

ROTEIRO DE MÉTRICAS DO SISP

Versão 3.0

MINUTA SUBMETIDA À
CONSULTA PÚBLICA



MINISTÉRIO DA
GESTÃO E DA INOVAÇÃO
EM SERVIÇOS PÚBLICOS



SUMÁRIO

1.INTRODUÇÃO	10
2.OBJETIVO	12
3.TERMOS E DEFINIÇÕES	13
4.CONTAGEM DE PONTOS DE FUNÇÃO - MÉTODO APF	22
4.1. PROCESSO DE CONTAGEM - MÉTODO APF	23
5.CONTAGEM DE PONTOS DE FUNÇÃO - MÉTODO SFP	34
5.1. PROCESSO DE CONTAGEM DO MÉTODO SFP	36
5.2. CORRESPONDÊNCIA E CONVERSIBILIDADE ENTRE OS MÉTODOS	40
5.3. EXEMPLO DE APLICAÇÃO DA CONVERSIBILIDADE	41
6.CÁLCULO DE PONTOS DE FUNÇÃO PARA O SISP	43
6.1. PROJETO DE DESENVOLVIMENTO	44
6.2. LINHA DE BASE APÓS DESENVOLVIMENTO INICIAL	45
6.3. PROJETO DE MELHORIA	45
6.4. BASELINE APÓS MELHORIA - MÉTODO SFP	47
6.5. BASELINE APÓS MELHORIA – MÉTODO APF	48
6.6. PROJETOS DE MIGRAÇÃO DE DADOS	49
6.7. MANUTENÇÃO CORRETIVA	50
6.8. MUDANÇA DE PLATAFORMA	51
6.8.1. Mudança de Plataforma - Linguagem	51
6.8.2. Mudança de Plataforma – Banco de Dados	52
6.9. ATUALIZAÇÃO DE VERSÃO	54
6.9.1. Atualização de Versão - Linguagem	54
6.9.2. Atualização de Versão - Browser	55
6.9.3. Atualização de Versão – Banco de Dados	56
6.10. MANUTENÇÃO EM INTERFACE	56
6.11. ADAPTAÇÃO SEM ALTERAÇÃO DE REQUISITOS FUNCIONAIS	57
6.12. APURAÇÃO ESPECIAL	58
6.12.1. Apuração especial – Base de dados	59

6.12.1.a.	Atualização de dados sem consulta prévia	60
6.12.1.b.	Consulta prévia sem atualização	60
6.12.1.c.	Atualização de dados com consulta prévia	61
6.12.2.	Apuração especial – Geração de Relatórios.....	61
6.12.3.	Apuração especial – Alteração	62
6.12.4.	Apuração especial – Reexecução	62
6.13.	DESENVOLVIMENTO, MANUTENÇÃO E PUBLICAÇÃO DE PÁGINAS ESTÁTICAS.....	63
6.14.	MANUTENÇÃO DE DOCUMENTAÇÃO DE SISTEMAS LEGADOS.....	64
6.15.	VERIFICAÇÃO DE ERROS	64
6.16.	PONTOS DE FUNÇÃO DE TESTE.....	65
6.17.	COMPONENTE REUSÁVEL	66
7.	ORIENTAÇÕES COMPLEMENTARES	68
7.1.	CONTAGEM COM MÚLTIPLAS MÍDIAS.....	68
7.1.1.	CENÁRIO 1: Mesmos dados apresentados em tela e impressos.....	69
7.1.2.	CENÁRIO 2: Dados de saída idênticos	69
7.1.3.	CENÁRIO 3: Dados de saída distintos.....	69
7.1.4.	CENÁRIO 4: Dados de entrada batch e on-line idênticos	70
7.2.	ABAS E IDENTIFICAÇÃO DE PROCESSO ELEMENTAR	70
7.3.	PRINCIPAIS FALHAS DE CONTAGEM IDENTIFICADAS.....	70
7.3.1.	Dados de Código.....	71
8.	ORIENTAÇÕES PARA REALIZAR ESTIMATIVAS DE PROJETOS	74
8.1.	REUNIR A DOCUMENTAÇÃO DISPONÍVEL.....	75
8.2.	ESTIMATIVA DE TAMANHO COM SFP	75
8.2.1.	Exemplo de Estimativa de Tamanho com SFP	76
8.3.	ESTIMATIVA DE ESFORÇO DE PROJETOS DE SOFTWARE.....	77
8.3.1.	Exemplo de Estimativa de Esforço	77
8.3.2.	Distribuição de Esforço por fase de projeto	78
8.4.	ESTIMATIVA DE PRAZO DE PROJETOS DE SOFTWARE	79
8.4.1.	Passo 1: Definir a composição do time de referência	79
8.4.2.	Passo 2: Estimar a produtividade do time de referência.....	81
8.4.3.	Passo 3: Estimar o prazo do projeto	83
8.5.	ESTIMATIVA DE CUSTO DE PROJETOS.....	85
8.5.1.	Exemplo de Estimativa de Custo	86
8.6.	Estimativa de recursos de TIC	89

