de projetos, ou pessoa designada, deverá realizar apresentação das funcionalidades desenvolvidas para o requisitante, e demais interessados, no ambiente de homologação;
- Testar as funcionalidades implementadas, podendo-se levar em consideração os casos de teste;
- Verificar o atendimento às regras de negócio, aos requisitos funcionais e aos requisitos não-funcionais;
- Registrar as não conformidades das funcionalidades através de Termo de Desvio de Qualidade, estabelecendo-se um prazo para as correções.

Saídas:

- Termo de Desvio de Qualidade (TDQ);
- Aceite da Homologação.

Responsável:

- Requisitante.

Participante:

- Gerente de Projetos;
- Líder de Projetos;
- Analista de Requisitos;
- Analista de Testes.



A24. Elaborar ou Atualizar Manuais

Objetivo: Elaborar, na última iteração do projeto, os documentos para o suporte à operação, produção e treinamento dos usuários no sistema a ser implantado. O material produzido deve conter informações necessárias para a instalação, operação, uso e manutenção do sistema finalizado.

Entradas:

- Especificação de Casos de Uso;
- Sistema implantado em ambiente de homologação.

Descrição da Atividade:

Caso o projeto de software esteja sendo desenvolvido por empresa contratada, os Manuais deverão ser entregues, se solicitado na OS, após a homologação da última iteração do projeto, juntamente aos demais artefatos.

Desenvolver manual do sistema

- Elaborar o documento que descreva os procedimentos de instalação, configuração, operação e manutenção do sistema.

Desenvolver manual do usuário

- Elaborar o documento que descreva a utilização do sistema. Para a maioria dos produtos, projetados profissionalmente, são produzidos como aplicativos próprios, como os sistemas de ajuda ou sites da Web.

Opções de formato dos manuais

- Help on-line;
- Documento eletrônico.

Saídas:

- Manual do usuário;
- Manual do sistema.

Responsável:

- Analista de Requisitos (Manual do Usuário);
- Arquiteto de Software (Manual do Sistema).

Participantes:

- Líder de Projetos;



- Arquiteto de Software;
- Analista de Requisitos;
- Analista de Infraestrutura Tecnológica;
- Gerente de Projetos.

A25. Realizar Mensuração da Iteração

Objetivo: Realizar análise e mensuração da iteração e atualizar a linha de base da aplicação.

Entradas:

- Ordem de Serviço da Iteração;
- Todos os artefatos previstos na Ordem de Serviço da Iteração;
- Demais artefatos relacionados com a demanda (Ex. Documento de Visão do Sistema, Glossário, Casos de Uso, Regras de Negócio Gerais etc.)
- Declaração de Escopo do Projeto;
- Diagrama de Processos de Negócio;
- Termo de Recebimento Definitivo;
- Planilha de mensuração (estimada e detalhada);
- *Baseline* de medição da Aplicação;
- Descrição de caso de uso;
- Navegação no sistema em ambiente de homologação;
- Biblioteca de componentes;
- Atas de Reunião.

Descrição da Atividade:

- Coletar a documentação existente relacionada à demanda para auxiliar na análise da contagem;
- Identificar o propósito da contagem;
- Registrar o fundamento da contagem (versão CPM, versão do Roteiro de Métricas do Ibama);
- Identificar a fronteira da aplicação;
- Buscar o apoio dos envolvidos para entendimento do escopo e de possíveis impactos, se necessário;
- Realizar a contagem de pontos de função, registrando os elementos em planilha;
- Registrar demais campos da planilha de contagem, como nome do sistema, período da



realização da análise e contagem;

- Datar e assinar a planilha de Contagem de Pontos de Função;
- Sinalizar não conformidades ao identificar inconsistências nos artefatos ou necessidades adicionais de informação. Dependendo da gravidade, retornar para o gerente solicitar ao responsável, estabelecendo prazo para ajuste;
- Caso seja a última iteração, realizar, adicionalmente, a medição final do tamanho funcional do projeto, atualizando a base histórica de medições do projeto e do sistema;
- Em alguns casos, a pedido da chefia do CNT ou quando o Analista de Métricas avaliar ser necessário, poderá ser confeccionado um Parecer Técnico sobre a contagem realizada, contendo as premissas adotadas, documentação que fundamentou a contagem e outros itens pertinentes.

Saídas:

- Planilha de mensuração detalhada atualizada;
- *Baseline* de medição da Aplicação.

Responsável:

- Analista de Métricas.

Participantes:

- Gerente de Projetos;
- Líder de Projetos;
- Requisitante;
- Áreas de apoio (Administração de Dados, Infraestrutura, fiscalização de contratos de software, Área de Negócio).



A26. Encerrar a Iteração

Objetivo: Formalizar o fim da iteração.

Entradas:

- Ordem de Serviço da Iteração;
- Planilha de mensuração atualizada;
- *Baseline* de medição da Aplicação atualizada.

Descrição da Atividade:

- Coletar informações e indicadores da iteração;
- Atualizar cronograma do projeto;
- Revisar o plano de desenvolvimento;
- Coletar as lições aprendidas;
- Verificar a entrega dos manuais de usuário e de sistema, quando solicitados na OS;
- Coletar o Termo de Recebimento Definitivo junto ao requisitante, havendo conformidade aos requisitos estabelecidos, e encaminhá-lo ao Gestor do Contrato.

Saídas:

- Termo de Recebimento Definitivo;
- Plano de Desenvolvimento atualizado;
- Plano de Gerenciamento do Projeto atualizado;
- Lições Aprendidas.

Responsável:

- Gerente de Projetos.

Participantes:

- Todos os participantes da iteração.

6.3 TRANSIÇÃO

6.3.1 INTRODUÇÃO

Este subprocesso descreve as atividades que garantem que a liberação externa será disponibilizada no ambiente de produção para utilização dos usuários finais de acordo com as expectativas do requisitante, incluindo treinamento, migrações de dados, ajustes finais e procedimentos de instalação e encerrando o projeto.

6.3.2 OBJETIVOS

- Verificar e aplicar as estratégias de compatibilidade, conversão e migração de dados;
- Executar o *Script* de Implantação para a liberação externa;
- Estabilizar o sistema, de acordo com os critérios estabelecidos;
- Viabilizar ambiente para treinamento;
- Realizar treinamento para os multiplicadores e equipe de sustentação;
- Mapear possíveis ocorrências de incidentes;
- Coletar os indicadores finais do projeto e do processo;
- Encerrar o projeto, internalizando lições aprendidas.

6.3.3 FLUXO DE TRABALHO

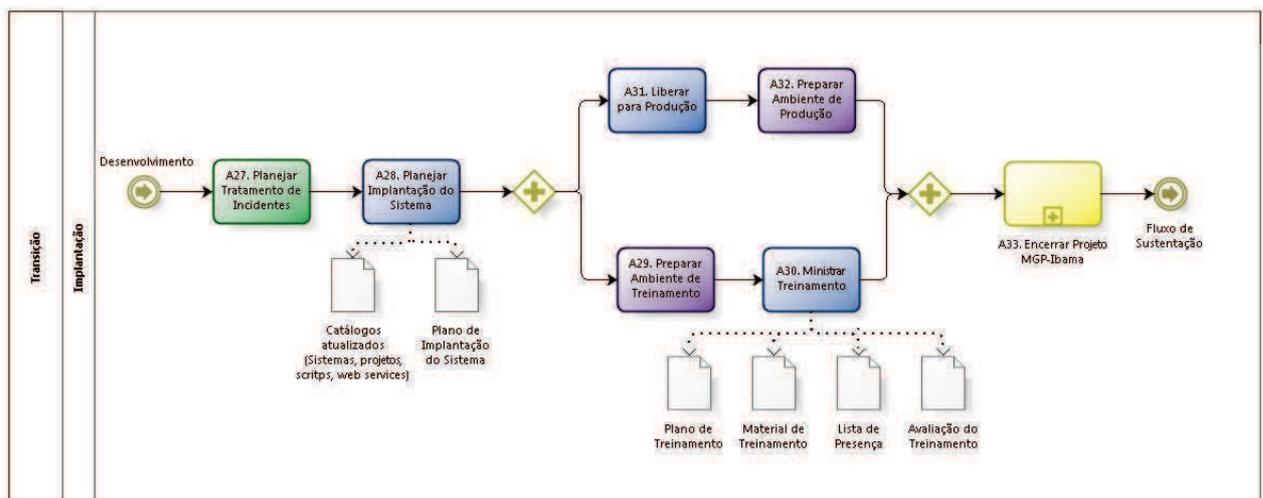


Figura 4: PDS-Ibama: Transição



A27. Planejar Tratamento de Incidentes

Objetivo: Planejar como os incidentes serão tratados, indicando qual a ação será tomada e quem será o responsável por tratar o incidente.

- Esse subprocesso corresponde a uma atividade do Processo de Gerenciamento de Tratamento de Incidentes Computacionais do Ibama.

A28. Planejar Implantação do Sistema

Objetivo: Elaborar o plano de implantação do sistema e manter atualizada a base de conhecimento de sistemas para promover a divulgação, treinamento e auxiliar a gestão de projetos.

Entradas:

- Artefatos do Projeto.

Descrição da Atividade:

- Elaborar o plano de implantação do sistema com o requisitante;
- Atualizar os dados do Portfólio de Projetos de TI;
- Atualizar o Catálogo de Sistemas;
- Atualizar o Catálogo de *Web Services* com as interfaces, caso necessário;
- Atualizar o Catálogo de *Scripts*, caso necessário;
- Atualizar a Ficha Detalhada do Sistema.

Saídas:

- MGP - Portfólio de Projetos atualizada;
- Catálogo de sistemas atualizado;
- Catálogo de *Web Services* atualizado;
- Catálogo de *Scripts* atualizado;
- Ficha Detalhada do Sistema atualizada;
- Plano de Implantação do Sistema.

Responsável:

- Gerente de Projetos.

Participantes:

- Líder de Projetos;
- Requisitante;
- Administrador de Dados;
- Analista de Infraestrutura Tecnológica.



A29. Preparar Ambiente de Treinamento

Objetivo: Esta atividade tem o propósito de preparar toda a infraestrutura necessária para a realização do treinamento na utilização do sistema no escopo do projeto.

Entradas:

- Documento de Arquitetura de sistema;
- Documento de Visão do Sistema;
- *Build*;
- *Script* de Implantação.

Descrição da Atividade:

- Definir e configurar os servidores necessários para suportar o ambiente de treinamento, conforme o Padrão dos Ambientes de Construção de Sistemas de Informação do Ibama;
- Instalar e configurar os sistemas, ferramentas, *plug-ins*, e acessórios necessários para o treinamento;
- Executar o *Script* de Implantação no ambiente de treinamento;
- Divulgar o endereço do ambiente de treinamento, comunicando que o ambiente está preparado.

Saídas:

- Endereço do Ambiente de Treinamento.

Responsável:

- Analista de Infraestrutura Tecnológica.

Participantes:

- Administrador de Banco de Dados;
- Líder de Projetos;
- Gerente de Projetos.



A30. Ministrando Treinamento

Objetivo: Esta atividade foca na elaboração e realização do treinamento, caso necessário, aos requisitantes, multiplicadores e equipe de suporte do CNT para transferência de conhecimento do sistema.

Entradas:

- Manual do Sistema;
- Manual do Usuário.

Descrição da Atividade:

Elaborar o treinamento

- Elaborar plano de treinamento conforme os padrões do Ibama;
- Elaborar apresentação sobre manutenção do sistema, baseado nos manuais;
- Planejar e dimensionar turmas de treinamento;
- Agendar treinamentos;
- Reservar sala, infraestrutura (projektor, computadores, etc);
- Convocar envolvidos para os treinamentos.

Ministrar o treinamento para Multiplicadores e Equipe Técnica

- Apresentar procedimentos de operação do sistema para os requisitantes, multiplicadores e atendentes dos Serviços Ibama;
- Apresentar procedimentos de suporte do sistema para equipe técnica de TI;
- Recolher assinaturas na lista de presença;
- Coletar sugestões dos participantes relacionadas às funcionalidades do sistema;
- Avaliar aderência dos manuais do sistema e do usuário;
- Avaliar o treinamento por meio de pesquisa de opinião dos participantes;
- Controlar a presença dos participantes do treinamento.

Ministrar o treinamento para os Usuários do Sistema

- Apresentar procedimentos de operação do sistema para os participantes;
- Recolher assinaturas na lista de presença;
- Coletar sugestões dos participantes relacionadas às funcionalidades do sistema;
- Avaliar o treinamento por meio de pesquisa de opinião dos participantes;
- Controlar a presença dos participantes do treinamento.

Saídas:

- Plano de treinamento;
- Material de treinamento;



- Lista de Presença;
- Sugestões dos participantes;
- Avaliação do Treinamento.

Responsável:

- Líder de Projeto (treinamento para multiplicadores e equipe técnica);
- Requisitante (treinamento para os usuários do sistema).

Participantes:

- Multiplicadores;
- Atendentes dos Serviços Ibama;
- Usuários do sistema;
- Gerente de Projetos;
- Arquiteto de Software.

A31. Liberar para Produção

Objetivo: Aprovar a entrada do projeto em ambiente de produção.

Entradas:

- Termos de Recebimento Definitivos das Ordens de Serviço do Projeto;
- *Script* de Implantação;
- Plano de Implantação do Sistema.

Descrição da Atividade:

- Verificar com a Área Requisitante o momento adequado de se colocar o sistema em produção, alinhado ao Plano de Implantação do Sistema;
- Aprovar a implantação do sistema em produção.

Saídas:

- Autorização de implantação em produção.

Responsável

- Gerente de Projetos.

Participantes:

- Requisitante;
- Comitê de Gestão de Mudanças.



A32. Preparar Ambiente de Produção

Objetivo: Elaborar o *checklist* de implantação do sistema com a definição das atividades para a implantação do sistema em ambiente de produção bem como as necessidades, requisitos e disponibilidade de artefatos e recursos. Analisar a infraestrutura do ambiente de produção para verificar a sua adequabilidade às exigências do sistema.

Entradas:

- Documento de Arquitetura;
- Documento de Visão do Sistema;
- *Build* do sistema;
- *Scripts* de Banco;
- *Script* de Implantação.

Descrição da Atividade:

- Definir e configurar os servidores necessários para suportar o ambiente de produção, conforme o Padrão dos Ambientes de Construção de Sistemas de Informação do Ibama;
- Instalar e configurar os sistemas, ferramentas, *plug-ins*, e acessórios necessários para o ambiente de produção;

Criar unidade de implantação

- Controlar linha de base (*baseline, label, build, release*) dos códigos fontes/itens de configuração e verificar a correta definição da linha de base conforme os itens de configuração do escopo do sistema com apoio do líder de projeto;

Preparar plano de contingência

- Elaborar plano de recuperação do servidor e retorno da versão anterior em caso de problema na instalação.

Implantar em Ambiente de Produção e Comunicar aos envolvidos

- Identificar as características técnicas do ambiente de produção, seus recursos e limitações;
- Identificar as características do ambiente de produção relevantes para o sistema a ser implantado, tais como: servidores, capacidades de processamento, memória, armazenamento de dados, de banda de rede e backup, sistemas operacionais e softwares existentes e suas licenças de uso, bem como capacitação do pessoal;
- Verificar, também, a projeção de utilização dessas capacidades pelos sistemas atuais e futuros (previstos);



- Verificar as necessidades do sistema decorrentes dos requisitos não-funcionais;
- Executar o *script* de implantação no ambiente de produção;
- Divulgar o endereço do ambiente de produção, comunicando que o ambiente está preparado.

Providenciar Correção de Erros Encontrados

- Analisar e providenciar correção dos erros encontrados na implantação em ambiente de produção;
- Retornar a versão anterior em caso de problema na instalação, após autorização do Gerente de Projetos.
- Atuar em conjunto com equipe específica de acordo com a natureza do problema (Desenvolvimento, Banco de Dados, entre outros);
- Registrar os erros ou problemas encontrados e as soluções adotadas.

Saídas:

- *Script* de Implantação atualizado;
- Endereço do Ambiente de Produção.

Responsável:

- Analista de Infraestrutura Tecnológica.

Participante:

- Gerente de Projetos;
- Líder de Projetos;
- Administrador de Banco de Dados.

A33. Encerrar Projeto

Objetivo: Registrar formalmente o encerramento do projeto.

- Esse subprocesso corresponde ao grupo de processos de Planejamento da Metodologia de Gestão de Projetos do Ibama (MGP-Ibama)

Ver Metodologia de Gerenciamento de Projetos - MGP-Ibama.



7. MONITORAMENTO COM INDICADORES

Neste tópico serão descritas algumas métricas de supervisão, controle e monitoramento de projetos de software no âmbito do PDS-Ibama, com o objetivo de informar, periodicamente, ao gerente do projeto e gestores do Instituto como está o andamento dos trabalhos. Para isto é conveniente que a coleta, computação e apresentação das informações estejam automatizadas. É fundamental a facilidade de obtenção dos valores que alimentam os indicadores, para resultar em informações mais consistentes e precisas.

A coleta das métricas deve ser armazenada em um repositório do projeto sob a responsabilidade do Gerente do Projeto, seguindo as normas de controle de versão/gerência de configuração.

Como o Processo de Desenvolvimento de Software do Ibama utiliza uma abordagem iterativa, os indicadores devem focar no processo e não apenas no produto. Portanto, deve-se considerar não apenas o resultado (ex.: tamanho do projeto em Pontos de Função), mas também o processo em si, sua variação no tempo, onde se possa acompanhar uma tendência, auxiliando uma tomada de decisão gerencial.

Assim, foram definidos dois Indicadores, conforme descrito a seguir.

ISC – INDICADOR DE SATISFAÇÃO DO CLIENTE	
Tópico	Descrição
Finalidade	Medir o grau de satisfação da Área de Negócio (gestores) quanto ao processo, serviço e participação das áreas requisitantes e de TI no projeto.
Instrumento de medição	Questionários de forma impressa, ou de forma eletrônica (preferencialmente) com a utilização da ferramenta LimeSurvey.
Forma de acompanhamento	A avaliação será realizada após o término do projeto, com a aplicação do questionário à área de negócio e posterior cálculo do indicador. Porém, os requisitantes, equipe de projetos e demais envolvidos poderão antecipadamente, com o conhecimento do indicador e questionário, promover ações necessárias para maximizar a satisfação e sucesso do projeto.
Periodicidade de apuração	Uma única vez, após o término do projeto.
Mecanismo de Cálculo (métrica)	$\text{ISC} = \frac{\sum_{1}^n \text{Nota}_n}{n}$ <p>Onde: ISC – Indicador de Satisfação do Cliente; n – Quantidade de questões aplicáveis do Formulário; Nota_n – Nota obtida para a questão n, considerando os valores: Péssimo: 1; Ruim: 2; Razoável: 3; Bom: 4; Ótimo: 5.</p>
Início de Vigência	A partir do início do projeto, com o Termo de Abertura do Projeto.



Ministério do Meio Ambiente
Instituto Brasileiro do Meio Ambiente e dos Recursos Naturais Renováveis
Diretoria de Planejamento, Administração e Logística
Centro Nacional de Telemática

IQD – INDICADOR DE QUALIDADE DA DOCUMENTAÇÃO	
Tópico	Descrição
Finalidade	Medir o nível de qualidade da documentação produzida e entregue, em conformidade com os requisitos e normas do Instituto.
Instrumento de medição	<ul style="list-style-type: none">- Lista dos artefatos obrigatórios a serem produzidos ou atualizados durante o Processo de Desenvolvimento de Software do projeto em questão;- Templates do PDS-Ibama, Normas e Padrões do Instituto relacionadas, como Padrão de Codificação de Sistemas de Informação, Padrão de Objetos e Estrutura de Banco de Dados do Ibama, Padrão de Ambiente de Construção de Sistemas de Informação, etc.- Ordens de Serviço do projeto.
Forma de acompanhamento	Verificação da lista de artefatos obrigatórios a serem elaborados ou atualizados para o projeto, definido pelo Gerente de Projetos, registrando-se os itens entregues e em conformidade com as normas e padrões adotados pelo Instituto.
Periodicidade de apuração	Após o encerramento do projeto. Porém, os requisitantes, equipe de projetos e demais envolvidos poderão antecipadamente, com o conhecimento do indicador e das normas e dos padrões, promover ações necessárias para maximizar o atendimento do indicador.
Mecanismo de Cálculo (métrica)	$IQD = \frac{TAQ}{TAE}$ <p>Onde: IQD – Indicador de Qualidade da Documentação; TAQ – Total de Artefatos entregues e dentro dos Padrões e Normas do Instituto; TAE – Total de Artefatos Exigidos para o projeto.</p>
Início de Vigência	A partir do início do projeto, com o Termo de Abertura do Projeto.

Desta forma, as áreas gestoras do Instituto poderão definir metas para seus projetos de software, no Plano de Gerenciamento do Projeto (MGP), com base nestes indicadores, definindo valores a serem atingidos, de acordo com a especificidade dos seus projetos, levando em considerando prazo, equipe, riscos, entre outros.

7.1 REVISÃO DO PROCESSO DE DESENVOLVIMENTO DE SOFTWARE

Com vistas a promover a melhoria contínua do PDS-Ibama, é necessária a sua revisão periódica, pelo menos a cada 2 (dois) anos. Assim, processos, atividades, tarefas, artefatos e outros elementos poderão ser incluídos, retirados ou alterados, contribuindo para o alcance da melhoria contínua do processo.



8. ARTEFATOS DO PDS-IBAMA

Tendo em vista a padronização da documentação, foram elaborados os *templates* (modelos) dos artefatos (documentos) a serem produzidos nas diversas fases PDS-Ibama.

Os artefatos do PDS-Ibama são classificados em obrigatórios e complementares. Os artefatos complementares passam a ser obrigatórios a medida que a gestão de projetos avalie a sua necessidade para o projeto.

Excepcionalmente os artefatos indicados como obrigatórios, após análise do contexto e dos riscos pelo gerente de projetos, podem ter sua elaboração dispensada.

Os termos de aceite e as validações dos artefatos podem ser substituídos por registros em ferramenta automatizada definidas pelo Ibama.

N	Nome do artefato	Ferramenta	Novo Sistema / Módulo	Manutenção (exceto corretiva)
1	Planilha de Mensuração (Estimada ou detalhada)	LibreOffice Calc	Obrigatório	Obrigatório
2	Diagrama de Processo de Negócio	Bizagi, BPA, Xmind	Complementar	Complementar
3	Glossário do Sistema	LibreOffice Writer	Complementar	Complementar
4	Documento de Visão	LibreOffice Writer	Obrigatório	Complementar
5	Modelo de Dados Conceitual	Datamodeler	Obrigatório	Obrigatório
6	Diagrama de Casos de Uso	Astah LibreOffice Writer	Obrigatório	Obrigatório
7	Plano de Desenvolvimento	LibreOffice Writer	Obrigatório	Obrigatório
8	Ordem de Serviço da Iteração	LibreOffice Writer	Obrigatório	Obrigatório
9	Ata de Reunião	LibreOffice Writer	Obrigatório	Obrigatório
10	Especificação de Casos de Uso	LibreOffice Writer	Obrigatório	Obrigatório
11	Documento de Mensagens	LibreOffice Writer	Obrigatório	Complementar
12	Casos de Teste automatizados	LibreOffice Writer	Obrigatório	Complementar
13	Matriz de Rastreabilidade	A definir	Obrigatório	Complementar
14	Regras de Negócio Gerais	LibreOffice Writer	Obrigatório	Complementar
15	Protótipos	A definir	Obrigatório	Obrigatório
16	Modelo de Dados Lógico	Datamodeler	Obrigatório	Obrigatório
17	Modelo de Dados Físico	Datamodeler	Obrigatório	Obrigatório



Ministério do Meio Ambiente
Instituto Brasileiro do Meio Ambiente e dos Recursos Naturais Renováveis
Diretoria de Planejamento, Administração e Logística
Centro Nacional de Telemática

18	Diagrama de Classes - somente se orientado a objeto	Astah	Complementar	Complementar
19	Biblioteca de Componentes	LibreOffice Writer	Obrigatório	Complementar
20	Documento de Arquitetura de Sistema	LibreOffice Writer	Obrigatório	Complementar
21	Dicionário de Dados	Datamodeler	Obrigatório	Obrigatório
22	Código-Fonte	Eclipse, Git, Jenkins, Flyway	Obrigatório	Obrigatório
23	Plano de Testes	LibreOffice Writer	Complementar	Complementar
24	Manual do Sistema	LibreOffice Writer	Obrigatório	Complementar
25	Manual do Usuário	LibreOffice Writer	Obrigatório	Complementar
26	Plano de Treinamento	LibreOffice Writer	Complementar	Complementar
27	Lista de Presença	LibreOffice Writer	Obrigatório	Complementar
28	Avaliação de Treinamento	Lime Survey	Obrigatório	Complementar
29	Script de Implantação em Ambientes	LibreOffice Writer	Obrigatório	Obrigatório
30	Plano de Implantação do Sistema	LibreOffice Writer	Obrigatório	Complementar



9. GLOSSÁRIO

Neste documento serão consideradas as seguintes definições:

9.1 GLOSSÁRIO DE ARTEFATOS DO PDS-IBAMA

N	Nome do artefato	Objetivo
1	Planilha de Mensuração (Estimada e Detalhada)	Registrar as estimativas e a medição do tamanho funcional do projeto, as derivações em esforço, prazo e ponderações dos pacotes de desenvolvimento e respectivos elementos.
2	Diagrama de Processo de Negócio	No DPN é construído o diagrama de fluxo de trabalho, contendo os processos e as atividades relacionadas, identificando os atores responsáveis. Os passos de cada atividade devem ser descritos da forma mais clara possível.
3	Glossário do Sistema	Identificar e descrever o vocabulário do negócio/sistema.
4	Documento de Visão	Fornecer uma base completa e de alto nível para os requisitos funcionais e não funcionais do sistema, bem como o que não é escopo do sistema. Este artefato é escrito a partir da perspectiva dos clientes, focando nas características essenciais do sistema, compreendendo a lista dos requisitos, solução tecnológica e nos níveis aceitáveis de qualidade. Deve evoluir ao longo do ciclo de vida do produto, mas cada versão deve conter a expectativa da visão completa que o sistema deve ter após a entrega do projeto e deve ser atualizada caso sejam identificadas necessidades de adequação nos requisitos ao longo do projeto, para que ao final esteja representando o produto entregue. Funciona como um regulador com base no qual todas as decisões futuras deverão ser validadas.
5	Modelo de Dados Conceitual	Modelo de dados com representação de alto nível que considera o ponto de vista do usuário criador dos dados.
6	Diagrama de Casos de Uso	Descreve as principais funcionalidades do sistema, que foram divididas em Casos de Uso, e a interação dessas funcionalidades com os usuários do mesmo sistema. Geralmente são compostos por: Cenário, atores, casos de uso e comunicação (entre atores e casos de uso).
7	Plano de Desenvolvimento	Documentar a estratégia de desenvolvimento do projeto a partir do agrupamento dos Casos de Uso em pacotes, que corresponderão às iterações e respectivas liberações internas. O Plano de Desenvolvimento deve planejar a estrutura do projeto de forma a facilitar sua gestão.



Ministério do Meio Ambiente
Instituto Brasileiro do Meio Ambiente e dos Recursos Naturais Renováveis
Diretoria de Planejamento, Administração e Logística
Centro Nacional de Telemática

8	Ordem de Serviço da Iteração	Formaliza a autorização da execução do desenvolvimento/manutenção de demanda de sistema (pacote de Casos de Uso), que engloba as fases desde o requisito até o encerramento da iteração com a emissão do Termo de Recebimento Definitivo. Contém a identificação do sistema, a área requisitante, o escopo da demanda, a data de início e duração da execução, o <i>checklist</i> de artefatos a serem entregues, nome e assinatura do requisitante e gestor do contrato.
9	Ata de Reunião	Registrar os participantes e as deliberações realizadas nas reuniões do projeto.
10	Especificação de Casos de Uso	Capturar o comportamento necessário ao sistema sob a perspectiva do usuário final, para alcançar um ou mais objetivos. Deve conter: fluxo principal, alternativos e de exceção; pré-condições e pós-condições; regras de negócio específicas; especificações de interface; diagramas; e requisitos não funcionais específicos. Diferentes usuários se beneficiam de formas diferentes:- Requisitantes usam para descrever, ou pelo menos aprovar a descrição do comportamento do sistema.- Arquitetos usam para identificar a funcionalidade arquiteturalmente significativa.- Desenvolvedores usam para compreender o comportamento requerido do sistema de forma que eles possam identificar classes e métodos a partir do fluxo de eventos dos Casos de Uso.- Analistas de Teste usam como uma base para identificar um subconjunto de Casos de Teste necessários.- Gerentes de Projeto e Líderes Técnicos usam para replanejar e avaliar o trabalho de cada iteração.- Os responsáveis pela elaboração dos manuais usam para compreender a sequência de comportamento do sistema que eles necessitam descrever na documentação.
11	Documento de Mensagens	Registrar as mensagens a serem apresentadas pelo sistema nas situações previstas pelos requisitos do usuário.
12	Casos de Teste	Descreve os cenários de testes, pré-condições e resultados esperados, visando orientar a avaliação da implementação dos casos de uso em termos de atendimento aos requisitos especificados.
13	Matriz de Rastreabilidade	Registrar as informações para a rastreabilidade entre as tabelas, os Casos de Uso, as funcionalidades, os componentes e os elementos necessários para a técnica de medição de tamanho funcional.
14	Regras de Negócio Gerais	Especificar e detalhar as regras de negócio que podem ser comuns a mais de um caso de uso.



Ministério do Meio Ambiente
Instituto Brasileiro do Meio Ambiente e dos Recursos Naturais Renováveis
Diretoria de Planejamento, Administração e Logística
Centro Nacional de Telemática

15	Protótipos	Protótipo é um modelo de tela sem funcionalidades implementadas, contendo apenas apresentação gráfica. Utilizado para fins de ilustração e melhor entendimento.
16	Modelo de Dados Lógico	Representar a estrutura lógica (entidades de negócio) das informações persistentes mantidas e referenciadas pelo sistema, além de seus relacionamentos, independente do armazenamento físico e da tecnologia do banco de dados.
17	Modelo de Dados Físico	Representar a estrutura física de armazenamento dos dados persistentes que são mantidos e referenciados pelo sistema. Depende da tecnologia do banco de dados.
18	Diagrama de Classes	Representação das classes do sistema que implementam as suas funcionalidades. As classes representam um conjunto de objetos que compartilham as mesmas responsabilidades, relacionamentos, operações, atributos e semântica.
19	Biblioteca de Componentes	Apresenta os componentes de sistemas como classes, pacotes de banco, elementos de software, entre outros. Seu registro serve para facilitar sua manutenção e sua reutilização por outras aplicações, informando sobre as regras de negócio, formas de utilização, parâmetros, entidades envolvidas, dados manipulados etc..
20	Documento de Arquitetura de Sistema	Fornecer o contexto e as orientações, decisões e restrições (com justificativas) para que os desenvolvedores construam o sistema. Pode conter, ainda: referências aos requisitos arquiteturalmente significantes; referências aos elementos de design arquiteturalmente significantes; instruções de empacotamento para subsistemas e componentes; camadas e subsistemas críticos; interfaces de sistema críticas; principais abstrações; classes de análise importantes; principais cenários que descrevam comportamento crítico do sistema. O DAS deve indicar, também, quem deve manter e alterar a arquitetura após a primeira concepção do sistema e das motivações técnicas.
21	Dicionário de Dados	Apresentar uma descrição de todos os elementos do banco de dados do sistema.
22	Código-Fonte	Arquivos de Instruções escritas em uma linguagem de programação, em formato texto, utilizado para gerar o sistema executável para o usuário.
23	Plano de Testes	O objetivo do plano de testes é determinar as diretrizes, ambiente de teste, ferramentas, artefatos, pessoas e cronograma para a execução dos testes.
24	Manual do Sistema	Material que auxilia e orienta a equipe de sustentação nos



Ministério do Meio Ambiente
Instituto Brasileiro do Meio Ambiente e dos Recursos Naturais Renováveis
Diretoria de Planejamento, Administração e Logística
Centro Nacional de Telemática

		procedimentos de instalação, operação e manutenção do sistema.
25	Manual do Usuário	Material que auxilia e orienta os usuários finais no processo de aprendizagem e utilização do sistema.
26	Plano de Treinamento	Planejar as ações de capacitação dos multiplicadores do sistema no que tange ao escopo do projeto (funcionalidades e operações do sistema). Envolve identificar o que fazer e como fazer, ou seja, o projeto pedagógico contendo: justificativa, objetivos, participantes, período, localização, metodologia (recursos pedagógicos, didáticos e instrucionais que serão utilizados para apoio), conteúdo programático, carga-horária, custos, avaliação e outras informações pertinentes.
27	Lista de Presença	Registra a assiduidade dos participantes do treinamento.
28	Avaliação de Treinamento	Avalia o treinamento ministrado visando sua contínua melhoria. Podem ser avaliados o material, a metodologia, o instrutor, a duração do curso, entre outros.
29	<i>Script</i> de Implantação	Descrever o conjunto das tarefas necessárias para instalar e validar o produto desenvolvido no projeto de maneira que ele possa ser efetivamente transferido para o ambiente de desenvolvimento, teste, homologação, treinamento e produção.
30	Plano de Implantação do Sistema	Descrever os atores envolvidos e suas responsabilidades na implantação, premissas e ações necessárias para a implantação do sistema. Requisitos operacionais, de infraestrutura e de suporte ao sistema e usuários, bem como, o cronograma de implantação.



9.2 GLOSSÁRIO GERAL

APF: É a técnica que mede o tamanho funcional de um software, a partir de uma perspectiva funcional e independente da tecnologia adotada. É independente da linguagem, dos métodos de programação, hardware ou plataforma utilizada. Ou seja, a contagem de pontos de função de um sistema permanece constante em qualquer local que seja contado, diferenciando apenas a quantidade de esforço necessário para desenvolvê-lo. Ponto de função (PF) é a unidade de medida desta técnica.

Artefatos: Documentos ou guias produzidos durante a realização das atividades deste PDS, sendo utilizados basicamente para o registro de informações do projeto, bem como, para orientar sobre a execução de atividades do projeto. São produtos de trabalhos tangíveis e bem definidos consumidos, produzidos ou modificados por tarefas. Artefatos podem ser compostos por outros artefatos. São exemplos de artefatos: um modelo, como o Modelo de Casos de Uso ou o Modelo de Design. Um elemento do modelo, ou seja, um elemento existente em um modelo, como uma classe ou um subsistema.

Atividades: Atividades são grupos de tarefas, normalmente são utilizadas para efeito de planejamento e controle dos projetos.

Banco de dados: (1) Uma coleção de dados relacionados armazenados juntos, com redundância controlada de acordo com um esquema, para ser utilizado por um ou mais aplicativos. (2) Todos os arquivos de dados armazenados no sistema. (3) Um conjunto de dados armazenados juntos e gerenciados por um sistema de gerenciamento de banco de dados.

Base histórica de medições do projeto e do sistema: Consolidar as estimativas e medidas do projeto, por iteração, para: geração dos indicadores, revisão dos métodos estabelecidos, ajuste do planejamento ao aumento de conhecimento dos processos de negócio envolvidos e soluções na automatização. Envolve os elementos necessários para as métricas de tamanho e respectivas derivações de esforço, prazo, custo e ponderações das iterações subsequentes, além da geração dos indicadores do projeto estabelecidos.

