



NORMA DE PROCEDIMENTO - PRODEST Nº 004

Tema:	Desenvolvimento e Manutenção de Sistemas				
Emitente:	Instituto de Tecnologia da Informação e Comunicação do Espírito Santo – PRODEST				
Sistema:				Código:	
Versão:	2	Aprovação:	Instrução de Serviço Nº 085-N de 14/10/2019	Vigência:	15/10/2019

1. OBJETIVO

- 1.1** Estabelecer procedimentos para atender às demandas de desenvolvimento de novas aplicações e de manutenção nos sistemas existentes.

2. ABRANGÊNCIA

- 2.1** Instituto de Tecnologia da Informação e Comunicação do Estado do ES - Prodest.

3. FUNDAMENTAÇÃO LEGAL

- 3.1** Não se aplica.

4. DEFINIÇÕES

4.1 Sistema (software de sistema)

No contexto desta normativa, pode ser definido como um conjunto de programas (aplicações) de computador que interagem para atingir um objetivo em comum. Uma solução abrangente que envolve várias partes interligadas, oferecendo um composto de funcionalidades para atender as necessidades do usuário.

4.2 Aplicação

Programa de computador que permite ao usuário realizar uma tarefa específica, geralmente relacionada a processamento de dados.

4.3 Diagnóstico de Demanda

Compreende as atividades necessárias à identificação de problema ou necessidade de software, análise de viabilidade, especificação de requisitos, proposição de soluções, estimativa de tamanho funcional para o escopo definido, planejamento de projeto e mapeamento de riscos para as necessidades de novos desenvolvimentos ou de manutenções preventivas, evolutivas e adaptativas.

4.4 Manutenções



4.4.1 Manutenção Adaptativa

Compreende toda alteração de caráter impositivo, que visa adaptar o software a uma nova realidade ou novo ambiente externo. Assim, manutenções do tipo adaptativas referem-se à adequação do sistema a mudanças legislativas, mudanças em regras ou procedimentos, mudanças tecnológicas ou a quaisquer outras necessárias a adequar o sistema ao ambiente alterado.

4.4.2 Manutenção Evolutiva

Toda mudança de caráter eletivo em requisitos funcionais do sistema em ambiente de produção, que implique em inclusão, alteração e/ou exclusão de funcionalidades.

4.4.3 Manutenção Preventiva

As mudanças relacionadas à reengenharia, onde se busca melhorar a confiabilidade ou oferecer uma estrutura melhor para futuras manutenções. Por meio de observações da equipe técnica, busca-se agir preventivamente para evitar erros futuros no software ou facilitar outras manutenções.

4.5 Tamanho Funcional

Métrica de medida de software obtida pela totalização de suas funcionalidades, sendo que a mais difundida tem como unidade o Ponto de Função (PF).

4.6 Contagem Indicativa

Estima o tamanho em pontos de função de um sistema baseado apenas na identificação de seus arquivos lógicos e aplicação do peso de 35 PF para cada Arquivo Lógico Interno (ALI) e 15 PF para cada Arquivo de Interface Externa (AIE).

4.7 Contagem Estimativa

Estima o tamanho em pontos de função de um sistema baseado na identificação de todas as suas funções, sem a necessidade de identificar suas complexidades. Nesse caso, a complexidade para os arquivos é tratada como baixa e para as transações, tratada como média.

4.8 Deploy

Deploy é a atividade de implantação do software. Consiste em disponibilizar a aplicação para utilização ou testes em um determinado ambiente.

5. UNIDADES FUNCIONAIS ENVOLVIDAS

5.1 Gerência de Sistemas de Informação – GESIN

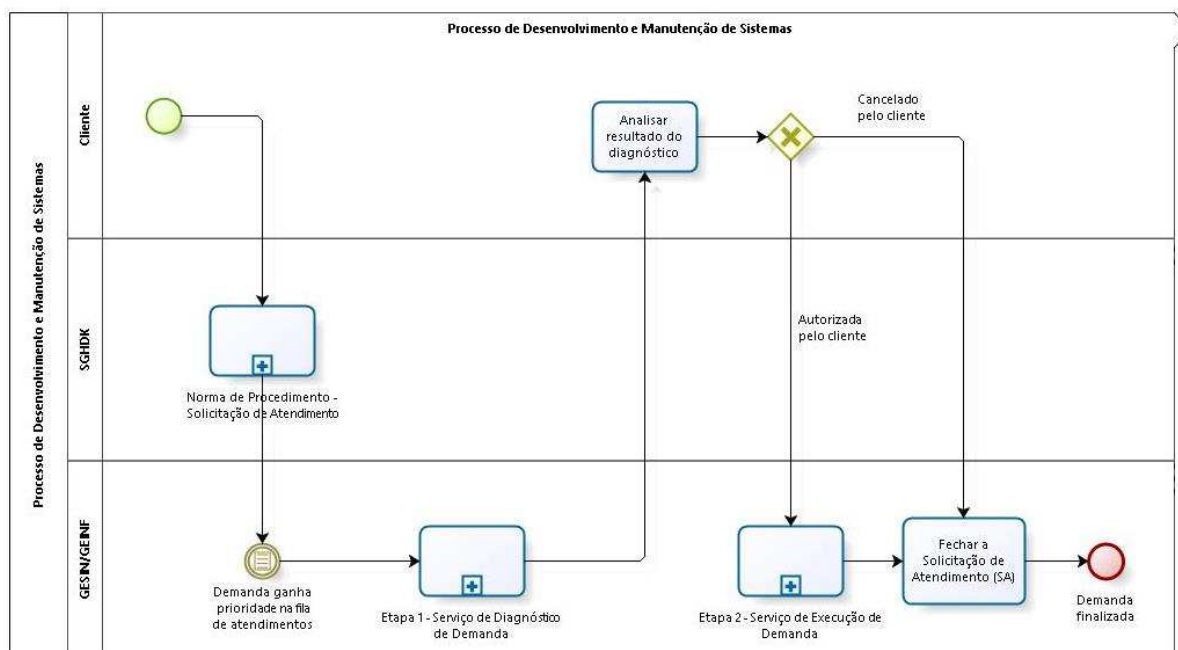
5.2 Gerência de Gestão da Informação – GEINF

5.3 Gerência de Suporte – GESUP

5.4 Subgerência de Help Desk – SGHDK

5.5 Subgerência de Integração – SGITG

6. PROCEDIMENTOS

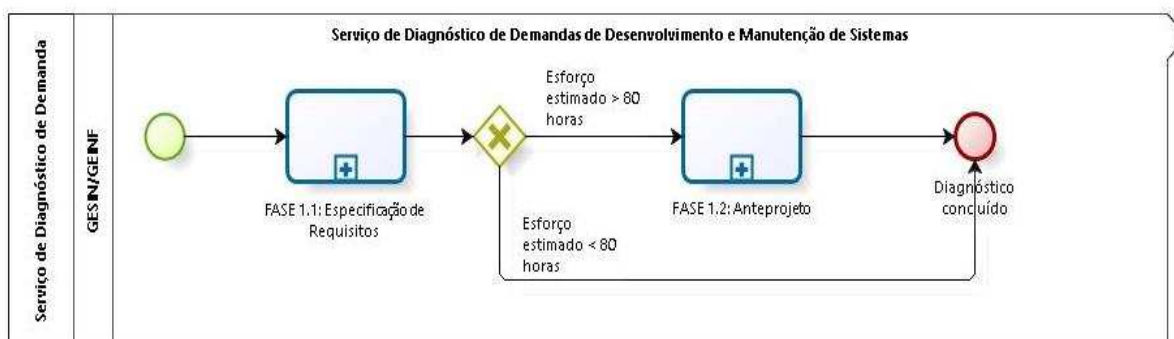


Powered by
bizagi
Modeler

6.1 Etapa 1 – Serviço de Diagnóstico de Demanda

Esta etapa compreende as atividades necessárias à identificação de problema ou necessidade de software, análise de viabilidade, especificação de requisitos, proposição de soluções, estimativa de tamanho funcional para o escopo definido, planejamento de projeto e mapeamento de riscos para as necessidades de novos desenvolvimentos e manutenções preventivas, evolutivas e adaptativas.