9.DIRETRIZES PARA DEFINIÇÃO DE FATORES DE IMPACTO	90
9.1.DEFINIÇÃO DE FATOR DE IMPACTO	92
10. CONTAGEM E ACOMPANHAMENTO DE PROJETOS ÁGEIS	100
10.1. ORIENTAÇÕES GERAIS	102
10.2. MUDANÇAS EM FUNCIONALIDADES.....	103
10.2.1. Fatores que influenciam o número de mudanças	104
10.3. REGISTRO E CONTROLE	105
10.3.1. Exemplo de aplicação - Contagem Estimada da <i>Release N</i>	105
10.3.2. Exemplo de aplicação - Contagem Final da <i>Release</i>	107
10.3.3. Exemplo de aplicação - Contagem para a <i>Baseline</i> da Aplicação	110
10.4. RECOMENDAÇÕES PARA O USO DO FATOR ÁGIL	111
10.4.1. Exemplo de Remuneração da Contratada	112
10.5. CANCELAMENTO DE PROJETOS ÁGEIS.....	112
11. CONTAGEM DE PF EM CONTEXTOS ESPECÍFICOS	114
11.1. CONTAGEM DE IA (Inteligência Artificial)	114
11.1.1. Determinar a Fronteira da Aplicação	114
11.1.2. Escopo da contagem de IA	115
11.1.3. Contagem para Processamento Batch de IA	116
11.1.4. Contagem para a Execução Interativa de IA	123
11.1.5. Manutenções de Soluções de IA.....	126
11.2. CONTAGEM DE PF EM CHATBOT	126
11.2.1. Visão <i>ChatBot</i> para contagem PF.....	126
11.2.2. Escopo da contagem PF para <i>ChatBot</i>	128
11.2.3. Contagem de funções de dados para <i>ChatBot</i>	128
11.2.4. Contagem de funções transacionais pra <i>Chatbot</i>	129
11.2.5. Manutenção em soluções de <i>Chatbot</i>	130
11.2.6. Itens sem contagem de PF em soluções de <i>Chatbot</i>	131
11.3. CONTAGEM PF DE PAINÉIS ANALÍTICOS	132
11.3.1. Criação e carga de tabelas.....	132
11.3.2. Leitura de dados de outras aplicações.....	133
11.3.3. Gráficos e tabelas dinâmicas.....	134
11.4. CONTAGEM PF PARA PROJETOS BASEADOS EM GEOTECNOLOGIAS.....	134
11.4.1. Medição de camadas georreferenciadas	135
11.4.2. Apresentação de dados em formatos diferentes	135

11.4.3.	Apresentação simultânea de vários elementos visuais	136
11.4.4.	Funcionalidades complexas de georreferenciamento.....	137
11.5.	CONTAGEM DE DESENVOLVIMENTO MULTIPLATAFORMA	140
11.5.1.	Plataforma de desenvolvimento híbrida.....	141
11.5.1.a.	Contagem de Projetos de Desenvolvimento	141
11.5.1.b.	Contagem de Projetos de Melhoria	141
11.5.2.	Plataforma de desenvolvimento nativa	142
11.5.2.a.	Contagem de Projetos de Desenvolvimento	142
11.5.2.b.	Contagem de Projetos de Melhoria	143
12.	CONTAGEM PF DE PROJETOS DE DW	144
12.1.	OBJETIVO	144
12.2.	MODELO DE ARQUITETURA DE REFERÊNCIA DE DATA WAREHOUSE/DATA MART	145
12.3.	ESTIMATIVA DE PROJETO DE DW.....	148
12.4.	VISÃO GERAL DA CONTAGEM DE PROJETO DW	150
12.5.	ESCOPO DA CONTAGEM E FRONTEIRA DA APLICAÇÃO.....	152
12.6.	CONTAGEM DE FUNÇÕES DE DADOS	153
12.6.1.	Contagem funções de dados internas do DW	153
12.6.2.	Contagem de funções de dados do processo ETL	156
12.6.3.	Considerações finais sobre funções de dados em projetos DW	157
12.7.	CONTAGEM DE FUNÇÕES TRANSACIONAIS.....	158
12.7.1.	Contagem de funções de carga de dados.....	158
12.7.2.	Contagem de funções de geração de relatórios ou gráficos.....	160
12.7.3.	Contagem de funções de extração de dados do Sistema Transacional de Origem.....	161
12.8.	PROJETO DE MELHORIA DE DW	167
12.9.	DEMANDAS TÍPICAS EM PROJETOS DE DW	167
12.9.1.	Alteração de Dados de Dimensões Estáticas	167
12.9.2.	Criação/Alteração de Campos em Tabelas Fato e Dimensão	168
12.9.3.	Alteração de Dados em Tabelas Fato e Dimensão	169
12.9.4.	Extração de Dados em Tabelas Fato e Dimensão	169
12.9.5.	Criação, Configuração e Disponibilização de um Filtro de Relatório.....	169
12.9.6.	Criação de Métricas.....	170
12.9.7.	Reorganização de Bancada.....	170
12.9.8.	Exclusão de Dados Antigos do DW.....	170
12.9.9.	Metadados Relacionados ao Controle do Processo de ETL.....	171