Baseline de medição da Aplicação - Planilha de mensuração do sistema contendo a contagem de aplicação como está em produção. Deverá ser atualizada ao final de cada projeto entregue em produção.

Builds: Componentes executáveis que fazem parte de uma versão operacional do sistema.

Business Case: Documento necessário para incluir demanda no PDTI. Registra a necessidade de TI e seu alinhamento com os objetivos estratégicos do Ibama. Propõe nível de prioridade (GUT), metas e ações, plano de investimento e gestão de risco.

Catálogo de Scripts: Armazenar rotinas que foram desenvolvidas para a realização de apurações especiais, de maneira que possam ser reutilizadas (ex.: correção de problemas ou atualizações na base de dados, geração de relatórios ou arquivos). Além do código, deve ser armazenada uma documentação básica para orientar seu uso, como descrição, objetivos, parâmetros, diferentes



formas de utilização, entre outras informações pertinentes.

Catálogo de sistemas: Repositório que identifica e fornece descrição básica dos sistemas existentes no Ibama (sistemas legados) e dos sistemas que estão em processo de desenvolvimento (sistemas novos). Deve estar alinhado ao PDTI e indicar o Gerente de Projetos responsável pelo sistema.

Catálogo de Web Services: Descreve os serviços disponibilizados pelo Instituto juntamente com sua documentação de implementação contendo as interfaces, parâmetros de entrada e saída, mensagens e ambientes de homologação e produção, juntamente com suas regras de segurança.

CMM: *Capability Maturity Model*. Modelo para avaliação da maturidade dos processos de software de uma organização e para identificação das práticas-chave que são requeridas para aumentar a maturidade desses processos.

Desenvolvedor (Programador): Uma pessoa responsável pelo desenvolvimento da funcionalidade necessária de acordo com os procedimentos e os padrões adotados no projeto.

Documento de Oficialização de Demanda (DOD): Formalizar a demanda do requisitante, sendo obrigatório para demandas que necessitem de planejamento de nova contratação, como orienta a IN 04/2014.

Entrada: Um artefato usado por um processo.

Estrutura Analítica do Projeto (EAP): é um processo de subdivisão das entregas e do trabalho do projeto em componentes menores e mais facilmente gerenciáveis. É estruturada em árvore exaustiva, hierárquica (de mais geral para mais específica) orientada às entregas, fases de ciclo de vida ou por sub-projetos (*deliverables*) que precisam ser feitas para completar um projeto.

Fase: O tempo entre dois marcos primários do projeto, durante o qual um conjunto bem definido de objetivos é atendido, artefatos são concluídos e decisões são tomadas sobre passar ou não para a próxima fase.

Ficha Detalhada do Sistema: Descrever as características técnicas gerais do sistema, como linguagem utilizada, equipe envolvida, endereços da aplicação nos diversos ambientes, tamanho funcional, endereço de versionamento, entre outros.

Fluxo de trabalho: A sequência de atividades realizadas em um negócio que produz um resultado de valor observável para um ator individual do negócio.

Framework: Uma microarquitetura que fornece um *template* extensível para aplicativos dentro de um determinado domínio.

Interface do usuário (UI): (1) O hardware ou software (ou ambos) que permite que um usuário interaja com um computador. (2) O termo "interface do usuário" geralmente se refere à apresentação visual e o software que a fundamenta, com os quais interage o usuário.

Interface gráfica de usuário (GUI): Um tipo de interface que permite a comunicação dos usuários



com o programa através da manipulação de recursos gráficos, em vez da digitação de comandos. Normalmente, a GUI inclui uma combinação de elementos gráficos, dispositivos apontadores, barras de menu e outros menus, janelas sobrepostas e ícones.

Lições Aprendidas: Registrar o aprendizado obtido durante a realização do projeto. Devem ser incluídas na base de conhecimento da organização para melhorar os futuros projetos.

Linguagem Unificada de Modelagem (UML): Uma linguagem para visualizar, especificar, construir e documentar os artefatos de um sistema intensivo de software.

Lista de Demandas Priorizadas pelo CTI: Lista de demandas de TI (sistemas, contratações, entre outras) deliberadas pelo Comitê de TI como prioritárias que visam atender as estratégias do Instituto.

Localizador de recurso uniforme (URL) : Um identificador padrão de um recurso na World Wide Web, usado por navegadores da Web para iniciar uma conexão. O URL inclui o protocolo de comunicação a ser usado, o nome do servidor e informações sobre caminho que identificam os objetos a serem recuperados no servidor.

MGP - Análise de Viabilidade do Projeto: Analisar uma nova demanda por serviço ou produto, apresentando subsídios técnicos e financeiros para avaliar sua viabilidade de modo a embasar a decisão pela continuidade da demanda como um novo projeto.

Minuta de Termo Aditivo do Contrato: Proposta de texto para adequação do contrato, podendo ser usado para efetuar acréscimos ou supressões no objeto (alterações quantitativas do objeto), prorrogações, além de outras modificações admitidas em lei.

Papel (Role): define o comportamento e as responsabilidades de um indivíduo ou de um conjunto de indivíduos que trabalham juntos como uma equipe, no contexto de uma organização de engenharia de software. Os papéis não são pessoas; pelo contrário, eles descrevem como as pessoas se comportam no negócio e quais são as responsabilidades que elas têm.

Plano Diretor de Tecnologia da Informação (PDTI): Instrumento de diagnóstico, planejamento e gestão dos recursos e processos de TI. Visa atender às necessidades tecnológicas e de informação do Ibama para um determinado período, norteador, assim, as ações de TI do Instituto.

Portfólio de Projetos: Repositório que identifica e fornece descrição básica dos projetos de TI existentes no Ibama. Deve estar alinhado ao PDTI e indicar o Gerente de Projetos responsável pelo projeto. Caso o projeto seja de desenvolvimento ou manutenção de sistemas, o projeto deve estar alinhado ao Catálogo de Sistemas.

Processo de desenvolvimento: Um conjunto de passos ordenados parcialmente e executados para uma determinada finalidade durante o desenvolvimento de software, como modelos de construção ou de implementação.

Projeto: Projetos são realizados por pessoas, restringidos por recursos limitados, planejados, executados e controlados. Um projeto é um esforço temporário empreendido para criar um serviço



ou produto exclusivo. Temporário significa que todo projeto tem começo e fim definidos. Exclusivo significa que o produto ou serviço é de alguma forma diferente de todos os produtos e serviços semelhantes. Em geral, os projetos são componentes críticos da estratégia de negócios de organizações executoras.

Release: Um subconjunto do produto final que é o objeto de avaliação em um marco principal. Um *release* é uma versão estável e executável do produto, que vem acompanhada dos artefatos necessários para sua utilização (como notas de *release* ou instruções de instalação, por exemplo). Um *release* pode ser interno ou externo. Um *release* interno é usado apenas pela organização de desenvolvimento, como parte de um marco, ou para fazer uma demonstração para usuários ou clientes. Um *release* externo é liberado para os usuários finais. Um *release* não é necessariamente um produto completo, mas pode ser apenas uma fase ao longo do caminho, com sua utilidade avaliada apenas do ponto de vista da engenharia. Uma das funções dos *releases* é forçar a equipe de desenvolvimento a fazer fechamentos em intervalos regulares, evitando a síndrome do "90% pronto, 90% faltando".

Requisito de software: Uma especificação de um comportamento do sistema que pode ser observado externamente; por exemplo, entradas e saídas do sistema, funções e atributos do sistema ou atributos do ambiente do sistema.

Requisitos funcionais: Os requisitos funcionais são aqueles que fazem parte do sistema, como um relatório específico, um campo a mais em um cadastro, etc. Eles normalmente têm finalidade de agregar valor ao usuário ou facilitar o trabalho que ele desenvolve.

Requisitos não-funcionais: Requisitos não-funcionais são aqueles relacionados ao ambiente onde o sistema está inserido. Um servidor mais robusto, um *firewall*, ou um usuário especializado em determinado procedimento pode ser visto como requisitos não funcionais.

Tarefa: É uma unidade de trabalho que um indivíduo, desempenhando o papel descrito, pode ser chamado a realizar. A tarefa tem uma finalidade clara, normalmente expressa em termos da criação ou atualização de alguns artefatos como um modelo, uma classe, um plano. Toda tarefa é atribuída a um papel específico. Em geral, a granularidade de uma tarefa é de duração de algumas horas a alguns dias e, em geral, envolve um papel e afeta um ou alguns artefatos.

Template: Uma estrutura predefinida de um artefato / documento.

Termo de Abertura do Projeto: Formalizar o início do projeto, apresentando seu nome, seus objetivos, a equipe básica e as informações necessárias para iniciar o planejamento.

Termo de Encerramento do Projeto: Formaliza o encerramento do projeto. Verifica se todos os serviços ou produtos do projeto foram recebidos definitivamente, de acordo com os requisitos, registrando o nível de sucesso do projeto.

Teste Baseado em Erros: Consiste em incluir propositalmente algum erro no programa e observar o comportamento do programa com erro, comparando-o com o comportamento do programa



original.

Teste Caixa Preta: Teste Funcional usado para demonstrar que as funções do Sistema estão operacionais, a entrada está adequadamente aceita, a saída está corretamente produzida e a integridade das informações externas é mantida. Atividade complementar aos testes de caixa branca, com a finalidade de descobrir tipos/classes de erros. Procura descobrir erro em: funções incorretas ou ausentes, erros de interface, erros nas estruturas de dados, erros em acesso a bancos de dados externos, erros de desempenho, erro de inicialização e término.

Teste de Aceitação: A validação é bem-sucedida quando: o Sistema funciona de uma maneira razoavelmente esperada pelo cliente, são atendidas as Expectativas dos clientes e a documentação é usada efetivamente.

Teste de Regressão: Teste necessário para assegurar que modificações no programa não causaram novos erros baseado em arquivo de 'log'.

Teste de Sistema: É utilizado para comparar o sistema com seus objetivos originais. Enfatiza a análise do comportamento da estrutura hierárquica de chamadas de módulos. fase mais complexa, devido à quantidade de informações envolvidas.

Teste de Unidade: Deve ser escrito pelo mesmo programador que desenvolveu o código a ser testado ou pelo analista de teste. Serve como documentação complementar do sistema e é essencial para análise de desempenho.

Teste Estrutural (Caixa Branca): São testes completos que verificam todos os caminhos lógicos de componentes ou programas, fornecendo casos de teste que põem a prova conjuntos específicos de condições e/ou garantem que todos os caminhos independentes, dentro de um módulo, tenham sido exercitados pelo menos uma vez. Executa todas as decisões lógicas para valores falsos ou verdadeiros. Executa todos os laços em suas fronteiras. Exercita as estruturas de dados internas.

10. GUIAS TÉCNICOS

- Metodologia de Gerenciamento de Projetos do Ibama (MGP-Ibama);
- Processo de Gestão de Riscos de Segurança da Informação;
- Política de Gerenciamento de Configuração de Ativos de Tecnologia da Informação;
- Processo de Gerenciamento de Configuração de Ativos de Tecnologia da Informação;
- Processo de Gerenciamento de Mudanças;
- Padrão de Ambiente de Construção de Sistemas de Informação;
- Padrão de Codificação de Sistemas de Informação;
- Padrão de Objetos e Estrutura de Banco de Dados do Ibama;
- Padrão de Interface e Usabilidade de Sistemas de Informação;
- Padrões de Governo - Padrões de Interoperabilidade de Governo Eletrônico (ePing), Modelo de Acessibilidade em Governo Eletrônico (eMag), Padrões Web em Governo Eletrônico



(ePWG), disponíveis em www.governoeletronico.gov.br, Modelo de Requisitos para Sistemas Informatizados de Gestão Arquivística de Documentos (eArq Brasil), disponível em www.conarq.arquivonacional.gov.br, entre outros;

- Política de Segurança da Informação e Informática do Ibama (POSIC) e suas normas complementares;
- Manual de Redação da Presidência da República.

11. REFERÊNCIA BIBLIOGRÁFICA

- RUP – Rational Unified Process®, Rational Software Corporation, <http://www.wthreex.com/rup/portugues/index.htm>;
- NBR ISO/IEC 12207 - Tecnologia de informação - Processos de ciclo de vida de software; ABNT Associação Brasileira de Normas Técnicas; Outubro, 1998;
- ISO/IEC 25010:2011-Systems and software engineering -- Systems and software Quality Requirements and Evaluation (SQuaRE) -- System and software quality models;
- PMBOK Guide - A Guide to the Project Management Body of Knowledge. Fifth Edition.ed. Project Management Institute;
- Glossário Padrão de Termos Utilizados em Teste de Software. Versão 1.3br (01 de Julho de 2007). Produzido pelo “Glossary Working Party” - International Software Testing Qualification Board;
- Object Oriented Analysis and Design - OOA & OOD de G. Booch, 1994;
- SLTI/MPOG. Instrução Normativa N° 04/2014. Secretaria de Logística e Tecnologia da Informação, Ministério do Planejamento, Orçamento e Gestão. Dispõe sobre o processo de contratação de serviços de Tecnologia da Informação pela Administração Pública Federal direta, autárquica e fundacional. Brasília, 2014;
- PRESSMAN, Roger S. Engenharia de software. 6ª ed. Rio de Janeiro: McGraw-Hill, 2006;
- SOMMERVILLE, Ian. Engenharia de software. 8ª ed. São Paulo: Addison-Wesley, 2007;
- BOOCH, Grady; RUMBAUGH, James; JACOBSON, Ivar. UML Guia do Usuário.2ª ed. Rio de Janeiro: Campus, 2006;
- BARTIÉ, A.. Garantia da Qualidade de Software. Elsevier Editora. Rio de Janeiro. 2002;
- Metodologia de Desenvolvimento de Sistemas - Instituto Brasileiro do Meio Ambiente e dos Recursos Naturais Renováveis (Ibama), Boletim de Serviço Especial nº 06 de 31.05.2011.
- Metodologia de Desenvolvimento de Sistemas - Poder Judiciário - Justiça do Trabalho -



Ministério do Meio Ambiente
Instituto Brasileiro do Meio Ambiente e dos Recursos Naturais Renováveis
Diretoria de Planejamento, Administração e Logística
Centro Nacional de Telemática

Tribunal Regional do Trabalho da 11ª Região - Secretaria de Tecnologia da Informação (SETI), 2010;

- Metodologia de Gestão e Desenvolvimento de Software - Superintendência de Seguros Privados - Coordenação Geral de Tecnologia da Informação, versão 1,0, 2011;
- Processo de Software para o SISP (PSW-SISP), versão 1.0, 2012.

ANEXO U DO TERMO DE REFERÊNCIA

**PADRÃO DOS AMBIENTES DE CONSTRUÇÃO DE
SISTEMAS DE INFORMAÇÃO**

Versão 1.0



Ministério do Meio Ambiente
Instituto Brasileiro do Meio Ambiente e dos Recursos Naturais Renováveis
Diretoria de Planejamento, Administração e Logística
Centro Nacional de Telemática

PADRÃO DOS AMBIENTES DE CONSTRUÇÃO DE SISTEMAS DE INFORMAÇÃO

Versão 1.0

- 2013 -



Sumário

1. Introdução.....	2
2. Ambiente de Desenvolvimento.....	3
2.1. Criação e manutenção do projeto de sistema.....	3
2.1.1. Estrutura de Diretórios/Pastas para Codificação.....	5
2.1.2. Estrutura de Diretórios/Pastas para Documentação do sistema.....	5
2.2. Versionamento de codificação e documentação.....	5
3. Ambiente de Teste.....	6
4. Ambiente de Homologação.....	7
5. Ambiente de Treinamento	8
6. Ambiente de Produção	9



1. Introdução

Um ambiente é um conjunto de hardware e software, com uma configuração específica, estabelecida para um propósito pré-definido. Com o intuito de documentar e uniformizar os procedimentos no âmbito do Ibama, este documento visa a padronização dos ambientes de desenvolvimento, teste, homologação, treinamento e produção para a construção de sistemas de informação. Para cada um dos ambientes, a plataforma a ser utilizada e os recursos de hardware e software precisam ser registrados, assim como os procedimentos específicos de integração e uniformidade entre os ambientes, cabe ressaltar que a versão dos aplicativos dos servidores de aplicação e base de dados devem ser iguais em todos os ambientes, afim de garantir homogeneidade, integridade e integração entre os ambientes.

A integração tem sido considerada o elemento central para prover um suporte efetivo ao desenvolvimento de software, pois é ela que define todas as regras e diretrizes que governarão o uso das ferramentas. A integração é responsável pela combinação de ferramentas de forma que elas trabalhem em harmonia em todas as etapas do processo. Existem vários níveis de integração de ferramentas, diferindo no suporte fornecido. Dentre eles, pode-se citar:

- **Integração de Dados:** A chave para integrar ferramentas é a habilidade de compartilhar informação de projeto.
- **Integração de Apresentação:** A integração de apresentação tem como objetivo tornar as interfaces do ambiente de construção de sistemas consistentes, permitindo que o usuário utilize as ferramentas, alternando de uma para a outra, sem sofrer um impacto na interface. Os componentes utilizados são os mesmos, com as mesmas funcionalidades, não havendo a necessidade de aprender como utilizar cada ferramenta.
- **Integração de Controle:** Para integrar um conjunto de ferramentas é preciso ter um foco forte no gerenciamento do processo. Engenheiros de software têm se tornado mais orientados a processos, usando métodos, técnicas e ferramentas para controlar e guiar seu trabalho.
- **Integração de Conhecimento:** Com o aumento da complexidade dos processos de software faz-se necessário considerar informações de natureza semântica, sem as quais a integração não é plenamente obtida. O conhecimento, assim como os dados, deve estar disponível e representado de forma única no ambiente, de forma a poder ser acessado por todas as



ferramentas que dele necessitarem, evitando redundância e inconsistências

2. Ambiente de Desenvolvimento

Será o ambiente a ser utilizado pela equipe de desenvolvimento de software, com o objetivo de implementar a solução. No desenvolvimento dos sistemas deverão ser seguidos os padrões de codificação, interface, nomenclatura de banco de dados e a metodologia de desenvolvimento de sistemas do Ibama (MDS). Assim como, estar em conformidade com a política de segurança e o gerenciamento de mudanças do Ibama.

2.1. Criação e manutenção do projeto de sistema

O projeto de sistema será criado no ambiente de desenvolvimento seguindo o padrão: <http://tucunare.ibama.gov.br/desenv/siglasistema>, onde:

- desenv = Indicação do nome do ambiente de desenvolvimento.
- siglasistema = Nome (sigla) do sistema em letras minúsculas.

Por exemplo: <http://tucunare.ibama.gov.br/desenv/docibama>

A base de dados a ser utilizado para o ambiente de desenvolvimento será uma base com as tabelas corporativas e as tabelas específicas do sistema, com os usuários com senha e perfil de desenvolvimento. A nomenclatura para o nome do esquema do banco de dados do projeto será: `bd_siglasistema`, onde:

- bd = Indicação que se trata de um base de banco de dados.
- siglasistema = Nome (sigla) do sistema em letras minúsculas.

Somente os desenvolvedores que irão trabalhar no projeto de sistema serão habilitados para realização de chek-out e check-in dos códigos fontes do projeto. Enquanto o desenvolvedor estiver codificando o código do módulo (funcionalidade), este deverá utilizar o ambiente local de desenvolvimento na sua estação de trabalho: <http://localhost/siglasistema>

onde:

- localhost = Computador local do desenvolvedor.
- siglasistema = Nome (sigla) do sistema em letras minúsculas.

Por exemplo: <http://10.5.2.1/~1111111111/docibama>

Somente após os testes iniciais e verificação de consistência da aplicação, que deverá ser



realizado o chek-out do módulo (funcionalidade) para o ambiente de desenvolvimento. No caso específico do ambiente de desenvolvimento, é interessante identificar que o desenvolvedor será o responsável pela geração dos releases e pela execução de testes básicos de integração. Estes testes são muito mais simples do que os testes a serem executados no ambiente de testes. O objetivo é assegurar que o release a ser encaminhado ao ambiente de testes está funcionando minimamente, isto é, assegurar que a instalação, a configuração inicial e algumas funcionalidades básicas se apresentam sem problemas.

Este repositório e sua infra-estrutura completa, são consequência direta das características das ferramentas de desenvolvimento utilizadas e dos processos de desenvolvimento adotados. Segue na tabela abaixo, a configuração para este ambiente de desenvolvimento no Ibama:

Servidor de Aplicação – Ambiente de Desenvolvimento			
Endereço	Software	Finalidade	Usuário
http://tucunare.ibama.gov.br/desenv/siglasistema	Eclipse, CVS	- Desenvolvimento do sistema - Geração de releases	Equipe de Desenvolvimento
http://localhost/siglasistema	Eclipse, CVS	- Codificação modularizada - Geração de releases	Equipe de Desenvolvimento
http://tucunare.ibama.gov.br/documentacao/siglasistema	CVS, Editor de Texto	- Consulta à documentação do sistema - Documentação atualizada	Desenvolvedores, Analista de sistemas e Gerente do Projeto

Base de Dados - Ambiente de Desenvolvimento			
Base	Software	Finalidade	Usuário
bd_siglasistema Base nova, exceto para o esquema corporativos (acesso-segurança, estados, municípios, espécies...) Para a tabela de pessoas (usuários) se houver, realizar a cópia do ambiente de produção e alterar todas as senhas para teste123.	SQL Developer	- Manutenção da base de dados do projeto	Equipe de Desenvolvimento
	DB Designer	- Elaboração e validação do modelo lógico de dados do projeto	Analista de Sistema e Analista de Banco de dados



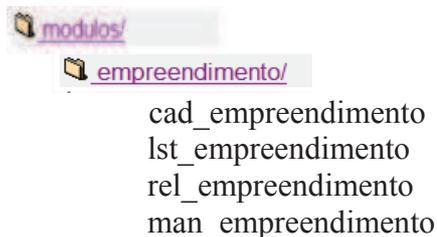
2.1.1. Estrutura de Diretórios/Pastas para Codificação

O nome das pastas e dos arquivos deverão ser em letras minúsculas e não conter espaço ou acentos e os módulos devem estar no diretório `modulos/` podendo conter subdiretórios para uma melhor organização dos arquivos. Por exemplo a estrutura de um projeto ficaria assim:

siglasistema



Cada módulo da aplicação deverá ter um diretório de mesmo nome sem a extensão, onde deverá estar o código fonte do módulo.

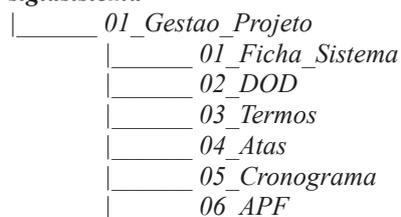


onde os prefixos correspondem:

- cad = cadastro (incluir)
- lst = lista
- rel = relatório
- man = manter (alterar e excluir)

2.1.2. Estrutura de Diretórios/Pastas para Documentação do sistema

siglasistema





|_____ 02_Mapeamento_Processo

|_____ 03_Viabilidade_Requisito

|_____ 04_Analise_Designer

|_____ 01_Diagrama_Caso_de_Uso

|_____ 02_Casos_de_Uso

|_____ 03_MER

|_____ 04_Diagrama_Sequencia

|_____ 05_Homologacao_Implantacao

|_____ 01_Manuais

|_____ 02_Testes

|_____ 03_Treinamento

|_____ 06_Documento_Apoio

2.2. Versionamento de codificação e documentação

Será utilizado um sistema de versionamento para o controle de códigos fonte e versionamento de releases pelos desenvolvedores, assim como, para a documentação do projeto pelos analistas de sistemas.

Para cada release de codificação a ser gerada para os ambientes, deve seguir o seguinte padrão: `release_siglasistema_xx.yy_data_hora`, onde:

- siglasistema = Nome (sigla) do sistema em letra minúscula.
- xx.yy = xx (versão do sistema) . yy (número da alteração)
- data= Data no padrão americano que a release foi gerada
- hora= Será o horário que a release foi gerada

Por exemplo: `release_docibama_01.17_20130820_1015`

Para a release de documentação manter o nome do arquivo original, fazer o versionamento manual da atualização no documento e utilizar o versionamento somente como repositório.

3. Ambiente de Teste

Este é o ambiente utilizado pela equipe de testes, com o objetivo de identificar defeitos na solução, a partir dos *releases* gerados. Neste ambiente será aplicado o plano de teste do projeto com



os seus casos de testes. Para isto, é importante criar uma infra-estrutura de testes integrados na qual versões estáveis dos componentes ou rotinas comuns são instaladas. Neste ambiente também devem ser instaladas versões estáveis dos produtos em desenvolvimento, possibilitando a análise de interação entre as diferentes versões.

Este ambiente é também importante para testes integrados quando uma versão de rotina ou de componente compartilhado é atualizada e é necessário ter uma idéia do impacto desta alteração nos produtos dela dependentes. Também pode ser utilizado para os primeiros testes de novas versões de componente adquiridos de terceiros e que façam parte da solução que está sendo desenvolvida.

Este ambiente de teste deve ser acessível a todas as estações de trabalho dos desenvolvedores, mas sua gestão deve estar sob o domínio de um time (composto por membros da equipe de Projetos e Arquitetura) dedicado a seu controle, pois as versões de sistemas nela instalados devem ser conhecidas e estáveis. Este ambiente serve como confirmação ou aperfeiçoamento dos processos da análise de impacto necessária na gerência de mudanças.

Recomenda-se que neste ambiente de testes integrados os desenvolvedores não façam uso de usuários privilegiados para seus testes, assim aproximando os resultados dos testes com aqueles que serão observados pelos usuários finais quando da instalação do produto em homologação e produção.

Segue, na tabela abaixo, a configuração para este ambiente.

SERVIDOR DE APLICAÇÃO – Ambiente de teste			
Endereço	Software	Finalidade	Usuário
http://tucunare.ibama.gov.br/teste/siglasistema	Selenium	- Realização de testes funcionais e de casos de testes - Realização de testes gerais no projeto	Equipe de Teste

BASE DE DADOS – Ambiente de teste			
Base	Software	Finalidade	Usuário
Base nova, exceto para o esquema corporativos (acesso-segurança, estados, municípios, espécies...) Para a tabela de pessoas (usuários) se houver, então deverá	SQL Developer	- Consultar a base de dados do projeto	Equipe de Desenvolvimento



ser realizado a cópia do ambiente de produção e alterar todas as senhas para teste123.			
--	--	--	--

Após a realização de todos os testes e suas correções, deverá ser gerado uma release no formato citados no item 1.2 para ser inserida no ambiente de homologação.

4. Ambiente de Homologação

Este ambiente simula o ambiente de produção do IBAMA. É utilizado para validar o produto antes de colocá-lo em real funcionamento.

O ambiente de homologação deve ser trabalhado e mantido para ser similar ao ambiente de produção. Seu objetivo é oferecer aos futuros usuários do sistema a possibilidade de testar as funcionalidades dos novos produtos de desenvolvimento e encontrar possíveis incorreções de resultado ou comportamento. Estes desvios devem ser relatados à equipe de desenvolvimento responsável pela solução que providenciará as alterações necessárias. Da mesma forma que este ambiente serve para a análise prévia das funcionalidades do novo sistema, ele também deve servir para que as equipes de desenvolvimento e suporte a rede e banco executem seus procedimentos de instalação e atualização antes que os produtos de desenvolvimento sejam implantados em produção.

É importante ressaltar os seguintes pontos:

- Ambientes de homologação não devem ser utilizados pelas equipes de desenvolvimento para testes.
- Desenvolvedores não devem possuir acesso privilegiado ao ambiente de homologação. Para evitar que os desenvolvedores alterem configurações ou efetuem cópias não autorizadas de versões de sistema ou massas de dados, os desenvolvedores não devem ter característica de acesso privilegiado ao ambiente de homologação. Isso permite com que o desenvolvedor, ao acessar sistemas em homologação, tenha a mesma experiência que o usuário final, podendo reproduzir eventuais problemas encontrados por ele.
- O ambiente de homologação será uma cópia do ambiente de produção mais a nova versão do produto ou projeto. Assim os testes do usuário serão muito mais eficientes.

Segue, na tabela abaixo, a configuração para este ambiente.



SERVIDOR DE APLICAÇÃO – Ambiente de homologação		
Endereço	Finalidade	Usuário
http://tucunare.ibama.gov.br/homolog/siglasistema	- Realização da homologação (testes) pelos usuários.	Gestor de Negócio, Requisitantes e Usuários
	- Executar os procedimentos de instalação e atualização antes que os produtos de desenvolvimento sejam implantados em produção.	Equipes de suporte a rede e banco de dados.

BASE DE DADOS – Ambiente de homologação			
Base	Software	Finalidade	Usuário
Cópia da base do ambiente de produção, atualizada mensalmente. Em casos excepcionais poderá ocorrer atualização específica para o projeto, caso seja solicitado pelo Gestor de negócio.	SQL Developer	- Consultar a base de dados do projeto	Equipe de banco de dados

Após a realização de todos os testes e suas correções, deverá ser gerado uma release no formato citados no item 1.2 para ser inserida no ambiente de treinamento se for o caso, e no ambiente de produção.

5. Ambiente de Treinamento

Este é o ambiente que não obrigatório para todos os projetos de sistemas, será criado a medida que for solicitado pelo gestor do sistema ou gerente do projeto, quando for identificado que não é possível realizar o treinamento no ambiente de homologação.

O ambiente de treinamento deverá ser um cópia do ambiente de homologação com a última versão de release concluída para o sistema, sendo a versão que será implantada em ambiente de produção. Após conclusão do treinamento o ambiente deverá ser excluído.

Segue, na tabela abaixo, a configuração para este ambiente.

SERVIDOR DE APLICAÇÃO – Ambiente de treinamento		
Endereço	Finalidade	Usuário
http://tucunare.ibama.gov.br/treina/siglasistema	- Realização de treinamentos para capacitação dos usuários na utilização do sistema.	Usuários do sistema



BASE DE DADOS – Ambiente de treinamento			
Base	Software	Finalidade	Usuário
Cópia da base do ambiente de homologação da release solicitada.	SQL Developer	- Consultar a base de dados do projeto	Equipe de banco de dados

6. Ambiente de Produção

Este é o ambiente no qual os sistemas do Ibama estão operacionais e funcionais, em que o produto é efetivamente colocado em funcionamento. Os recursos deste ambiente deverão estar alinhados com os recursos da homologação, tendo como adendo o hardware e software complementar (firewall, licenças, etc). Atualmente o Ibama utiliza a hospedagem em um datacenter externo, no qual seguindo o indicado no plano de implantação do projeto e das releases, são abertos chamados pela equipe de rede e banco de dados para implantação de releases em ambiente de produção.

SERVIDOR DE APLICAÇÃO – Ambiente de produção		
Endereço	Finalidade	Usuário
Internos: http://ibamanet.ibama.gov.br/siglasistema Externos: http://www.ibama.gov.br/siglasistema	- Utilização e operacionalização do sistema.	Usuários do sistema

BASE DE DADOS – Ambiente de produção			
Base	Software	Finalidade	Usuário
Base produção	Oracle, MySQL, Postgres	- Disponibilizar a base de dados do projeto	Equipe de banco de dados



Integração entre os ambientes de construção de sistemas de informação



ANEXO V DO TERMO DE REFERÊNCIA

**PADRÃO DE OBJETOS E ESTRUTURAS DE
BANCO DE DADOS**

Versão 1.0



Ministério do Meio Ambiente
Instituto Brasileiro do Meio Ambiente e dos Recursos Naturais Renováveis
Diretoria de Planejamento, Administração e Logística
Centro Nacional de Telemática

PADRÃO DE OBJETOS E ESTRUTURAS DE BANCO DE DADOS

Versão 1.0

- 2014 -



1. Introdução

A definição de padrões para criação de objetos e estruturas de banco no âmbito do Ibama faz parte da iniciativa de padronização da área de tecnologia da informação, buscando garantir uma melhor organização das base de dados, contribuir de forma considerável para uma plena administração de dados, permitir o planejamento global afim de evitar aplicações isoladas e a proliferação de dados incompatíveis e redundantes, obter maior disponibilidade e facilidade de interpretação, aumentar a utilização compartilhada e conseqüentemente a diminuição do seu custo, contribuir para a rapidez na resolução de problemas, facilidade de entendimento e conhecimento dos aplicativos e finalmente a melhoria da qualidade dos produtos e serviços prestados.

Este documento estabelece nomeclaturas, regras e procedimentos que devem ser seguidos para a criação de objetos no banco de dados, objetivando a padronização das definições das estruturas físicas e lógicas relacionadas com o armazenamento dos dados.

Entenda-se aqui por definição tanto a questão da nomenclatura quanto a questão de como devem ser constituídos os objetos de banco de dados e estruturas de armazenamento, abrangendo, inclusive, regras sobre os scripts de criação e manipulação de dados.

Todos os servidores ou colaboradores deverão seguir as regras desta norma quanto se tratar de bases de dados dos sistema de informação desenvolvido ou mantido no âmbito do Ibama.



2. Nomeclatura

2.1. TABLE

O nome da tabela, deverá expressar o que será armazenado em seu conteúdo, para facilitar a leitura e entendimento do modelo de dados (MER) e também a criação dos outros objetos que a utilizarem. Deve ser prefixado com TB_, não ultrapassar 30 caracteres, em caixa alta e não deve conter acentos ou plural.

Exemplo: TB_PESSOA, TB_FUNCIONARIO, TB_USUARIO.

2.2. COLUMN

O nome de uma coluna de uma tabela deve ser alusivo a sua finalidade dentro da tabela; no caso de um nome muito extenso, deve-se abreviá-lo mantendo, no entanto, a finalidade da Coluna. Deve, obrigatoriamente, começar com um grupo de três letras indicando o tipo de dado mais específico da Coluna, conforme tabela a seguir:

Descrição	Prefixo	Exemplo	Tipo de dado utilizado
Código	COD	COD_MUNICIPIO	Number/Varchar2
Data	DAT	DAT_NASCIMENTO	Date
Descrição	DES	DES_ENDERECO	Varchar2
Hora	HOR	HOR_CADASTRO	Timestamp
Índice, Indicador	IND	IND_ESTRUTURAL	Varchar2
Campos binários	LOB	LOB_FOTO	Blob
Nome	NOM	NOM_PESSOA	Varchar2
Número	NUM	NUM_CPF	Number/Varchar2
Quantidade	QTD	QTD_PARCELA	Number
Seqüência	SEQ	SEQ_UNIDADE	Number
Sigla	SIG	SIG_ORGAO	Varchar2
Situação	SIT	SIT_CANELADO	Char
Tipo	TIP	TIP_REGISTRO	Varchar2
Dinheiro	VAL	VAL_SALARIO	Number
Texto Longo	TXT	TXT_COMENTARIO	Clob



2.3. OWNER

A denominação do owner de um schema contendo todos os objetos de banco de dados diretamente relacionados a um sistema ou aplicação deve ser igual à sigla pela qual este mesmo sistema ou aplicação é conhecido.

Exemplo: para o esquema SISREG o owner será SISREG.

3. Utilização dos Tipos de Dados

3.1. Varchar2 - Utilizado para dados que representam cadeias de caracteres com um tamanho variável mas com possibilidade de se estimar o tamanho máximo. Cadeias de caracteres deste tipo podem variar de tamanho mas sem ultrapassar o tamanho máximo.

3.2. Char - Utilizado para dados que representam cadeias de caracteres com tamanho fixo e invariável, ou seja, todas as cadeias de caracteres representadas por este tipo possuem um tamanho único.