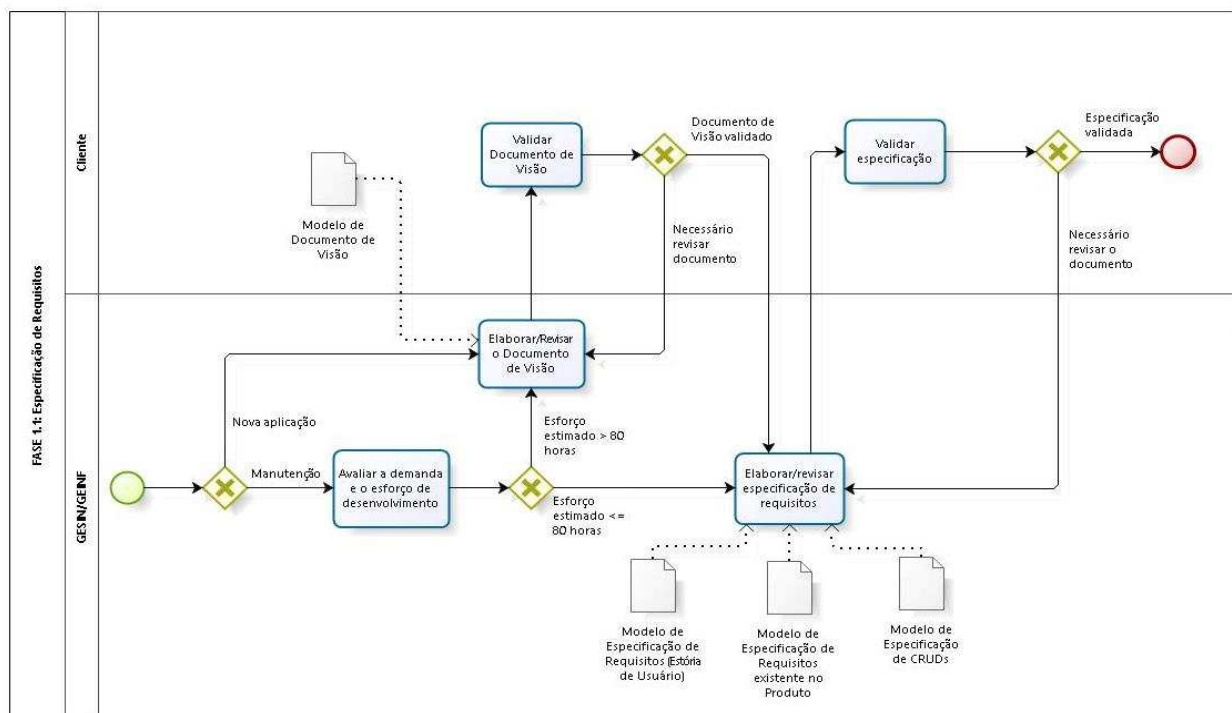
As fases da Etapa de Diagnóstico encontram-se discriminadas a seguir:



Powered by
bizagi
Modeler

6.1.1 Fase 1.1 – Especificação de Requisitos

Esta fase tem como objetivo realizar a especificação dos requisitos considerando o escopo da demanda de manutenção ou de desenvolvimento de nova aplicação.



6.1.1.1 Atividade: Avaliar a demanda e o esforço de desenvolvimento

Essa atividade trata de uma análise preliminar da demanda, onde o esforço do desenvolvimento é realizado utilizando analogia com outros projetos anteriores a partir da experiência do analista.

6.1.1.2 Atividade: Elaborar/Revisar o Documento de Visão

Essa atividade tem o objetivo de produzir a especificação do Documento de Visão, artefato que servirá para fornecer uma visão global das funcionalidades contempladas no projeto de desenvolvimento da nova aplicação ou no projeto de manutenção. Portanto, seu uso será restrito ao âmbito do projeto (da aplicação ou da manutenção), não sofrendo atualizações ao longo do ciclo de vida do produto de software.

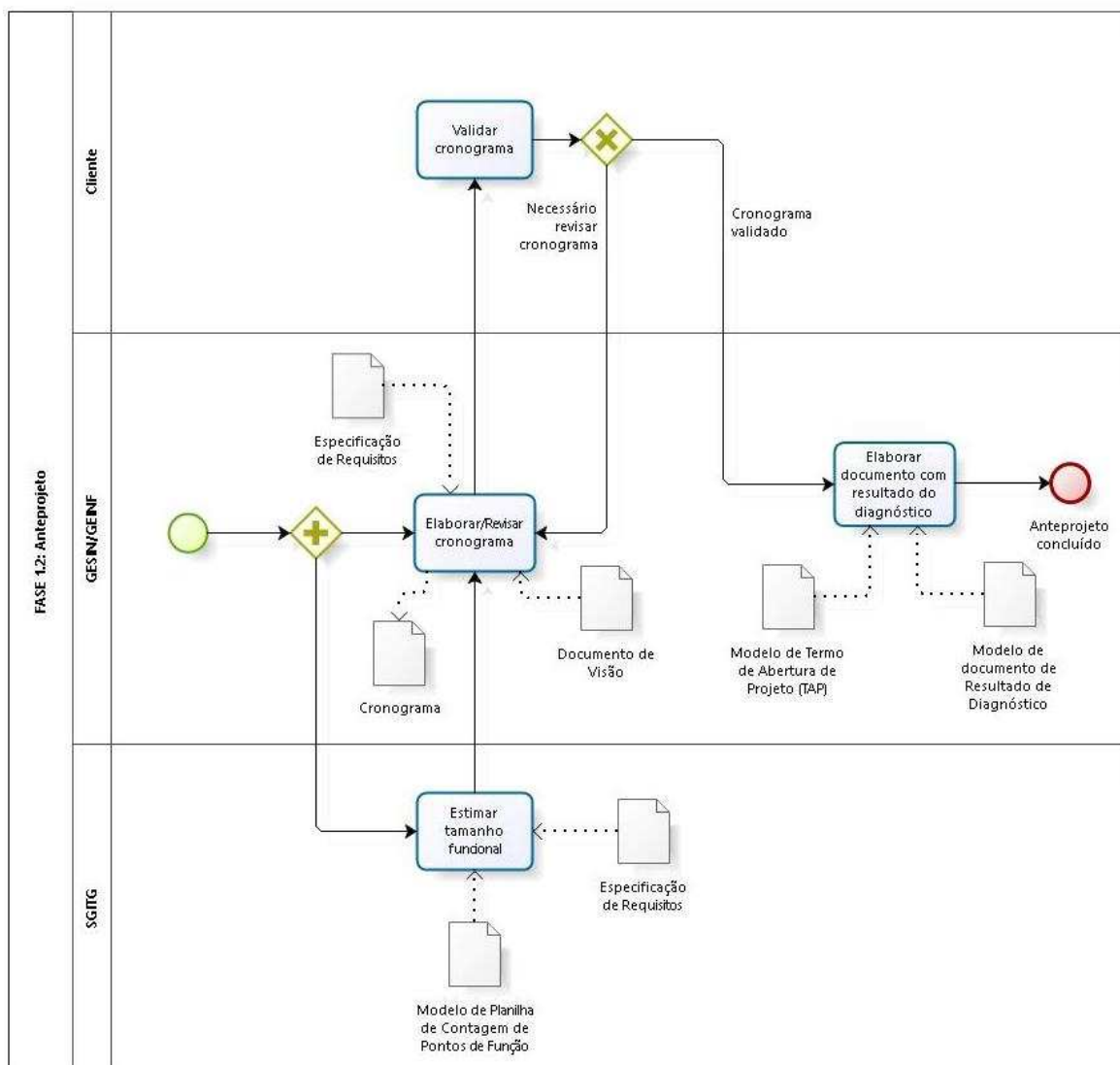
6.1.1.3 Atividade: Elaborar/Revisar a especificação de requisitos

Essa atividade não tem, necessariamente, o objetivo de esgotar todo o detalhamento dos requisitos. Em caso de projetos de nova aplicação ou de demanda de manutenção com esforço estimado superior a 80 (oitenta) horas, o que se espera é, pelo menos, um nível de detalhamento que permita a elaboração de um cronograma o mais próximo possível da realidade e com menos incertezas. Uma especificação completa deve ser buscada em caso de manutenções com esforço estimado em até 80 (oitenta) horas.

6.1.2 Fase 1.2 – Anteprojeto

O anteprojeto deve apresentar artefatos que demonstram aspectos inerentes à gestão e desenvolvimento da demanda. São artefatos mínimos que devem

ser desenvolvidos nessa fase: cronograma, planilha de contagem de pontos de função, documento com resultado do diagnóstico.



6.1.2.1 Atividade: Elaborar/Revisar cronograma

Nesta atividade, devem ser definidas as tarefas necessárias para a execução do escopo mapeado para a demanda, bem como seu sequenciamento lógico, prazos e responsáveis. Deve ser evitada a superalocação de recursos de trabalho.

6.1.2.2 Atividade: Estimar tamanho funcional

Nesta atividade, deve ser estimado o tamanho funcional do projeto de desenvolvimento da aplicação ou da manutenção aplicando a técnica de contagem indicativa ou estimativa. Neste momento, o objetivo principal é validar os prazos definidos no cronograma.

6.1.2.3 Atividade: Elaborar documento com resultado do diagnóstico

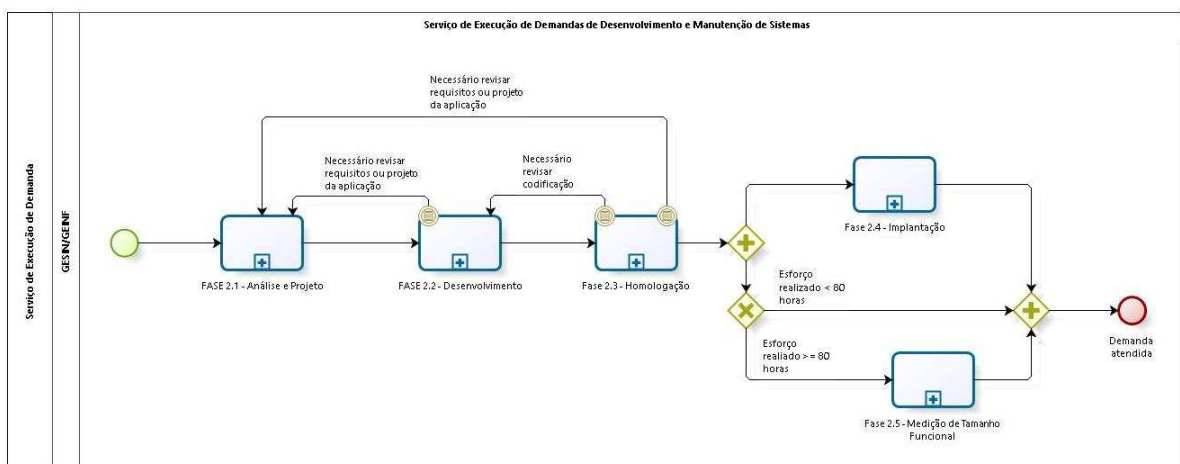
Essa atividade visa formalizar o resultado das análises realizadas na Fase



de Anteprojeto. Poderá ser adotado um dos seguintes modelos de documento: Termo de Abertura do Projeto (TAP) ou Resultado de Diagnóstico.

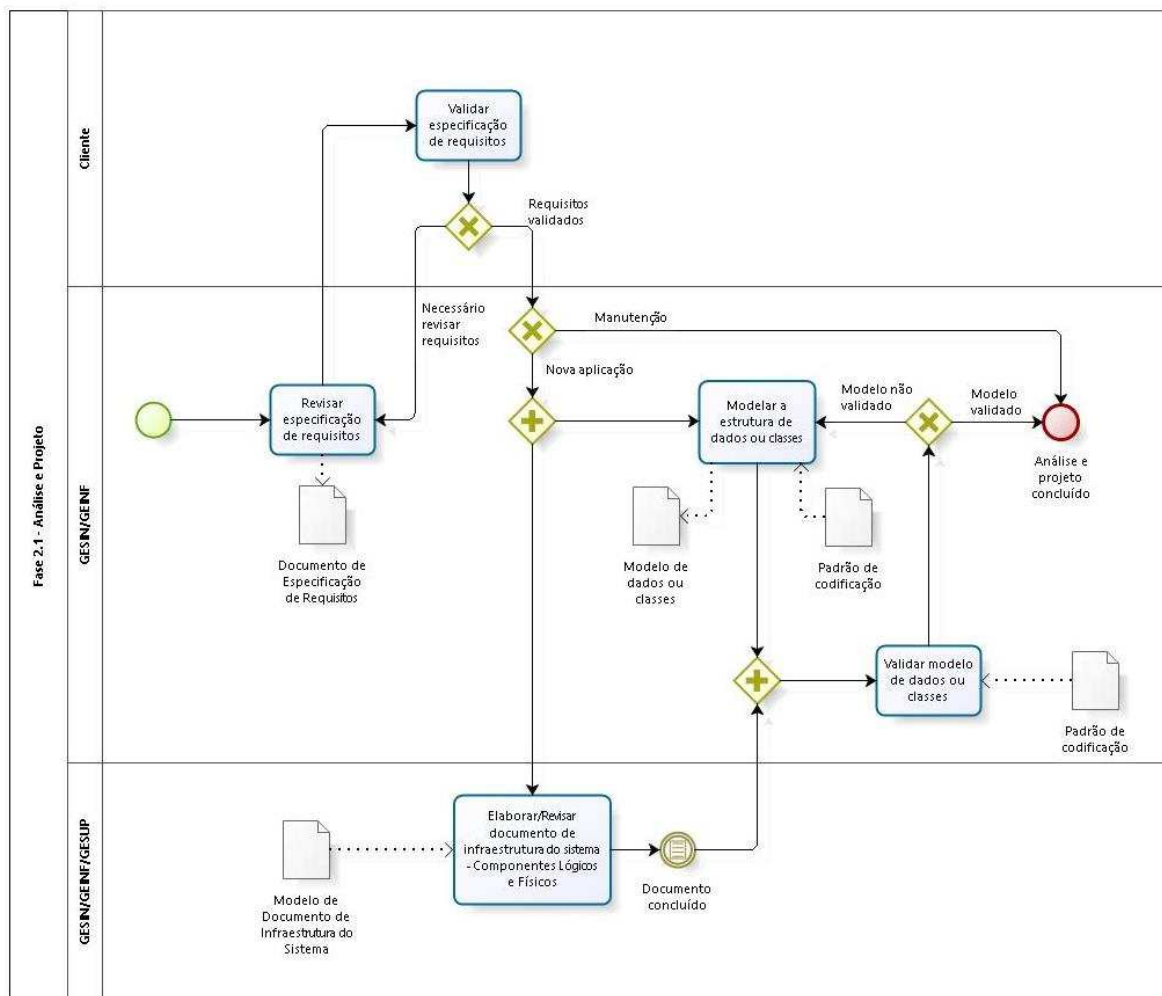
6.2 Etapa 2 – Serviço de Execução de Demanda

Esta etapa compreende as atividades necessárias para o desenvolvimento de nova aplicação ou manutenção em uma aplicação em produção.



6.2.1 Fase 2.1 – Análise e Projeto

Fase destinada a detalhar o escopo da demanda a ser atendida e a elaborar os artefatos técnicos a serem utilizados na Fase de Desenvolvimento.



6.2.1.1 Atividade: Elaborar/Revisar documento de infraestrutura do sistema – Componentes Lógicos e Físicos

Esta atividade consiste em documentar arquiteturalmente os componentes lógicos da aplicação. Deve ter um diagrama geral das camadas da aplicação e suas integrações com repositórios de dados e sistemas externos. Deve conter também um diagrama de pacotes da aplicação englobando suas camadas. Também deve prever a estrutura física necessária para a implantação do sistema. Caso o sistema tenha alguma particularidade relacionada a acessos de rede, um diagrama de redes deve ser fornecido.

6.2.1.2 Atividade: Modelar a estrutura de dados ou classes

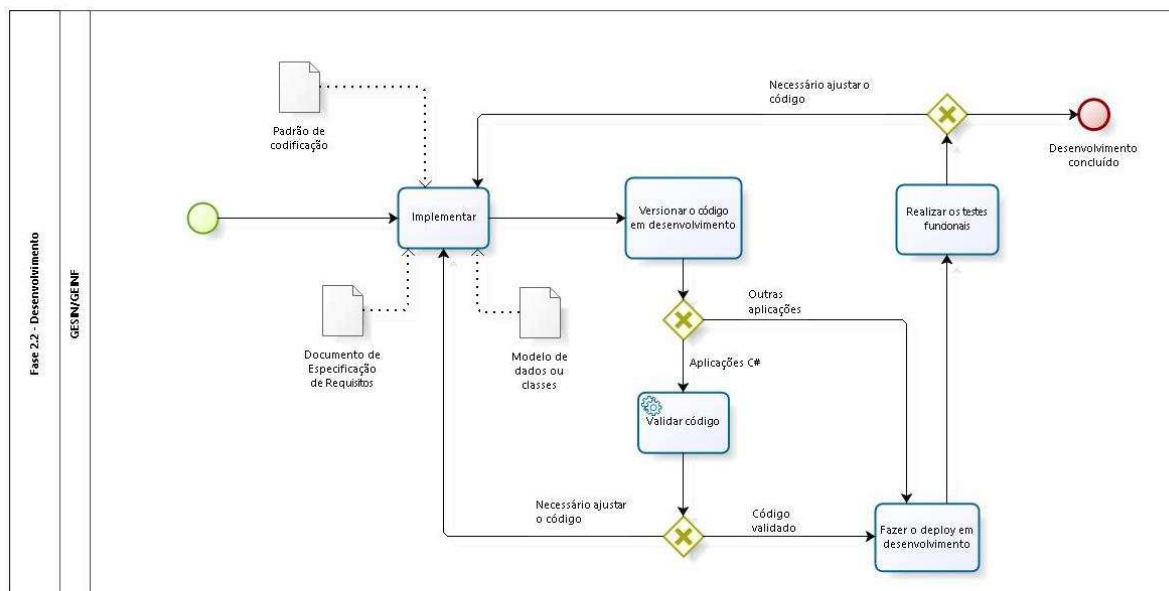
Deve ser documentada a modelagem de dados ou diagrama de classes do sistema. É importante definir os tipos dos atributos e os relacionamentos existentes entre os objetos do modelo.

6.2.1.3 Atividade: Validar modelo de dados ou classes

Nesta atividade os modelos definidos na atividade anterior serão validados e confrontados com os padrões e regras do PRODEST.

6.2.2 Fase 2.2 – Desenvolvimento

Esta fase tem como objetivo codificar as funcionalidades conforme aprovado nas fases anteriores, realizando a construção dos códigos fontes e os testes que se fizerem necessários à garantia do funcionamento adequado da solução desenvolvida.



6.2.2.1 Atividade: Implementar

Esta atividade cuida do desenvolvimento ou da manutenção da aplicação a partir dos requisitos definidos. O código das novas aplicações C# deve seguir o padrão de codificação adotado pelo PRODEST.

6.2.2.2 Atividade: Versionar o código em desenvolvimento

Deve-se manter um histórico de todas as modificações de cada linha dos arquivos do projeto desde o estado inicial até o estado atual, mantendo todas as mudanças intermediárias.