3.3. Number - Utilizado para dados que representam números de uma forma genérica.

3.4. Date - Utilizado para dados que representam datas cronológicas informando obrigatoriamente dia, mês e ano. O formato a ser utilizado será o padrão brasileiro dd/mm/aaaa.

3.5. Clob - Utilizado para dados que representam cadeias de caracteres com um tamanho variável e sem possibilidade de se estimar o tamanho máximo. Cadeias de caracteres deste tipo constituem textos que podem ser longos ou curtos e devem ser utilizados quando o conteúdo puder exceder 4.000 caracteres, caso contrário utilizar o tipo varchar2.

3.6. Blob - Utilizado para dados que representam imagens estáticas, vídeos, documentos etc, ou seja, qualquer conteúdo que deva ser armazenado no formato binário.



3.7. Timestamp – utilizado para dados que representam data e hora. O formato a ser utilizado será o padrão brasileiro dd/mm/aaaa hh:mm:ss

4. Scripts

Os scripts para criação de Tabelas devem ter as Colunas ordenadas segundo os critérios a seguir :

a) Primeiro devem vir as Colunas cujo preenchimento é obrigatório (NOT NULL) na seguinte ordem:

- 1) Chave Primária
- 2) Chave(s) Estrangeira(s)
- 3) Demais colunas obrigatórias

b) Segundo devem vir as Colunas cujo preenchimento não é obrigatório (NULL) mas que tem maior probabilidade de serem preenchidas futuramente.

c) Terceiro devem vir as Colunas cujo preenchimento não é obrigatório (NULL) e que tem maior probabilidade de não serem preenchidas.

5. TRIGGER

Para a criação do nome de um trigger, deve ser utilizado o prefixo TG_ + nome da tabela + tipo de gatilho (INC, ALT ou EXC) caso a tabela tenha mais de uma trigger, seguindo a mesma regra de abreviação utilizada para a criação do nome da tabela quando este ultrapassar o limite de 30 caracteres.

Ex: TG_MUNICIPIO_INC, TG_UF_INC, TG_UF_ALT



6. SEQUENCE

Para a criação do nome de uma *sequence*, deve ser utilizado o prefixo SQ_ + nome da tabela, seguindo a mesma regra de abreviação utilizada para a criação do nome da tabela quando este ultrapassar o limite de 30 caracteres.

Ex: SG_MUNICIPIO, SQ_UF, SQ_TIPO.

7. VIEW

Para a criação do nome de uma visão, deve ser utilizado o prefixo VW_ + nome da tabela ou nome curto da tabela + complemento se for necessário, seguindo a mesma regra de abreviação utilizada para a criação do nome da tabela quando este ultrapassar o limite de 30 caracteres.

O nome curto da tabela pode ser criado obedecendo qualquer critério, desde que seja único entre todas as tabelas do esquema

Ex: VW_MUNICIPIO, VW_MUN_REGIAO, VW_UF, VW_PESSOA_ENDERECO.

8. MATERIALIZED VIEW

Para a criação do nome de uma visão, deve ser utilizado o prefixo MV_ + nome da tabela ou nome curto da tabela + complemento se for necessário, seguindo a mesma regra de abreviação utilizada para a criação do nome da tabela quando este ultrapassar o limite de 30 caracteres.

Ex: MV_MUNICIPIO, MV_MUN_REGIAO, MV_UF, MV_PESSOA_ENDERECO.

9. PACKAGE

A nome de um pacote deve ser o mesmo do módulo do sistema ao qual ele será referenciado adicionado prefixo PKG_, e seguindo a mesma regra de abreviação utilizada para a criação do nome da tabela quando este ultrapassar o limite de 30 caracteres.



Esta metodologia visa facilitar a manutenção dos sistemas, bem como a implementação de critérios de segurança de acesso e manipulação das informações necessárias.

Quando um módulo necessitar utilizar uma função de um outro módulo do mesmo esquema, então esta função deverá ser recriada no pacote base do esquema, chamada PKG_BASE, e executada a partir dos pacotes de cada módulo.

Também será permitida a criação de pacotes específicos para agrupamento de funções de propósito comum a todo o sistema, estilo biblioteca de funções, neste caso o nome do pacote deverá refletir o objetivo de suas funções, como por exemplo, um pacote com funções de cálculos, deverá chamar ser criado com nome PKG_CALCULO.

Para troca de informações entre os sistemas, nunca, um esquema deve acessar diretamente dados de um outro esquema, para isso deve ser criado o pacote público, chamado PKG_PUBLICO, que servirá de interface de comunicação com os demais esquemas.

Como exemplo de nomenclatura, podemos citar o pacote utilizado para o módulo de cadastramento de material, o nome do módulo será cad_material.php e o seu pacote correspondente chamaria PKG_CAD_MATERIAL.

Os nomes das funções criadas dentro dos pacotes também deverão seguir os seguintes padrões:

a) funções para selecionar informações devem iniciar com o prefixo “**sel_**”;

Exemplo: sel_material, sel_material_disponivel, sel_usuario.

b) funções para inclusão de informações devem iniciar com o prefixo “**inc_**”;

Exemplo: inc_material, inc_status, inc_cliente

c) funções para exclusão de dados devem iniciar com o prefixo “**exc_**”

Exemplo: exc_material, exc_cliente

d) funções que podem tanto incluir quanto alterar devem iniciar com o prefixo “**inc_alt_**”.

Exemplo: inc_alt_material, inc_alt_cliente



10. ÍNDICE

Uma tabela pode ter um ou mais índices, sendo o nome do índice formado pelo nome curto da tabela + nome da coluna(s) + “_I”, seguindo a mesma regra de abreviação utilizada para a criação do nome da tabela, quando este ultrapassar o limite de 30 caracteres. O nome curto da tabela pode ser criado obedecendo qualquer critério, desde que seja único entre todas as tabelas do esquema.

Para a criação de um índice na tabela de município para a coluna NOM_MUNICIPIO, poderíamos utilizar o nome curto MNCP (retirado as vogais da palavra MUNICIPIO) e criar o nome do índice como: MNCP_NOM_MUNICIPIO_I

Para os indexes compostos, a nomenclatura deve seguir o seguinte padrão:

Nome ou nome curto da tabela + Nome ou nome curto da Coluna 1 + “_” + Nome ou Nome curto da coluna 2 + “_I”

Exemplo: FUNC_NOM_FUNC_DAT_NASC_FUNC_I (índice referente às colunas nom_funcionario e dat_nascimento do funcionário)

11. FOREIGN KEY

Uma tabela pode ter uma ou mais chaves estrangeiras, sendo o nome da chave estrangeira formada pelo nome curto da tabela + nome da coluna + ‘_FK’, seguindo a mesma regra de abreviação utilizada para a criação do nome da tabela quando este ultrapassar o limite de 30 caracteres

O nome curto da tabela pode ser criado obedecendo qualquer critério, desde que seja único entre todas as tabelas do esquema.

Para a criação de uma chave estrangeira na tabela de município para a coluna cod_uf,



poderíamos utilizar o nome curto MUN e criar o nome da chave estrangeira como: FK_MUN_COD_UF.

Índices de chaves estrangeiras - o nome do índice referente à chave estrangeira deve ser formado pelo nome ou nome curto da tabela de referência + nome da coluna estrangeira + ‘_FK_I’

12. UNIQUE KEY

Uma tabela pode ter uma ou mais chaves únicas, sendo que o nome da chave única deve ser formada pelo nome curto da tabela + nome da(s) coluna(s) + ‘_UK’, seguindo a mesma regra de abreviação utilizada para a criação do nome da tabela quando este ultrapassar o limite de 30 caracteres

O nome curto da tabela pode ser criado obedecendo qualquer critério, desde que seja único entre todas as tabelas do esquema.

Para a criação de uma chave única na tabela de município para as colunas cod_uf e cod_municipio, poderíamos utilizar o nome curto MUN e criar o nome da chave estrangeira como: MUN_COD_UF_COD_MUN_UK

13. RULE

O nome da RULE deverá ser em maiúsculo, no singular, sem caracteres especiais, de acordo com o seguinte padrão: RUL_ + Nome significativo para a RULE

Ex.: RUL_SEXO, RUL_SIM_NAO



14. TYPE

O nome de um Type deverá ser em maiúsculo, no singular, sem caracteres especiais, de acordo com o seguinte padrão: TP_ + nome do Type

Ex: TP_FUNCIONARIO

15. SINÔNIMO

O nome de um Sinônimo deve ser exatamente igual ao nome dado ao objeto do banco de dados sem a parte do Owner do Schema. Apenas a Equipe de DBAs deve poder criar, alterar ou excluir Sinônimos.

16. PROCEDURE

O nome de uma Procedure deverá ser em maiúsculo, no singular, sem caracteres especiais, de acordo com o seguinte padrão: P_ + nome da procedure

Ex: P_CALCULAR_JURO

17. FUNCTION

O nome de uma Function deverá ser em maiúsculo, no singular, sem caracteres especiais, de acordo com o seguinte padrão: F_ + nome da function

Ex: F_CONVERTER_TEMPERATURA



18. PARÂMETROS DE ENTRADA DAS FUNÇÕES

Os nomes dos parâmetros utilizados na criação das procedures/functions devem ser em minúsculo, sem caracteres especiais, de acordo com o seguinte padrão:

p_ + nome do parâmetro

Sempre que possível deve ser utilizado o mesmo nome da coluna do banco de dados ao qual o parâmetro fizer referência, seja para consulta, alteração ou inclusão de dados.

Exemplo: p_cod_uf, p_num_pessoa

19. CONSTRAINTS TIPO CHECK

O nome de uma Constraint tipo Check, deverá ser em maiúsculo, no singular, sem caracteres especiais, de acordo com o seguinte padrão: CK_ + nome da constraint

EX: CK_DEPARTAMENTO_DEPTNO

20. TABLESPACES

O nome utilizado na criação da tablespace deve o mesmo nome utilizado para o schema em maiúsculo, sem caracteres especiais, de acordo com o seguinte padrão:

- Tablespace de Dados: igual ao nome do esquema

Ex: SISREG, SISTRUT

- Tablespace de Índices: adicionar “_I” ao nome do esquema.

Ex: SISREG_I, SISTRUT_

- Tablespace de Dados Binários (LOB): adicionar “_LOG” ao nome do esquema.

Ex: DOCIBAMA_LOB, SIGER_LOB, DOF_LOB



21. SCHEMA

O nome de um *schema* deve ser em maiúsculo, no singular, sem caracteres especiais, e, quando possível, igual a sigla do sistema ao qual pertencerá.

22. Glossário

Banco de Dados Corporativo: é um banco de dados relacional contendo objetos (tabelas, visões, procedimentos, etc.) que são compartilhados e usados por diversas aplicações ou sistemas ao mesmo tempo, evitando assim redundância de dados e garantindo integridade das informações, além de outras vantagens. Ele é composto por outros três bancos de dados independentes, cada qual constituído por uma instância e uma base de dados. São assim denominados : Banco de Dados de Desenvolvimento, Banco de Dados de Homologação e Banco de Dados de Produção.

Banco de Dados de Desenvolvimento : corresponde a uma instância e a uma base de dados com o objetivo de armazenar objetos de banco de dados e dados para desenvolvimento dos novos sistemas ou aplicações, sendo um banco de dados relacional utilizado unicamente pelas equipes de desenvolvimento, devidamente autorizadas. Todo novo sistema ou aplicação deve ser desenvolvido utilizando este banco de dados, antes de ser colocado em homologação.

Banco de Dados de Homologação : corresponde a uma instância e a uma base de dados com o objetivo de armazenar objetos de banco de dados e dados para testes dos novos sistemas ou aplicações, sendo um banco de dados relacional. É utilizado pelos usuários finais e as equipes de desenvolvimento, devidamente autorizadas. Todo novo sistema ou aplicação deve, antes de ser colocado em produção, ser colocado neste ambiente para testes e validação por parte do usuário final.

Banco de Dados de Produção : corresponde a uma instância e a uma base de dados com o objetivo de armazenar objetos de banco de dados e dados de sistemas ou aplicações já em produção, constituindo assim dados válidos e que são acessados por diversos usuários do banco de dados, devidamente autorizados. Também é um banco de dados relacional.

Column (coluna) : é uma parte de uma Tabela que contém dados do mesmo tipo e que, semanticamente, são de mesma natureza.

Constraint (regra de integridade) : correspondem a uma regra de integridade aplicada sobre uma determinada tabela, mantida no banco de dados. Em linhas gerais, constraints, são utilizadas para prevenir entrada inválida de dados, implementando regras a nível de tabela que são verificadas toda



vez que uma linha é inserida, alterada ou excluída, se a regra não for respeitada, a operação não é concluída no banco de dados.

Datafile (arquivo de dados): é uma estrutura física (um arquivo do sistema operacional) onde fisicamente o conteúdo de uma *tablespace* é armazenado. Uma *tablespace* é constituída por um ou mais *datafiles*.

DBA : O termo DBA é uma sigla de origem inglesa para *Database Administrator*. Como no jargão técnico, no comércio, em empresas, enfim, em tudo que se relaciona com administração de banco de dados no mundo (inclusive no Brasil) se utiliza o termo DBA, ao invés da tradução para a língua local, resolvemos adotá-lo para referenciar aquele que é responsável por gerenciar e administrar o banco de dados.

Equipe de DBAs : é um grupo formado por dois ou mais técnicos e/ou analistas responsáveis pela administração do Banco de Dados Corporativo, devidamente treinados para tal tarefa de administração. A necessidade de uma equipe deve-se ao fato de que a administração de um banco de dados não deve ficar centralizada em uma única pessoa.

Equipe de Desenvolvimento : é qualquer grupo de dois ou mais técnicos, analistas e/ou programadores responsáveis por definir, projetar, desenvolver e implantar sistemas, sejam de uso local e restrito ao IBAMA (intranet), seja de âmbito mundial (internet).

Function (função): possui a mesma definição de uma *procedure*, diferenciando apenas na obrigatoriedade de retornar um resultado no final de sua execução.

Index (índice): é um objeto de banco de dados pertencente a um *schema*, utilizado para acelerar o acesso e a recuperação de linhas em uma tabela através do uso de ponteiros, provendo assim um acesso direto e rápido às suas linhas.

Line (linha): corresponde a uma instância ou registro de uma Tabela.

Owner (proprietário): um *owner*, é um usuário do banco de dados, que tem poder total de ação sobre os objetos dos *schema* a que está associado, sem precisar de algum direito ou privilégio especial para isso, podendo executar comandos DML e DDL naturalmente.

Package (pacote): é um conjunto de procedimentos na linguagem PL/SQL, agrupados numa mesma estrutura armazenada no banco de dados. Um pacote tem duas partes : a especificação do pacote e o corpo do pacote.

Package Specification (especificação do pacote): também é conhecido como cabeçalho do pacote; e a parte do pacote que contém informações sobre o conteúdo do pacote (assinatura das funções), não contendo nenhum código PL/SQL .

Package Body (corpo do pacote): é a parte do Pacote que contém o código PL/SQL de todas as



funções definidas na Especificação do Pacote.

Procedure (procedimento): *Procedures* são trechos de código na linguagem PL/SQL que são separados do fluxo principal de execução e podem ser chamadas uma ou mais vezes, desviam o fluxo do processamento para o seu corpo, permitem manipular diretamente os dados do banco de dados e ao final de sua execução retornam para o lugar de onde foi chamada sem retornar resultado.

Role (papel/cargo): é um grupo de privilégios (de sistema e/ou sobre objetos do banco de dados) que podem ser concedidos a um ou mais usuários.

Rule (regra): é uma regra que restringe o conjunto de valores (domínio) que podem ser inseridos em campos do banco de dados.

Schema (esquema) : um *schema* na terminologia Oracle é um conjunto de vários objetos de bancos de dados que estão associados a um usuário específico do banco de dados (também chamado de *owner* ou proprietário).

Sequence (sequência) : é um objeto de banco de dados usado para gerar números únicos e seqüenciais. É geralmente utilizada para implementar chave primária em uma Tabela. Ela é gerada e incrementada (ou decrementada) internamente por uma rotina do próprio banco de dados Oracle.

Synonyms (sinônimo) : é um nome alternativo para um objeto do banco de dados.

Table (tabela): é a unidade básica de armazenamento de dados sendo composta por linhas e colunas.

Tablespace: corresponde a uma área lógica para armazenamento. Dentro de uma tablespace são armazenados objetos do banco de dados além dos próprios dados, inclusive os dados sobre o banco de dados (dicionário de dados).

Terceiros: toda e qualquer entidade (empresa, micro-empresa, conjunto de um ou mais técnicos em informática tais como analistas, programadores, etc. que atuem como profissional liberal, consultores independentes ou ligados a alguma corporação, etc.) que porventura presta ou venha a prestar serviços para o IBAMA, contratada temporariamente ou não, independente da atividade que desempenhe ou venha a desempenhar relacionada ao desenvolvimento de sistemas e banco de dados.

Trigger (gatilho) : é um conjunto de comandos PL/SQL que são executados quando se realiza uma inserção, alteração e/ou exclusão em uma Tabela. A *trigger* é executada (disparada) antes da consumação da operação sobre a tabela. Se, por algum motivo, a execução da *trigger* não for completada, a operação sobre a tabela também não será. Geralmente são utilizadas para implementar regras de integridade complexas e que por isso não podem ser implementadas por *Constraints*; também são utilizadas para execução de ações diversas desencadeadas pela operação de inserção, alteração ou exclusão sobre a tabela.



Ministério do Meio Ambiente
Instituto Brasileiro do Meio Ambiente e dos Recursos Naturais Renováveis
Diretoria de Planejamento, Administração e Logística
Centro Nacional de Telemática

Type (tipo): são tipos de dados de alias ou um tipo definido pelo usuário no banco de dados atual, baseado em um tipo nativo do banco de dados.

Username (nome do usuário): corresponde a um nome pelo qual o usuário é identificado e reconhecido no banco de dados, não sendo necessariamente parte do nome da pessoa que é o usuário do banco de dados. Por padrão, deve ser utilizado o número do cpf ou cnpj da pessoa sem a formatação, ou seja, somente os caracteres numéricos.

View (visão) : representam logicamente subconjuntos de uma ou mais tabelas e se comportam como tabelas quanto a inserção, alteração, exclusão e consulta de dados; ou seja, pode-se inserir, alterar, excluir ou consultar dados em uma *view*, respeitando sempre as regras de integridade e de segurança que estiverem definidas sobre as Tabelas envolvidas na constituição da *view*.

ANEXO W DO TERMO DE REFERÊNCIA

**PADRÃO DE CODIFICAÇÃO DE SISTEMAS DE
INFORMAÇÃO**

Versão 1.0



Ministério do Meio Ambiente
Instituto Brasileiro do Meio Ambiente e dos Recursos Naturais Renováveis
Diretoria de Planejamento, Administração e Logística
Centro Nacional de Telemática

PADRÕES DE CODIFICAÇÃO DE SISTEMAS DE INFORMAÇÃO

Versão 1.0

- 2013 -



Ministério do Meio Ambiente
Instituto Brasileiro do Meio Ambiente e dos Recursos Naturais Renováveis
Diretoria de Planejamento, Administração e Logística
Centro Nacional de Telemática



1. Introdução

1.1 Padrões de codificação são um grupo de recomendações para linguagens de programação, indicando estilos de programação, praticas e métodos para cada item de um programa escrito em uma linguagem específica.

1.2 O padrões definem organização de arquivos, indentação, comentários, declarações, estados, espaços, nomenclatura, praticas de programação, princípios de programação e algumas regras de ouro. É extremamente recomendado que programadores sigam estas regras para ajudar a melhorar a legibilidade e entendimento do código criado também facilitando as futuras manutenções deste código.

1.3 Todo sistema de informação desenvolvido ou mantido no âmbito do Ibama deverá observar as regras desta norma.



2. Objetivos

2.1 O padrão de codificação visa estabelecer as regras de codificação de sistemas aplicadas às novas implementações e às evoluções dos sistemas em produção.

2.2 O objetivo de um padrão único de codificação é prover a melhoria contínua do processo de desenvolvimento com vistas a aprimorar a qualidade dos sistemas corporativos do Ibama, proporcionando:

- a) elevação da legibilidade do código;
- b) elevação da qualidade do código fonte;
- c) maior legibilidade do código;
- d) mudanças com menor impacto no código;
- e) melhoria do conhecimento técnico do programador;
- f) aumento de produtividade;
- g) aumento da manutenibilidade, ou seja, facilidade de um software de ser modificado a fim de corrigir defeitos.



3. Os Padrões de Codificações

3.1. Formato de Arquivos

A identificação deve possuir as seguintes características:

- a) O TAB deve possuir no tamanho de 4 espaços.
- b) O Tamanho máximo da Linha de ser de 80 caracteres, admitindo-se até 120 caracteres, em casos especiais.
- c) As linhas devem terminar apenas com Linefeeds (LF) – [n].
- d) Não usar Carriage Return (CR) – [r], padrão Macintosh.
- e) Não usar a combinação (CR)(LF) – [r\n], padrão Windows.

O Bloco de Documentação deve ser construído nos padrões JAVAdoc e PHPdoc, conforme tecnologia adotada, no início do arquivo com informações pertinentes ao mesmo.

Exemplo:

```
/**
 * [Descrição do arquivo].
 *
 * [mais informações precisa ter 1 [ENTER] para definir novo parágrafo]
 *
 * [pode usar quantas linhas forem necessárias]
 * [linhas logo abaixo como esta, são consideradas mesmo parágrafo]
 *
 * @package [Nome do pacote de Classes, ou do sistema]
 * @category [Categoria a que o arquivo pertence]
 * @name [Apelido para o arquivo]
 * @author [nome do autor] <[e-mail do autor]>
 * @link [link de onde pode ser encontrado esse arquivo]
 * @version [Versão atual do arquivo]
 * @since [Arquivo existe desde: Data ou Versao]
 */
```

3.2. Nomeclaturas

3.2.1. Padrão CamelCase

O padrão refere-se à prática de escrever palavras compostas ou frases, onde cada palavra é iniciada com Maiúsculas, e unida sem espaços.

O padrão divide-se em:



- a) UpperCamelCase - iniciado por letra Maiúscula (ex: GameCube, OpenOffice.org, StarCraft).
- b) lowerCamelCase - iniciado por letra Minúscula (ex: iPod, iMac)

3.2.2. UpperCamelCase

Deve ser utilizado o padrão UpperCamelCase para a nomenclatura de Classes (Exemplos: Uploads, FileUploads, ImageFileUploads).

3.2.3. lowerCamelCase

Deve ser utilizado o padrão lowerCamelCase para:

- a) Variáveis (Exemplos: `$tmpQry`, `$objUpload`, `$arrUF`, `$iCount`)
- b) Propriedades (Exemplos: `$this->tableName`, `$this->fieldId`, `$this->fields`)
- c) Funções e Métodos (Exemplos: `$this->getName()`, `$this->setFields()`, `$this->getById()`).

3.3.4. UPPERCASE

Deve ser utilizado o padrão UPPERCASE para a nomenclatura de Constantes (Exemplo: NOT_EXIST, OVERFLOW_MAX_SIZE, DB_HOST).

3.3. Estilo de Código

3.3.1. Demarcação de Código PHP

- Não usar `short_tags` (`<? e <?=>`).
- Usar sempre `<?php ?>` para delimitar código PHP, pois é a forma mais portátil de incluir código PHP em diferentes sistemas operacionais e configurações.

3.3.2. Comentários

- Utilizar `/**/` para comentário de um bloco (mais de uma linha).



- Utilizar // para comentário de uma única linha.
- Utilizar // para comentário ao final da linha.

3.3.3. Quebra de Linhas

Quando a expressão não couber em uma única linha, deve-se quebrar a linha de acordo com as seguintes regras:

- Quebrar a linha após uma vírgula;
- Quebrar a linha antes de um operador;
- Alinhar a nova linha com o mesmo nível do início da expressão na linha anterior;

3.3.3.Classes

- ⤴ Deve-se nomear em UpperCamelCase.
- ⤴ As chaves "{" e "}" devem constar na linha abaixo ao da linha nome da Classe.
- ⤴ Toda classe deve ter um bloco de documentação em conformidade com o Padrão do PHPDocumentor ou JAVAdoc.
- ⤴ Qualquer código dentro da classe precisa ser indentado com quatro espaços.
- ⤴ Admite-se apenas uma classe por arquivo.

Exemplo:

```
class Exemplo
{
    protected $name = null;
    /**
     * Seta o um valor à propriedade Name.
     * @access public
     * @param string $pValue Nome Completo
     */
    public function setName( pValue )
    {
        // Qualquer conteúdo
        // precisa ser indentado com [Tab|4 espaços].
    }
    /**
     * Retorna o valor da propriedade Name.
```



```
* @access public
* @return string
*/
public function getName()
{
    // Qualquer conteúdo
    // precisa ser indentado com [Tab|4 espaços].
}
}
```

3.3.4. Variáveis de Classes (Propriedades)

- ⤴ Deve-se nomear em lowerCamelCase.
- ⤴ Devem ser declaradas no topo da classe, precedendo as demais declarações de métodos.
- ⤴ Deve-se declarar a visibilidade para todas as classes: private, protected ou public.
- ⤴ Não utilizar declaração de variáveis de classes como public, para incentivar o uso de (set/get), salvo em decorrência de especificidades técnicas que exijam o uso de classes públicas.

3.3.5. Funções e/ou Métodos

- ⤴ Deve-se nomear em lowerCamelCase.
- ⤴ As chaves "{" e "}" devem constar na linha abaixo a do nome da Função/Método.
- ⤴ Toda Função/Método deve possuir um bloco de documentação em conformidade com o Padrão do PHPDocumentor ou JAVAdoc.
- ⤴ Qualquer código dentro da Função/Método precisa ser indentado com quatro espaços.
- ⤴ Deve-se declarar a visibilidade para todos os métodos e funções: private, protected ou public.

3.3.6. Estruturas de Controle

⤴ if / else / elseif

- a) Deve-se usar espaço simples depois do "(" e antes do ")" da condição no IF e no ELSEIF.
- b) A chave "{" deve constar da mesma linha da expressão, a chave "}" virá na linha após a última linha de conteúdo.



c) Qualquer código entre as chaves "{" e "}" precisa ser indentado com quatro espaços.

d) Exemplos:

```
if ( /*condicao*/ ) {  
    // Qualquer conteúdo  
    // precisa ser indentado com [Tab|4 espaços].  
} elseif ( ( /*condicao1*/ ) or ( /*condicao2*/ ) ) {  
    // Qualquer conteúdo  
    // precisa ser indentado com [Tab|4 espaços].  
} else {  
    // Qualquer conteúdo  
    // precisa ser indentado com [Tab|4 espaços].  
}
```

▲ while/for

a) Deve-se usar espaço simples depois do "(" e antes do ")" da expressão.

b) A chave "{" virá na mesma linha da expressão, a chave "}" virá na linha abaixo da última linha de conteúdo.

c) Qualquer código entre as chaves "{" e "}" precisa ser indentado com quatro espaços.

Exemplo:

```
while ( /*Condição*/ ) {  
    // Qualquer conteúdo  
    // precisa ser indentado com [Tab|4 espaços].  
}  
  
do {  
    // Qualquer conteúdo  
    // precisa ser indentado com [Tab|4 espaços].  
} while ( /*Condição*/ ) ;  
  
for ( /*inicialização*/; /*Condição*/; /*atualização*/ ) {  
    // Qualquer conteúdo  
    // precisa ser indentado com [Tab|4 espaços].  
}
```

▲ switch/case

a) Usar espaço simples depois do "(" e antes do ")".

b) A chave "{" virá na mesma linha da Expressão, a chave "}" virá na linha abaixo da última linha de conteúdo.

c) Qualquer código entre as chaves "{" e "}" precisa ser indentado com quatro espaços.

d) Qualquer código dentro de: case e default, precisa ser indentado com



quatro espaços, inclusive a palavra reservada: break.

Exemplo:

```
switch ( /*variavel*/ ) {  
    case 1:  
        // Qualquer conteúdo  
        // precisa ser indentado com [Tab|4 espaços].  
        break;  
    case 2:  
        // Qualquer conteúdo  
        // precisa ser indentado com [Tab|4 espaços].  
        break;  
    default:  
        // Qualquer conteúdo  
        // precisa ser indentado com [Tab|4 espaços].  
}
```

4. Arquitetura

Para as soluções orientadas à transação deve-se utilizar as linguagens de programação PHP ou JAVA.

O desenvolvimento dos sistemas transacionais deve ser realizado em camadas (Apresentação, Negócio e Integração).

Deve-se admitir apenas componentes, bibliotecas e pacotes de dados livres, ou seja, desenvolvidos sob a licença GNU General Public License – GPL ou Lesser General Public License – LGPL ou Creative Commons – CC.

Para as soluções orientadas à processo deve-se utilizar tecnologia baseada na notação BPMN com integração em base de dados ORACLE.

Na construção da solução, deve-se adotar as técnicas de programação orientada a objetos e prover o baixo acoplamento e alta coesão entre os módulos e componentes implementados.

5. Referências

- Sun Microsystem. Code Conventions for the Java™ Programming Language, Revisado em 20/04/1999: <http://www.oracle.com/technetwork/java/javase/documentation/codeconvtoc-136057.html>
- Zend Framework. Coding Style: <http://framework.zend.com/manual/en/coding-standard.coding-style.html>
- Wikipedia. CamelCase: <http://pt.wikipedia.org/wiki/Camelcase>



Ministério do Meio Ambiente
Instituto Brasileiro do Meio Ambiente e dos Recursos Naturais Renováveis
Diretoria de Planejamento, Administração e Logística
Centro Nacional de Telemática

- OLIVEIRA, Walker de Alencar. PHP Coding Standards v0.6, 1999.
- Metodologia de Desenvolvimento de Sistemas do Ibama V1.1.

ANEXO X DO TERMO DE REFERÊNCIA

**PADRÃO DE INTERFACE E USABILIDADE DE
SISTEMAS DE INFORMAÇÃO**

Versão 2.0



Ministério do Meio Ambiente
Instituto Brasileiro do Meio Ambiente e dos Recursos Naturais Renováveis
Diretoria de Planejamento, Administração e Logística
Centro Nacional de Telemática

PADRÃO DE INTERFACE E USABILIDADE DE SISTEMAS DE INFORMAÇÃO

Versão 2.0

2014



Histórico de Revisão

Data	Versão	Descrição	Autor
15/04/2011	1.0	Elaboração do documento	Rosana de Souza Ribeiro Freitas
02/05/2014	2.0	Revisão e alteração do documento pela Ascom	Luciana V. Araújo, Talitha Monfort Pires, Rodrigo Vasconcellos e Jucier Costa Lima



1. Introdução

A interface é o meio pelo qual o usuário intervém, dialoga, modifica o objeto da interação (no caso, um sistema de informação). Se a interface for fácil de aprender, simples, direta e amigável, ele estará inclinado a fazer uso dela. Desenvolver sistemas de informação com usabilidade é pensar do ponto de vista do usuário.

A usabilidade pode ser definida como o estudo ou a aplicação de técnicas que proporcionem a facilidade de uso de um dado objeto, no caso, um sítio na internet. A usabilidade busca assegurar que qualquer pessoa consiga usar o sítio e que este funcione da forma esperada. Em resumo, a usabilidade tem como objetivos a:

- facilidade de uso;
- facilidade de aprendizado;
- facilidade de memorização de tarefas;
- produtividade na execução de tarefas;
- prevenção, visando à redução de erros;
- satisfação do indivíduo.

Tanto a acessibilidade quanto a usabilidade têm como foco de atenção o usuário (cidadão) e, muitas vezes, se sobrepõem a áreas do saber. No entanto são áreas distintas. Acessibilidade trata do acesso a locais, produtos, serviços ou informações efetivamente disponíveis ao maior número e variedade possível de pessoas independentemente de suas capacidades físico-motoras, perceptivas, culturais e sociais. Já a usabilidade trata da facilidade de uso. Um sistema de informação pode acabar sendo acessível mas difícil de ser utilizado ou fácil de ser utilizado mas inacessível a parte dos usuários.

No entanto a acessibilidade e a usabilidade são disciplinas complementares e é mais provável que um sítio com boa acessibilidade tenha uma boa usabilidade assim como um sítio com boa usabilidade seja mais acessível.

O aumento das demandas de aplicações, a crescente participação de diversos colaboradores de empresas terceirizadas para a construção de sistemas de informação do Ibama e a utilização de diferentes interfaces para cada aplicação dificultam o uso do sistema pelos usuários, pois o mesmo recurso pode ser realizado de forma



completamente diferente, ocasionando a diminuição da capacidade de assimilação, acessibilidade e memorização de tarefas. As tarefas simples passam a ser complexas por serem feitas de formas diferentes em cada aplicação. Além disso, essa falta de padronização impacta negativamente a identidade visual do órgão, fazendo com que o usuário não identifique todos os sistemas como sendo do Ibama.

O CNT (Centro Nacional de Telemática), responsável pelo setor de TI do Instituto Brasileiro do Meio Ambiente e dos Recursos Naturais Renováveis (Ibama), visando a agilizar o desenvolvimento, facilitar a navegabilidade e principalmente padronizar um *layout* mínimo dos sistemas de informação do Ibama, estabelece, com este documento, um padrão de interface e usabilidade.

Cabe ressaltar que este padrão não substitui (mas complementa) o Modelo de Acessibilidade de Governo Eletrônico (eMAG), que consiste em um conjunto de recomendações a ser considerado para que o processo de acessibilidade dos sítios e portais do governo brasileiro seja conduzido de forma padronizada e de fácil implementação, e os Padrões Web em Governo Eletrônico (ePWG), que são recomendações de boas práticas agrupadas em formato de cartilhas com o objetivo de aprimorar a comunicação e o fornecimento de informações e serviços prestados por meios eletrônicos pelos órgãos do governo federal.

Todo sistema de informação web desenvolvido ou mantido no âmbito do Ibama deverá seguir as regras desta norma, exceto para os sítios do Ibama e Ibamanet, que possuem particularidades específicas.



2. Objetivos

Este documento tem como objetivo orientar os desenvolvedores, *web designers* e usuários requisitantes na construção e na manutenção da interface de sistemas de informação no âmbito do Ibama.

Os princípios apresentados a seguir são baseados em parâmetros de usabilidade, ou seja, com foco no usuário, a fim de se promover uma melhor interação do usuário com as funcionalidades do sistema, orientando-se sua construção para que os padrões de usabilidade sejam mantidos. Estas orientações visam a abranger o máximo de soluções de interfaces necessárias para o correto desenvolvimento de sistemas.

O objetivo de um padrão mínimo de interface e usabilidade é prover a melhoria contínua do processo de desenvolvimento com o propósito de se aprimorar a qualidade dos sistemas corporativos do Ibama, buscando-se:

- estabelecer padrões de qualidade de uso, desenho, arquitetura de informação e navegação;
- consolidar a acessibilidade;
- criar artefatos de acordo com os padrões estabelecidos pelo W3C (*World Wide Web Consortium*);
- aumentar a capacidade de manutenção de um *software*, ou seja, a facilidade de modificá-lo a fim de se corrigirem defeitos;
- garantir a consistência entre as aplicações;
- facilitar o trabalho em conjunto de grupos;
- aumentar a eficiência na utilização do aplicativo;
- reduzir o tempo e os custos de desenvolvimento de novas aplicações.