6.2.2.3 Atividade: Validar código

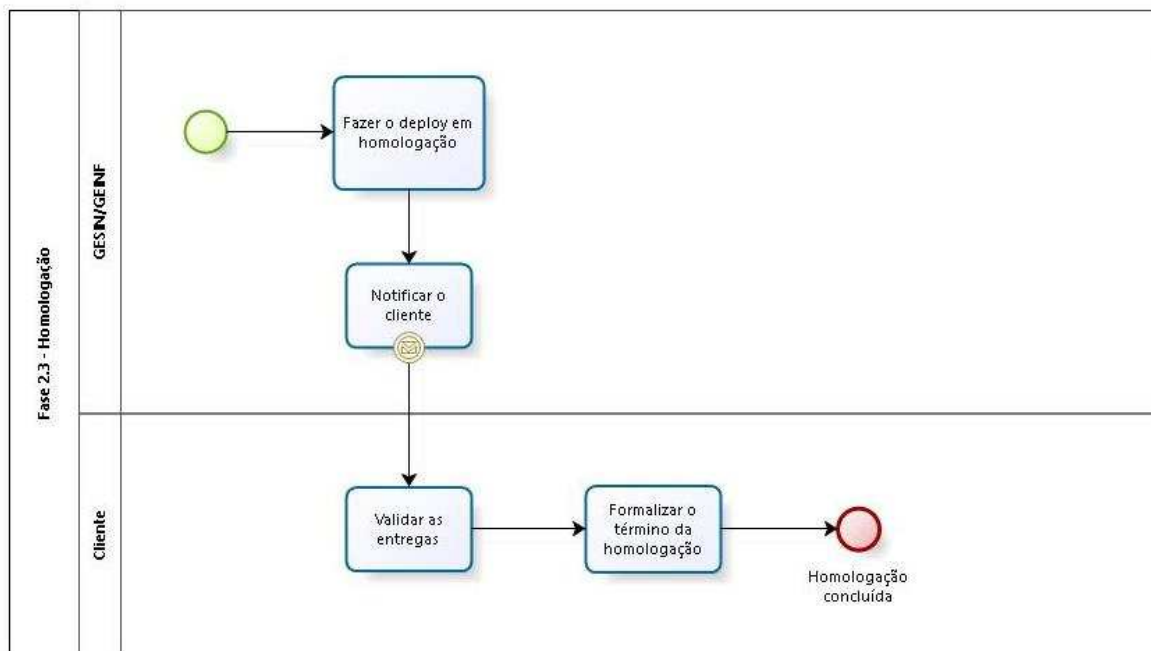
Esta atividade executará, obrigatoriamente, os testes automatizados de compilação e de qualidade de código. Outros testes recorrentes podem ser executados nesta fase, se for o caso, como testes unitários, de integração e de carga. Os testes que não atingirem a meta definida para a aplicação devem interromper o fluxo e retornar para o desenvolvimento.

6.2.2.4 Atividade: Fazer o *deploy* em desenvolvimento

Essa atividade envolve publicar o sistema no servidor do ambiente, com as configurações corretas; publicar as alterações dos modelos de dados nos bancos de dados do ambiente; publicar algum aplicativo nas lojas (se for o caso); verificar as integrações externas do sistema (armazenamento, acessos HTTP, cache distribuído); verificar se a aplicação como um todo ficou disponível.

6.2.3 Fase 2.3 – Homologação

Na Fase de Homologação as funcionalidades desenvolvidas são validadas pelo demandante (cliente), visando atestar a aderência do que foi construído ou codificado com o que foi previamente especificado.

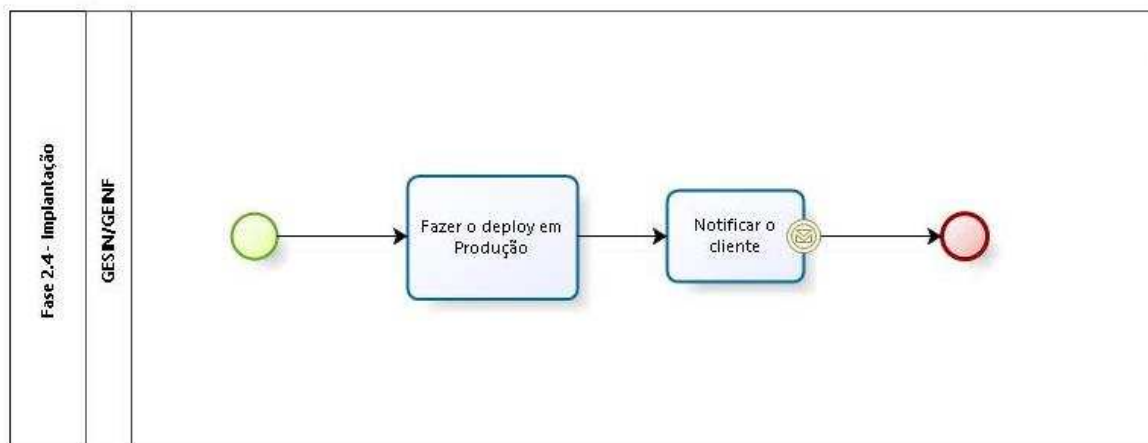


6.2.3.1 Atividade: Fazer o *deploy* em homologação

Essa atividade envolve publicar o sistema no servidor do ambiente, com as configurações corretas; publicar as alterações dos modelos de dados nos bancos de dados do ambiente; publicar algum aplicativo nas lojas (se for o caso); verificar as integrações externas do sistema (armazenamento, acessos HTTP, cache distribuído); verificar se a aplicação como um todo ficou disponível.

6.2.4 Fase 2.4 – Implantação

Nessa fase, a manutenção ou a nova aplicação desenvolvida é disponibilizada em ambiente de produção, pronta para uso.

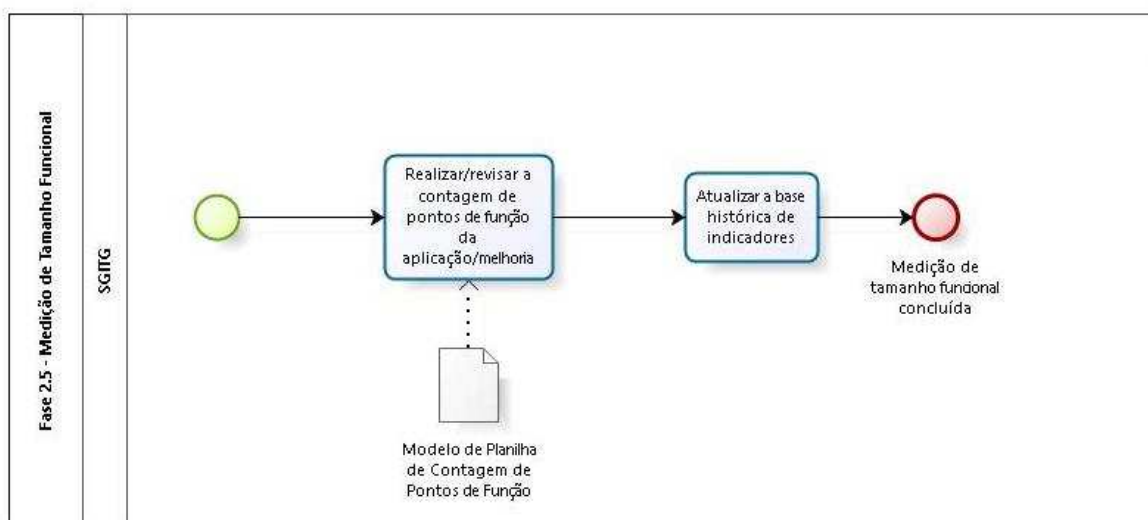


6.2.4.1 Atividade: Fazer o *deploy* em produção

Essa atividade envolve publicar o sistema no servidor do ambiente, com as configurações corretas; publicar as alterações dos modelos de dados nos bancos de dados do ambiente; publicar algum aplicativo nas lojas (se for o caso); verificar as integrações externas do sistema (armazenamento, acessos HTTP, cache distribuído); verificar se a aplicação como um todo ficou disponível.

6.2.5 Fase 2.5 – Medição de Tamanho Funcional

Nesta fase, será realizada a contagem de Pontos de Função (PF) do escopo da aplicação ou da manutenção (projeto de melhoria) entregue, com o objetivo de gerar indicadores do processo de desenvolvimento. Adicionalmente, as contagens das manutenções poderão ser utilizadas para atualizar o tamanho funcional das aplicações instaladas.





6.2.5.1 Atividade: Realizar/Revisar a contagem de pontos de função da aplicação/manutenção

Realizar a medição de tamanho funcional da aplicação ou manutenção utilizando a técnica de Análise de Pontos de Função, conforme regras de contagem previstas no Roteiro de Métricas de Software do SISP/SLTI, versão 2.2 ou superior e, na ausência de regra neste Roteiro, o Manual de Práticas e Contagens (CPM, sigla para “Counting Practices Manual”) versão 4.3.1 ou superior, publicado pelo IFPUG (sigla para “International Function Point Users Group”).

6.2.5.2 Atividade: Atualizar a base histórica de indicadores

O analista deve incluir o resultado da contagem na base de indicadores do processo de desenvolvimento.

7. INFORMAÇÕES ADICIONAIS

7.1 Não se aplica.

8. ANEXOS

8.1 ANEXO I – Modelo de Documento de Visão

8.2 ANEXO II – Modelo de Especificação de Requisitos (Estória de Usuário)

8.3 ANEXO III – Modelo de Especificação de CRUD

8.4 ANEXO IV – Modelo de Termo de Abertura de Projeto

8.5 ANEXO V – Modelo de Documento de Resultado de Diagnóstico

8.6 ANEXO VI – Modelo de Planilha de Contagem de Pontos de Função

8.7 ANEXO VII – Modelo de Documento de Infraestrutura do Sistema

8.8 ANEXO VIII – Padrão de Codificação

**Governo do Estado do Espírito Santo**

Secretaria de Estado de Gestão e Recursos Humanos – SEGER

Instituto de Tecnologia da Informação e Comunicação do Estado do Espírito Santo

**9. ASSINATURAS**

EQUIPE DE ELABORAÇÃO:	
Renato Machado Albert Gerente de Gestão da Informação	Nivaldo Helmer Júnior Gerente de Gestão da Informação
Caio Martins Barbosa Subgerente de Sistemas de Informação	Francisco de Assis Favarato Cutini Subgerente de Data Warehouse
Vinícius Gazzoli Rangel Subgerente de Projetos	Thiago Wotikoski Lourenço Subgerente de Atendimento ao Cliente
Álvaro Luiz Lago de Menezes Analista de Tecnologia da Informação	Elaborado em 10/10/2019
Marcos Tadeu de Araújo Torres Analista de Tecnologia da Informação	
APROVAÇÃO:	
Tasso de Macedo Lugon Diretor Presidente	Aprovado em 11/10/2019



[NOME DO PROJETO]

ESPECIFICAÇÃO DE VISÃO



Tipo do Documento:

Especificação de Visão

Código: PPS-EV

Revisão: 01

Data: 02/10/2015

Sumário

1. Identificação.....	3
2. Título	3
3. Objetivo	3
4. Necessidade de Negócio	3
5. Visão Atual do Negócio.....	3
6. Visão da Solução	4
7. Estórias de Usuário.....	4
7.1. <código da estória de usuário> - <nome da estória de usuário>	4
7.1.1. Narrativa	4
7.1.2. Premissas	4
7.1.3. Fora do Escopo.....	5
7.2. <código da estória de usuário> - <nome da estória de usuário>	5
7.2.1. Narrativa	5
7.2.2. Premissas	5
7.2.3. Fora do Escopo.....	5
7.3.	5
8. Anexos	5



1. Identificação

Sistema: <nome do sistema>		Módulo: <nome do módulo>
Cliente: <sigla do órgão>	Demandante: <nome do demandante>	Nº SA: <número da SA>

2. Título

<nome do projeto>

3. Objetivo

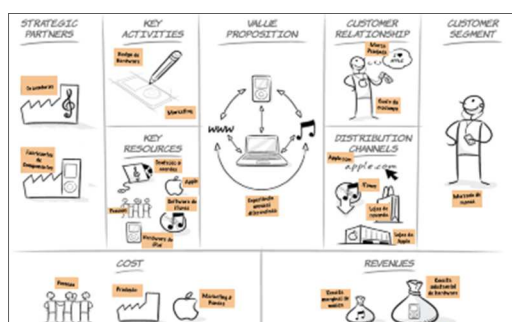
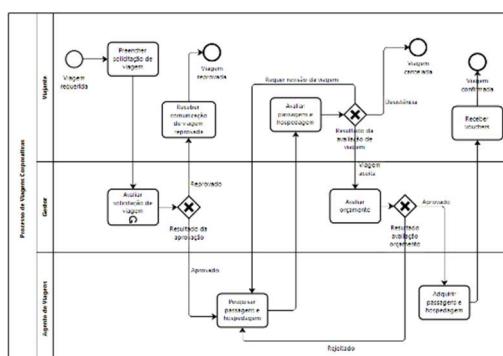
O documento tem por objetivo descrever a situação atual do negócio e fornecer uma visão geral do escopo do produto de software a ser desenvolvido.

4. Necessidade de Negócio

<Descrever o problema/necessidade que o software irá solucionar/atender> [RMA1]

5. Visão Atual do Negócio

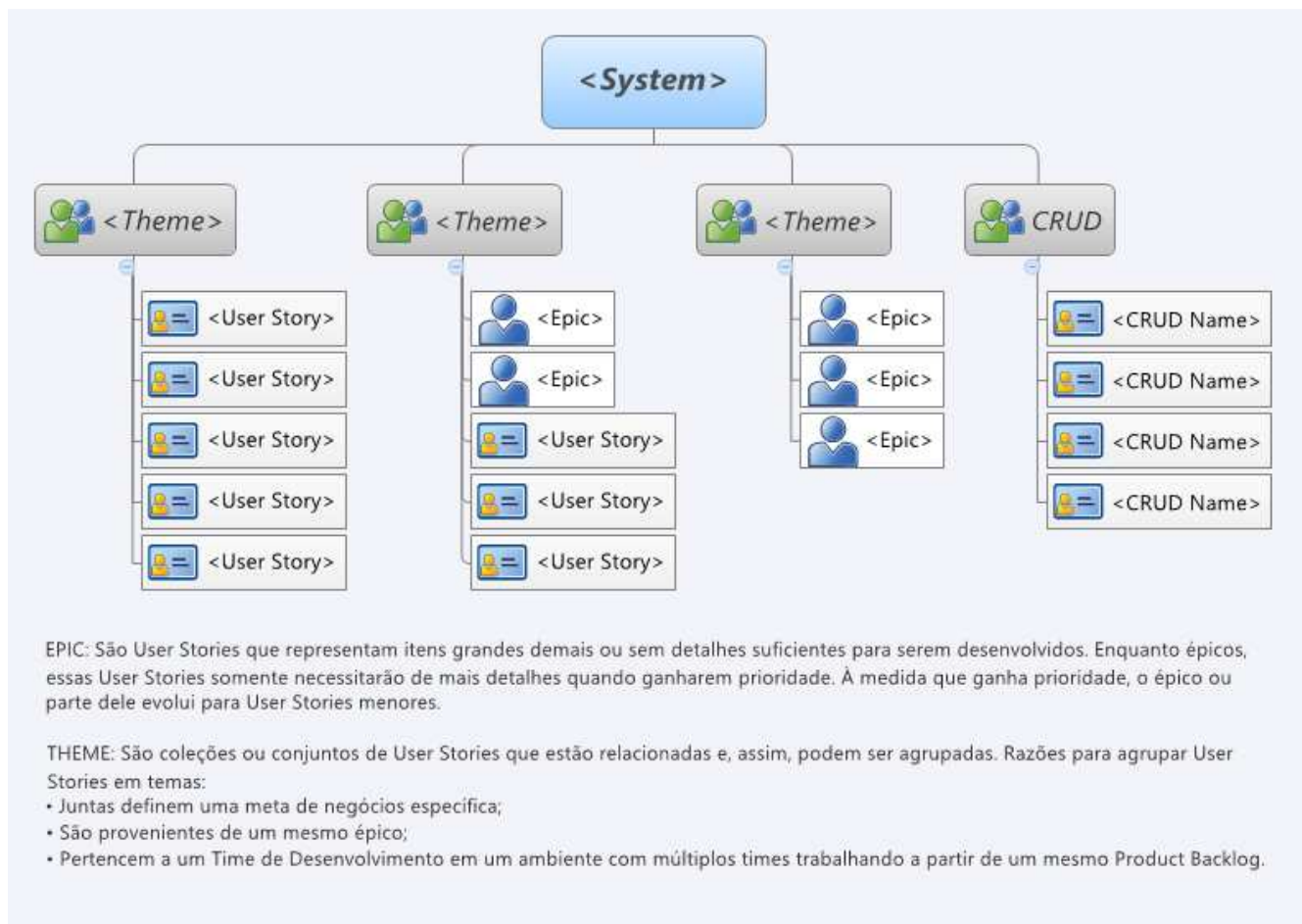
<Exemplos> [RMA2]



6. Visão da Solução

Segue o diagrama hierárquico da solução contendo o agrupamento das histórias de usuário identificadas durante o levantamento de requisitos.

<Exemplo>



7. Estórias de Usuário

7.1. <código da história de usuário> - <nome da história de usuário>

7.1.1. Narrativa

Como um <papel/ator> devo/quero/posso/gostaria <ação a ser realizada> para que <objetivo a ser alcançado> [RMA3]

7.1.2. Premissas

<itens óbvios ou que tenham relevância no sistema> [RMA4]

<formatos e validações que sejam relevantes para o cliente> [RMA5]



7.1.3. Fora do Escopo

<itens que não serão atendidos por esta estória de usuário>[RMA6]

7.2. <código da estória de usuário> - <nome da estória de usuário>

7.2.1. Narrativa

Como um <papel/ator> devo/quero/posso/gostaria <ação a ser realizada> para que <objetivo a ser alcançado>

7.2.2. Premissas

<itens óbvios ou que tenham relevância no sistema>

<formatos e validações que sejam relevantes para o cliente>

7.2.3. Fora do Escopo

<itens que não serão atendidos por esta estória de usuário>

7.3. ...

8. Anexos[FXdA7]



Governo do Estado do Espírito Santo

Secretaria de Estado de Gestão e Recursos Humanos – SEGER

Instituto de Tecnologia da Informação e Comunicação do Estado do Espírito Santo



Tipo do Documento:

Especificação de Requisitos

Código: PPS-ER

Revisão: 01

Data: 16/12/2014

1. Identificação

<código da estória de usuário> - <nome da estória de usuário>

2. Objetivo

O documento tem por objetivo registrar e controlar toda a especificação da estória de usuário em referência, que será descrita utilizando a técnica de critérios de aceite.

3. Narrativa

Como um <papel/ator> devo/quero/posso/gostaria <ação a ser realizada> para que <objetivo a ser alcançado> [RMA1]

3.1. Premissas

<itens óbvios ou que tenham relevância no sistema> [RMA2]

<formatos e validações que sejam relevantes para o cliente> [RMA3]

3.2. Fora do Escopo

<itens que não serão atendidos por esta estória de usuário> [RMA4]

4. Critérios de Aceite

Cenário <#>: <título>

Dado <pré-condição>

Quando <ação>

Então <resultado esperado>

Caso <exceções e validações>

<Exemplo:

Cenário 1: Listar Capacitação Estagiário

Dado que o gestor de estágios esteja na tela de Capacitação de Estagiário

Quando ele selecionar obrigatoriamente um órgão, um setor e um estagiário

Então ele visualiza uma tabela com todas as capacitações, exibindo os dados de carga horária, CPF do supervisor de estágio e data de conclusão da capacitação.

Caso não haja nenhuma capacitação cadastrada ele visualiza a mensagem “Nenhuma capacitação encontrada.”.