3. Padrão de interface e usabilidade

O padrão de interface das aplicações desenvolvidas para os sistemas de informação do Ibama em ambiente web deverá ser implementado utilizando-se uma estrutura simples de apresentação dividida nas seguintes partes:

- Login
- Cabeçalho
- Menu
- Área de Trabalho
- Rodapé
- Formulários

3.1 Tela de login

Será composta de duas áreas:

3.1.1 Área de identificação do sistema

Deve disponibilizar a logomarca do Ibama e o nome do sistema (obrigatório) com fonte arial 14 em negrito, na cor #000080 e com alinhamento centralizado. Os sistemas não podem possuir logomarcas próprias.

3.1.2 Área de trabalho

Deve disponibilizar as informações para que o usuário possa *logar* no sistema. A fonte do texto deverá ser arial, tamanho 12, na cor #000080 e alinhado à esquerda e os objetos centralizados.

Para o acesso com senha comum, a tela de login deverá possuir os campos para entrada do “CPF”, “senha”, “caracteres *captcha*”, o *link* “Esqueceu a senha?” e o botão “Entrar”.

Para o acesso com certificação digital, a tela de login deverá possuir o botão “Acessar com certificado digital” e o *link* para uma página com “Informações sobre certificação digital”.



Nome do Sistema

Instituto Brasileiro do Meio Ambiente e dos Recursos Naturais Renováveis

Informe CPF e Senha para acessar o sistema

CPF

Senha

cdce **Entrar**

> Esqueceu a senha?

Com certificação digital para acessar o sistema

Acessar com Certificado Digital

> Informações sobre a Certificação digital

Figura 01: exemplo da tela de login

Para os usuários que possuírem mais de um perfil no sistema, deverá ser apresentada uma tela para a escolha do perfil conforme tela abaixo:

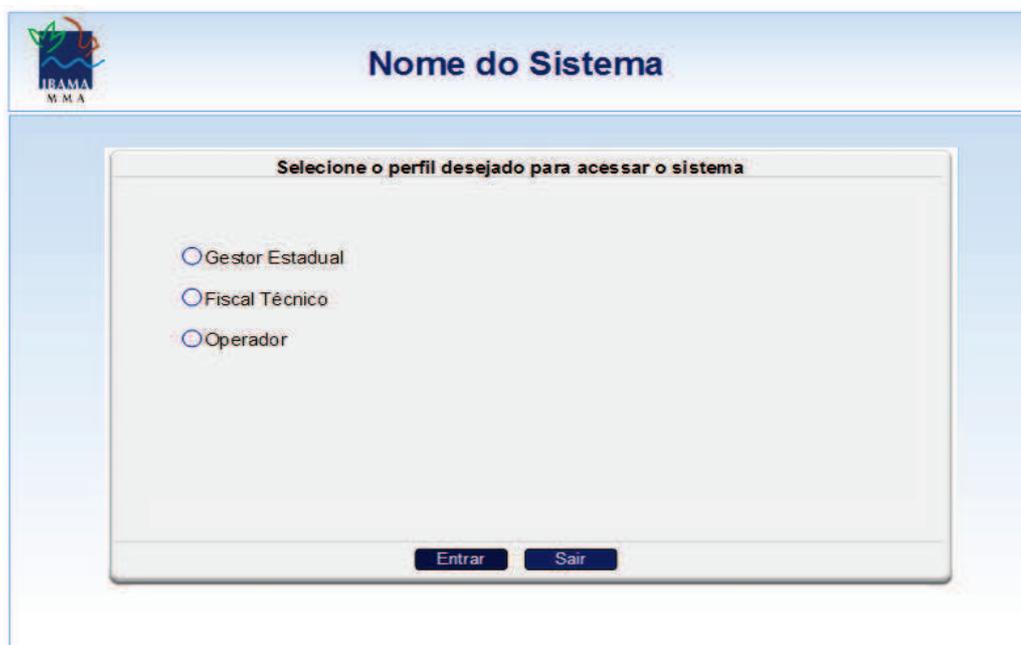


Figura 02: exemplo da tela de seleção de perfil

3.2 Cabeçalho

O cabeçalho deve ter uma altura de 55 pixels (55px), preenchido com a cor azul-claro (#c9dffd) em degradê.

Deverá ser subdividido em três áreas:

- Área de identificação do sistema;
- Área para apresentação das informações do usuário que está acessando;
- Área de ícones rápidos.

3.2.1 Área de identificação do sistema

Deve disponibilizar a logo do Ibama e o nome do sistema (obrigatório) com fonte arial 14 em negrito, na cor #000080 e com alinhamento centralizado.

3.2.2 Área para apresentação das informações do usuário que está acessando

Deve apresentar, no campo superior à direita, o nome do usuário e o perfil que está acessando o sistema, com fonte arial 12 na cor preta e alinhado à esquerda.

3.2.3 Área de ícones rápidos

Deve disponibilizar, logo abaixo das informações do usuário, os ícones/botões “Página Inicial”, “Ajuda”, “Fale Conosco”, “Trocar Perfil” e “Sair”.



Figura 03: exemplo de cabeçalho antes da autenticação do usuário



Figura 04: exemplo de cabeçalho após a autenticação do usuário

3.3 Menu

O menu do sistema organizado por módulos e funcionalidades deverá ser posicionado abaixo do cabeçalho, à esquerda da tela, em posição vertical, com submenus horizontais em até no máximo dois subníveis se necessário. O *background* do menu será na cor #e1e9f4 e a fonte arial tamanho 12 na cor #000080 para o menu principal e o secundário e a cor preta para os demais itens de menu.

Os módulos devem trazer o símbolo > indicando que há submenus a serem acessados e, ao se passar o mouse no menu/submenu, este deverá ser aberto e a cor do *background* alterada para #0084d1.

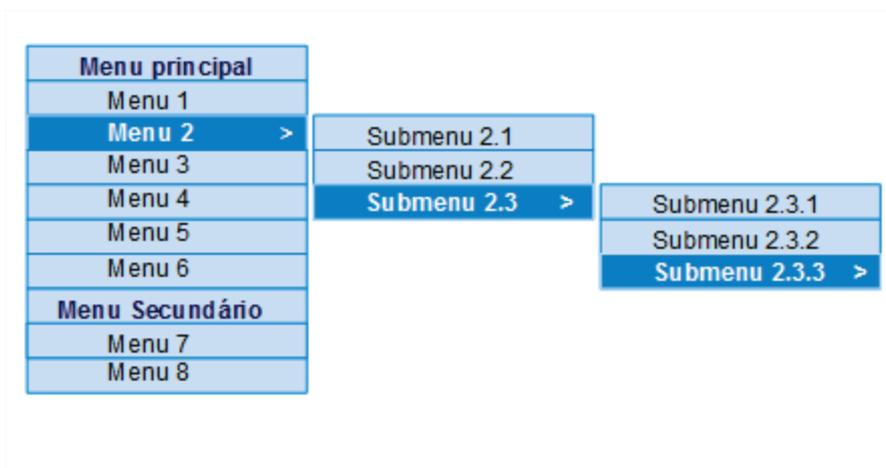


Figura 05: exemplo de menu principal e secundário

3.4 Área de Trabalho

A área de trabalho da tela inicial deve disponibilizar a apresentação do sistema (obrigatório) ou painel geral do sistema (opcional). Após a seleção de algum menu, deve ser apresentado neste espaço a *grid* contendo os registros cadastrados para a



funcionalidade selecionada ou a tela de consulta.

Deverá ser subdividida em duas áreas:

- Área do caminho em que o usuário se encontra (migalha de pão) e pesquisa geral
- Área do conteúdo

3.4.1 Área do caminho em que o usuário se encontra (migalha de pão)

Nesta área, deve ser identificado o caminho em que o usuário se encontra enquanto navega no sistema. Esta “migalha” deverá ser apresentada em uma barra ao lado do menu principal. Quando a área disponível não for suficiente para mostrar todo o caminho, os primeiros itens do caminho devem desaparecer, mostrando-se o símbolo de “.....” indicando que o caminho visível não é o completo. O texto deverá ser escrito com a fonte arial tamanho 12, na cor preta e alinhado à esquerda.

Também nesta área, poderá ser disponibilizada a pesquisa geral em todo o site, que é opcional. Deverá conter um campo texto curto e o ícone de busca e deverá ser alinhado à esquerda.



Figura 06: exemplo da utilização do caminho (migalha de pão) e busca



3.4.2 Área do conteúdo

Deverá ser na cor branca com a logomarca do Ibama ao centro, como marca d'água, e ocupar todo o espaço entre o cabeçalho e o rodapé da aplicação, sendo redimensionado de acordo com a largura e a altura da janela do navegador.

Os títulos de cada página serão no formato de formulário-padrão.



Figura 07: exemplo da área do conteúdo

3.5 Rodapé

O rodapé, que é opcional, deverá ter uma altura de 20 pixels (20px), preenchido com cinza claro degrade na cor #e1e9f4.

Deverá ser subdividido em três áreas, sendo a primeira à esquerda para mensagens livres da aplicação, a segunda, para a sigla do sistema e versão, e a terceira, reservada para logomarcas pequenas de parceiros (se houver e mediante aprovação).



Figura 08: Exemplo de rodapé



3.6 Formulários

Os formulários do sistema deverão ser divididos em seções que obedecerão ao critério de agrupamento de atributos logicamente relacionados.

O formulário deverá apresentar, quando expandido, somente a primeira seção (grupo) com que o usuário tiver interação. As demais devem ser apresentadas fechadas.

Todas as seções poderão ser apresentadas expandidas desde que a quantidade de seções apresentadas no formulário não prejudique a visualização do sistema nas resoluções atendidas pelo projeto e desde que não criem barras de rolagem desnecessárias. Todo formulário deve possuir ao menos uma seção.

Os formulários deverão ser exibidos centralizados na área de conteúdo da aplicação e deverão possuir as seguintes características:

- O título alinhado à esquerda e na cor (#000080) com tamanho de fonte 12px em negrito e fundo na cor azul-escuro (#c9dffd) em degradê. O corpo do formulário será na cor branca.

- Os botões Minimizar, Maximizar e Fechar deverão estar na parte superior esquerda com a imagem padrão 15x28 pixels (px).

- Os botões das ações deverão ficar alinhados na parte inferior do formulário e centralizados. O *background* do botão será na cor azul-escuro (#000080) e a fonte arial na cor branca com tamanho 12.

Cadastrar Empreendimento

▼ **Dados do Empreendimento**

Rótulo:

Rótulo:

Rótulo:

^ **Dados do Responsável pelo Empreendimento**

^ **Documentos anexos**

Cadastrar **Limpar**

Figura 09: exemplo de formulário de Cadastro



The screenshot shows a web application interface. On the left is a vertical menu with items: Menu principal, Menu 1, Menu 2 (highlighted), Menu 3, Menu 4, Menu 5, Menu 6, Menu Secundário, Menu 7, and Menu 8. The main content area is titled 'Empreendimentos' and contains a search form with fields for CNPJ, Nome, and UF, and buttons for 'Consultar', 'Cadastrar', and 'Limpar'. Below the form is a table titled 'Lista de Empreendimentos' with columns 'Coluna 1', 'Coluna 2', 'Coluna 3', and 'Ação'. The table contains four rows of data. At the bottom of the table, it shows 'Total de registros: 16' and 'Página: < 2/4 >'. The user's name 'Rosana Ribeiro Gestor Estadual' is visible in the top right corner.

Figura10: exemplo de formulário Consultar e Listar

3.6.1 Rótulos dos campos

Caso os campos tenham rótulos, estes deverão ser na cor preta com alinhamento à esquerda, fonte arial, tamanho 12 e exibidos de forma alinhada à frente dos campos.

O campo obrigatório deverá apresentar um asterisco na cor vermelha a sua frente.

Rótulo:*

Figura 11: exemplo de campo obrigatório com asterisco em vermelho

Caso o usuário se esqueça de preencher o campo, após mensagem, o rótulo pode ser apresentado em cor vermelha e com asterisco.



3.6.2 Campo texto curto

Os atributos do tipo “alfanumérico” curtos deverão ser compostos por campos do tipo *Input*. Deverão contemplar dois (2) tamanhos de campos para inserção de textos curtos:

- largura 220px: até 30 caracteres

Rótulo:

- largura 360px: de 31 a 80 caracteres

Rótulo:

Figura 12: exemplos de campo texto curto

3.6.3 Campo texto longo

Os atributos do tipo “alfanumérico” longos deverão ser compostos por campos do tipo *Textarea*.

- Largura 360px: a partir de 81 caracteres

Rótulo:

Figura 13: exemplo de campo texto longo

3.6.4 Campo data

O campo data deverá apresentar o ícone de calendário para possibilitar a exibição do calendário. E deverá apresentar oito (8) caracteres com a máscara no formato dd/mm/aaaa (dia, mês e ano).

Campo data: 

Figura 14: exemplo de campo data

Ao selecionar o ícone calendário, deverá abrir um calendário com a data do dia selecionada e com opção para se selecionar outra data desejada.

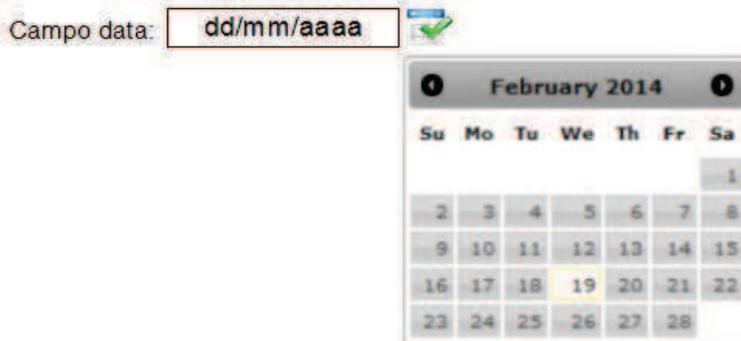


Figura 15: Exemplo de calendário

Input hora – largura 92px: 4 caracteres com a máscara “hh:mm”.



Figura 16: exemplo de campo hora

3.6.5 Seleção única – combo – largura 92px: devem ser disponibilizadas as horas. O intervalo seguido deverá constar do documento de especificação de requisitos junto com o formato de hora a ser utilizado.

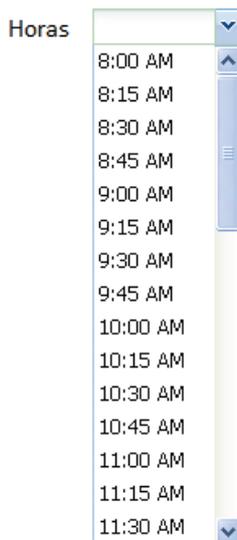


Figura 17: exemplo de campo hora

3.6.6 Campo de seleção única ou múltipla

Os atributos do tipo “domínio” podem possuir valores fixos ou valores provenientes de tabelas do sistema, de seleção única ou de múltipla seleção.

Os atributos do tipo “domínio fixo”, com seleção única e apresentados em ordem alfabética, poderão ser inseridos por quatro tipos de campos.

As opções deverão estar inseridas em um *fieldset* próprio e com o rótulo do campo no título.

Radio – Disponibilização de campos com até três opções mutuamente excludentes, ou seja, seleção única.

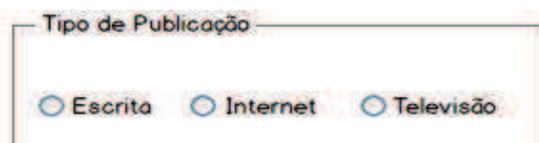


Figura 18: exemplo de seleção única – Radio button

Quando a regra definir um valor padrão, esta opção deverá ser apresentada pré-selecionada. Caso contrário, todos os itens devem ser apresentados desmarcados.

Os atributos do tipo “domínio fixo” e “domínio em banco”, com seleção única e apresentados em ordem alfabética, poderão ser inseridos por quatro tipos de campos.

Combo pq – largura 220px: disponibilização para mais de três opções mutuamente excludentes em que o maior registro não ultrapasse 30 caracteres.

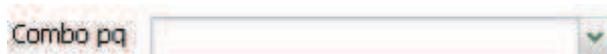


Figura 19: exemplo de combo pequeno

Combo gd – largura 360px: disponibilização de mais de três opções mutuamente excludentes em que o maior registro possua entre 31 e 100 caracteres.



Figura 20: exemplo de combo grande

Seleção única – Combo *Inline* pq – largura 220px: para casos onde o maior registro não ultrapasse 30 caracteres (em arial 12)

Combo *Inline* gd – largura 360px: para casos onde o maior registro possua entre 31 e 100 caracteres (em arial 12)

Item (ComboBox) : 334

SKU	Item
33431400	Manilla Folders Foolsap 5 Tab Buff
33476704	Manilla Folders Foolsap Dark Blue
33476705	Manilla Folders Foolsap Red
33476706	Manilla Folders Foolsap Dark Green
33476707	Manilla Folders Foolsap Pink
33476708	Manilla Folders Foolsap Yellow
33476709	Manilla Folders Foolsap Orange
33476713	Manilla Folders Foolsap Grey
33476733	Manilla Folders Foolsap Lilac
33476756	Manilla Folders Foolsap Light Blue
33476757	Manilla Folders Foolsap Light Green

Figura 21: exemplo de seleção única grande

Seleção múltipla – Checkbox para até três opções

Check1

Check2

Check3

Figura 22: exemplo de seleção múltipla – checkbox

Seleção múltipla – *Checkbox list* para mais de três opções

Selecionar todos

Seguro de Vida

Casa Própria

Carro popular

Loja comercial

Geladeira

Televisão de LED

Figura 23: exemplo de seleção múltipla com mais de três opções – *checkbox list*

A primeira opção deverá permitir marcar e desmarcar todos e deverá ser apresentada conforme a figura 23.

As opções devem ser disponibilizadas em ordem alfabética, exceto em situações que a regra defina outro critério de ordenação.

Quando a regra definir um valor padrão, esta opção deverá ser apresentada

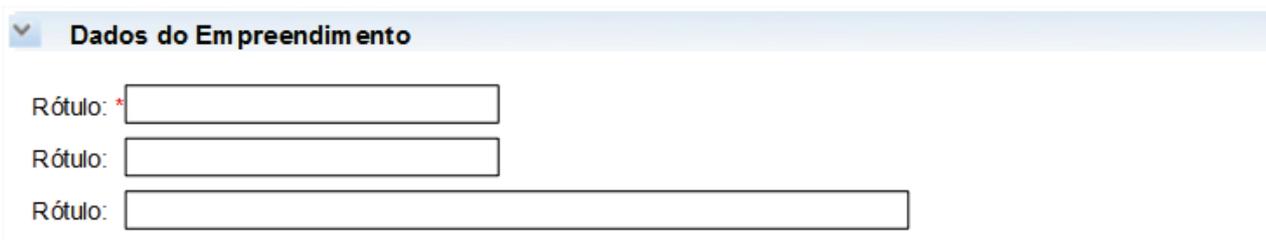


selecionada. Caso contrário, todos os itens devem ser apresentados desmarcados.

As determinações deste item podem ser modificadas de acordo com a especificação das regras de negócio.

3.6.7 Campo grupo

Deverá ter o nome dos itens que irão compor o grupo e este poderá abrir e fechar ao selecionar os ícones  e .



▼ **Dados do Empreendimento**

Rótulo: *

Rótulo:

Rótulo:

Figura 24: exemplo de campo grupo

3.6.8 Utilização de abas

A aba selecionada será evidenciada na cor (#c9dffd).

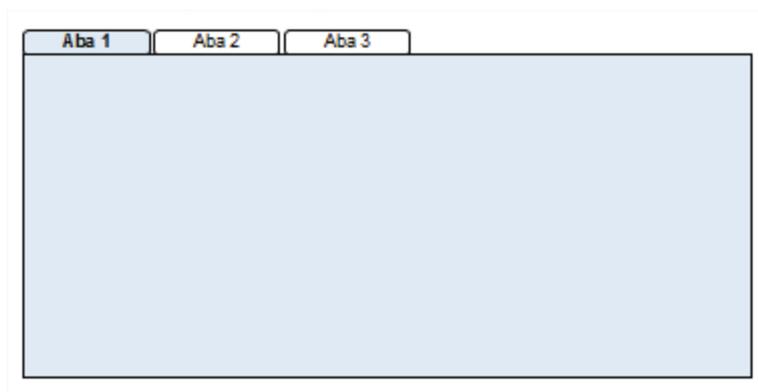


Figura 25: exemplo de utilização de abas

3.6.9 Utilização de gride

A estrutura de exibição de dados no formato de grid deve ser utilizada seguindo as seguintes características:

- O título deverá ser centralizado, na cor branca, fonte arial tamanho 12 e em



negrito. O título das colunas deve ser na cor preta, fonte arial tamanho 12 e o resultado listado deverá ser na fonte arial tamanho 12.

– As colunas que possuem o recurso de ordenar devem ser sublinhadas. As listas de registros do sistema deverão possibilitar a customização da ordenação pelos usuários, por qualquer atributo disponibilizado, a partir da seleção do cabeçalho do atributo pelo qual o usuário deseja ordenar. Esta função deverá permitir ainda que a ordem seja ascendente ou descendente.

– A grid deverá possuir paginação no rodapé. O sistema carregará do banco de dados os registros à medida que o usuário navegar usando as barras de rolagem ou quando selecionar avançar ou retroceder, apresentando o total de registros (ex: 16) e a página e a quantidade de páginas (ex: 2/4).

Lista de Empreendimentos			
<u>Coluna 1</u>	<u>Coluna 2</u>	<u>Coluna 3</u>	Ação
Coluna 1.1	Coluna 2.1	Coluna 3.1	   
Coluna 1.2	Coluna 2.2	Coluna 3.2	   
Coluna 1.3	Coluna 2.3	Coluna 3.3	   
Coluna 1.4	Coluna 2.4	Coluna 3.4	   

 Total de registros: 16 Página: < 2/4 >

Figura 26: exemplo de utilização de grid

Ainda na barra de paginação, deverá existir a opção de Atualização  . Esta opção deverá atualizar a lista de registros e, caso não seja encontrado nenhum registro, deverá ser emitida a mensagem “Não foram encontrados dados com os parâmetros informados!” no local em que se apresenta o resultado da pesquisa.

Lista de Empreendimentos	
Não foram encontrados dados com os parâmetros informados!	
	Total de registros: 00



Figura 27: exemplo de tela da grid quando não for encontrado nenhum registro

3.6.10 Utilização de upload

Para funcionalidades que permitem apenas um arquivo anexado:

Apenas a descrição do anexo e o botão que permite adicionar anexo será apresentada na tela do formulário.



Figura 28: exemplo de upload para apenas um arquivo

Quando existir um anexo, serão apresentados o nome do arquivo, a extensão, o tamanho e os botões de ações: Excluir  Visualizar  ou, se for em pdf, o símbolo 

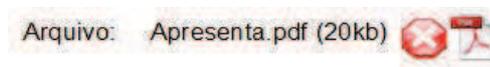


Figura 29: exemplo de upload quando já houver um arquivo inserido

Para a opção excluir, deverá aparecer uma tela para confirmação da operação. Ex. O arquivo será excluído. Deseja continuar com a operação?

Quando a funcionalidade permitir a inclusão de diversos anexos:

Serão apresentados a descrição do anexo e o botão que permite adicionar anexo e a *grid* com aqueles que foram anexados. A cada inserção de anexo, ele limpa o campo arquivo e atualiza a grid.

Para os anexos que já foram inseridos, haverá a opção para excluir e visualizar o anexo.



Documentos anexos

Arquivo:  [Adicionar arquivo](#)

Lista de documentos anexados			
Nome	Tamanho	Data Inserção	Ação
apresenta.pdf	20 kb	20/01/2014	 
imagem.jpg	05 kb	18/02/2014	 

Total de registros: 02 Página: < 1/1 >

Figura 30: exemplo de upload para diversos anexos

3.6.11 Relatórios

Os resultados de todos os relatórios deverão ser disponibilizados em formato PDF, e o opcional no formato CSV (Comma-separated values).

Para os relatórios no formato PDF, o Acrobat Reader deverá ser disponibilizado dentro da área de conteúdo do sistema, mantendo sua estrutura de navegação.

3.6.12 Validação de campos

O campo obrigatório deverá possuir o rótulo com um asterisco na cor vermelha a sua frente.



Rótulo:

Figura 31: exemplo de campo obrigatório com asterisco em vermelho

3.6.13 Validação *inline*

Validações que não necessitam de requisições ao banco deverão ser feitas após ser retirado o foco do campo. Caso seja verificado algum problema no conteúdo informado (ou a ausência de conteúdo em campos obrigatórios), será apresentado o ícone indicativo com a explicação do erro no *hint* do ícone em um balão de diálogo.



Figura 32: exemplo de validação campo obrigatório

Os formulários do sistema devem disponibilizar o botão de conclusão da ação habilitado para o usuário somente quando todas as validações *inline* tiveram sido contempladas.

Se o usuário limpar o conteúdo do campo, este deverá retornar ao formato padrão, indicando a obrigatoriedade do preenchimento.

3.6.14 Validação em uma transação

Ao executar a validação dos atributos e/ou regras de negócio em uma transação, caso o sistema identifique algum atributo inválido e/ou a infração de alguma regra, deverá ser apresentada mensagem ao usuário informando-o.

Em seguida, o sistema deve retornar à tela de preenchimento do formulário, mantendo os dados informados.

Nos casos onde ocorrer falha na conexão com a base de dados no momento da validação, o sistema deve apresentar a mensagem:

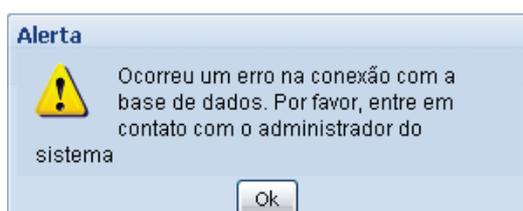


Figura 33: exemplo de mensagem de falha na conexão com a base de dados

3.6.15 Desconectar do sistema



Apenas as funcionalidades e/ou as tarefas do sistema que não forem apresentadas em *pop-up* deverão possuir opção de desconectar do sistema. Esta opção deverá cancelar a tarefa executada mediante *logoff* do usuário no sistema.

3.6.16 Ações dependentes de indicação de registro

A ação que dependa da indicação de um registro deverá ficar desabilitada até que um registro seja selecionado. Após a seleção do registro, o ícone da ação deverá ser habilitado.

3.6.17 Campos dependentes

Os campos que dependam de outro para ser informados devem ser exibidos em tela somente após a informação daquele que o habilita.

3.6.18 Atributos não editáveis

Os campos dos atributos apresentados sem a possibilidade de alteração deverão ser não editáveis, com padrão visual diferenciado (fonte na cor #808080).

3.6.19. Foco

A posição do foco do usuário não deverá ser alterada sem que ele acione algum comando que execute esta ação.

3.6.20 Apresentação das funcionalidades

As funcionalidades do sistema devem ser apresentadas na janela de navegação utilizada pelo sistema. As *div modais* ou *pop-ups* não deverão ser abertas em outro *browser* ou aba e deverão se sobrepor caso sejam abertos vários níveis.

Apenas deverão ser habilitadas as ações da última *div* ou *pop-up* aberta. As anteriores deverão permanecer abertas, esmaecidas, mas não habilitadas.

3.6.21 Apresentação das mensagens de operação

Todas as mensagens de operação deverão ser apresentadas em janelas definidas pela *framework* adotada ou os *alert javascript*.

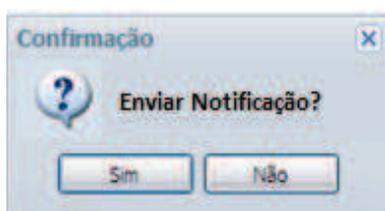




Figura 34: exemplo de mensagem de operação

3.6.22 Execução de função

O acionamento de uma tecla só deve executar uma ação caso o foco esteja em objeto que possua uma ação associada.

3.6.23 Mensagens de sucesso

O sistema deverá apresentar mensagem de sucesso da operação realizada sempre que concluir as transações sem erros.

Esta mensagem deverá desaparecer automaticamente após 3 segundos sem a necessidade de confirmação.

Os textos das mensagens para os fluxos serão:

Inclusão = Inclusão realizada com sucesso!

Alteração = Alteração realizada com sucesso!

Exclusão = Exclusão realizada com sucesso!

Para fluxos diferentes dos citados, a mensagem deverá seguir o mesmo padrão: <fluxo> realizado<a> com sucesso!

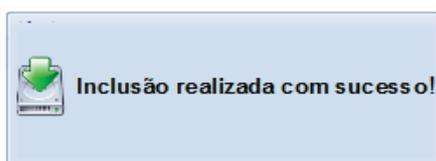


Figura 35: exemplo de mensagem de sucesso

3.6.24 Campo com máscara

Todo campo que possuir máscara definida na especificação de requisitos deverá ter a validação para erro no formato do campo feita *inline*, ou seja, após ser retirado o foco do campo, o sistema já deverá indicar se o campo foi preenchido corretamente.

Caso o usuário tente sair do campo sem preenchê-lo corretamente, deverá ser apresentado o ícone indicativo com a mensagem de erro no *hint* do ícone. A mensagem de erro será: "O <nome do campo> deve estar no formato <máscara correta>."

3.6.25 Validação para quantidade de caracteres



Toda vez que o usuário tentar incluir dados que violem as quantidades de caracteres definidas para o campo, o sistema deverá fazer a validação *inline* e mostrar a mensagem no *hint* do ícone indicativo, conforme exemplos que seguem:

Quantidade máxima: <XX> caracteres.

Quantidade mínima: <XX> caracteres. Ou

A quantidade deve estar entre <mínimo> e <máximo> caracteres.

3.6.26. Elementos visuais

A usabilidade será composta por diversos elementos visuais (ícones, cores e textos), nos quais são observados aspectos de contraste, repetição, alinhamento e proximidade com o objetivo de orientar o usuário por intermédio de pistas visuais, facilitando-se a utilização do sistema.

A biblioteca de ícones será a [Crystal Clear](http://commons.wikimedia.org/wiki/Crystal_Clear) licença (LGPL), disponível na url http://commons.wikimedia.org/wiki/Crystal_Clear. Os ícones deverão ter o tamanho de 70x70 mm. Os ícones que serão mais utilizadas estão listadas abaixo:

Ícones rápidos		
Ícone	Indicação <Alt>	Indicação
	Página Inicial	Este ícone será apresentado: <ul style="list-style-type: none"> No cabeçalho.
	Ajuda	Este ícone será apresentado: <ul style="list-style-type: none"> No cabeçalho.
	Fale Conosco	Este ícone será apresentado: <ul style="list-style-type: none"> No cabeçalho.
	Trocar Perfil	Este ícone será apresentado: <ul style="list-style-type: none"> No cabeçalho.
	Sair	Este ícone será apresentado: <ul style="list-style-type: none"> No cabeçalho.

Ícones informativos		
Ícone	Indicação <Alt>	Indicação
	Erro! O <i>hint</i> deste ícone dependerá do erro ocorrido no campo.	Este ícone será apresentado: <ul style="list-style-type: none"> Na mensagem onde o preenchimento não estiver correto.
	Alerta	Este ícone será apresentado: <ul style="list-style-type: none"> Nas mensagens de alerta
	Ajuda no campo. O <i>hint</i> desse ícone dependerá dos dados necessários para o preenchimento de determinado campo.	Este ícone será apresentado: <ul style="list-style-type: none"> Do lado direito dos campos, com texto de ajuda para preenchimento do campo.
	Informação no campo. O <i>hint</i> desse ícone dependerá dos dados necessários para o preenchimento de determinado campo.	Este ícone será apresentado: <ul style="list-style-type: none"> Para complementar informações do campo a fim de facilitar o preenchimento.



Ícones/botões de ações		
Ícone	Indicação <Alt>	Ação
	Inserir	Aciona o fluxo de inserção de um novo registro.
	Alterar	Aciona o fluxo de alteração de um registro.
	Excluir	Exclui o registro indicado.
	Cancelar	Cancela o fluxo acionado.
	Validar	Validar.
	Visualizar	Visualiza o registro indicado.
	Inativar/Bloquear	Inativa/bloqueia o registro indicado.
	Ativar/Desbloquear	Ativa/desbloqueia o registro indicado.
	Atualizar	Atualizar registro selecionado.
	Buscar	Executa a busca com os parâmetros informados.
	Operação realizada com sucesso	Aparece quando a operação é realizada com sucesso.

3.7 Palheta de cores

Azul claro	Azul médio	Azul escuro	Cinza
R - 201 G - 223 B - 253	R - 0 G - 132 B - 209	R - 0 G - 0 B - 128	R - 225 G - 233 B - 244
#C9DFFD	#0084D1	#000080	#E1E9F4



4. Considerações gerais

4.1 Redimensionamento da tela

Os *layouts* utilizados para o sistema usarão as funcionalidades nativas da biblioteca JavaScript, que constrói as telas. Esses *layouts* devem ser dinâmicos, ou seja, devem se adaptar ao redimensionamento da tela e reorganizar automaticamente os componentes apresentados, seja em computadores ou dispositivos móveis (tablets, celulares, PDA's)

4.2 Utilização de temas

As customizações da aparência das telas devem ser feitas utilizando-se os recursos de temas da biblioteca. Esses temas devem conter folhas de estilo CSS (*Cascading Style Sheets*) e imagens. A vantagem de se usarem temas é que a identidade visual do órgão se mantém e pode ser replicada facilmente de um sistema para outro.

As folhas de estilos que não forem nativas da *biblioteca de componentes* deverão ser nomeadas de forma a identificar sua função na estrutura do sistema.

A nomenclatura das folhas de estilo deverá seguir o padrão: "sistema_<Identificação>.css".

4.3 Código

O código do sistema não deverá conter características aplicadas diretamente a ele. Sempre que for possível, atribuir as características por meio das folhas de estilos.

O código deverá apresentar endentação padrão, de forma a facilitar o entendimento da hierarquia utilizada na construção.

Comentários deverão ser utilizados somente para identificação das áreas da página, de forma a facilitar a localização dos elementos.

Os componentes criados (*pop-ups*, *grids*, campos de formulários...) devem possuir identificadores para permitir a automação de testes.

O padrão de codificação de sistemas do Ibama deve ser seguido.

4.4 CSS (*Cascade Style Sheet*)

Com o objetivo de garantir a flexibilidade da formatação do sistema, não devem ser atribuídas características diretas ao código, devendo ser atribuídas as características às tags associadas às divs, utilizando-se, para isso, o padrão cascata na atribuição das características.

4.5 Resolução de vídeo

Os sistemas atuais deverão ter seu leiaute de, no mínimo, 1280 x 720 *pixels*.



4.6 Navegadores

O sistema deverá ter suas características visuais e funcionais mantidas nas versões atualizadas e estáveis dos browser Microsoft Internet Explorer, Mozilla Firefox e Google Chrome.

5. Referências

W3C "Checklist of Checkpoints for Web Content Accessibility Guidelines 1.0", publicado no site do W3C (<http://www.w3.org/>).

Diretrizes de Usabilidade V2.1. Empresa Basis Tecnologia da Informação Ltda.

Site do governo eletrônico (<http://www.governoeletronico.gov.br/acoes-e-projetos/e-MAG>).

Site do governo eletrônico (<http://www.governoeletronico.gov.br/acoes-e-projetos/padrees-brasil-e-gov>).