Cenário 2: Cadastrar Capacitação Estagiário



Governo do Estado do Espírito Santo

Secretaria de Estado de Gestão e Recursos Humanos – SEGER

Instituto de Tecnologia da Informação e Comunicação do Estado do Espírito Santo



Dado que o gestor de estágios tenha realizado o Cenário 1 e solicitado a inclusão de uma nova capacitação

Quando ele inserir o nome da capacitação, a carga horária, o CPF do supervisor, a data de conclusão e solicitar a gravação

Então o registro deve ser inserido e exibido um aviso de registro salvo.

Caso o gestor de estágios não informe o nome da capacitação, a carga horária, o CPF do supervisor ou a data de conclusão e solicite a gravação, deve ser exibida a mensagem de campo obrigatório nos campos que não foram preenchidos.

Caso ele tente informar uma data de conclusão posterior à data atual, não permitir a gravação do registro e informar a mensagem “A data de conclusão da capacitação deve ser anterior à data atual”.

Cenário 3: Alterar Capacitação Estagiário

Dado que o gestor de estágios tenha executado o Cenário 1

Quando ele selecionar um registro na lista de capacitação do estagiário, não será permitido alterar o campo data de conclusão. Ao alterar o texto de capacitação, a carga horária ou o CPF do supervisor e solicitar a gravação

Então o registro deve ser alterado, exibindo um aviso de registro alterado.>

5. Anexos

<Listar, conforme o caso, protótipos de telas, legislação de referência ou outros documentos relevantes>



Tipo do Documento: Especificação de Requisitos CRUD	Código: PPS-EC	
	Revisão: 01	Data: 14/09/2015

1. Definição

Os requisitos CRUD são estórias de usuários relacionadas às operações básicas dos dados (*criação, consulta, atualização e destruição*), sem necessidade de regras de negócio.

Definição (Wikipédia): CRUD (acrônimo de Create, Read, Update e Delete na língua Inglesa) para as quatro operações básicas utilizadas em bases de dados.

2. Objetivo

O documento tem por objetivo registrar todas as **entidades** que possuem estórias de usuário com “padrão” CRUD.

3. Narrativas

3.1. Criação (CREATE)

Como um **papel/ator** quero criar nova **entidade** para que dados sejam gravados.

3.2. Consulta (READ)

Como um **papel/ator** quero pesquisar uma **entidade** para que dados sejam exibidos.

3.3. Atualização (UPDATE)

Como um **papel/ator** quero atualizar a **entidade** selecionada para que dados sejam gravados.

3.4. Destruição (DELETE)

Como um **papel/ator** quero remover a **entidade** selecionada para que dados sejam excluídos.

4. Listagem de Entidades

Operações	Papel/ Ator	Nome da Entidade	Atributos da Entidade	Observações (restrições, comentários, etc.)
<Exemplos: C, R, U, D>			<Exemplos: Nome (texto), Data Criação (data), Situação (texto)...>	<Exemplos: - A deleção da entidade será lógica (ativo/inativo); - Permitir deleção somente quando não existirem entidades relacionadas; - etc.>

Legenda Operações: C = Create, R = Read, U = Update, D = Delete.

5. Anexos

<Protótipos CRUD de telas ou outros documentos relevantes>



TERMO DE ABERTURA DO PROJETO

Título:

<Título do Projeto>

Unidade Responsável:

<Sigla Gerência/Sigla Subgerência>

Coordenador:

<Nome do coordenador>

Objetivo:

<Descrição do objetivo do projeto>

Justificativa:

<Descrição da justificativa do projeto>

Descrição Geral:

<Descrição do projeto>

Premissas:

- <Premissa1>;
- <Premissa2>;
- ...

Restrições:

- <Restrição1>;
- <Restrição2>;
- ...

Parte Interessada	Representante	Papel no projeto
PRODEST	<Nome do Diretor Presidente>	Patrocinador
	<Nome do Diretor Técnico>	<Cliente> ou <Apoio à gestão>
	<Nome do Gerente da Área>	Apoio à gestão
	<Nome do Subgerente >	Apoio à gestão
<Sigla do Órgão>	<Nome do cliente>	<Cliente>
...		

Principais Riscos:

- <Risco1>;
- <Risco2>;
- ...

Principais Marcos	Data Alvo
-------------------	-----------

**Governo do Estado do Espírito Santo**

Secretaria de Estado de Gestão e Recursos Humanos – SEGER

Instituto de Tecnologia da Informação e Comunicação do Estado do Espírito Santo



1. <Marco1>	
2. <Marco2>	
3. ...	
Estimativas	
Custo	Prazo
<NA> ou <Custo com contratação de serviços>	<x> meses

Autorizações:

<Nome do Diretor Presidente>Diretor Presidente
(PRODEST)

<Nome do Diretor Técnico>Diretor Técnico
(DITEC)

<Nome do Gerente da Área><Nome da Gerência>
(<Sigla>)

<Nome do Cliente><Cargo do cliente>
(<Sigla do Órgão>)**Data:** dd/mm/aaaa



Governo do Estado do Espírito Santo

Secretaria de Estado de Gestão e Recursos Humanos – SEGER

Instituto de Tecnologia da Informação e Comunicação do Estado do Espírito Santo



Tipo do Documento:	RESULTADO DE DIAGNÓSTICO		Código: PPS-RD
		Revisão: 01	Data: 14/08/2018

1. Identificação

Sistema: <nome do sistema>		Módulo: <nome do módulo>
Cliente: <sigla do órgão>	Demandante: <nome do demandante>	Nº SA: <número da SA>

2. Estimativa de Prazo e Tamanho

Início da Etapa de Execução: <data inicial>	Fim da Etapa de Execução: <data final>	Quantidade de PF: <qt. de PF>
---	--	---

3. Itens da SA não Avaliados

Item	Descrição Resumida da Solicitação
01	<Título resumido da solicitação>
02	<Título resumido da solicitação>

4. Itens da SA Avaliados

Item	Descrição Resumida da Solicitação	Pontos de Função
01	<Título resumido da solicitação>	<qt. de PF>
02	<Título resumido da solicitação>	<qt. de PF>

5. Instruções/Observações Complementares

<Informações consideradas relevantes>



Tipo do Documento:

Medição de Tamanho Funcional

Código:
PPS-MF

Revisão:
01

Data:
01/10/2015

<Identificação do Sistema>

Projeto..... : _____
Responsável..... : _____
Revisor..... : _____

Data : _____
Data : _____

Identificação da Contagem

Tipo de Contagem : ☐ Projeto de Desenvolvimento
☐ Projeto de Manutenção
☐ Aplicação (Baseline)

R\$/PF..... _____
Total de PF Ajustados.....: _____ 0
Custo total.....: _____ R\$ 0,00

Propósito da Contagem

Escopo da Contagem



Governo do Estado do Espírito Santo

Secretaria de Estado de Gestão e Recursos Humanos – SEGER

Instituto de Tecnologia da Informação e Comunicação do Estado do Espírito Santo



Tipo do Documento: Infraestrutura do Sistema	Código: PPS-IS	Nº de Páginas: 4
	Revisão: 01	Data: 08/05/2015

1. Histórico de Revisões

Data	Versão	Descrição	Autor	Revisor

2. Projeto

<Nome do projeto>

3. Definição da Plataforma Tecnológica

Segue abaixo a definição da linguagem de desenvolvimento, banco de dados e demais tecnologias envolvidas no projeto.

Plataforma de Desenvolvimento
<Descrição 1>
<Descrição 2>

< Exemplo:

Plataforma de Desenvolvimento
Linguagem de Desenvolvimento: ASP, HTML, JAVASCRIPT, CSS, XML
Servidor Reporting Services
Browser para execução da aplicação (Explorer 7.0 acima, Mozilla Firefox 3 acima, Google Chrome, Safari)
Servidor Web de Desenvolvimento com IIS (Microsoft Internet Information Services – versão 7 ou superior)
Framework .NET 4.0
Ferramenta Visual Studio 2008 para geração de Relatórios (Analysis Services Project)
Banco de Dados MSSQL Server 2008 R2

>

4. Arquitetura Tecnológica do Sistema

Nessa seção há uma descrição e um diagrama para demonstrar como será a arquitetura do sistema. A arquitetura serve para organizar o desenvolvimento, estruturar a solução, minimizar riscos técnicos e identificar oportunidade de reuso de código.

< Exemplo:

4.1. Arquitetura de Hardware

- Sistema utiliza plataforma Web, sob uma arquitetura cliente-servidor 3 camadas, conforme a figura a seguir.



Governo do Estado do Espírito Santo

Secretaria de Estado de Gestão e Recursos Humanos – SEGER

Instituto de Tecnologia da Informação e Comunicação do Estado do Espírito Santo

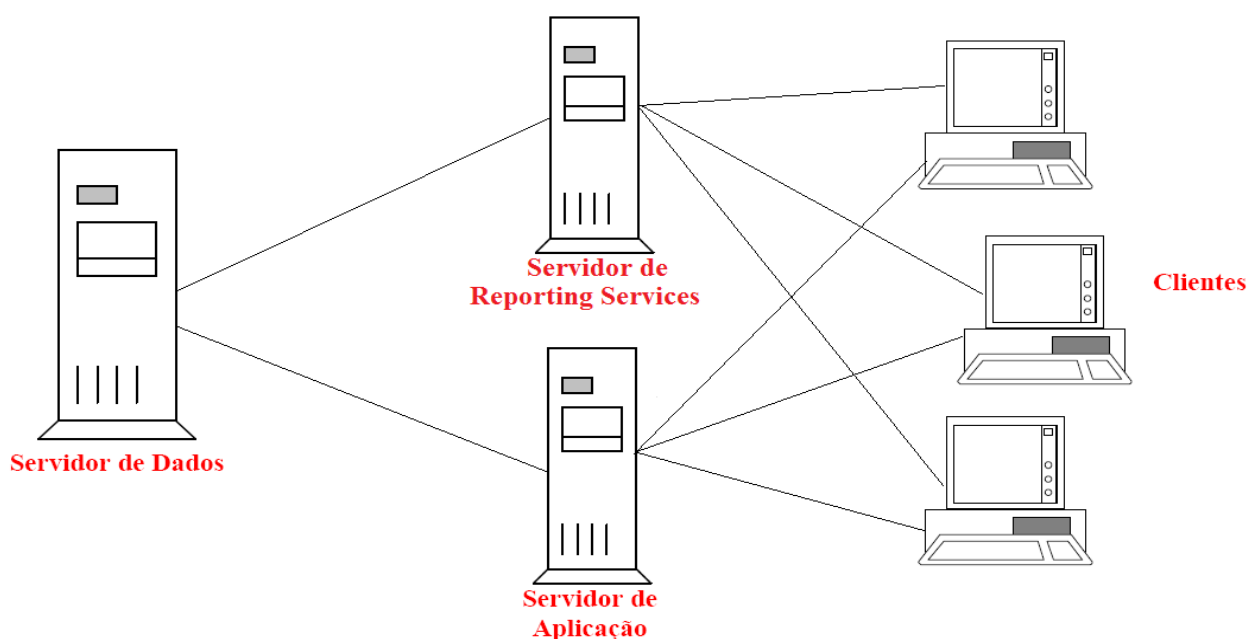


Figura1: Camada 1 com o Servidor de Banco de Dados. Camada 2 com o Servidor de Aplicação. Camada 3: Servidor de Reporting Services. Camada 4 com os clientes acessando o sistema na Internet.

- Atualização do Portal da Transparência, a SECONT, órgão gestora dos dados do Portal da Transparência, recebe arquivos de carga de diversos órgãos do Estado e solicita as cargas mensais no ambiente de homologação. Algumas cargas já são efetuadas diretamente no ambiente de produção, como Receitas, Despesas. A Figura 2 ilustra as etapas da atualização do Portal da Transparência.

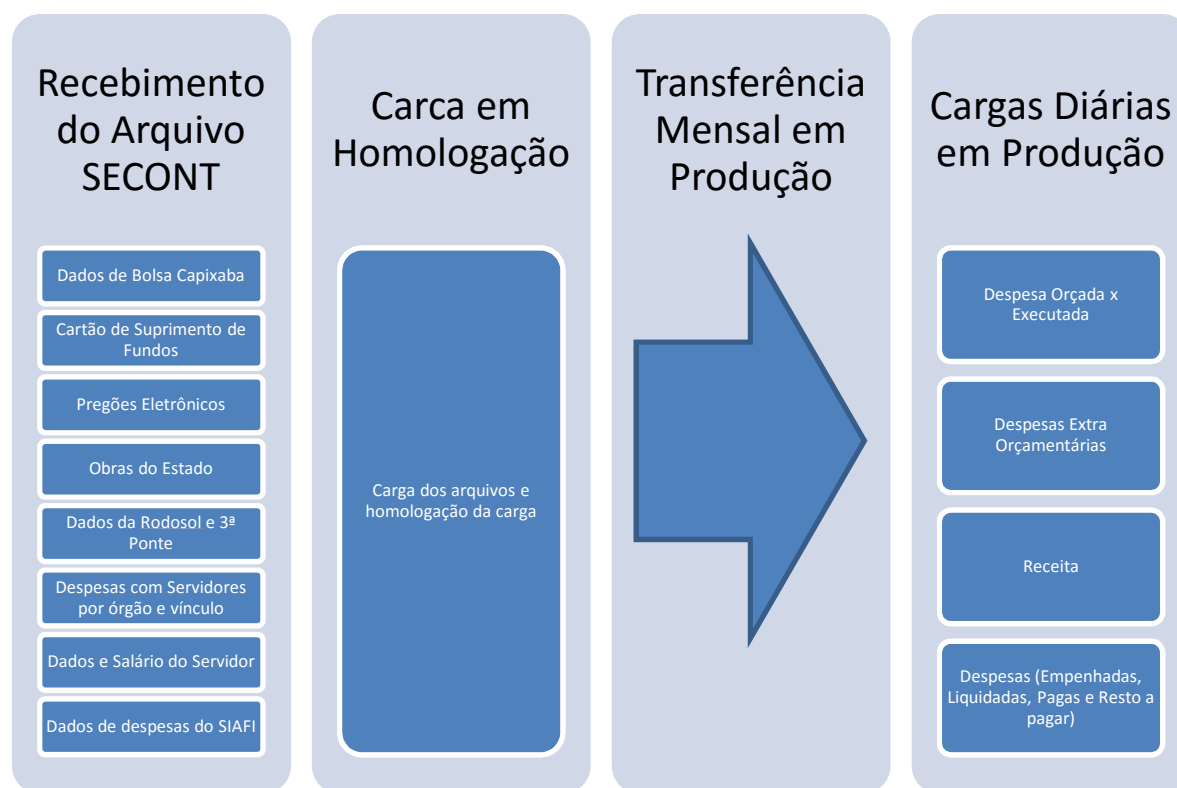


Figura2: Atualizações e processos de carga do Portal da Transparência.

4.2. Arquitetura de Software

Nessa seção, há uma descrição das camadas e subsistemas existentes no ambiente de produção do Portal da Transparência. A Figura 3 apresenta a arquitetura de software.



Governo do Estado do Espírito Santo

Secretaria de Estado de Gestão e Recursos Humanos – SEGER

Instituto de Tecnologia da Informação e Comunicação do Estado do Espírito Santo



- **Gráficos:** os gráficos do portal são gerados com a consulta no Repositório de Gráficos e Data Set Compartilhados. Esses componentes são responsáveis por controlar as regras dos gráficos. Quando um cliente faz uma requisição de acesso para um gráfico, o servidor de Reporting Services é responsável por renderizar os dados em formato HTML, executado por Browsers.
- **Consultas da camada de Visão:** A camada de visão contém todos os programas reesponsáveis por layouts em navegadores de Websites. Tais informações são consultadas no banco de dados, via camada de gerência de Dados.
- **Camada de Gerência de Dados:** Camada responsável pela interface com o banco de dados da aplicação do Portal da Transparência. Ela é gerencia inserções, deleções e consultas de dados.
- **Programa de atualização mensal:** Esse programa é responsável por atualizar os dados no ambiente de produção, posteriormente à homologação feita pelos gestores dos dados do Portal da Transparência. Vale destacar que tais dados são armazenados em tabelas totalizadoras, estratégia utilizada para otimizar o tempo do cliente durante uma consulta de dados, evitando cálculos desnecessários durante a navegação do usuário.
- **Programa de Carga:** responsável pelas cargas diárias. Esse programa é do estilo arquitetônico Dutos e Filtros e agendado para ser executado diariamente. Também usa a estratégia de carga em tabelas totalizadoras.