ANEXO Y DO TERMO DE REFERÊNCIA

**PROCESSO DE GERENCIAMENTO E
CONFIGURAÇÃO DE ATIVOS DE
TECNOLOGIA DA INFORMAÇÃO**

Versão 1.0



Ministério do Meio Ambiente
Instituto Brasileiro do Meio Ambiente e dos Recursos Naturais Renováveis
Diretoria de Planejamento, Administração e Logística
Centro Nacional de Telemática

PROCESSO DE GERENCIAMENTO DE CONFIGURAÇÃO DE ATIVOS DE TECNOLOGIA DA INFORMAÇÃO

Versão 1.0

- 2013 -



Sumário

1. Introdução.....	3
2. O processo de Gerenciamento de Configuração.....	4
2.1. Gerenciamento e Planejamento	4
2.1.1. Atualizar a Política e a Metodologia de Gerenciamento da Configuração	4
2.2. Identificação da Configuração	4
2.2.1. Avaliar e registrar elementos incorporados à infraestrutura da Rede IBAMA	4
2.3. Controle de Configuração	7
2.3.1. Acompanhar alterações nos registros de itens de configuração	7
2.3.2. Atualizar registros de itens de configuração	9
2.4. Verificação da Configuração	9
2.4.1. Monitorar e avaliar alarmes das ferramentas de Gerenciamento de Eventos	9
2.4.2. Monitorar e avaliar novos elementos de infraestrutura encontrados	11
2.4.3. Emitir relatórios sobre os itens de configuração gerenciados	12



1. Introdução

O Gerenciamento de Configuração é um processo cujo principal objetivo é manter controle sobre os ativos de tecnologia da informação de uma organização, prioritariamente sobre aqueles mais vitais para a organização cumprir suas funções institucionais. Manter controle pode ser entendido como manter registros consistentes de atributos desses componentes, além de descobrir e documentar os relacionamentos entre esses ativos da organização.

A gerência de configuração é extremamente importante, é por isso que diversos modelos de qualidade e maturidade como ISO, CMMI, MPS.Br e SPICE dão tanto valor à esta disciplina. Na busca de um maior controle sobre os ativos de tecnologia existente no Ibama, este processo de gerenciamento de configuração de ativos foi elaborado como um modelo inicial que deve ser revisto e melhorado de forma contínua conforme a necessidade.



2. O processo de Gerenciamento de Configuração

O processo e os procedimentos de Gerenciamento de Configuração, abreviado GC, utilizados pela Coordenação do Centro Nacional de Telemática - CNT, foram organizados por área de atividade e são listados a seguir:

- Gerenciamento e Planejamento;
- Identificação da Configuração;
- Controle de Configuração; e
- Verificação da Configuração.

2.1. Gerenciamento e Planejamento

O objetivo desse subprocesso do GC é planejar o nível de gerenciamento de configuração desejado e como ele será obtido.

2.1.1. Atualizar a Política e a Metodologia de Gerenciamento da Configuração

A política e a metodologia de gerenciamento da configuração são os principais mecanismos de planejamento. Tais documentos devem ser atualizados com a periodicidade mínima de 1 (um) ano, garantindo que eles estarão sempre aderentes à realidade.

2.2. Identificação da Configuração

O objetivo principal do grupo de atividades de “Identificação da Configuração” é definir como os novos elementos de infraestrutura devem ser tratados no Gerenciamento de Configuração.

2.2.1. Avaliar e registrar elementos incorporados à infraestrutura da Rede IBAMA

Os novos elementos que surgirem na infraestrutura da Rede IBAMA deverão ser avaliados pelo Gerente de Configuração, para decidir se eles representam novos itens de configuração. Essa avaliação irá definir se o novo elemento se encaixa nos critérios constantes na Seção III da “Política de Gerenciamento de Configuração”, que define quais elementos da infraestrutura devem participar

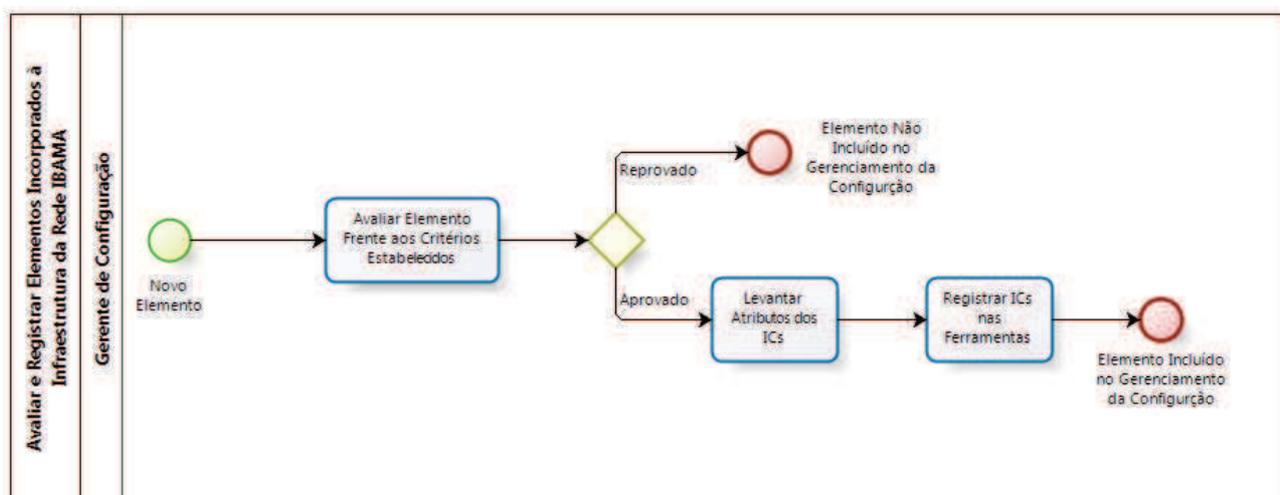
do gerenciamento de configuração.

O Gerente de Configuração também deve considerar nessa avaliação os tipos de componentes que estão sob controle, definidos conforme o inciso II da Seção III da “Política de Gerenciamento de Configuração de Ativos Componentes de Tecnologia da Informação”.

Se o resultado dessa avaliação for positivo, o Gerente de Configuração deverá registrar o novo elemento de rede como um novo item de configuração (IC). Os registros realizados até o momento estão organizados em diferentes sistemas e documentos, de acordo com a tabela constante no Quadro II – “Registros do Gerenciamento de Configuração” deste documento.

O Gerente de Configuração deverá levantar e registrar os dados apropriados desse novo elemento nas ferramentas de gerenciamento de redes e de configuração (vide Anexo III – “Ferramentas”).

A figura abaixo sintetiza o processo de avaliar e registrar elementos novos incorporados à infraestrutura da Rede do Ibama:





2.3. Controle de Configuração

O principal objetivo dessa área de atividades é detectar e manter registros de mudanças feitas nos itens de configuração. Essas mudanças incluem modificações de versões de softwares, mudanças de localização ou proprietário do ativo, entre outros aspectos que forem úteis. Sem esse controle, os dados que constam no sistema de gerenciamento de configuração ficarão desatualizados com o passar do tempo.

2.3.1. Acompanhar alterações nos registros de itens de configuração

O Gerente de Configuração deverá acompanhar as alterações realizadas nos itens de configuração gerenciados e garantir que será mantido um histórico dessas modificações.

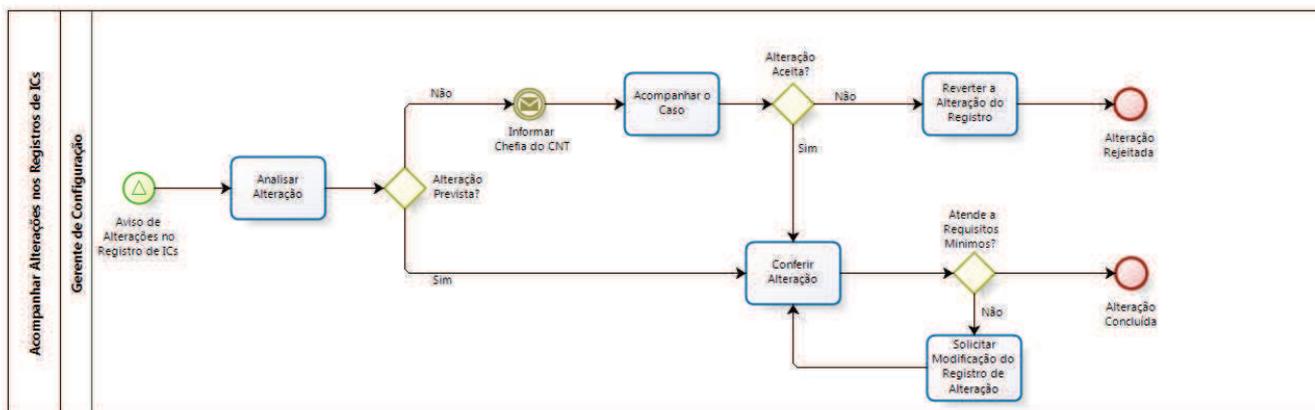
Os itens de configuração mantidos nos sistemas de gerenciamento de configuração deverão estar sob esse controle de alterações. Toda vez que se alterar uma informação nos sistemas de gerenciamento de configuração, a modificação deverá ser registrada no sistema, e a antiga versão do registro deverá ser armazenada em um histórico, para permitir a comparação com a nova. Os sistemas de gerenciamento de configuração deverão ser configurados para emitir notificações sempre que uma alteração for realizada em algum registro de item de configuração. É responsabilidade do Gerente de Configuração assegurar-se de que ele será notificado sobre alterações nos registros desses sistemas.

Os sistemas de gerenciamento de eventos também deverão enviar notificações aos responsáveis, informando sobre alterações nos registros dos itens de configuração. É obrigação do Gerente de Configuração assegurar-se de que ele será alertado sobre modificações nesses registros.

No caso de alterações não previstas nos registros, o Gerente de Configuração deverá alertar à Chefia do CNT e acompanhar o caso, revertendo a alteração do registro, se necessário. Se a alteração for prevista, deverá conferir se está de acordo com os requisitos mínimos registrados nas ferramentas utilizadas para o gerenciamento de configuração, conforme Quadro II. Se não atender os requisitos, o Gerente de Configuração deve solicitar ao responsável pela alteração que deixe sua

modificação nos registros dentro do esperado. Após corrigida, a alteração nos registros será aceita pelo Gerente de Configuração.

Na figura abaixo, o fluxo de trabalho que representa essa atividade:



2.3.2. Atualizar registros de itens de configuração

Os elementos sob gerenciamento de configuração que fizeram parte de uma mudança programada pelo CNT, devem ter seus registros atualizados nos sistemas utilizados pelo gerenciamento de configuração. É responsabilidade do Gerente de Configuração informar-se sobre as Solicitações de Mudanças em andamento e atualizar os devidos registros de itens de configuração para manter os dados de configuração consistentes com as mudanças.

2.4. Verificação da Configuração

Esse grupo de atividades tem o objetivo de garantir que exista conformidade entre os itens de configuração documentados e os itens de configurações reais.

2.4.1. Monitorar e avaliar alarmes das ferramentas de Gerenciamento de Eventos

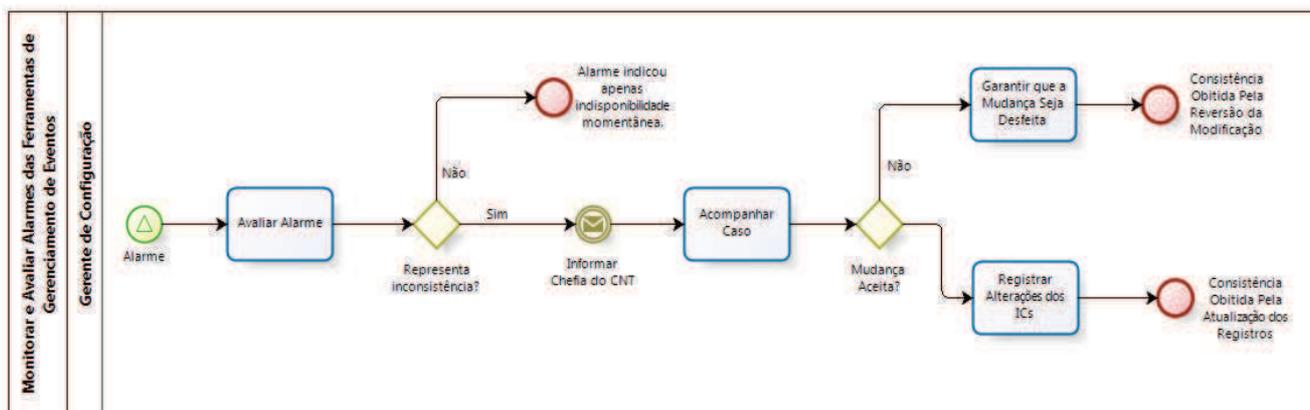
As ferramentas de gerenciamento de eventos serão utilizadas para atividades de verificação da configuração. Os alarmes emitidos por essas ferramentas podem sinalizar inconsistências entre o que está registrado e o que existe na realidade.



As notificações que estiverem relacionadas à ausência de resposta de um item de configuração devem ser verificadas pelo Gerente de Configuração, que concluirá se as alterações comunicadas pelo alarme representam uma modificação de configuração do elemento, ou apenas uma parada momentânea de operação.

Caso o Gerente de Configuração avalie que o alarme trata-se de uma inconsistência, deverá comunicar o fato à Chefia do CNT e acompanhar o caso, para constatar se a modificação na infraestrutura deve ser aceita, ou não. Se for aceita, o Gerente de Configuração deve alterar o registro dos itens de configuração modificados, tornando-os consistentes com a realidade. Caso contrário, deve-se garantir que a modificação na infraestrutura seja desfeita, seguindo a configuração registrada pelo gerenciamento de configuração.

O procedimento “Monitorar e avaliar alarmes das ferramentas de Gerenciamento de Eventos” é representado no fluxo de atividades abaixo:



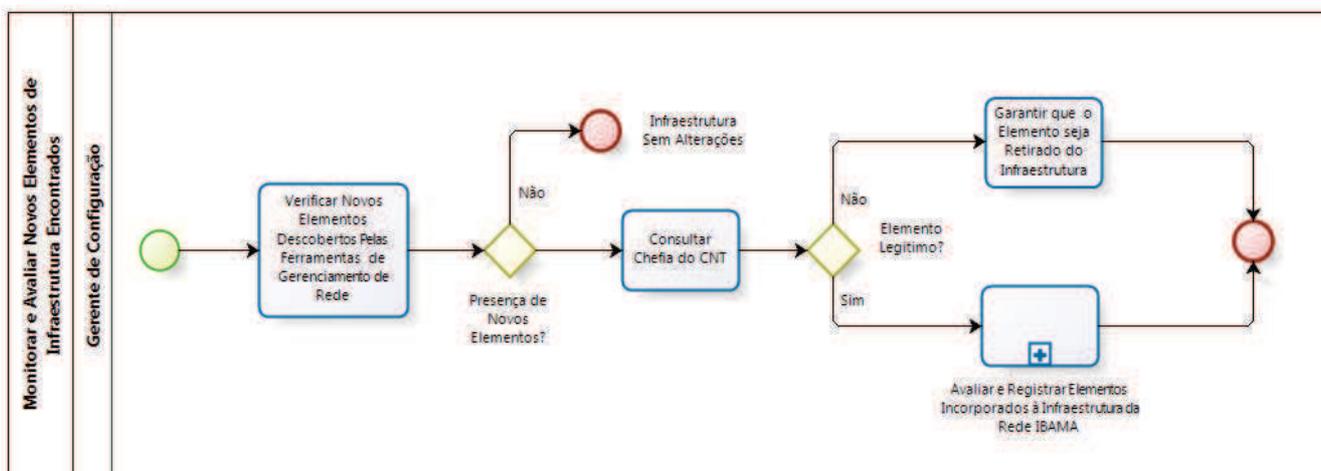


2.4.2. Monitorar e avaliar novos elementos de infraestrutura encontrados

É tarefa do Gerente de Configuração verificar se não existem elementos indevidos presentes na infraestrutura de produção.

No caso da detecção de novos elementos, o Gerente de Configuração deve informar-se sobre a sua legitimidade com a Chefia do CNT. Se o elemento for legítimo, deve adotar o procedimento de “Avaliar e registrar elementos incorporados à infraestrutura da Rede Ibama”, para adequar os registros de itens de configuração. Se o elemento não for autorizado a integrar a infraestrutura de rede, o Gerente de Configuração deve garantir que o mesmo seja retirado desse ambiente.

O procedimento “monitorar e avaliar novos elementos de infraestrutura encontrados” é representado no fluxo de atividades abaixo:





2.4.3. Emitir relatórios sobre os itens de configuração gerenciados

O Gerente de Configuração deverá emitir relatórios sobre os itens de configuração cadastrados nas ferramentas de gerenciamento de rede, para subsidiar as atividades de verificação da configuração. Para esse mesmo fim, também deve emitir relatórios sobre os registros de itens de configuração constantes nas ferramentas de gerenciamento de configuração.



Quadro I - Registros do Gerenciamento de Configuração

Sistemas e documentos que armazenam registros do Gerenciamento de Configuração:

Lista de Ativos	Relação de todos os ativos existentes no Ibama.
Matriz Serviços x Servidores.	Identifica que serviços são prestados e em qual equipamento encontra-se. Pode ser acessado em G:\comum\cnt\cntadm\Metodologia&Padroes\gerenciamento_configuracao\Docs
Lista de Procedimentos/Serviços do CNT.	Contém os procedimentos/serviços disponibilizados pelo CNT. Localização: “G:\cnt\cntadm\Metodologia&Padroes\Gerenciamento_de_Configuracao\Docs\Lista_de_Servicos_do_IBAMA”.
Relação de Setores por Prédio - IBAMA X Pilhas de Swithes.	Levantamento de quais setores são atendidos por quais switches. Localização: “G:\cnt\cntadm\Metodologia&Padroes\Gerenciamento_de_Configuracao\Docs\Relacao_de_Setores_por_Predio_IBAMA_Pilhas_de_Switches”
Configurações de Impressoras.	Localização das impressoras do IBAMA e potência de energia requerida por elas. Localização: “G:\cnt\cntadm\Metodologia&Padroes\Gerenciamento_de_Configuracao\Docs\Configuracoes_de_Impressoras”
Documentação do VNC na sede.	Software VNC utilizado para acesso remoto “G:\cnt\cntadm\Metodologia&Padroes\Gerenciamento_de_Configuracao\Docs\Acesso_remoto.odt”.
Configurações de Switches.	Nessa pasta estão planilhas e arquivos que concentram dados de configuração dos switches. Localização: “G:\cnt\cntadm\Metodologia&Padroes\Gerenciamento_de_Configuracao\Docs\Configuracoes_de_Switches”
Configurações de Access Points.	Essa pasta contém arquivos com dados de configuração dos pontos de acesso sem-fio da rede do IBAMA. Localização: “G:\cnt\cntadm\Metodologia&Padroes\Gerenciamento_de_Configuracao\Docs\Configuracoes_de_Access_Points”
Documentação Pública sobre o gerenciamento de configuração de ativos de tecnologia da informação.	WIKi – http://wikibama.ibama.gov.br



Quadro II – Ferramentas

Ferramentas utilizadas pelo CNT no gerenciamento de configuração:

Sistema Zabbix. Gerenciamento de Rede (Eventos / Incidentes)	Software que monitora diversos parâmetros de uma rede como a integridade e desempenho dos servidores. Oferece excelentes relatórios e visualização de dados de recursos com base nos dados armazenados, e usa um mecanismo de notificação que permite aos usuários configurar e-mail com alertas para qualquer evento, o que permite uma reação rápida para os problemas do servidor. Pode ser acessado por meio da URL - http://10.1.25.15/zabbix/
Sistema OCS.	OCS Inventory NG é uma aplicação desenvolvida para apoiar os administradores de TI a gerenciar seus parques computacionais. Com o OCS Inventory NG é possível conhecer precisamente todos os componentes de hardware e software que estão instalados em cada computador de sua rede. Graças a sua arquitetura baseada no conceito de agentes, a coleta de informações e a consequente alimentação de dados é realizada totalmente de forma automatizada e sem intervenção humana. Pode ser acessado por meio da URL - http://guara.ibama.gov.br/ocsreports/
openNMS	OpenNMS é um projeto open-source (código aberto) dedicado à criação de uma plataforma de gerência de rede voltada principalmente para a camada de aplicação;

ANEXO Z DO TERMO DE REFERÊNCIA

**PROCESSO DE GERENCIAMENTO
DE MUDANÇAS**

Versão 1.0



Ministério do Meio Ambiente
Instituto Brasileiro do Meio Ambiente e dos Recursos Naturais Renováveis
Diretoria de Planejamento, Administração e Logística
Centro Nacional de Telemática

PROCESSO DE GERENCIAMENTO DE MUDANÇAS

Versão 1.0

- 2013 -



Sumário

1. Introdução.....	3
2. O Processo de Gerenciamento de Mudanças.....	4
2.1. Implantação/atualização de sistemas de informação.....	5
2.2. Atualização de objetos em nível de banco de dados.....	5
2.3. Extração de informações em nível de banco de dados.....	6
2.4. Intervenção em ativos ou serviços de rede	7



1. Introdução

São processos, ferramentas e técnicas para gerenciar os vários aspectos envolvidos em um processo de mudança a fim que os resultados previstos sejam atingidos e da forma mais eficaz possível.

Este documento visa padronizar os procedimentos de gerenciamento de intervenção em ambiente de produção de tecnologia da informação, no âmbito do Centro Nacional de Telemática - CNT.



2. O Processo de Gerenciamento de Mudanças

O processo de Gerenciamento de Mudanças em ambiente de produção faz-se necessária para a implantação/atualização de sistemas de informação e atualização de objetos em nível de banco de dados.

Para fins dos procedimentos considera-se:

- a) Ambiente de produção: estrutura computacional composta por hardware e software destinada a manter sistemas e serviços para os usuários.
- b) Ambiente de homologação: estrutura computacional composta por hardware e software destinada a simulação de funcionalidades de sistemas e serviços a serem implantados em ambiente de produção.
- c) Intervenção em ambiente de produção: a implantação e atualização de sistemas de informação, alteração de objetos em banco de dados, alteração de configuração de serviços e aplicativos em hardwares servidores de aplicação e de dados.
- d) Termo de Recebimento Definitivo: declaração formal, por parte do requisitante, de que as funcionalidades desenvolvidas atendem aos requisitos estabelecidos e foram homologadas.
- e) Solicitação de Mudança (SM): requisição de alteração em elementos de software ou hardware do ambiente de produção autorizado pelo gestor do sistema, com a aquiescência da autoridade máxima da diretoria requisitante. A solicitação de mudança deve conter no mínimo:
 - Área Requisitante
 - Nome do Servidor Requisitante
 - Descrição detalhada da mudança
 - Justificativa em termos de impactos
- f) Plano de Implantação: documento previsto na Metodologia de Desenvolvimento de Sistemas (MDS) do Ibama que deverá conter no mínimo:
 - Identificador único ou número da Ordem de Serviço;
 - Tipo de intervenção (Ordem de Serviço ou Ordem de serviço Corretiva);
 - Nome ou Sigla do Sistema;



- Orientações para implantação, contendo, no mínimo, o número sequencial da ordem de execução e a descrição do procedimento de implantação, contando do comando de publicação a ser executado na ferramenta de versionamento.
- Demais considerações, tais como: restrições de horário e relação de impacto, a exemplo da verificação de objetos inválidos.

2.1. Implantação/atualização de sistemas de informação

Para o gerenciamento de mudanças para as implantações/atualizações de sistemas de informação deve-se executar as seguintes atividades:

- a) A área requisitante deve homologar a funcionalidade desenvolvida em ambiente de homologação e, havendo conformidade com os requisitos estabelecidos, deve-se encaminhar à secretaria do CNT o Termo de Recebimento Definitivo devidamente assinado.
- b) A fábrica de software deverá registrar as informações do Plano de Implantação em assentamento próprio no chamado, via sistema de gestão de demandas do Ibama, anexando o Termo de Recebimento Definitivo, direcionado ao núcleo de banco de dados ou à área de rede corporativa, caso haja necessidade de intervenção na configuração de ativos ou serviços de rede.
- c) A área de banco de dados ou de rede corporativa deverá verificar as informações do Plano de Implantação e proceder com a implantação no ambiente de produção.
- d) Após a execução do processo de implantação em produção, a área de banco de dados ou de rede corporativa deverá registrar em assentamento próprio o resultado dessa execução e atualizar o inventário de ativos.
- e) A fábrica de software, de posse do resultado positivo da implantação, deverá atualizar o sistema de gestão de demanda, ou em caso negativo deverá promover os ajustes necessários e reencaminhar o chamado ao núcleo de banco de dados ou de rede corporativa.

2.2. Atualização de objetos em nível de banco de dados

Para o gerenciamento de mudanças de atualização de objetos em nível de banco de dados



deve-se executar as seguintes atividades:

- a) A área requisitante deverá abrir um chamado no sistema de gestão de demandas do Ibama e registrar em assentamento próprio as informações da Solicitação de mudança e encaminhá-las ao núcleo de banco de dados, anexando o documento comprovando a aquiescência da autoridade máxima da diretoria requisitante.
- b) O núcleo de banco de dados verificará a autorização do requisitante para a execução da mudança, bem como a sua viabilidade.
- c) Havendo questionamentos ou obstáculos, o núcleo de banco de dados deverá encaminhar o chamado à gerência de liberação para as devidas providências administrativas.
- d) Caso a execução da solicitação seja viável, a gerência de liberação encaminhará ao núcleo de banco de dados para implementação, informando-se as medidas administrativa que ensejaram sua viabilidade, inclusive eventuais condições e dados complementares, e iniciando-se o prazo para execução da solicitação a partir deste encaminhamento;
- e) Caso a execução da demanda seja inviável, a gerência de liberação cancelará a solicitação, registrando-se os motivos.
- f) Após execução do chamado, o núcleo de banco de dados deverá atualizar o inventário de ativos e informar os resultados no sistema de gestão de demandas, ou encaminhá-los ao requisitante em caso de informação sensível.

2.3. Extração de informações em nível de banco de dados

Para a extração de informações em nível de banco de dados deve-se executar as seguintes atividades:

- a) A área requisitante deverá abrir um chamado no sistema de gestão de demandas do Ibama e registrar em assentamento próprio as informações a serem extraídas.
- b) O núcleo de banco de dados verificará a autorização do requisitante para acesso as informações solicitadas.
- c) Havendo questionamentos ou obstáculos, o núcleo de banco de dados deverá encaminhar o



chamado à gerência de liberação para as devidas providências administrativas.

d) Caso a execução da solicitação seja viável, a gerência de liberação encaminhará ao núcleo de banco de dados para implementação, informando-se as medidas administrativa que ensejaram sua viabilidade, inclusive eventuais condições e dados complementares, e iniciando-se o prazo para execução da solicitação a partir deste encaminhamento;

e) Caso a execução da demanda seja inviável, a gerência de liberação cancelará a solicitação, registrando-se os motivos.

f) Após execução do chamado a área de banco de dados deverá encaminhar à área requisitante as informações solicitadas.

2.4. Intervenção em ativos ou serviços de rede

Para o gerenciamento de mudanças de intervenção em ativos ou serviços de rede em ambiente de produção deve-se executar as seguintes atividades:

a) A área requisitante deverá abrir um chamado no sistema de gestão de demandas do Ibama e registrar em assentamento próprio as informações da Solicitação de mudança e encaminhá-las ao núcleo de rede corporativa,

b) O núcleo de rede corporativa verificará a autorização do requisitante para a execução da mudança, bem como a sua viabilidade.

c) Havendo questionamentos ou obstáculos, o núcleo de rede corporativa deverá encaminhar o chamado à gerência de liberação para as devidas providências administrativas.

d) Caso a execução da solicitação seja viável, a gerência de liberação encaminhará ao núcleo de rede corporativa para implementação, informando-se as medidas administrativa que ensejaram sua viabilidade, inclusive eventuais condições e dados complementares, e iniciando-se o prazo para execução da solicitação a partir deste encaminhamento;

e) Caso a execução da demanda seja inviável, a gerência de liberação cancelará a solicitação, registrando-se os motivos.

f) Após execução do chamado, o núcleo de rede corporativa deverá informar os resultados no

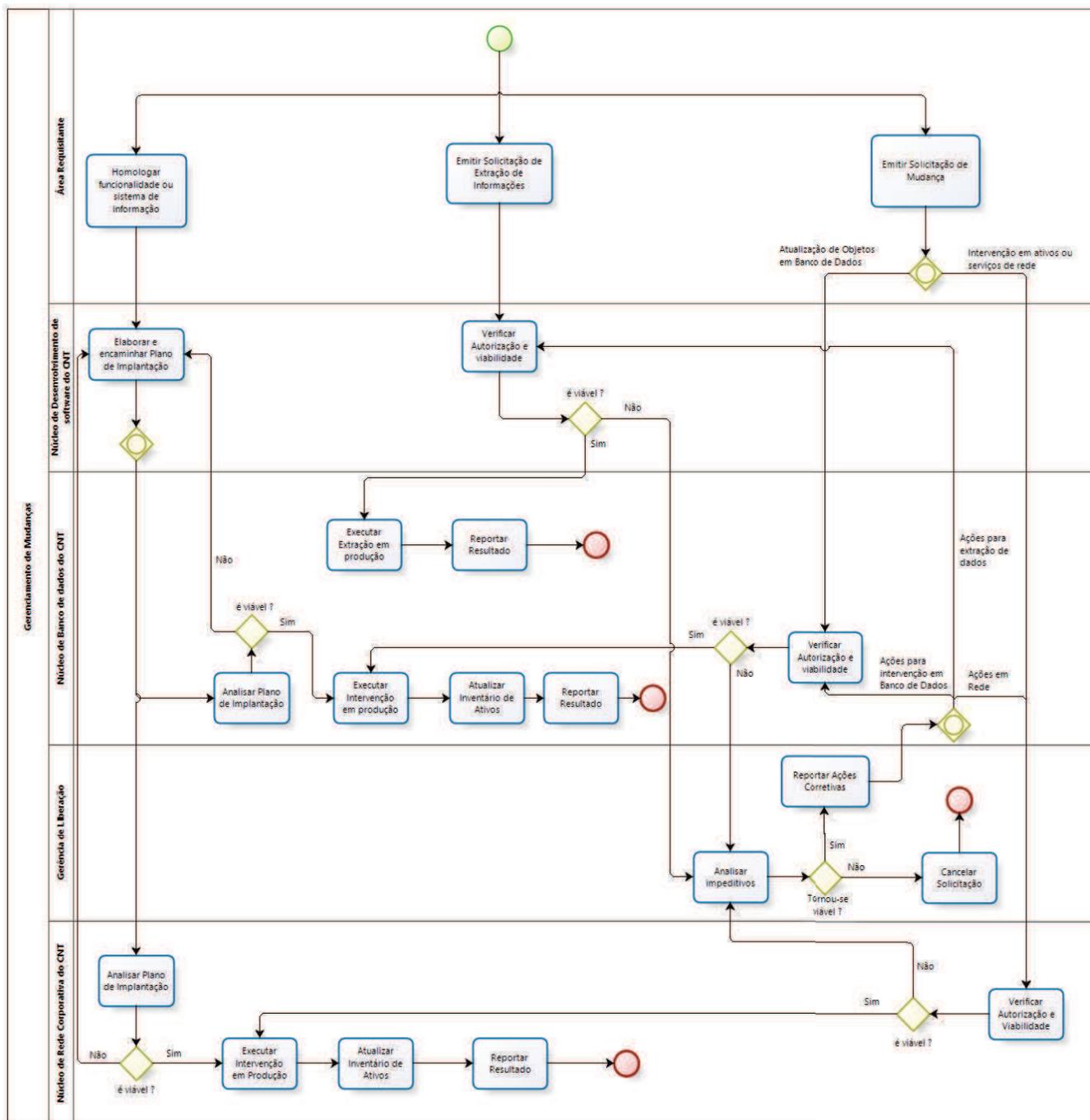


Ministério do Meio Ambiente
Instituto Brasileiro do Meio Ambiente e dos Recursos Naturais Renováveis
Diretoria de Planejamento, Administração e Logística
Centro Nacional de Telemática

sistema de gestão de demandas, ou encaminhá-los ao requisitante em caso de informação sensível.

g) O núcleo de rede corporativa deverá atualizar o inventário de ativos após a execução da mudança.