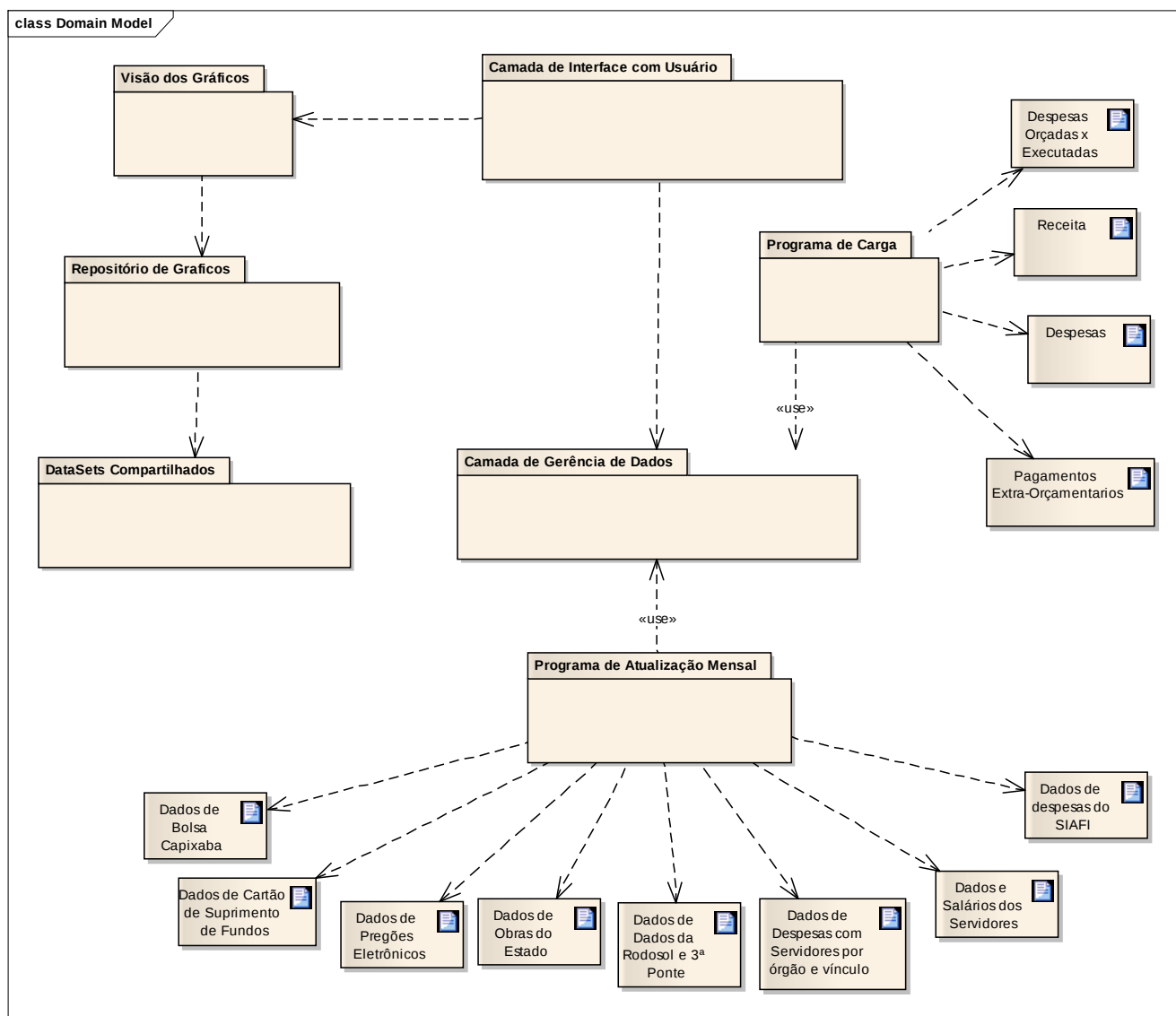


Figura3: Arquitetura de Software do Portal da Transparência.



Governo do Estado do Espírito Santo

Secretaria de Estado de Gestão e Recursos Humanos – SEGER

Instituto de Tecnologia da Informação e Comunicação do Estado do Espírito Santo



5. Observações Gerais

<Observações gerais sobre a especificação tecnológica do sistema >



PADRÃO DE CODIFICAÇÃO

APLICAÇÕES .NET C#



Governo do Estado do Espírito Santo

Secretaria de Estado de Gestão e Recursos Humanos – SEGER

Instituto de Tecnologia da Informação e Comunicação do Estado do Espírito Santo



Tipo do Documento:

Padrão de Codificação

Código: PPS-PC

Revisão: 01

Data: 14/09/2018

Sumário

1. Histórico de revisões.....	2
2. Resumo.....	2
3. Convenções de Nomenclatura.....	2
3.1. Estilos de nomenclaturas (Notação)	2
3.1.1. Pascal Case.....	2
3.1.2. Camel Case	2
3.2. Regras Gerais de Nomeclatura.....	3
3.3. Namespace	3
4. Estilo	3
4.1. Geral	3
4.2. Comentário.....	5
5. Qualidade.....	5
6. Referências.....	6



1. Histórico de revisões

Data	Versão	Descrição	Autor (es)
14/09/2018	1.0	Versão Inicial	Marcos Tadeu Torres Álvaro Menezes

2. Resumo

Este documento descreve os padrões de codificação que devem ser adotados pelos desenvolvedores do Instituto de Tecnologia da Informação e Comunicação do Estado do Espírito Santo – PRODEST. Estão incluídos padrões de nomenclatura, identificação e de estruturação de classes, além das definições de alguns indicadores de qualidade do código-fonte. O documento é voltado para a linguagem C# .NET.

3. Convenções de Nomenclatura

Nesta seção descrevemos as regras para padronização de nomes utilizados no código.

3.1. Estilos de nomenclaturas (Notação)

3.1.1. Pascal Case

É a denominação em inglês para a prática de escrever palavras compostas ou frases, onde a primeira letra do identificador e a letra inicial de cada palavra dentro do identificador devem estar em maiúsculo. Usada para todos os identificadores, exceto os nomes de parâmetro e variáveis locais.

Exemplos:

1. PropertyDescriptor
2. HtmlTag

Um caso especial é feito para as abreviações de duas ou mais letras em que as letras iniciais estão em maiúsculas, conforme mostrado no seguinte identificador:

1. IOStream
2. CPF

3.1.2. Camel Case

É a denominação em inglês para a prática de escrever palavras compostas ou frases, onde a primeira letra de cada palavra dentro do identificador deve estar em maiúsculo, todas as outras em minúsculo, inclusive a primeira letra do identificador. Usada somente para os nomes de parâmetro e variáveis locais.

Exemplos:

1. propertyDescriptor
2. htmlTag
3. ioStream
4. cpf



3.2. Regras Gerais de Nomeclatura

- Não utilizar acentos.
- Não utilizar espaços.
- Não utilizar sublinhado (subtraço), hifens, ou qualquer outro caractere não alfanumérico.
- Não é recomendado utilizar abreviações. Siglas são aceitáveis. Embora a clareza nos nomes seja essencial mesmo que o nome fique longo. Sendo assim, siglas só devem ser utilizadas quando forem uma referência muito comum no modelo de negócio sendo tratado, por exemplo: CPF (Cadastro de Pessoas Físicas).
- Usar PascalCasing para todos os membros, tipos e namespaces públicos onde os identificadores são palavras compostas ou frases.
- Usar camelCasing para nomes de parâmetro.
- Qualquer API acessada externamente não deve depender da diferenciação entre maiúsculas e minúsculas para distinguir dois nomes no mesmo contexto. C# é uma linguagem case-sensitive, mas isso não pode ser assumido como regra para qualquer linguagem externa.
- Não utilizar notação húngara.
- Prefixar os nomes das interfaces com I, para indicar que o tipo é uma interface.

3.3. Namespace

A regra geral para nomenclatura de namespaces é:

```
1. <EmpresaOuSecretaria>.(<Produto>|<Sistema>)[.<Função|Módulo>][.<Subnamespace>]
```

4. Estilo

4.1. Geral

- Caracteres de tabulação não devem ser utilizados.
- Nunca declare mais de um namespace por arquivo.
- Evite colocar mais de uma classe por arquivo.
- Não codifique o texto de mensagens direto no código. Use arquivos de recurso (resource files).
- Sempre use chaves ({ e }) numa nova linha em instruções condicionais. Ex:

```
1. protected void Page_Load(object sender, EventArgs e)
2. {
3.     bool IsTrue;
4.     if (IsTrue == true)
5.     {
6.         // Do something;
7.         //....
8.         return false;
9.     }
10. }
```



- Use uma linha branca para separar grupos lógicos de código. Ex:

```
1. public void SayHello(string name)
2. {
3.     string fullMessage = "Hello " + name;
4.     DateTime currentTime = DateTime.Now;
5.
6.     string message = fullMessage + ", time:" + currentTime.ToShortTimeString();
7.
8.     MessageBox.Show(message);
9. }
```

- Use regiões (#region) para agrupar partes relacionadas de código. Ex:

```
1. namespace SampleApplication
2. {
3.     public partial class _Default : System.Web.UI.Page
4.     {
5.
6.         #region Private Members
7.         // code here
8.         #endregion
9.
10.        #region Constructors
11.        // code here
12.        #endregion
13.
14.        #region Public Properties
15.        // code here
16.        #endregion
17.
18.        #region Private Properties
19.        // code here
20.        #endregion
21.
22.        #region Public Methods
23.        // code here
24.        #endregion
25.
26.        #region Private Methods
27.        // code here
28.        #endregion
29.    }
```

- Não se devem utilizar blocos no interior de métodos. A possível necessidade de organização de código nestes casos deve ser superada com a criação de novos métodos e com a utilização de blocos para agrupar estes métodos. Ex:

```
1. #region Controle do Formulário
2. public void metodoInicio()
3. {
4.     ...
5.     metodoInterno();
6. }
7. private void metodoInterno()
8. {
9.     ...
10. }
11. #endregion
```



4.2. Comentário

- Comentário deve estar no mesmo nível do código (mesma indentação), mas numa linha separada, nunca no final da linha de código. Ex:

```
1. // Format a message and display.  
2. string fullMessage = "Hello " + name;  
3. DateTime currentTime = DateTime.Now;
```

- Insira um espaço entre o delimitador de comentário e o texto do comentário. Ex:

```
1. // The following declaration creates a query. It does not run  
2. // the query.
```

5. Qualidade

A qualidade do código-fonte será medida e avaliada através de análise estática de código com a ferramenta SonarQube 7.0 ou superior. Alguns dos indicadores de qualidade do código-fonte que podem ser utilizados estão relacionados a seguir:

Grupo	Indicador	Unidade	Meta Sugerida
Arquitetura	Directory Tangle Index (ciclos de dependências entre pacotes e classes)	%	<= 2%
Projeto	Complexity / file	média total	<= 10
	Complexity / class	média total	<= 10
	Complexity / function	média total	<= 3
	Duplications	%	<= 4%
	Security Issue Tags	unidades	= 0
	Technical Debt ratio	%	<= 2,5%
	Sqale Rating	Nota	= A
Violações de código (possíveis bugs, estilo de codificação, más práticas de codificação):	Rules Compliance Index	%	>= 95%
	Critical Issues	unidades	= 0
	Blocker Issues	unidades	= 0
	Unit Tests Coverage - camada negócio	%	>= 70%



Governo do Estado do Espírito Santo

Secretaria de Estado de Gestão e Recursos Humanos – SEGER

Instituto de Tecnologia da Informação e Comunicação do Estado do Espírito Santo



Indicadores relacionados a testes	Unit Test Success	%	= 100%
	Skipped Tests	unidades	= 0

6. Referências

- <https://docs.microsoft.com/en-us/dotnet/standard/design-guidelines/naming-guidelines>
- <https://rules.sonarsource.com/csharp>