Segue abaixo o processo de Gerenciamento de Mudanças em ambiente de produção de tecnologia da informação do Ibama:



ANEXO AA DO TERMO DE REFERÊNCIA

**ROTEIRO DE MÉTRICAS DE
SOFTWARE DO IBAMA**

Versão 1.0



ROTEIRO DE MÉTRICAS DE SOFTWARE DO IBAMA

**Instituto Brasileiro de Meio Ambiente e dos
Recursos Naturais Renováveis – IBAMA
Planejamento da Contratação de
Tecnologia da Informação**

Versão 1.0

Histórico da Revisão

Data	Versão	Descrição	Autor
07/12/15	1.0	Criação do documento	José Augusto Luís de Carvalho Loidianne Alves Marinho Márcio Pereira Lima

Sumário

1. Introdução.....	5
2. Objetivo.....	7
3. Contagem de Pontos de Função.....	7
3.1 Determinar Propósito, Tipo e Escopo da Contagem e Fronteira da Aplicação.....	8
3.2 Identificar Funções de Dados e Funções Transacionais.....	9
3.3 Calcular Tamanho Funcional.....	10
4. Cálculo de Pontos de Função para o Ibama.....	11
4.1 Projeto de Desenvolvimento.....	12
4.2 Projeto de Melhoria.....	12
4.3 Projeto de Migração de Dados.....	15
4.4 Manutenção Corretiva.....	16
4.5 Mudança de Plataforma.....	17
4.5.1 Mudança de Plataforma – Linguagem de Programação.....	17
4.5.2 Mudança de Plataforma – Banco de Dados.....	18
4.6 Atualização de Versão.....	19
4.6.1 Atualização de Versão – Linguagem de Programação.....	19
4.6.2 Atualização de Versão – <i>Browser</i>	20
4.6.3 Atualização de Versão – Banco de Dados.....	20
4.7 Manutenção em Interface.....	20
4.8 Adaptação em Funcionalidades sem Alteração de Requisitos Funcionais.....	22
4.9 Apuração Especial.....	23
4.9.1 Apuração Especial – Base de Dados.....	23
4.9.2 Apuração Especial – Geração de Relatórios.....	25
4.9.3 Apuração Especial – Reexecução.....	25
4.10 Atualização de Dados.....	25
4.11 Manutenção de menus, telas e listas de seleção estáticas Desenvolvimento, Manutenção e Publicação de Páginas Estáticas de <i>Intranet</i> , <i>Internet</i> ou Portal.....	26
4.12 Manutenção de Documentação de Sistemas Legados.....	27
4.13 Verificação de Erros.....	27
4.14 Pontos de Função de Teste.....	28
4.15 Componente Interno Reusável.....	29
5. Orientações Complementares para Contagem.....	30
5.1 Divisão de um mesmo Processo Elementar em Transações Distintas.....	30
5.2 Contagem de Pontos de Função com Múltiplas Mídias.....	31
5.2.1 Cenário 1: Mesmos dados apresentados em tela e impressos.....	32
5.2.2 Cenário 2: Mesmos dados de saída como dados em arquivo e relatório impresso.....	33
5.2.3 Cenário 3: Mesmos dados de entrada batch e on-line.....	33
5.2.4 Cenário 4: Múltiplos canais de entrega da mesma funcionalidade.....	34
5.2.5 Cenário 5: Relatório em múltiplos formatos.....	34
5.3 Aferição de Pontos de Função em ambiente BPMS.....	35
5.3.1 Fronteira de Aplicação.....	36
5.3.2 Critérios de Contagem.....	36
5.4 Aferição de Pontos de Função em ambiente de Apoio à Decisão (Business Intelligence e Dataware House).....	38
5.4.1 Fronteira de Aplicação.....	38
5.4.2 Critérios de Contagem.....	39
5.5 Aferição de Pontos de Função em Ambiente Baseado em Geotecnologias.....	40
5.5.1 Critérios de Contagem.....	41

6. Considerações Especiais para Planejamento e Acompanhamento de Projetos.....	41
6.1 Diretrizes para Planejamento: Estimativas de Projetos de Software.....	41
6.1.1 Contagem Estimativa de Pontos de Função (CEPF).....	43
6.1.2 Distribuição de Esforço por Fase do Projeto.....	48
6.1.3 Estimativa de Prazo de Projetos de Software.....	48
6.2 Diretrizes para Acompanhamento de Projetos.....	50
6.2.1 Considerações sobre Mudança de Requisitos.....	51
6.2.2 Considerações sobre Projetos Cancelados.....	51
6.2.3 Considerações sobre Redução de Cronograma.....	52
7. Atividades Sem Contagem de Pontos de Função.....	54
8. Processo de Revisão do Roteiro de Contagem.....	55
8.1 Revisão para Correção de Inconsistências e Situações não Previstas.....	55
8.2 Revisão para Adoção de Novas Versões do CPM e do Roteiro de Métricas de Software do SISP.....	55
9. Dicas para otimizar o custo das manutenções.....	56
10. Conclusão.....	56
11. Referencias Bibliográficas.....	57
Orientações para preenchimento da Planilha de Contagem.....	58
ANEXO 1 – Identificação da Planilha de Contagem de Ponto de Função.....	58
ANEXO 2 – Planilha de Contagem de Ponto de Função.....	59
ANEXO 3 – Fundamentação da Contagem de Ponto de Função.....	61

Índice de figuras

Figura 1: Procedimento de Contagem de Pontos de Função.....	9
Figura 2: Processo de Estimativas de Projetos de Software [Hazan, 2008].....	43
Figura 3: Modelo Lógico da Análise de Pontos de Função.....	47
Figura 4: Relação entre a Estimativa de Prazo e de Esforço.....	53

Índice de tabelas

Tabela 1: Contribuição Funcional dos Tipos Funcionais (Fonte: CPM 4.3).....	11
Tabela 2: Identificação dos Arquivos Lógicos Internos da Aplicação.....	48
Tabela 3: Identificação dos Arquivos de Interface Externa da Aplicação.....	48
Tabela 4: Identificação das Entradas Externas da Aplicação.....	49
Tabela 5: Identificação das Consultas Externas da Aplicação.....	49
Tabela 6: Identificação das Saídas Externas da Aplicação.....	50
Tabela 7: Distribuição de Esforço por Macroatividades do Projeto.....	52
Tabela 8: Expoente t por Tipo de Projeto.....	53
Tabela 9: Estimativa de Prazo de Projetos menores que 100 PF.....	54
Tabela 10: Percentuais definidos para a mudança de requisitos.....	59
Tabela 11: Percentuais de redução de prazo.....	67

1. Introdução

O Instituto Brasileiro do Meio Ambiente e dos Recursos Naturais Renováveis – IBAMA – têm utilizado a métrica Ponto de Função (PF) nas estimativas e dimensionamento de tamanho funcional de projetos de software devido aos diversos benefícios de utilização desta métrica, destacando-se: regras de contagem objetivas, independência da solução tecnológica utilizada e facilidade de estimativa nas fases iniciais do ciclo de vida do software.

Por permitir a medição objetiva de serviços de desenvolvimento de soluções de software, a utilização da métrica Pontos de Função é uma boa prática na contratação de serviços e está aderente ao estabelecido na Instrução Normativa nº 04/2014 SLTI/MP, conforme dispõe a Portaria nº 31 SLTI/MP nº 31, de 29 novembro de 2010.

Esta Portaria, no seu Art. 3º, também recomenda que os órgãos integrantes do Sistema de Administração dos Recursos de Informação e Informática (SISP), adotem o roteiro de contagem nas suas contratações de serviços de desenvolvimento e manutenção de soluções de software, consubstanciando-se então o objetivo deste Roteiro.

O Manual de Práticas de Contagem de Pontos de Função (CPM 4.3.1) [IFPUG, 2010], publicado pelo *International Function Point Users Group* (IFPUG), define as regras de contagem de pontos de função. É importante ressaltar que a métrica Ponto de Função foi concebida como uma medida de tamanho funcional para projetos de desenvolvimento e de melhoria (manutenção evolutiva) de software.

Além disso, a contagem de pontos de função é baseada no projeto lógico da aplicação (logical design). Nas fases iniciais do ciclo de vida do software, o insumo para a definição das estimativas do projeto é um documento inicial de requisitos, por exemplo: documento de visão ou algum outro tipo de especificação elaborada pelo analista de negócios. Assim, torna-se importante o estabelecimento de métodos para estimar o tamanho dos projetos de software nas fases iniciais do ciclo de vida.

Outro ponto a ser destacado é a importância da definição de métodos para geração de estimativas de prazo e custo dos projetos de software a partir do tamanho funcional estimado do projeto. Considerando que o objetivo principal do CPM não é esse, o Roteiro de Métricas de Software do Ibama foi criado para contemplar questões não abordadas pelo manual do IFPUG, mas vivenciadas pelo Instituto, como a utilização da métrica em contratos administrativos.

O restante deste documento encontra-se organizado da seguinte forma:

- o capítulo 2 descreve os objetivos e as referências consultadas para a elaboração deste documento;
- o capítulo 3 apresenta algumas definições básicas para a contagem de pontos de função;
- o capítulo 4 define métricas baseadas em Ponto de Função para dimensionar projetos de desenvolvimento e vários tipos de projetos de manutenção de software;
- o capítulo 5 estabelece diretrizes para contagem de múltiplas mídias;
- o capítulo 6 define um processo de estimativas e recomendações para o gerenciamento de projetos contratados com base em métricas, tais como: tratamento de mudanças de requisitos, projetos cancelados e redução de cronograma;
- o capítulo 7 apresenta algumas atividades que não devem ser consideradas nas contagens de pontos de função;
- o capítulo 8 apresenta o processo de revisão deste guia de contagem;
- o capítulo 9 dicas para otimizar o custo das manutenções;
- o capítulo 10 conclui o documento.

2. Objetivo

Este documento tem como objetivo principal apresentar um roteiro de métricas para ser utilizado na mensuração de softwares/sistemas do Ibama, incluindo as estimativas de tamanho de projetos de software e para utilização em contratos administrativos relacionados a desenvolvimento e manutenção de sistemas.

Usa como base o roteiro de métricas do SISP 2.0 e as regras de contagem de pontos de função do Manual de Práticas de Contagem (CPM 4.3.1) para vários tipos de projetos de desenvolvimento e de manutenção de sistemas, promovendo o uso de métricas objetivas nos contratos de prestação de serviços desses projetos.

3. Contagem de Pontos de Função

A métrica PF mede o tamanho funcional de um projeto de software, observando as funcionalidades implementadas, considerando a visão do usuário. O tamanho funcional é definido como “tamanho do software derivado pela quantificação dos requisitos funcionais do usuário” [Dekkers, 2003].

A métrica PF é independente da metodologia e tecnologia utilizadas. A Análise de Pontos de Função (APF) é um método padrão para a medição de projetos de desenvolvimento e de manutenção de sistemas, visando estabelecer uma medida de tamanho do software em pontos de função, com base na quantificação das funcionalidades solicitadas e entregues, sob o ponto de vista do usuário. Assim, a APF tem como objetivo medir o que o software faz, por meio de uma avaliação padronizada dos requisitos de negócio do sistema.

O Manual de Práticas de Contagem (CPM) [IFPUG, 2010b] apresenta as regras de contagem de pontos de função de projetos de desenvolvimento, projetos de melhoria e aplicações implantadas. A Figura 1 ilustra o procedimento de contagem de pontos de função, descrito nas seções seguintes.

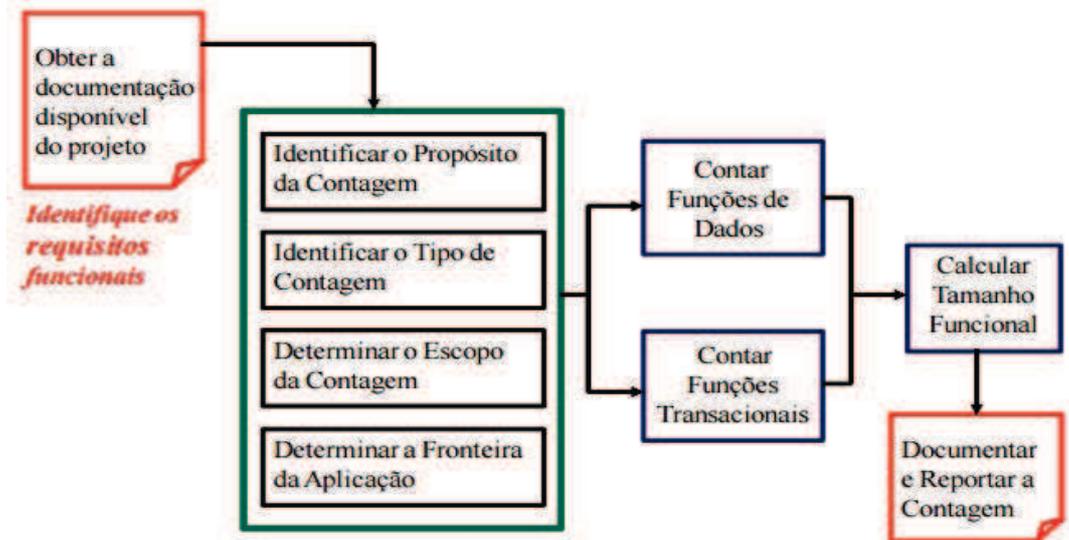


Figura 1: Procedimento de Contagem de Pontos de Função

3.1 Determinar Propósito, Tipo e Escopo da Contagem e Fronteira da Aplicação

A contagem de pontos de função se inicia com a análise da documentação disponível do projeto em questão, visando a identificação dos requisitos funcionais.

O próximo passo é o estabelecimento do propósito da contagem, o qual fornece uma resposta para uma questão de negócio a ser resolvida, por exemplo: necessidade de dimensionar um projeto de um novo sistema para auxiliar o processo de contratação do mesmo.

Com base no propósito da contagem são definidos o escopo da contagem e o tipo de contagem. O escopo da contagem identifica quais funcionalidades serão incluídas na contagem de pontos de função, e o tipo de contagem identifica se o projeto é de desenvolvimento, de melhoria ou aplicação instalada.

A fronteira da aplicação, que é a interface conceitual que indica o limite lógico entre o sistema sendo medido e os usuários (também entre outras aplicações), deve ser definida com base na visão do usuário, desconsiderando questões de implementação. Deve-se ressaltar que toda contagem de pontos de função é realizada dentro de uma fronteira estabelecida.

3.2 Identificar Funções de Dados e Funções Transacionais

Uma vez estabelecida a fronteira da contagem, o próximo passo é o mapeamento dos

requisitos de dados e de funções transacionais para os tipos funcionais da APF, a saber:

- **Arquivo Lógico Interno (ALI):** é um grupo de dados, logicamente relacionados, reconhecido pelo usuário, mantido por meio de um processo elementar da aplicação que está sendo contada.

- **Arquivo de Interface Externa (AIE):** é um grupo de dados, logicamente relacionados, reconhecido pelo usuário, mantido por meio de um processo elementar de uma outra aplicação e referenciado pela aplicação que está sendo contada. O AIE é obrigatoriamente um ALI de outra aplicação.

- **Entrada Externa (EE):** é um processo elementar que processa dados ou informação de controle que entram pela fronteira da aplicação. Seu objetivo principal é manter um ou mais ALI ou alterar o comportamento do sistema.

- **Consulta Externa (CE):** é um processo elementar que envia dados ou informação de controle para fora da fronteira da aplicação. Seu objetivo principal é apresentar informação para o usuário através da recuperação de dados ou informação de controle de ALI ou AIE.

- **Saída Externa (SE):** é um processo elementar que envia dados ou informação de controle para fora da fronteira da aplicação. Seu objetivo principal é apresentar informação para um usuário ou outra aplicação através de um processamento lógico adicional à recuperação de dados ou informação de controle. O processamento lógico deve conter cálculo, ou criar dados derivados, ou manter ALI ou alterar o comportamento do sistema.

Após a identificação dos tipos funcionais para cada requisito funcional definido no documento de requisitos do sistema, deve-se avaliar a complexidade (Baixa, Média, Alta) e a contribuição funcional do mesmo para a contagem de pontos de função, observando as regras de contagem de pontos de função descritas no CPM.

A identificação e a avaliação das complexidades dos tipos funcionais não podem ser realizadas de maneira subjetiva. A contagem de pontos de função deve seguir rigorosamente as regras de contagem do CPM e as definições complementares deste roteiro de métricas.

A Tabela 1 apresenta a contribuição dos tipos funcionais na contagem de pontos de função.

Tipo Funcional	Complexidade		
	Baixa	Média	Alta
Arquivo Lógico Interno (ALI)	7 PF	10 PF	15 PF
Arquivo de Interface Externa (AIE)	5 PF	7 PF	10 PF
Entrada Externa (EE)	3 PF	4 PF	6 PF
Saída Externa (SE)	4 PF	5 PF	7 PF
Consulta Externa (CE)	3 PF	4 PF	6 PF

Tabela 1: Contribuição Funcional dos Tipos Funcionais (Fonte: CPM 4.3.1).

3.3 Calcular Tamanho Funcional

O Manual de Práticas de Contagem do IFPUG define dois tipos de projetos de software, a saber:

- **Projeto de Desenvolvimento:** projeto para desenvolver e entregar a primeira versão de uma aplicação de software. Seu tamanho funcional é a medida das funcionalidades entregues ao usuário no final do projeto. Também considera-se as funcionalidades de conversão de dados, caso seja requisitado no projeto a migração ou carga inicial de dados para a nova aplicação.

- **Projeto de Melhoria:** projeto de manutenção evolutiva ou melhoria funcional. Seu tamanho funcional é a medida das funcionalidades incluídas, alteradas e excluídas ao final do projeto. Também considera-se as funcionalidades de conversão de dados, caso seja requisitado a migração ou carga inicial de dados no projeto de melhoria.

Seguem abaixo as definições dos termos técnicos da Análise de Pontos de Função utilizados nas fórmulas de dimensionamento de projetos de software propostas neste roteiro:

- **PF_INCLUÍDO:** pontos de função associados às novas funcionalidades que farão parte da aplicação após um projeto de desenvolvimento ou de manutenção.

- **PF_ALTERADO:** pontos de função associados às funcionalidades existentes na aplicação que serão alteradas no projeto de manutenção.

- **PF_EXCLUÍDO:** pontos de função associados às funcionalidades existentes na aplicação que serão excluídas no projeto de manutenção.

- **PF_CONVERSÃO:** pontos de função associados às funcionalidades de conversão de dados dos projetos de desenvolvimento ou de manutenção. Exemplos de funções de conversão incluem: migração ou carga inicial de dados para popular as novas tabelas criadas (Entradas Externas) e relatórios associados à migração de dados, caso requisitado pelo usuário (Saídas

Externas ou Consultas Externas). Observe que os dados carregados em um processo de migração não devem ser contados como Arquivos de Interface Externa.

Quando for caracterizado um esforço relativamente maior da atividade de conversão de dados em projetos de desenvolvimento e de melhoria, a atividade deve ser tratada como projeto separado de migração de dados, conforme seção 4.3. Por exemplo, os projetos que envolvem a migração de dados de banco de dados hierárquico para banco de dados relacional e o tratamento de funções complexas de migração de dados. Nesses casos, recomenda-se tratá-los como projetos separados de migração de dados, descritos na seção 4.3.

4. Cálculo de Pontos de Função para o Ibama

Este capítulo tem como propósito descrever os diversos tipos de projetos de software e definir métricas para seu dimensionamento baseado nas regras de contagem de pontos de função do CPM.

Quanto à documentação de projetos de manutenção, deve-se registrar a solicitação e documentar os requisitos do projeto de manutenção e da aplicação impactada pela demanda, de forma detalhada, visando apoiar a contagem de pontos de função da demanda e futuras manutenções. É importante também documentar as estimativas e a contagem de pontos de função.

Quando não houver necessidade de contratar todas as fases do ciclo de vida do software a contratada será remunerada pela contagem de pontos de função considerando apenas os percentuais das fases contratadas, conforme os níveis percentuais sugeridos na Tabela 7. Exemplo: para um novo projeto de desenvolvimento de um sistema de treinamentos, que não exista a intenção de contratar a fase de requisito, a contratada será remunerada pela contagem de pontos de função desconsiderando o percentual dessa fase.

4.1 Projeto de Desenvolvimento

Projeto de Desenvolvimento é o projeto para desenvolver e entregar a primeira versão de uma aplicação de software. Seu tamanho funcional é a medida das funcionalidades entregues ao usuário no final do projeto. Também considera-se as funcionalidades de conversão de dados. Segue a fórmula de cálculo utilizada no dimensionamento de projetos de desenvolvimento de software:

$$\text{PF_DESENVOLVIMENTO} = \text{PF_INCLUIDO} + \text{PF_CONVERSÃO}$$

Quando for caracterizado um esforço relativamente maior da atividade de conversão de dados em projetos de desenvolvimento, a atividade deve ser tratada como projeto separado de migração de dados, conforme seção 4.3.

4.2 Projeto de Melhoria

O Projeto de Melhoria (*enhancement*), também denominado de projeto de melhoria funcional ou manutenção evolutiva, está associado às mudanças em requisitos funcionais da aplicação, ou seja, à inclusão de novas funcionalidades, alteração ou exclusão de funcionalidades em aplicações.

Segundo o padrão IEEE Std 1219 [IEEE, 1998], esta manutenção seria um tipo de manutenção adaptativa, definida como: modificação de um produto de software existente para mantê-lo funcionando adequadamente em um ambiente que sofre mudanças.

O projeto de melhoria é considerado um tipo de projeto de manutenção adaptativa com mudanças em requisitos funcionais da aplicação, ou seja, com funcionalidades incluídas, alteradas ou excluídas na aplicação, segundo o CPM 4.3.1.

Este roteiro separa o projeto de melhoria (quando as mudanças são associadas aos requisitos funcionais) do projeto de manutenção adaptativa (quando as mudanças estão associadas aos requisitos não funcionais da aplicação).

Um projeto de melhoria consiste em demandas de criação de novas funcionalidades (grupos de dados ou processos elementares), demandas de exclusão de funcionalidades (grupos de dados ou processos elementares) e demandas de alteração de funcionalidades (grupos de dados ou processos elementares) em aplicações implantadas em produção.

Segue a fórmula de cálculo utilizada no dimensionamento de projetos de melhoria de software:

$$\text{PF_MELHORIA} = \text{PF_INCLUIDO} + (\text{FI} \times \text{PF_ALTERADO}) \\ + (0,40 \times \text{PF_EXCLUIDO}) + \text{PF_CONVERSÃO}$$

- **FI (Fator de Impacto)** pode variar de 50% a 90% conforme condições abaixo:
- **FI = 50%** para **funcionalidade de sistema desenvolvida ou mantida** por meio de um projeto de melhoria pela empresa contratada.
- **FI = 75%** para **funcionalidade de sistema não desenvolvida ou mantida** por meio de um projeto de melhoria pela empresa contratada e **sem necessidade de redocumentação da funcionalidade**.

● **FI = 90%** para **funcionalidade de sistema não desenvolvida ou mantida** por meio de um projeto de melhoria pela empresa contratada e **com necessidade de redocumentação da funcionalidade**. FI = 90% representa a adição de 15% como fator de redocumentação ao Fator de Impacto anterior (75%). Nesse caso, **a contratada deve redocumentar a funcionalidade mantida, gerando a documentação completa da mesma, aderente ao processo de software da contratante**. Se houver uma nova demanda de projeto de melhoria na funcionalidade em questão, será considerado que a contratada desenvolveu a funcionalidade. Observe que o percentual de 90% apenas será considerado na primeira demanda de projeto de melhoria em cada funcionalidade.

Este roteiro define um fator de redocumentação menor para projetos de manutenção (melhoria, corretiva e adaptativa) do que o fator proposto em projetos específicos de redocumentação (seção 4.12 deste roteiro). Isso porque, em projetos de manutenção de uma funcionalidade sem documentação, é necessário realizar o entendimento da funcionalidade para poder modificá-la e testá-la, ou seja, é necessário realizar a engenharia reversa da funcionalidade para executar os testes corretamente. Assim sendo, a redocumentação requisitada em projetos de melhoria requer um esforço menor do que em projetos de redocumentação, descritos na seção 4.12, onde é necessário remunerar todo o esforço de engenharia reversa e a atividade de documentação. Em projetos de manutenção, o fator de 15% está remunerando apenas a atividade de documentação.

Os percentuais de FI acima correspondem à contratação de todas as fases do processo de desenvolvimento de software. Caso alguma fase não seja contratada, deve-se aplicar ao FI um redutor que corresponde ao percentual da fase não contratada, conforme percentuais sugeridos na Tabela 7.

Quando for caracterizado um esforço relativamente maior da atividade de conversão de dados em projetos de melhoria, a atividade deve ser tratada como projeto separado de migração de dados, conforme seção 4.3.

Seguem algumas considerações importantes para serem analisadas em projetos de melhoria.

Observação 1: Função Alterada

Uma função de dados (Arquivo Lógico Interno ou Arquivo de Interface Externa) é considerada alterada quando houver inclusão ou exclusão de Tipos de Dados (TD). De acordo com o glossário do CPM 4.3.1, um Tipo de Dados (DET – *Data Element Type*) é um atributo único, reconhecido pelo usuário e não repetido. Também é considerada alterada se algum tipo de dado sofrer mudança de tamanho (número de posições) ou tipo de campo (por exemplo: mudança de numérico ou alfanumérico), caso a mudança decorra de alteração de regra de negócio.

Uma função transacional (Entrada Externa, Consulta Externa e Saída Externa) é considerada alterada, quando a alteração contemplar:

- Mudança de tipos de dados;
- Mudança de arquivos referenciados;
- Mudança de lógica de processamento.

O CPM 4.3.1 define **lógica de processamento** como requisitos especificamente solicitados pelo usuário para completar um processo elementar. Esses requisitos devem incluir uma ou mais das seguintes ações:

- Validações são executadas;
- Fórmulas matemáticas e cálculos são executados;
- Valores equivalentes são convertidos;
- Dados são filtrados e selecionados através da utilização de critérios;
- Condições são analisadas para verificar quais são aplicáveis;
- Um ou mais ALIs são atualizados;
- Um ou mais ALIs ou AIEs são referenciados;
- Dados ou informações de controle são recuperados;
- Dados derivados são criados através da transformação de dados existentes, para criar dados adicionais;
- O comportamento do sistema é alterado;
- Preparar e apresentar informações para fora da fronteira;
- Receber dados ou informações de controle que entram pela fronteira da aplicação;
- Dados são reordenados.

Observação 2: Outros Tipos de Funções Alteradas

Este roteiro considera como função alterada qualquer mudança em funcionalidades da aplicação devido às mudanças de regras de negócio. Por exemplo, uma funcionalidade de cadastro envolvia a inclusão de um telefone do gerente. Devido a mudanças no processo de negócio, a funcionalidade deve sofrer uma manutenção para cadastrar dois telefones do gerente. Desta forma, o roteiro considera esta função como uma Entrada Externa alterada, PF_ALTERADO

em um projeto de melhoria, mesmo que não existam mudanças de lógica de processamento, de tipos de dados ou de arquivos referenciados.

Serão tratadas como manutenções adaptativas apenas as manutenções que implicarem exclusivamente em mudanças em requisitos não funcionais. Se uma mesma funcionalidade tiver mudanças em requisitos funcionais e não funcionais, esta deve ser contada apenas uma vez, como função alterada em um projeto de melhoria.

4.3 Projeto de Migração de Dados

Os projetos de migração de dados devem ser contados como um novo projeto de desenvolvimento de um sistema, seguindo a fórmula abaixo:

$$\text{PF_CONVERSÃO} = \text{PF_INCLUIDO}$$

Um projeto de migração deve contemplar minimamente: os ALI mantidos pela migração, as Entradas Externas – considerando as cargas de dados nos ALI – e, caso seja solicitado pelo usuário, os relatórios gerenciais das cargas, que serão contados como Saídas Externas. Todas as contagens de PF devem ser realizadas com base nas funcionalidades requisitadas e recebidas pelo usuário.

4.4 Manutenção Corretiva

Mesmo com a execução de atividades de garantia da qualidade, pode-se identificar defeitos na aplicação entregue. A manutenção corretiva altera o software para correção de defeitos. Encontra-se nesta categoria, as demandas de correção de erros (*bugs*) em funcionalidades de sistemas em produção.

Quando o sistema em produção tiver sido desenvolvido pela contratada, a manutenção corretiva será do tipo Garantia se estiver no período de cobertura e em conformidade com as demais condições de garantia previstas em contrato.

Quando o sistema estiver fora da garantia ou não tenha sido desenvolvido pela empresa contratada, deverá ser estimado e calculado o tamanho do projeto de manutenção corretiva. Nestes casos, a aferição do tamanho em pontos de função da funcionalidade ou das funcionalidades corrigidas deve considerar um fator de impacto (FI) sobre o PF_ALTERADO.

$$\text{PF_CORRETIVA} = \text{FI} \times \text{PF_ALTERADO}$$

Fator de Impacto (FI):

- 50% quando estiver fora da garantia e a correção for feita pela mesma empresa que desenvolveu a funcionalidade.
- 75% quando estiver fora da garantia e a correção for feita por empresa diferente daquela que desenvolveu a funcionalidade.

As demandas de manutenção corretiva não contemplam atualização de documentação da funcionalidade corrigida, pois este roteiro considera que, normalmente, manutenção corretiva não se refere a erros de requisitos. Caso seja erro em requisitos, essa demanda deve ser tratada como projeto de melhoria (alteração de funcionalidade), descrito na seção 4.2. Porém, quando o erro for causado por documentação dúbia ou imprecisa (elaborada pela contratada) da funcionalidade corrigida, a manutenção corretiva poderá contemplar os ajustes na documentação, mesmo fora da garantia.

Caso seja demandada a redocumentação da funcionalidade corrigida, porque a documentação não existe ou está desatualizada, deve-se adicionar ao FI um fator de redocumentação de 15%, conforme descrito na seção 4.2.

4.5 Mudança de Plataforma

São considerados nesta categoria, projetos que precisam ser migrados para outra plataforma. Por exemplo, um sistema legado em COBOL que necessita ser redesenvolvido em JAVA; o banco de dados de um sistema legado que precisa ser migrado para o DB2.

Recomenda-se enfaticamente a realização da análise de impacto das mudanças propostas, para efeito de determinação do percentual adequado para aplicação sobre o total de pontos de função das funcionalidades impactadas. Por exemplo, em uma análise de impacto pode ser identificado que não haverá mudanças no código-fonte ou em função transacional, sendo necessário apenas testar o sistema, então deve-se utilizar um percentual contemplando apenas a fase de testes. No caso do teste apontar a necessidade de atualizar alguma função transacional, não deve ser contado o esforço do teste, mas sim o esforço abordado nesta seção, conforme as fórmulas apresentadas nos tópicos seguintes.

As próximas subseções apresentam os tipos de projetos de mudança de plataforma. Os projetos de mudança de plataforma que se enquadram em mais de uma subseção, devem ser contados apenas uma vez, considerando o tipo de projeto com maior contagem de pontos de função.

4.5.1 Mudança de Plataforma – Linguagem de Programação

Nesta categoria encontram-se as demandas de redesenvolvimento de sistemas em outra linguagem de programação. Como os projetos legados, frequentemente, não possuem documentação, devem ser considerados como novos projetos de desenvolvimento. Assim, será utilizada a fórmula de projetos de desenvolvimento do CPM 4.3.1.

Observe que caso não exista mudança nas funções de dados, ou seja, o banco de dados da aplicação seja mantido, as funções de dados não devem ser consideradas para fins de pagamento. No entanto, nesse caso, deve ser realizada a contagem das funções de dados a fim de compor a documentação da contagem final do projeto.

Outro ponto a ser observado são as fases contratadas. Caso o projeto já possua documentação de requisitos, a fase de requisitos não será contratada. Deve-se considerar apenas os percentuais das fases contratadas.

$$\text{PF_REDESENVOLVIMENTO_LINGUAGEM} = \text{PF_INCLUÍDO} + \text{PF_CONVERSÃO}$$

Quando for caracterizado um esforço relativamente maior da atividade de conversão de dados em projetos de redesenvolvimento, a atividade deve ser tratada como projeto separado de migração de dados, conforme seção 4.3.

4.5.2 Mudança de Plataforma – Banco de Dados

Nesta categoria encontram-se as demandas de redesenvolvimento de sistemas para utilizar um outro sistema gerenciador de banco de dados.

Observe que caso não exista mudança nas funções de dados, ou seja, o banco de dados da aplicação seja mantido, então as funções de dados não devem ser consideradas para fins de pagamento. No entanto, nesse caso, deve ser realizada a contagem das funções de dados a fim de compor a documentação da contagem final do projeto.

Caso a demanda de redesenvolvimento seja de um sistema gerenciador de banco de dados relacional para outro relacional, deve ser utilizada a seguinte fórmula:

$$\text{PF_REDESENVOLVIMENTO_BD_RELACIONAL} = (\text{PF_ALTERADO} \times 0,30) + \text{PF_CONVERSÃO}$$

O PF_ALTERADO deve considerar apenas as funcionalidades impactadas. As funcionalidades que possuem apenas demandas de testes, devem ser contadas usando o

percentual da fase de testes (ver Tabela 7).

Nos projetos de redesenvolvimento de banco de dados relacional para outro relacional, recomenda-se tratar o PF_CONVERSÃO dentro do mesmo projeto.

Na mudança de banco relacional para relacional, geralmente a estrutura de dados não é alterada, desta forma não contamos as funções de dados.

4.6 Atualização de Versão

São consideradas nesta categoria, demandas para uma aplicação existente – ou parte de uma aplicação existente – executar em versões diferentes de *browsers* (ex: *Internet Explorer*, *Firefox*, *Chrome*, etc) ou de linguagens de programação (ex: versão mais atual do JAVA). Também são consideradas nesta categoria atualização de versão de banco de dados.

Nesta categoria foram observadas demandas de diferentes tipos de projetos, descritos nas próximas subseções.

Outro ponto a ser observado é a classificação, em alguns casos, dessas demandas como componente interno reusável (seção 4.15).

Recomenda-se enfaticamente a realização da análise de impacto das mudanças propostas para efeito de determinação do percentual adequado para aplicação sobre o total de pontos de função das funcionalidades impactadas. Por exemplo, em uma análise de impacto, pode ser identificado que não haverá mudanças no código-fonte ou em função transacional, sendo necessário somente testar o sistema, então deve-se utilizar um percentual contemplando apenas a fase de testes. No caso do teste apontar a necessidade de atualizar alguma função transacional, não deve ser contado o esforço do teste, mas sim o esforço abordado nesta seção, conforme as fórmulas apresentadas nas subseções seguintes.

4.6.1 Atualização de Versão – Linguagem de Programação

Nesta categoria encontram-se as demandas de atualização de versão de linguagem de programação de sistemas. As funções de dados não devem ser contadas. Estas demandas devem ser dimensionadas de acordo com a fórmula abaixo.

$$\text{PF_ATUALIZAÇÃO_VERSÃO_LINGUAGEM} = \text{PF_ALTERADO} \times 0,30$$

O PF_ALTERADO deve considerar apenas as funcionalidades impactadas. As funcionalidades que possuem apenas demandas de testes, devem ser contadas usando o percentual da fase de testes (ver Tabela 7).

4.6.2 Atualização de Versão – Browser

Nesta categoria encontram-se as demandas de atualização de aplicações Web para executar em novas versões de um mesmo *browser* e para suportar a execução em mais de um *browser*. É importante destacar que este tipo de procedimento usualmente é realizado quando é necessário resolver algum problema de incompatibilidade. As funções de dados não devem ser contadas. Estas demandas devem ser dimensionadas de acordo com a fórmula abaixo.

$$\text{PF_ATUALIZAÇÃO_VERSÃO_BROWSER} = \text{PF_ALTERADO} \times 0,30$$

O PF_ALTERADO deve considerar apenas as funcionalidades impactadas. As funcionalidades que possuem apenas demandas de testes, devem ser contadas usando o percentual da fase de testes (ver Tabela 7).

Essas atualizações podem implicar em manutenções em componentes específicos da plataforma utilizada. Nesse caso, a demanda deve ser contada como componente interno reusável, descrita na seção 4.15 deste roteiro.

4.6.3 Atualização de Versão – Banco de Dados

Nesta categoria encontram-se as demandas de atualização de versão do sistema gerenciador de banco de dados. As funções de dados não devem ser contadas. Estas demandas devem ser dimensionadas de acordo com a fórmula abaixo.

$$\text{PF_ATUALIZAÇÃO_VERSÃO_BD} = \text{PF_ALTERADO} \times 0,30$$

O PF_ALTERADO deve considerar apenas as funcionalidades impactadas. As funcionalidades que possuem apenas demandas de testes, devem ser contadas usando o percentual da fase de testes (ver Tabela 7).

4.7 Manutenção em Interface

A manutenção em interface, denominada na literatura de manutenção cosmética, é associada às demandas de alterações de interface, por exemplo: fonte de letra, cores de telas, logotipos, mudança de botões na tela, mudança de posição de campos ou texto na tela. Também se enquadram nessa categoria as seguintes manutenções:

- Mudanças de texto em mensagens de erro, validação, aviso, alerta, confirmação de cadastro ou conclusão de processamento;

- Mudança em texto estático de e-mail enviado para o usuário em uma funcionalidade de cadastro. A demanda deve ser contada como manutenção em interface na funcionalidade de cadastro;

- Alteração de título de um relatório;
- Alteração de *labels* de uma tela de consulta.

Nestes casos, a aferição do tamanho em pontos de função das funções transacionais impactadas será realizada com a aplicação de um fator de redução de modo a considerar 20% da contagem de uma função transacional de mais baixa complexidade (3 PF), ou seja 0,6 PF, independentemente da complexidade da funcionalidade alterada. Neste tipo de manutenção não são contadas funções de dados.

PF_INTERFACE = 0,6 PF x QUANTIDADE DE FUNÇÕES TRANSACIONAIS IMPACTADAS

Está contemplada a atualização da documentação das funcionalidades da aplicação impactadas pela manutenção nas demandas desta categoria. Assim, a documentação (documento de requisitos, documento de interface, protótipo, entre outros) das funcionalidades alteradas deve ser atualizada. Caso não exista documentação para as funcionalidades alteradas, não será contemplada a redocumentação das funcionalidades da aplicação impactadas pela manutenção nas demandas desta categoria.

Observação 1 – Help: As demandas de projetos de desenvolvimento de sistemas ou de manutenção de funcionalidades já contemplam o desenvolvimento ou atualização do *help* da funcionalidade em questão, sendo tratada como uma atividade de documentação no processo de software.

No caso de demandas específicas de desenvolvimento ou atualização de *help* estático de funcionalidades, estas podem ser enquadradas nesta seção e deverá ser usado o valor de multiplicação de 0,3 PF. Em caso de requisitos de usuário para o desenvolvimento de funcionalidades de manutenção de *help*, deve-se contar a função de dados de *help* e as funcionalidades de manutenção de *help* (por exemplo: incluir *help* de tela, consultar *help* de campo) de acordo com o CPM 4.3.1.

4.8 Adaptação em Funcionalidades sem Alteração de Requisitos Funcionais

São consideradas nesta categoria as demandas de manutenção adaptativa associadas a solicitações que envolvem aspectos não funcionais, sem alteração em requisitos funcionais.

Seguem alguns exemplos:

- Aumentar a quantidade de linhas por página em um relatório;
- Colocar paginação em um relatório;
- Limitar a quantidade de linhas por página em uma consulta existente;
- Permitir exclusões múltiplas em uma funcionalidade que antes só possibilitava a exclusão de um item;
- Adaptação de uma funcionalidade para possibilitar a chamada por um *webservice* ou para outro tipo de integração com outros sistemas;
- Replicação de funcionalidade: chamar uma consulta existente em outra tela da aplicação;
- Alteração na aplicação para adaptação às alterações realizadas na interface com rotinas de integração com outros softwares, por exemplo, alteração em sub-rotinas chamadas por este software;
- Modificar o servidor a ser acessado em uma funcionalidade de *download* de arquivo;
- Adequar mensagem do sistema que em algumas telas apresenta “Usuário Não está Habilitado a ver esta Página”, para que passe a enviar uma mensagem mais adequada ao fato do usuário não possuir mais uma sessão ativa e ainda navegar no sistema. A demanda deve ser contada como manutenção adaptativa considerando as funcionalidades impactadas. Observe que trata-se de mudança em validação com regra de negócio não funcional.

Nestes casos, a aferição do tamanho em pontos de função da funcionalidade ou das funcionalidades que sofreram impacto deve considerar um fator de impacto (FI) sobre o PF_ALTERADO, seguindo os conceitos do CPM 4.3.1, apresentados na seção 4.2.

$$\text{PF_ADAPTATIVA} = \text{FI} \times \text{PF_ALTERADO}$$

FI (Fator de Impacto) pode variar conforme condições abaixo:

- **FI = 50%** para **funcionalidade de sistema desenvolvida ou mantida** por meio de um projeto de melhoria pela empresa contratada.
- **FI = 75%** para **funcionalidade de sistema não desenvolvida ou mantida** por meio de um projeto de melhoria pela empresa contratada.

Caso exista a documentação das funcionalidades impactadas, estas deverão ser atualizadas sem ônus adicional à contratante. Caso contrário, se for demandada a

redocumentação dessas funcionalidades, deve-se adicionar ao FI um fator de redocumentação de 15%, conforme descrito na seção 4.2.

4.9 Apuração Especial

São funcionalidades executadas apenas uma vez para: corrigir problemas de dados incorretos na base de dados das aplicações ou atualizar dados em bases de dados de aplicações, detalhados na subseção 4.9.1; gerar um relatório específico ou arquivo para o usuário por meio de recuperação de informações nas bases da aplicação, detalhados na subseção 4.9.2. A subseção 4.9.3 considera os casos de reexecução de uma apuração especial.

Caso a apuração seja de correção de dados devido a erros de funcionalidades de aplicações desenvolvidas pela contratada, observar as cláusulas contratuais com relação a garantias e prazos de correção.

Para estes casos, deve ser solicitada formalmente à empresa contratada o armazenamento do *script* para permitir posterior reexecução.

4.9.1 Apuração Especial – Base de Dados

Este tipo de apuração especial é um projeto que inclui a geração de procedimentos para atualização da base de dados. Deve-se destacar que estas funções são executadas apenas uma vez, não fazendo parte da aplicação, visando a correção de dados incorretos na base de dados da aplicação ou atualização em função de modificação da estrutura de dados, por exemplo inclusão de valor “sim” ou “não” no campo “indicador de matriz” referente ao CNPJ.

Normalmente, nesse tipo de atualização são afetados múltiplos registros. Nestes casos, considera-se a contagem de pontos de função das funcionalidades desenvolvidas. Geralmente, estas funcionalidades são classificadas como Entradas Externas. Nesse caso, como artefato de homologação da demanda, deve ser gerado um relatório para validação do usuário.

É importante ressaltar que as funções de dados associadas aos dados atualizados não devem ser contadas, considerando que não há mudanças nas estruturas dos Arquivos Lógicos Internos.

Foram identificados três tipos de Apuração Especial – Base de Dados, cujas fórmulas de cálculo são apresentadas a seguir:

a) Atualização de Dados sem Consulta Prévia

$$PF_APURAÇÃO_BD = PF_INCLUÍDO$$

b) Consulta Prévia sem Atualização

Em alguns casos de Apuração Especial – Base de Dados, o usuário solicita uma consulta prévia das informações. Deve-se ressaltar que essa consulta deve ser realizada antes da construção da funcionalidade, não se trata de homologação. A consulta prévia não é definida pela empresa contratada, obrigatoriamente essa deve ser solicitada pelo órgão contratante para a avaliação da viabilidade de implementar a Apuração Especial – Base de Dados. De fato, é uma prática interessante para evitar informações errôneas na base de produção dos sistemas. Esta consulta prévia, classificada como Consulta Externa ou Saída Externa deve ser dimensionada considerando-se o tamanho da funcionalidade em questão, conforme a fórmula abaixo:

$$\text{PF_CONSULTA_PRÉVIA} = \text{PF_INCLUÍDO}$$

c) Atualização de Dados com Consulta Prévia

Caso a Apuração Especial – Base de Dados seja solicitada após uma demanda de consulta prévia, deve-se aplicar um fator de 60% na fórmula de contagem da Apuração Especial – Base de Dados, seguindo a fórmula abaixo.

$$\text{PF_APURAÇÃO_BD_PÓS_CONSULTA_PRÉVIA} = \text{PF_INCLUÍDO} \times 0,60$$

4.9.2 Apuração Especial – Geração de Relatórios

Este tipo de apuração especial é um projeto que inclui a geração de relatórios em uma ou mais mídias para o usuário. Em alguns casos, são solicitadas extrações de dados e envio dos dados para outros sistemas. Caso, neste envio de dados, sejam requisitadas atualizações no sistema de origem, então essas funções transacionais são Saídas Externas, devido à atualização do Arquivo Lógico Interno.

Deve-se destacar que essas funções são executadas apenas uma vez, não fazendo parte da aplicação. Nestes casos, considera-se contagem de pontos de função das funcionalidades desenvolvidas. Frequentemente, estas funcionalidades são classificadas como Saídas Externas. Também podem ser classificadas como Consultas Externas, caso não possuam cálculos ou criação de dados derivados.

É importante ressaltar que as funções de dados associadas aos dados atualizados não devem ser contadas, considerando que não há mudanças nas estruturas dos Arquivos Lógicos.

$$\text{PF_APURAÇÃO_RELATÓRIOS} = \text{PF_INCLUÍDO}$$

4.9.3 Apuração Especial – Reexecução

Em alguns casos, a contratante pode ter interesse em executar uma apuração especial mais de uma vez. Nestes casos, ela deve solicitar formalmente à contratada o armazenamento do *script* executado. Desta forma, se for solicitada a reexecução de uma apuração especial, esta deve ser dimensionada com a aplicação de um fator redutor de 10% na contagem de pontos de função da apuração especial em questão, da seguinte maneira:

$$\text{PF_REEXECUÇÃO_APURAÇÃO} = \text{PF_NÃO_AJUSTADO} \times 0,10$$

4.10 Atualização de Dados

Em alguns casos, as demandas de correção de problemas em base de dados estão associadas a atualizações manuais (de forma interativa), diretamente no banco de dados em um único registro, e que não envolvem cálculos ou procedimentos complexos. São exemplos desse tipo de demanda, a atualização do valor de um campo de uma tabela cadastrado erroneamente ou a exclusão de um registro de uma tabela.

Nestes casos, a aferição do tamanho em Pontos de Função deve considerar 10% do PF de uma Entrada Externa e os Tipos de Dados da Entrada Externa são todos os TD considerados na funcionalidade – campos atualizados e campos utilizados para a seleção do registro.

$$\text{PF_ATUALIZAÇÃO_BD} = \text{PF_INCLUÍDO} \times 0,10$$

Deve-se ressaltar que neste tipo de demanda não há gestão de configuração (armazenamento de *script*, versionamento, etc) das atualizações. Caso a contratante identifique a necessidade de realização de gestão de configuração das atualizações no banco de dados, então a demanda será classificada como Apuração Especial - Base de Dados (subseção 4.9.1).

4.11 Manutenção de menus, telas e listas de seleção estáticas

As alterações de menus, telas e listas de seleção estáticas, que não fazem parte de um projeto de melhoria, serão contadas com a aplicação de um fator de redução de modo a considerar 10% da contagem de uma função transacional de mais baixa complexidade (3 PF), ou seja 0,3 PF, para cada menu, tela ou lista de seleção estática, independentemente da complexidade da funcionalidade alterada. Neste tipo de manutenção não são contadas funções de dados.

As alterações de menus, telas e listas de seleção estáticas que fazem parte de um projeto

de melhoria não terão acréscimos na contagem, uma vez que o projeto de melhoria já abarca todas as alterações nas funcionalidades do seu escopo.

4.12 Manutenção de Documentação de Sistemas Legados

Nesta seção são tratadas demandas de documentação ou atualização de documentação de sistemas legados. Observe que o desenvolvedor deve realizar uma engenharia reversa da aplicação para gerar a documentação. Para este tipo de projeto foi definido o fator de impacto de 25% dos pontos de função da aplicação em questão, considerando a fase de requisitos e a geração de artefatos associados a requisitos, conforme a fórmula abaixo.

$$\text{PF_DOCUMENTAÇÃO} = \text{PF_NÃO_AJUSTADO} \times 0,25$$

4.13 Verificação de Erros

As verificações de erro ou análise e solução de problemas são as demandas referentes a todo comportamento anormal ou indevido apontado pelo cliente nos sistemas aplicativos. Neste caso, a equipe de desenvolvimento da contratada se mobilizará para encontrar as causas do problema ocorrido. Se for constatado algum erro de sistema, a demanda será atendida como manutenção corretiva (seção 4.4).

Entretanto, uma vez não constatado o problema apontado pelo cliente ou o mesmo for decorrente de regras de negócio implementadas ou utilização incorreta das funcionalidades, será realizada a aferição do tamanho em pontos de função das funcionalidades verificadas que o cliente reportou erro. Caso não exista documentação de testes disponível dessas funcionalidades verificadas, será considerado 20% do tamanho funcional dessas funcionalidades com solicitação de análise pelo órgão contratante, segundo a fórmula abaixo:

$$\text{PF_VERIFICAÇÃO} = \text{PF_Funcionalidade_Reportada_Com_Erro} \times 0,20$$

Caso exista documentação de testes das funcionalidades verificadas, então será considerado 15% (mesmo percentual da fase de Testes, conforme Tabela 7) do tamanho funcional das funcionalidades analisadas, segundo a fórmula abaixo:

$$\text{PF_VERIFICAÇÃO} = \text{PF_Funcionalidade_Reportada_Com_Erro} \times 0,15$$

É importante ressaltar que a demanda de verificação de erros deve ser associada a uma funcionalidade específica. Os casos de sistema fora do ar por conta de problemas de rede ou

banco de dados devem ser tratados como serviços de suporte e não serviços de desenvolvimento e manutenção de sistemas. Esses serviços de suporte não fazem parte do escopo desse roteiro de métricas, não se aplicando verificação de erros nestes casos.

4.14 Pontos de Função de Teste

Muitas vezes, em projetos de manutenção, o conjunto de funções transacionais a serem testadas é maior do que a quantidade de funções a serem implementadas, isto é, além das funcionalidades que são afetadas diretamente pelo projeto de manutenção, outras precisam ser testadas [NESMA, 2009]. O tamanho das funções a serem apenas testadas deve ser aferido em Pontos de Função de Teste (PFT). Não considerar as funcionalidades incluídas, alteradas ou excluídas do projeto de manutenção na contagem de Pontos de Função de Teste.

A contagem de PFT será o somatório dos tamanhos em pontos de função das funções transacionais envolvidas no teste:

PFT = Somatório dos Tamanhos das Funções Transacionais Testadas

A conversão do PFT em ponto de função deve ser feita de acordo com a fórmula abaixo:

$$\text{PF_TESTES} = \text{PFT} \times 0,15$$

É importante ressaltar que no caso de uma função ser testada várias vezes, com cenários diferentes, a função só pode ser contada uma vez. Outra observação é que as funções testadas, consideradas no PFT, devem ser documentadas pela contratada considerando-se a documentação de testes definida no processo de desenvolvimento da contratante. Observe que estas funções farão parte do escopo de projetos de manutenção e desenvolvimento.

4.15 Componente Interno Reusável

Em alguns casos são demandadas manutenções em componentes específicos de uma aplicação e estes são reusados por várias funcionalidades da aplicação.

Este roteiro define que o componente, o qual deverá ser desenvolvido ou mantido, seja considerado como um processo elementar independente e contado como uma funcionalidade. Além disso, as funcionalidades da aplicação que necessitem de teste devem ser requisitadas pela contratante e dimensionadas por meio da métrica Pontos de Função de Teste proposta na seção 4.14.

PF_COMPONENTE = PF_NÃO_AJUSTADO

Seguem alguns exemplos de manutenção de componentes:

- Alteração de valores de elementos internos de configuração que afetem o comportamento ou a apresentação do sistema de forma geral, tais como páginas de estilos (arquivos CSS de sistemas Web), arquivos com mensagens de erro, arquivos de configuração de sistema, arquivos de internacionalização;
- Mudança em tópico de um menu de um sistema em PHP que aparece em todas as telas da aplicação. A contagem pode ser realizada considerando o componente “Apresentar Menu”.

5. Orientações Complementares para Contagem

Este capítulo tem como propósito apresentar as diretrizes de contagem de pontos de função em relação a divisão de um mesmo processo elementar em transações distintas e Múltiplas Mídias.

As definições apresentadas sobre tema Múltiplas Mídias têm como base o artigo “*Considerations for Counting with Multiple Media*” Release 1.1 publicado pelo IFPUG [IFPUG, 2010a].

5.1 Divisão de um mesmo Processo Elementar em Transações Distintas

A divisão de um mesmo processo elementar em transações distintas fere o conceito disposto no item 5.5.2 do Manual de Práticas de Contagem de Pontos de Função (CPM 4.3.1):

“Para identificar cada processo elementar devem ser executadas as seguintes atividades:

a) Compor e/ou decompor os Requisitos Funcionais do Usuário nas menores unidades de atividade que satisfaçam todos os itens abaixo:

- 1) é significativa para o usuário;*
- 2) constitui uma transação completa;*
- 3) é autocontida;*

4) *deixa o negócio da aplicação contada em um estado consistente.*”

O item 3.7 do CPM explicita o conceito de autocontido, reforçando a propriedade de constituição de transação completa. O entendimento do exemplo a seguir facilitará o entendimento de um cenário envolvendo a divisão de um processo elementar em transações distintas.

“Nenhum passo anterior ou subsequente é necessário para iniciar ou concluir o(s) Requisito(s) Funcional(is) do Usuário.

EXEMPLO: O Requisito Funcional do Usuário estabelece que um empregado deve ser incluído e atualizado.

Poderiam existir várias partes que comporiam o conjunto completo de informações do empregado. Isto pode ser representado por telas físicas, janelas ou abas distintas, tais como

- Identificação do empregado,*
- Localização do empregado,*
- Informações de dependentes,*
- Informações de salário e*
- Instrução.*

Para incluir um empregado, uma ou mais abas devem ser preenchidas, dependendo das regras de negócio. O processo de inclusão não estará autocontido até que todas as informações obrigatórias tenham sido digitadas e recebidas pelo sistema.

Para atualizar um empregado, uma ou mais abas podem ser atualizadas a qualquer momento, mas todas elas constituem passos do processo que satisfaz o Requisito Funcional do Usuário, de atualização do empregado.

Incluir, alterar ou excluir informações de cada aba individual não constituem processos elementares distintos, mas sim passos de processo envolvidos na atualização de um empregado. Embora seja possível entrar com informações adicionais no registro de empregado, o conjunto total de informações é considerado parte do único processo elementar: atualizar empregado.

Incluir Empregado e Atualizar Empregado seriam, cada um, um processo autocontido.”

5.2 Contagem de Pontos de Função com Múltiplas Mídias

A contagem de PF de funcionalidades entregues em mais de uma mídia, na aplicação das

regras de contagem de pontos de função definidas no CPM, tem levado a duas abordagens alternativas, a saber: *single instance* e *multiple instance*.

É importante enfatizar que o IFPUG reconhece ambas abordagens, *single instance* e *multiple instance*, para a aplicação das regras definidas no CPM.

As estimativas e contagens de PF abordadas neste documento são baseadas em *multiple instance*, com exceção dos casos de consultas em .pdf, .doc, .xls e consultas idênticas em tela e papel, que serão consideradas uma única funcionalidade.

A seguir são descritos os termos comuns definidos pelo IFPUG [IFPUG, 2010]:

- **Canal:** também refere-se a mídia. Múltiplos canais é sinônimo de múltiplas mídias.

- **Mídia:** descreve a maneira como os dados ou informações se movimentam para dentro e para fora de uma fronteira de aplicação, por exemplo, apresentação de dados em tela, impressora, arquivo, voz. Este termo é utilizado para incluir, dentre outros, diferentes plataformas técnicas e formatos de arquivos como diferentes mídias.

- **Múltiplas Mídias:** quando a mesma funcionalidade é entregue em mais de uma mídia. Frequentemente, apenas uma mídia é requisitada para um usuário específico em um determinado momento, por exemplo consulta de extrato bancário via Internet como oposto a consulta de extrato bancário via terminal do banco.

- **Multi-Mídia:** quando mais de uma mídia é necessária para entregar a funcionalidade, por exemplo, uma nova notícia publicada na Internet que é apresentada em vídeo e texto. Observe que a notícia completa só é apresentada para o usuário se ele ler o texto e assistir o vídeo.

- **Abordagem *Single Instance*:** esta abordagem não reconhece que a mídia utilizada na entrega da função transacional é uma característica de diferenciação na identificação da unicidade da função transacional. Se duas funções entregam a mesma funcionalidade usando mídias diferentes, elas são consideradas a mesma funcionalidade em uma contagem de pontos de função.

- **Abordagem *Multiple Instance*:** esta abordagem especifica que o tamanho funcional é obtido no contexto do objetivo da contagem, permitindo uma função de negócio ser reconhecida no contexto das mídias que são requisitadas para que a funcionalidade seja entregue. A abordagem *multiple instance* reconhece que a mídia para entrega constitui uma característica de diferenciação na identificação da unicidade da função transacional.

Os cenários descritos nas seções seguintes não representam uma lista completa de situações de múltiplas mídias. O entendimento dos exemplos a seguir facilitará o entendimento de outros cenários envolvendo múltiplas mídias.

5.2.1 Cenário 1: Mesmos dados apresentados em tela e impressos

Neste cenário, uma aplicação apresenta uma informação em uma consulta em tela. A mesma informação pode ser impressa, caso requisitado pelo usuário, na tela em questão.

Nesses casos, utiliza-se a abordagem *single instance*, considerando que dados idênticos sendo apresentados em tela e em relatório impresso devem ser contados como uma única função.

Caso as lógicas de processamento da consulta em tela e do relatório em papel sejam distintas, o processo elementar não é único e, portanto, a funcionalidade será contada duas vezes (*multiple instance*). Neste caso, duas funções são contadas: apresentação de dados em tela e apresentação de dados impressos.

5.2.2 Cenário 2: Mesmos dados de saída como dados em arquivo e relatório impresso

Uma aplicação grava dados em um arquivo de saída e imprime um relatório com informações idênticas às gravadas no arquivo.

Nesses casos, utiliza-se a abordagem *single instance*, considerando que os dados impressos e os dados apresentados no arquivo de saída sejam idênticos. Assim, apenas uma funcionalidade será incluída na contagem de pontos de função.

Este caso ocorre com ou sem o apoio de ferramenta de desenvolvimento para a geração dessas múltiplas saídas.

Porém, caso as lógicas de processamento da geração do arquivo de saída e do relatório em papel sejam distintas, o processo elementar não é único e, portanto, a funcionalidade será contada duas vezes. Além disso, se a geração das múltiplas saídas não seguirem o padrão da ferramenta de desenvolvimento e tiverem que ser customizadas para o cliente, então será utilizada a abordagem *multiple instance*.

5.2.3 Cenário 3: Mesmos dados de entrada *batch* e on-line

Uma informação pode ser carregada na aplicação por meio de dois métodos: arquivo *batch* e entrada on-line. O processamento do arquivo *batch* executa validações durante o processamento, da mesma forma que o processamento da entrada on-line também executa validações das informações. Neste caso, utiliza-se a abordagem *multiple instance*, que conta duas funcionalidades: a entrada de dados *batch* e a entrada de dados on-line.

Geralmente, a lógica de processamento utilizada nas validações em modo *batch* é diferente da lógica de processamento das validações nas entradas de dados on-line.

5.2.4 Cenário 4: Múltiplos canais de entrega da mesma funcionalidade

Uma funcionalidade deve ser disponibilizada em múltiplos canais, por exemplo: consulta de dados em página Web e consulta de dados no telefone celular. Neste caso, utiliza-se a abordagem *multiple instance*, que conta duas funcionalidades: consulta de dados na Web e consulta de dados via celular.

Considera-se que a funcionalidade é desenvolvida duas vezes, uma para cada canal de saída. Algumas vezes, são até projetos de desenvolvimento distintos, um projeto relativo ao sistema Web e outro para o sistema via celular. Lembrando que caso o projeto seja claro o suficiente para dizer que o desenvolvimento é o mesmo, poderá ser utilizada a abordagem *single instance*.

5.2.5 Cenário 5: Relatório em múltiplos formatos

Em demandas que possuem a solicitação do usuário para a geração de relatórios que contenham mesmas informações (DER), acessem mesmos arquivos (ALR) e possuem mesma lógica de processamento (não havendo supressões de registros por filtros, por exemplo), utiliza-se a abordagem *single instance*, ou seja, conta-se apenas a criação de 1 (uma) funcionalidade.

Conforme o Manual de Práticas de Contagem de Pontos de Função 4.3.1, quando dois processos elementares (PEs) são comparados e se determina que eles contêm diferentes DERs, RLRs ou Processamento Lógico, eles são identificados como processos elementares separados se forem especificados como requisitos funcionais distintos pelo usuário. Nesse caso, utiliza-se a abordagem *multiple instance*.

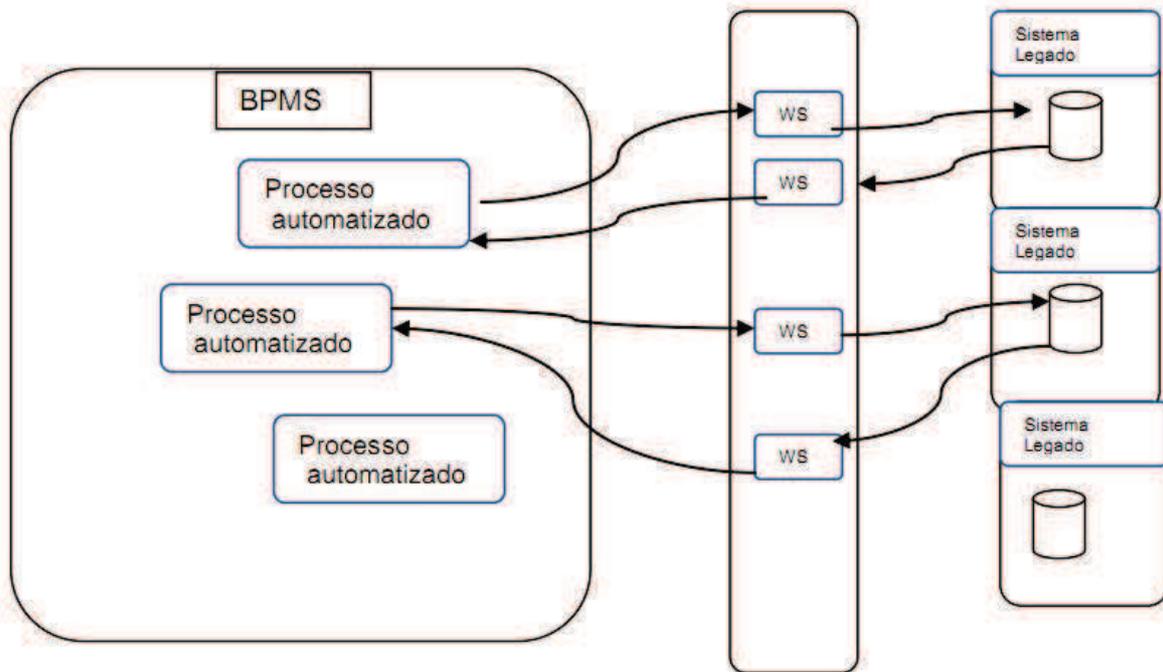
Assim, uma consulta que é exibida em tela constitui uma funcionalidade. Havendo opções de exportação destes mesmos dados da consulta em diversos formatos (pdf, doc, xls, html, xml, etc) ou mesmo a impressão direta em impressora, não se contam funcionalidades adicionais (a exibição das opções de exportação em botões separados ou em caixa de diálogo não invalida este conceito, por ser decisão de projeto/implementação).

Cabe ressaltar que quando houver diferença entre o relatório apresentado em tela e o relatório de exportação, deve ser especificado no documento de caso de uso o requisito funcional que ateste a diferença entre os relatórios.

5.3 Aferição de Pontos de Função em ambiente BPMS

Um BPMS é um Sistema no qual se pode desenvolver aplicativos integrados para gerenciamento de processos de negócios contemplando recursos como: documentação de processos, execução automatizada de processos, possibilidade de criação de indicadores

gerenciais de processos em painéis de controle, upload e trâmite de documentos eletrônicos, com possibilidade de certificação digital e integração com sistemas legados através da filosofia SOA (*Service Oriented Architecture*).



O ambiente BPMS mantém um conjunto de processos automatizado que se comunicam com os sistemas legados por meio de uma Interface de integração composta por tecnologias do tipo *webservice*, *view*, etc.

Os processos automatizados são executados dentro do ambiente BPMS. No Ibama este ambiente é proporcionado pelas ferramentas integradas Oracle.

5.3.1 Fronteira de Aplicação

Para a contagem de pontos de função de automação de processos, deve-se considerar como fronteira da aplicação apenas a interface conceitual do processo de negócio que será automatizado.

Não se deve considerar na contagem as funcionalidades que já integram e fazem parte do ambiente BPMS, a exemplo: funcionalidade de login, impressão nativa, Minhas Atividades, Meus Processos, Monitoramento de Processos, iniciação (*Start Point*) e finalização (*End Point*) dos processos.

Os grupos de dados mantidos nativamente pela ferramenta devem ser considerados como AIE – Arquivo de Interface Externa, a exemplo: cadastro de usuários e grupos organizacionais.

Não será considerada na contagem a interface de integração utilizada baseada apenas na tecnologia para acessar os dados mantidos em outras aplicações.

No caso do desenvolvimento de *webservices*, deve-se considerar:

- Como Entrada Externa aqueles consumidores de informação a ser mantida no escopo da aplicação.

- Como Consulta Externa ou Saída Externa aqueles consumidores de informação que não será mantida no escopo da aplicação.

- Como Consulta Externa ou Saída Externa aqueles provedores de informação para sistemas externos.

- Na primeira ocorrência de acesso a dados mantidos por aplicação externa, deve-se contabilizar os arquivos de interface externa associados.

Não serão consideradas na contagem funcionalidades pertencentes a outras aplicações que estão apenas sendo apresentadas na aplicação BPMS, uma vez que as regras de negócio e o processamento da informação são realizados nas aplicações de origem.

5.3.2 Critérios de Contagem

Os recursos de usabilidade não contabilizam pontos de função, pois constituem requisitos não funcionais, tais como:

- Mecanismos de navegação;
- Menus;
- Telas de Ajuda (*Help*), *tooltips*, links com manuais e roteiros;

As atividades e subprocessos que são executados fisicamente ou de modo externo à fronteira da Aplicação não contabilizam pontos de função, a exemplo: encaminhar memorando, assinar contrato.

As funcionalidades que existem exclusivamente para a manutenção de dados de código não devem ser consideradas processos elementares, assim como os dados de código não devem ser considerados como arquivos referenciados nos processos elementares que os leiam e/ou atualizem.

Deve-se considerar na contagem como um único processo elementar as Atividades modeladas em BPMN, que apesar de se apresentarem distintas no modelo, constituem uma transação completa para o negócio, com sentido de completude de determinado requisito funcional para o usuário.

Não são contabilizadas na contagem de pontos de função as atividades que foram

modeladas apenas para melhor entendimento do negócio e, que não constitui um processo elementar para o sistema, a exemplo das atividades “Selecionar Licenças” e “Renovar Licenças”. Apesar de serem duas atividades no modelo, para o negócio elas constituem apenas um processo elementar, individualmente elas não constituem uma transação completa.

Uma mesma atividade que pode ser realizada por pessoas/entidades diferentes deve ser contada apenas uma vez, a exemplo da atividade “Resolver Pendência” executada pela área jurídica, área de licitações e área técnica.

Não se deve contabilizar na contagem de pontos de função a notificação/lembrete (e-mail, sms ou outro meio) utilizadas em atividades que tenham um prazo para serem realizadas, onde após a expiração do prazo dessa notificação/lembrete é disparada.

Um exemplo consiste na atividade “Assinar Contrato” que tem um prazo de 01 (um) dia para ser realizada. Se o prazo expirar primeiro do que a conclusão da atividade, então será disparada uma atividade para notificar o responsável avisando do ocorrido e solicitando a sua realização. A notificação em si seja por email, sms ou algum outro meio, não tem sentido completo de negócio para o usuário, é apenas um procedimento integrante deste.

5.4 Aferição de Pontos de Função em ambiente de Apoio à Decisão (*Business Intelligence e Dataware House*).

Esta seção visa apresentar as diretrizes de Contagem de Pontos de Função utilizadas no Ibama quanto a aferição do tamanho funcional de ambientes de *Data Warehouses* e *Business Intelligence*. Esta abordagem é reconhecida pelo IFPUG. As definições apresentadas têm como base o artigo “*Function Points & Counting Enterprise Data Warehouses*” Release 1.0 publicado pelo IFPUG [IFPUG, 2007].

5.4.1 Fronteira de Aplicação

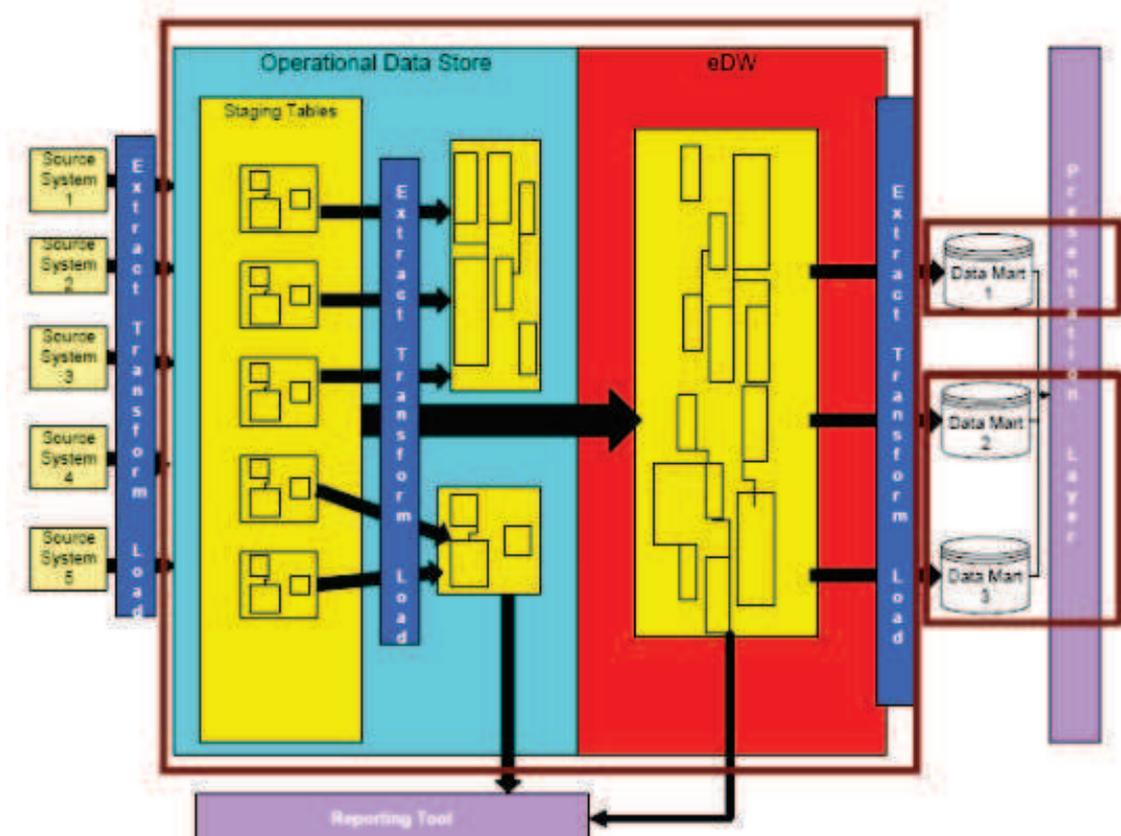
Para a contagem de pontos de função de *Data Warehouse* (DW) e *Business Intelligence* (BI) deve-se considerar os limites:

- A fronteira é determinada com base na visão do usuário. O enfoque é no que o usuário pode entender e descrever.
- A fronteira entre as aplicações relacionadas baseia-se em áreas funcionais separadas como visto pelo usuário, não em considerações técnicas.
- A fronteira inicial já estabelecida para a aplicação ou aplicações sendo modificadas não são influenciadas pelo escopo de contagem.

Não se deve considerar na contagem de pontos de função os seguintes limites da aplicação:

- Arquivos lógicos mantidos pelo(s) sistema(s) de origem, exceto aqueles referenciados nas funções transacionais;
- Tabelas Temporárias;
- *Staging Areas*;
- Tabelas de códigos;

Os *Data Marts* podem ser contados como fronteiras de aplicações separadas.



A *Staging Area*, usada para armazenar uma versão atual do *Data Warehouse* que existe no sistema original, não deve ser contabilizada na contagem pois não é reconhecida pelos usuários.

O ODS (Depósito de Dados Operacionais), área que contém dados transacionais detalhados que são tipicamente modificados, não deve ser contabilizado na contagem.

5.4.2 Critérios de Contagem

FUNÇÕES DE DADOS

Conte um Arquivo referenciado (ALR) para cada ALI mantido.

Conte um Arquivo referenciado (ALR) para cada ALI ou AIE referenciados durante o

processamento da entrada externa.

Conte um AIE para cada grupo lógico de dados que é copiado de um sistema de origem para o *Data Warehouse* sem nenhuma lógica especial de processamento.

Conte um ALI para cada tabela fato e dimensão mantidas pelo DW. Sendo que as tabelas dimensão essencialmente estáticas não devem ser contadas, pois configuram *CODE DATA*.

Não conte o mesmo grupo de dados com ALI e AIE.

Não conte duas vezes o mesmo grupo de dados como uma tabela dimensão que esteja modelado em duas estrelas do DW.

As hierarquias das tabelas dimensão devem ser consideradas como Tipo de Registro (TR) e não como ALI, a exemplo: departamento e setor.

FUNÇÕES TRANSACIONAIS

Cada processo completo de carga de dados em tabela fatos e dimensão é contado como apenas uma Entrada Externa.

Não conte três EE separadas para cada passo do processo (ex.: uma EE de Extração, uma EE de Transformação, e uma EE de Carregamento), uma vez que todos os três são requeridos para completar o processo elementar.

Conte pelo menos uma SE ou CE para cada relatório ou consulta desenvolvida e/ou suportada para satisfazer as necessidades do usuário.

As ferramentas de relatórios inerentes ao produto de BI ou DW não são contabilizadas, ou seja, as consultas e recursos gerados pelo usuário não devem ser contabilizados.

As funções de conversão de dados (carga inicial de dados) são contadas como EE.

As cargas de dados em tabelas dimensão são contadas como EE, mesmo que tais tabelas sejam consideradas Tipos de Registro (TR).

A geração de contexto de análise (Cubos) deve ser contada como SE.

5.5 Aferição de Pontos de Função em Ambiente Baseado em Geotecnologias

Sistemas baseados e geotecnologias são ferramentas utilizadas para análise de informações geográficas que usam funções de dados geométricos ligados a tabelas de atributos alfanuméricos.

5.5.1 Critérios de Contagem

Deve-se considerar os dados de uma aplicação de geoprocessamento como se estivessem em telas sem interfaces gráficas.

Deve-se considerar um *shapefile* mantido pelo usuário como um ALI.

Deve-se considerar um *shapefile* mantido por aplicação externa à fronteira e referenciado na aplicação contada como um AIE.

Os *shapefiles* que não são mantidos por nenhuma aplicação são considerados como *CODE DATA*.

Deve-se considerar as funções de transação (EE/CE/SE) relacionadas às operações em um *shapefile* mantido pelo usuário.

6. Considerações Especiais para Planejamento e Acompanhamento de Projetos

Este capítulo tem como propósito apresentar diretrizes para o planejamento e acompanhamento de projetos com o auxílio da métrica Ponto de Função e de técnicas relacionadas. Com base nesta finalidade é descrito um processo de estimativas de projetos de software aderente à área de processo de Planejamento de Projeto do CMMI (*Capability Maturity Model Integration*). Nesse contexto, é apresentado o método Contagem Estimativa de Pontos de Função (CEPF) para estimar o tamanho dos projetos de software em PF. Também são apresentadas recomendações para o gerenciamento de: mudanças de requisitos, projetos cancelados e progresso de projetos, e considerações sobre redução de cronograma e fator de criticidade de solicitação de serviços.

6.1 Diretrizes para Planejamento: Estimativas de Projetos de Software

Esta seção define métodos para estimativas de projetos de software.

O principal insumo (artefato de entrada) para um processo de estimativas é o documento de requisitos. Como as estimativas devem ser realizadas no início do processo de desenvolvimento de software, então o artefato a ser utilizado é um documento inicial de requisitos, por exemplo, o documento de visão ou formalização simples de requisitos.

O estimador deve analisar os requisitos para garantir a qualidade e então estimar o tamanho do projeto de software. O próximo passo é a derivação das estimativas de prazo (cronograma) e

custo (orçamento) com base na estimativa de tamanho e nos dados históricos de projetos concluídos da organização. Neste ponto, as principais estimativas foram geradas e precisam ser documentadas. As premissas e suposições utilizadas na geração das estimativas, dentre as quais: complexidade do projeto, plataforma de desenvolvimento, tipo do projeto, percentual de evolução de requisitos, também devem ser documentadas [Hazan, 2008].

A realização das estimativas por um analista de métricas que não atue na equipe do projeto, constitui uma prática recomendada. O analista de métricas deve analisar também a consistência da documentação utilizada na estimativa. No decorrer do processo de desenvolvimento, o projeto deve ser reestimado sempre que ocorrerem mudanças significativas nos requisitos funcionais ou não funcionais. Quando o projeto é concluído, deve-se aferir e documentar o tamanho, prazo e custo, assim como outros atributos relevantes do projeto, visando a coleta de dados para a melhoria do processo de estimativas. As lições aprendidas também devem ser documentadas [Hazan, 2008].

As estimativas devem ser realizadas em no mínimo dois marcos do processo de desenvolvimento de software, a saber:

● **Estimativa inicial:** realizada após o fechamento do escopo do projeto. Geralmente é baseada em um documento inicial de requisitos como, por exemplo, o documento de visão. Constitui uma boa prática a previsão de evolução de requisitos, especialmente em projetos de desenvolvimento de médio ou grande porte. Nessa etapa é importante destacar os seguintes conceitos na área de estimativas:

- Uma Estimativa é obtida por meio de uma atividade técnica, utilizando métodos de estimativas. Não deve sofrer interferências políticas;
- A Meta é um desejo, em função de necessidades de negócio, estabelecida politicamente;
- Um Compromisso é um acordo da gerência com as equipes técnicas para alcançar uma meta [Parthasarathy, 2007].

Em um cenário ideal, os resultados da estimativa atendem às metas de negócio. Quando este cenário não é real, é fundamental a redução de escopo do projeto, de modo que a meta se adapte aos resultados da estimativa.

● **Contagem de Pontos de Função Final:** realizada após a homologação da aplicação. Esta contagem considera as funcionalidades efetivamente entregues para o usuário pela aplicação. Neste caso, deve ser aplicada a contagem detalhada.

Para fins de faturamento, deve-se considerar a Contagem Final.

É importante ressaltar que as evoluções de requisitos (*scope creep*) devem ser consideradas na oficialização de abertura das demandas. Contudo, durante a execução do projeto, se estas mudanças forem significativas, maiores que a estimativa inicial, deverão ser tratadas como novo projeto.

As subseções seguintes apresentam os métodos de estimativas de tamanho, prazo e custo a serem utilizados nos projetos de software em contratos.

6.1.1 Contagem Estimativa de Pontos de Função (CEPF)

Antes de definir o método de estimativas – Contagem Estimativa de Pontos de Função (CEPF), é importante destacar que “estimar significa utilizar o mínimo de tempo e esforço para se obter um valor aproximado dos pontos de função do projeto de software investigado” [Meli, 1999]. Assim, é recomendável sempre fazer uma distinção entre os termos e conceitos: contagem de pontos de função e estimativa de pontos de função.

- **Contagem de Pontos de Função:** significa medir o tamanho do software por meio do uso das regras de contagem do IFPUG [IFPUG, 2010];

- **Estimativa de Pontos de Função:** significa fornecer uma avaliação aproximada do tamanho de um software utilizando métodos diferentes da contagem de pontos de função do IFPUG.

O método CEPF visa aferir o tamanho em PF de maneira simplificada, com base no conhecimento dos requisitos iniciais do projeto [Hazan, 2005]. A CEPF foi definida com base nas diretrizes adotadas no método Contagem Estimada de Pontos de Função da NESMA [NESMA, 2005]. A diferença é que o método da NESMA não recomenda a análise das funções identificadas, considerando todas as funções de dados identificadas com complexidade Baixa e as funções transacionais com complexidade Média. A CEPF propõe a análise das funcionalidades identificadas, e caso não seja possível determinar a complexidade, então são adotadas as diretrizes do método Contagem Estimada da NESMA. A CEPF também apresenta algumas dicas para ajudar um estimador no mapeamento dos requisitos iniciais nos tipos funcionais da Análise de Pontos de Função. Segue a descrição da CEPF [Hazan, 2005].

Primeiramente, conforme Figura 2, os requisitos funcionais iniciais documentados nas propostas comerciais, nos documentos de visão, formalização simples de requisitos ou em qualquer especificação inicial do sistema do usuário são mapeados nos tipos funcionais da Análise de Pontos de Função: Arquivo Lógico Interno (ALI), Arquivo de Interface Externa (AIE), Entrada Externa (EE), Consulta Externa (CE) e Saída Externa (SE). Posteriormente, os pontos de função são associados a cada função identificada, baseando-se nas tabelas de complexidade e

de contribuição funcional do CPM (Tabela 1).

O estimador deve realizar uma leitura do documento inicial de requisitos, buscando informações relevantes para a identificação de processos elementares. O processo elementar é definido como a menor unidade de atividade significativa para o usuário. O processo elementar deve ser completo em si mesmo, independente e deixar a aplicação em um estado consistente [IFPUG, 2010]. Em outras palavras, os processos elementares são funções transacionais independentes, isto é, funções sequenciais pertencem a um mesmo processo elementar e funções independentes constituem processos elementares diferentes.

Uma vez identificado o processo elementar, o estimador deve buscar o entendimento deste para classificá-lo em Entrada Externa, Consulta Externa ou Saída Externa. Adicionalmente, o estimador deve descobrir os dados associados ao processo elementar, visando a determinação da complexidade funcional da função identificada. Caso não seja possível a identificação da complexidade das funções transacionais ou de dados deve-se utilizar a complexidade Média.

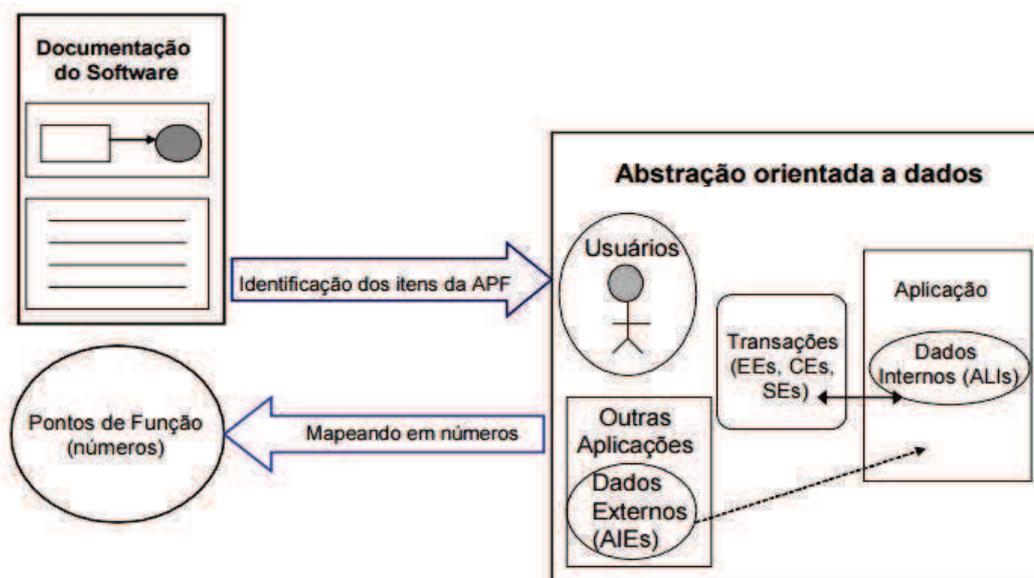


Figura 2: Modelo Lógico da Análise de Pontos de Função

A seguir são apresentadas dicas para ajudar no mapeamento dos requisitos funcionais da aplicação nos tipos funcionais da APF. As necessidades e funcionalidades especificadas para o projeto, contidas no documento inicial de requisitos, devem ser enquadradas em uma das seguintes tabelas:

Tabela 2 – Contagem dos Arquivos Lógicos Internos (ALI): banco de dados lógico da aplicação (tabelas e arquivos mantidos pela aplicação).

Considerações: Identifique os grupos de dados lógicos de aplicação nos modelos de dados

ou diagrama de classes ou a partir dos requisitos funcionais, descritos nos documentos de requisitos (documento de visão, relação de casos de uso, etc). Não considere arquivos físicos, arquivos de índices, arquivos de trabalho e tabelas de relacionamento sem atributos próprios (tabelas que existem para quebrar o relacionamento m x n e apenas transportam as chaves estrangeiras). As entidades fracas também não são consideradas um ALI. Se possível, tente descobrir os atributos lógicos, campos reconhecidos pelo usuário, e subgrupos de dados existentes para obter a complexidade funcional, segundo as regras de contagem do CPM. Caso não seja possível, deve-se utilizar a complexidade Baixa.

Nº ALI Baixa:	X 7 PF
Nº ALI Média:	X 10 PF
Nº ALI Alta:	X 15 PF
Total PF:	

Tabela 2: Identificação dos Arquivos Lógicos Internos da Aplicação

Tabela 3 – Contagem de Arquivos de Interface Externa (AIE): banco de dados de outras aplicações, **apenas referenciados** pela aplicação que está sendo estimada (tabelas e arquivos mantidos por outra aplicação).

Considerações: Identifique os grupos de dados lógicos de outras aplicações referenciados pela aplicação que está sendo estimada. Frequentemente, o referenciamento de dados ocorre para a validação de informações em cadastros ou consultas. Algumas vezes, relatórios ou consultas referenciam dados externos de outras aplicações, também considerados AIE. Não são considerados AIE arquivos físicos, arquivos de índice, arquivos de trabalho, tabelas de relacionamento sem atributos próprios e entidades fracas.

Nº AIE Baixa:	X 5 PF
Nº AIE Média:	X 7 PF
Nº AIE Alta:	X 10 PF
Total PF:	

Tabela 3: Identificação dos Arquivos de Interface Externa da Aplicação

Tabela 4 – Contagem de Entradas Externas (EE): funcionalidades que mantêm os Arquivos Lógicos Internos (ALI) ou alteram o comportamento da aplicação.

Considerações: Identifique as funcionalidades de manutenção de dados. Conte separadamente a inclusão, alteração e exclusão de dados, isto é, cada função independente de inclusão, alteração ou exclusão deve ser contada separadamente. A aplicação possui funções de entrada de dados que alteram o comportamento dela, por exemplo: processamentos *batch* ou processamento de informações de controle? Caso positivo, estas funções também devem ser identificadas como Entradas Externas. Se você não possui conhecimento sobre o processo elementar (funcionalidade analisada), considere as Entradas Externas identificadas com complexidade Média.

Nº EE Baixa:	X 3 PF
Nº EE Média:	X 4 PF
Nº EE Alta:	X 6 PF
Total PF:	

Tabela 4: Identificação das Entradas Externas da Aplicação

Tabela 5 – Contagem de Consultas Externas (CE): funcionalidades que recuperam dados de pelo menos 1 ALI ou 1 AIE e apresentam para o usuário sem a utilização de cálculos ou algoritmos. São os processos elementares do tipo “lê – imprime”, “lê – apresenta dados”, incluindo consultas, relatórios, geração de arquivos pdf, xls, *downloads*, entre outros.

Considerações: Você está desenvolvendo uma função para apresentar informações para o usuário: uma consulta, relatório, *listbox*, *download*, geração de um arquivo, geração de arquivo pdf, xls? Esta função **não** possui cálculos ou algoritmos para derivação dos dados referenciados nem altera um Arquivo Lógico Interno e nem muda o comportamento do sistema? Caso positivo, estas funções devem ser identificadas como Consultas Externas. Se você não possui conhecimento sobre o processo elementar (funcionalidade analisada), considere as Consultas Externas com complexidade **Média**.

Nº CE Baixa:	X 3 PF
Nº CE Média:	X 4 PF
Nº CE Alta:	X 6 PF
Total PF:	

Tabela 5: Identificação das Consultas Externas da Aplicação

Tabela 6 – Contagem de Saídas Externas (SE): funcionalidades que apresentam

informações para o usuário **com** utilização de cálculos ou algoritmos para derivação de dados ou atualização de Arquivos Lógicos Internos ou mudança de comportamento da aplicação. São as consultas ou relatórios com totalização de dados, relatórios estatísticos, gráficos, geração de arquivos com atualização *log*, *downloads* com cálculo de percentual, entre outros.

Considerações: Você está desenvolvendo uma funcionalidade para apresentar informações para o usuário: uma consulta ou relatório com totalização de dados, etiquetas de código de barras, gráficos, relatórios estatísticos, *download* com percentual calculado, geração de arquivo com atualização de *log*? Caso positivo, estas funções devem ser identificadas como Saídas Externas. Observe que esta função deve ter cálculos ou algoritmos para processar os dados referenciados nos arquivos lógicos ou atualizar campos (normalmente indicadores) nos arquivos ou mudar o comportamento da aplicação. Se você não possui conhecimento sobre o processo elementar (funcionalidade analisada), considere as Saídas Externas com complexidade **Média**.

Nº SE Baixa:	X 4 PF
Nº SE Média:	X 5 PF
Nº SE Alta:	X 7 PF
Total PF:	

Tabela 6: Identificação das Saídas Externas da Aplicação

A estimativa de tamanho do projeto em PF deve ser gerada com a totalização dos PF obtidos nas **Tabelas 2, 3, 4, 5 e 6**.

A fórmula de contagem ou de estimativa de pontos de função para projetos de desenvolvimento é a seguinte:

$$\text{PF_DESENVOLVIMENTO} = \text{PF_INCLUIDO} + \text{PF_CONVERSÃO}$$

6.1.2 Distribuição de Esforço por Fase do Projeto

O próximo passo é a definição da distribuição de esforço pelas macroatividades (fases) do projeto, visando definir o valor agregado ao projeto após cada fase do ciclo de vida.

A Tabela 7 apresenta a distribuição de esforço por macroatividade do Projeto.

Macroatividades do Processo de Desenvolvimento de Software	Percentual de esforço (%)
Engenharia de Requisitos	25%
Design / Arquitetura	10%
Implementação	40%
Testes	15%
Homologação	5%
Implantação	5%

Tabela 7: Distribuição de Esforço por Macroatividades do Projeto

6.1.3 Definição de Prazo de Projetos de Software

A tabela a seguir apresenta o prazo máximo de execução das OS, de acordo com o tamanho estimado do projeto constante na OS.

Tamanho do Projeto	Prazo Máximo
≤10 PF	9 dias úteis
> 10 PF e ≤ 20 PF	18 dias úteis
> 20 PF e ≤ 30 PF	27 dias úteis
> 30 PF e ≤ 40 PF	36 dias úteis
> 40 PF e ≤ 50 PF	45 dias úteis
> 50 PF e ≤ 60 PF	54 dias úteis
> 60 PF e ≤ 70 PF	63 dias úteis
> 70 PF e ≤ 85 PF	70 dias úteis
> 85 PF e ≤ 100 PF	79 dias úteis
> 100 PF e ≤ 125 PF	101 dias úteis
> 125 PF e ≤ 150 PF	123 dias úteis
> 150 PF e ≤ 175 PF	131 dias úteis
> 175 PF e ≤ 200 PF	138 dias úteis
> 200 PF e ≤ 225 PF	144 dias úteis
> 225 PF e ≤ 250 PF	149 dias úteis
> 250 PF e ≤ 275 PF	155 dias úteis
> 275 PF e ≤ 300 PF	160 dias úteis
> 300 PF e ≤ 350 PF	167 dias úteis
> 350 PF e ≤ 400 PF	175 dias úteis
> 400 PF e ≤ 450 PF	183 dias úteis
> 450 PF e ≤ 500 PF	191 dias úteis

> 500 PF e ≤ 600 PF	200 dias úteis
---------------------	----------------

Tabela 8: Definição de Prazo de Projetos para OS de desenvolvimento ou manutenção evolutiva.

O prazo considera todo o ciclo de vida do projeto, desde a fase de requisitos até a implantação. Demandas com volume maior que 600 Pontos de Função deverão ser divididas em OS de volumes menores e adequadas à tabela apresentada.

Caso seja necessário receber o projeto em um prazo menor, recomenda-se propor um processo de desenvolvimento incremental, priorizando funcionalidades em cada iteração de acordo com a necessidade dele. Caso, ainda assim, o prazo definido não atenda às necessidades do cliente, pode-se reduzi-lo prazo em até 25%, observando-se o conceito de Região Impossível proposto por Capers Jones [Jones, 2007]. Quanto mais perto da Região Impossível, o esforço e o custo do projeto aumentam de maneira exponencial. Assim, a redução de prazo de 10% implica no aumento de custo do esforço de 20%; a redução de prazo de 20% implica no aumento de custo do esforço de 50%; a redução de prazo de 25% implica em um aumento de custo do esforço de 70%. Não será permitida a redução de prazo superior a 25%, dada a Região do Impossível.

6.2 Diretrizes para Acompanhamento de Projetos

Esta seção apresenta considerações especiais sobre o gerenciamento de mudança de requisitos, projetos cancelados, progresso de projetos, assim como o tratamento de redução de cronograma e fator criticidade.

6.2.1 Considerações sobre Mudança de Requisitos

Em projetos de desenvolvimento e de manutenção de software é bastante observada a mudança de requisitos anterior à implantação do projeto, conforme o usuário e o desenvolvedor adquirem mais conhecimento sobre as necessidades e funcionalidades de negócio [Sommerville, 2007]. O CPM denomina este fenômeno de *Scope Creep*.

Nas estimativas iniciais de tamanho de projetos de desenvolvimento, após a fase de especificação, considerando-se o documento de visão inicial do projeto, será utilizado um percentual de 35% para evolução de requisitos. Por exemplo, suponha que após a análise do documento de visão de um projeto, aplicando-se a CEPF, foi obtido o tamanho de 200 PF, então o tamanho estimado desse projeto é de 270 PF (200 + 35%).

Uma mudança de requisito anterior à implantação do projeto gera retrabalho para a equipe de desenvolvimento, aumentando assim o esforço e o custo do projeto. Neste roteiro, as

mudanças de requisitos serão tratadas como uma nova demanda, em OS adicional, de forma que o projeto inicial deverá ser executado conforme o requisito anteriormente aprovado, mantendo-se os prazos e custos contratados. Desta forma, a OS adicional será estimada e seguirá o fluxo normal de uma demanda de desenvolvimento.

Cabe ressaltar que para evitar as solicitações de mudança de requisitos devido a falhas na execução da fase de engenharia de requisitos, é importante que seja dada atenção especial à atividade de validação e aceitação dos requisitos.

6.2.2 Considerações sobre Projetos Cancelados

Em alguns casos, devido a mudanças no ambiente da contratante, uma demanda ou parte de um projeto de desenvolvimento ou manutenção pode ser cancelado a critério da contratante. Nestes casos, o tamanho funcional das funcionalidades canceladas será aferido por meio da contagem de pontos de função das funcionalidades canceladas e um fator de impacto.

O fator de impacto é definido com base no percentual de esforço alocado à construção da funcionalidade em questão, observando a Tabela 7 de distribuição de esforço contida na subseção 6.1.2, considerando apenas o esforço das fases concluídas.

É importante ressaltar que em um processo de desenvolvimento incremental uma funcionalidade pode, por exemplo, estar em fase de requisitos e de testes, porque o plano de testes é construído na fase de requisitos, contudo, apenas serão consideradas na definição do percentual de esforço as fases de fato concluídas.

Segue um exemplo de aferição do tamanho funcional de um projeto cancelado:

Requisito	Tamanho	Engenharia de Requisitos	Design, Arquitetura	Implementação	Testes	Homologação	Implantação
Caso de Uso Relatório de Projetos	5 PF	100%	100%	50%	20%	0%	0%

O projeto encontrava-se com a engenharia de requisitos e design/arquitetura concluídos, a implementação estava na metade da execução e os testes estavam em 20% realizados.

Supondo o cancelamento do projeto por iniciativa da contratante, a contagem de PF a ser apurada é a seguinte:

Caso de Uso – Relatório de Projetos – 5 PF

Macroatividade	Esforço da Fase	Tamanho	Esforço realizado	Tamanho realizado
Engenharia de Requisitos	25%	1,25 PF	100%	1,25 PF
Design, Arquitetura	10%	0,5 PF	100%	0,5 PF
Implementação	40%	2 PF	50%	0PF
Testes	15%	0,75 PF	20%	0PF
Homologação	5%	0,25 PF	0%	0 PF

Implantação	5%	0,25 PF	0%	0 PF
Total:				1,75 PF

O tamanho realizado é de 1,75 PF, pois conforme definido neste roteiro, são consideradas apenas as fases concluídas.

6.2.3 Considerações sobre Redução de Cronograma

Alguns projetos, devido à legislação e a outros fatores externos, já se iniciam com um prazo imposto.

No entanto, se o projeto tiver um prazo imposto inferior ao prazo definido, conforme apresentado na subseção 6.1.3, então o requisitante deverá registrar a motivação para a abertura de OS com prazo reduzido. Os prazos reduzidos acarretam aumento de custos, conforme a definição da tabela 10:

Redução de prazo	Aumento de custo (PF)
1%	2%
2%	4%
3%	6%
4%	8%
5%	10%
6%	12%
7%	14%
8%	16%
9%	18%
10%	20%
11%	23%
12%	26%
13%	29%
14%	32%
15%	35%
16%	38%
17%	41%
18%	44%
19%	47%
20%	50%
21%	55%
22%	60%
23%	65%
24%	70%
25%	75%

Tabela 9: Percentuais de redução de prazo e aumento de custo.

Deve-se ressaltar que não é possível uma redução de prazo maior que 25%, devido aos

cálculos de Região Impossível e ainda, conforme nos aproximamos da Região Impossível, o esforço e o custo do projeto aumentam de maneira exponencial.

Como os riscos da redução de cronograma também são altos, não é recomendada a redução de cronograma. Deve-se tentar priorizar funcionalidades trabalhando com o processo incremental.

Não é recomendado o uso de redução de cronograma, sugere-se trabalhar com definição de prioridades.

7. Atividades Sem Contagem de Pontos de Função

Deve-se ressaltar que o processo de desenvolvimento de soluções possui várias atividades que precisam ser consideradas, mas que não serão aferidas pela métrica de Pontos de Função e, portanto, não fazem parte deste roteiro. São atividades categorizadas nessa condição:

- Treinamentos em Tecnologia da Informação: são as demandas de treinamentos em linguagens de programação, ferramentas de gestão, processos, modelos da qualidade, métricas, etc.

- Desenvolvimento de Cursos para EaD: são as demandas de desenvolvimento de um curso na modalidade de Ensino a Distância (EaD).

- Mapeamento de Processos de Negócio: são as demandas de elaboração de documentação contendo o mapeamento de processos de negócio de uma organização ou de parte de uma organização.

- Elaboração de Plano Diretor de Tecnologia da Informação (PDTI): são as demandas para elaboração de PDTI para clientes.

- Administração de Dados: este serviço requer uma equipe de administradores de dados (AD) dedicada para atender as demandas associadas à definição e manutenção do modelo de dados de negócio do cliente. Esta equipe fica disponível em horário comercial ou horário específico estabelecido no contrato para atendimento das demandas. Assim, estes serviços não possuem contagem de PF associada. **É importante ressaltar que as atividades de banco de dados associadas ao projeto de desenvolvimento ou de manutenção, por exemplo, preparação de ambiente (testes, homologação, implantação), desempenhadas pelos DBA da equipe de desenvolvimento, já estão consideradas dentro do projeto de software, não cabendo cobrança adicional.**

- Consultoria: São serviços especializados e tratados em contratações em separado, como

consultoria em métricas e consultoria para apoio na contratação de soluções de TI.

Essa relação não é taxativa, uma vez que existem outras atividades que precisam ser consideradas, mas que não serão aferidas pela métrica de Pontos de Função.

8. Processo de Revisão do Roteiro de Contagem

8.1 Revisão para Correção de Inconsistências e Situações não Previstas

A revisão deste roteiro será feita sempre que se verificarem inconsistências entre uma definição do CPM e uma regra constante deste documento, e situações não previstas neste roteiro. Essas situações, sempre que necessário, serão documentadas, gerando novas versões deste roteiro.

8.2 Revisão para Adoção de Novas Versões do CPM e do Roteiro de Métricas de Software do SISP

A adoção de nova versão do CPM ou do Roteiro de Métricas de Software do SISP como referência para este roteiro de contagem não será imediata às suas publicações. Nesse caso, haverá uma avaliação das novas versões para se decidir sobre a atualização deste documento. A atualização do roteiro para utilização em contrato deve ser negociada com a empresa contratada.

9. Dicas para otimizar o custo das manutenções

É importante que se tenha uma atenção especial na gestão de demandas de manutenção de sistemas. A seguir algumas dicas que podem ajudar a melhorar esse cenário:

- **Consolidar Manutenções na mesma funcionalidade em uma única demanda:** fazer uma manutenção para atender um único requisito ou para atender a vários requisitos na mesma função terá o mesmo tamanho funcional se elas forem solicitadas na mesma ordem de serviço. Se solicitadas em momentos distintos, as mesmas funções serão pagas várias vezes, para cada um dos projetos de melhoria/manutenção.

- **Análise crítica dos requisitos:** em muitas situações é possível ter uma única função que faça o papel de duas ou mais funções existentes. É muito comum existirem consultas e

relatórios com objetivo semelhante mas que apresentam uma diferença mínima entre os campos (atributos) apresentados. Nesses casos, é importante analisar criticamente os requisitos e, se possível, optar por uma transação mais completa e evitar a criação de funções semelhantes. É mais barato pagar pela criação de uma função nova, do que por uma manutenção em duas ou mais funcionalidades, principalmente a longo prazo.

10. Conclusão

Este documento apresentou um roteiro para o dimensionamento de tamanho de vários tipos de projetos de software, visando a aderência desses tipos de projetos desenvolvidos na instituição às diretrizes da Instrução Normativa SLTI nº 4, de 11 de Setembro de 2014. A estimativa de tamanho utiliza a métrica Ponto de Função Não Ajustado como unidade de medida, conforme recomendado nos Acórdãos 1.910/2007, 2.348/2009 e 1.647/2010 do Tribunal de Contas da União (TCU) e na Portaria SLTI/MP Nº 31, de 29 novembro de 2010.

É importante ressaltar que o uso de métricas em contrato de software é uma boa prática, visando proporcionar uma gestão efetiva dos contratos com base em dados quantitativos e objetivos. A implantação desta modalidade de contrato implica na definição de processos de gestão de requisitos e de gestão de projetos baseados nas melhores práticas.

Como trabalho futuro, recomenda-se a revisão e atualização deste roteiro sempre que for verificada inconsistência entre alguma definição do IFPUG publicada em versões futuras do CPM ou em *White Paper*, ou quando for detectado um novo tipo de serviço associado ao desenvolvimento de software não previsto neste roteiro.

11. Referencias Bibliográficas

- [FNDE.2010] FNDE. Guia de Contagem do FNDE, 2010
- [SISP 2.0] SISP. Roteiro de Métricas de Software do SISP. Versão 2.0, 2012
- [IFPUG, 2010] IFPUG. *Counting Practices Manual*. Version 4.3.1, January, 2010.
- [Jones, 2007] JONES, C. *Estimating Software Costs*. Second Edition, Mc Graw Hill, 2007.

Orientações para preenchimento da Planilha de Contagem

A planilha está dividida em três partes: Identificação, Contagem e Fundamentação (Anexos I, II, III).

Cada aba possui campos específicos que devem ser preenchidos adequadamente, conforme descrito nos anexos a seguir.

Observe nos anexos I e II que os campos de preenchimento automático (fundo cinza claro) não devem ser alterados, uma vez que seus valores são carregados com base em outros campos da planilha.

ANEXO 1 – Identificação da Planilha de Contagem de Ponto de Função

 Planilha de Contagem de Ponto de Função Versão 1.0			
Identificação da Contagem			
Empresa:		R\$/PF:	
Nome do Sistema:		PF:	
Nº da OS:		Custo:	R\$
Elaborador:		Data da Contagem:	DD/MM/AAAA
Revisor:		Data da Revisão:	DD/MM/AAAA
Tipo de Contagem	() Estimada	Tipo de Atividade:	() Manutenção
	() Detalhada		() Desenvolvimento
Histórico de Versão			
Versão	Data:	Descrição:	Responsável:
Propósito da Contagem			
Escopo da Contagem			

A primeira parte é a identificação da contagem. Nela são apresentadas as informações sobre a contagem, empresa, número da OS, o propósito da contagem, o tipo de contagem, etc.

ANEXO 2 – Planilha de Contagem de Ponto de Função

 <div style="text-align: center;"> Planilha de Contagem de Ponto de Função Versão 1.0 </div>									
Empresa:				PF:					
Nome do Sistema:				Nº da OS:					
Elaborador:				Revisor:					
Tipo de Projeto	Fator de Ajuste	Função	Tipo	DER	ALR/RL	Compl.	PF	Fund.	OBS

A segunda parte contém todos os itens necessários para a medição de pontos de função:

- **Campos automáticos:** Empresa, PF, Nome do Sistema, Nº da OS, Elaborador e Revisor.
- **Tipo do Projeto:** É o fato gerador para uma determinada função ter entrado no escopo da contagem. Ex: Desenvolvimento, Manutenção de Interface, Verificação de Erros, etc.
- **Fator de Ajuste:** Fator referente ao Tipo do Projeto. Exemplo: 0,20 é o fator de ajuste para Verificação de erros.
- **Função:** Preencha com o nome da função, seja ela uma função de dados ou uma função de transação. Se for função de dados, deve-se utilizar um substantivo ou grupo de palavras que identifiquem o grupamento de dados (ex: Pessoa, Remediador, Empreendimento, Unidade Organizacional, Processo), e não utilizar o nome físico da tabela associada. Se for função transacional, iniciar o nome da função com verbo significativo, ex: Incluir funcionário, Listar Empreendimentos, Exportar Dados do Relatório de Transferências, Visualizar Requerimento, Consultar Processo, Editar Documento, Visualizar dados do Documento (consulta implícita).
- **Tipo:** É o tipo de função. Preencha com as siglas abaixo:
 1. Entrada Externa – EE
 2. Saída Externa – SE

-
3. Consulta Externa – CE
 4. Arquivo Lógico Interno – ALI
 5. Arquivo de Interface Externa – AIE

- **DER:** Quantidade de DER necessária na determinação da complexidade de uma função de dados. A relação textual de DERs da função deve ser detalhada em caixas de comentário flutuante nas células correspondentes, ou em outro local que se avalie apropriado, mantendo-se a legibilidade e correspondência com o registro.

- **ALR/RL:** Quantidade de Arquivos Lógicos Referenciados ou quantidades de registros lógicos. A relação textual de ALR/RL da função deve ser detalhada em caixas de comentário flutuante nas células correspondentes, ou em outro local que se avalie apropriado, mantendo-se a legibilidade e correspondência com o registro.

- **Complexidade:** Calculado pela planilha de acordo com a quantidade de DER e ALR/RL e conforme o Tipo de Função.

- **PF:** Contribuição dos Pontos de Função da medição da respectiva função. Também calculado pela planilha.

- **Fundamentação:** número da tela de fundamentação (Anexo III).

- **Observação:** campo opcional para comentários adicionais. Utilize este espaço para documentar qual o insumo que teve para identificar a função relativa àquela linha. Qual caso de uso, qual relatório, enfim, algo que permita a você justificar a contagem com agilidade.

ANEXO 3 – Fundamentação da Contagem de Ponto de Função

 <div style="text-align: center;"> Planilha de Contagem de Ponto de Função FUNDAMENTAÇÃO </div>		
Fundamentação da Contagem		
ID	Título	Imagem tela

Na terceira parte são anexadas as telas de fundamentação para cada função transacional do anexo II:

- **ID:** Número da Tela;
- **Título:** Título da Tela;
- **Imagem tela:** imagem da tela correspondente a função transacional. A imagem pode ter o formato jpg, png ou gif.

Exemplo para a aba de fundamentação:

Fundamentação da Contagem		
ID	Título	Imagem Tela
1	Listar Instituição	