12.9.10.	Migração de Banco de Dados.....	171
12.9.11.	Migração de ferramenta ETL.....	171
13.	ATIVIDADES SEM CONTAGEM DE PF.....	172
13.1.	ATIVIDADES SEM CONTAGEM DE PF EM PROJETOS DE DESENVOLVIMENTO DE SW	172
13.2.	ATIVIDADES SEM CONTAGEM DE PF EM PROJETOS DE DW	174
13.2.1.	Tabelas Sem Contagem de Ponto de Função	174
13.2.2.	Customização de Ferramenta OLAP.....	174
13.2.3.	Integração da Ferramenta OLAP ao AD/LDAP	175
13.2.4.	Customização da Ferramenta ETL.....	175
13.2.5.	Criação de usuários e perfis de acesso às ferramentas OLAP e ETL.....	175
13.2.5.a.	Automação do Processo de ETL	175
14.	STORY POINTS.....	177
14.1.	ORIENTAÇÕES GERAIS	177
14.2.	BACKLOG DO PRODUTO	178
14.2.1.	Refinamento contínuo do backlog do produto.....	180
14.3.	BACKLOG DA SPRINT	180
14.4.	HISTÓRIAS DE USUÁRIO (<i>user stories</i>).....	181
14.4.1.	Formato das Histórias do usuário	182
14.4.2.	Métodos adicionais para levantamento das Histórias de Usuário	184
14.4.3.	Critérios de Aceitação	185
14.4.4.	Interface de usuário (ou UI, do inglês <i>User Interface</i>).....	187
14.4.5.	Tarefas (<i>tasks</i>).....	187
14.4.6.	Definição de Preparado.....	188
14.5.	INCREMENTO DO PRODUTO	189
14.5.1.	Definição de Pronto.....	189
14.6.	CONTAGEM POR STORY POINTS	191
14.6.1.	Foco da estimativa.....	193
14.6.2.	Passos para a contagem por <i>Story Points</i>	193
14.6.3.	Complexidade de histórias de usuário	194
14.6.4.	Exemplo de aplicação da métrica.....	196
14.6.5.	Boas práticas na estimativa por <i>Story Points</i>	197
14.6.6.	Boas práticas na definição do Backlog da Sprint	198
14.6.7.	Erros comuns na contagem por Story points.....	199
15.	CONTAGEM DE HORAS DE SERVIÇO TÉCNICO - HST	200

15.1.	COMPOSIÇÃO DO CATÁLOGO DE SERVIÇOS TÉCNICOS	200
15.2.	CATÁLOGO DE SERVIÇOS TÉCNICOS	204
15.3.	CUSTO DA HST	215
15.4.	VALOR A SER REMUNERADO PELO SERVIÇO	215
16.	DIRETRIZES DE CODIFICAÇÃO SEGURA	220
16.1.	DIRETRIZES GERAIS	220
16.2.	DIRETRIZES DE ESCRITA DE CÓDIGO	220
16.2.1.	Nomes Significativos.....	220
16.2.2.	Funções e Classes	221
16.2.3.	Comentários	221
16.2.4.	Estruturas.....	221
16.2.5.	Indentação	221
16.2.6.	Complexidade ciclomática.....	222
16.3.	DIRETRIZES DE SEGURANÇA DA INFORMAÇÃO	222
16.3.1.	Armazenamento de Dados.....	222
16.3.2.	Gerenciamento de Acessos e Permissões de Usuários.....	222
16.3.3.	Comunicação	223
16.3.4.	Medidas contra-ataques	223
16.3.5.	Auditoria, rastreamento e Logs.....	224
16.3.6.	Cópias de Segurança (Backups).....	225
16.3.7.	Testes.....	225
16.3.8.	Medidas contra falhas de segurança	226
16.3.9.	Diretrizes para a instalação, configuração e gerenciamento de ambientes de desenvolvimento de sistemas	226
16.3.10.	Diretrizes para a configuração de proteção a dados sensíveis.....	226
16.3.11.	Geração de Senhas	227
16.3.12.	Diretrizes para reforço da segurança de software nas fases de especificação, projeto e design	228
16.3.13.	Atualização de softwares.....	228
17.	REQUISITOS MÍNIMOS DE QUALIDADE E PADRONIZAÇÃO DOS ASPECTOS TÉCNICOS DA CODIFICAÇÃO	229
18.	ORIENTAÇÕES PARA O PROCESSO DE AMOSTRAGEM	230
19.	CONCLUSÃO.....	231
20.	COLABORADORES DA VERSÃO 3.0	232

21. REFERÊNCIAS BIBLIOGRÁFICAS	233
--------------------------------------	-----

FIGURAS E TABELAS

FIGURA 1: PROCESSO DE CONTAGEM DE PONTOS DE FUNÇÃO CONFORME CPM 4.3.1	23
TABELA 1: REGRAS PARA CONTAGEM DE FUNÇÕES DE DADOS (FONTE: CPM 4.3.1)	26
TABELA 2: REGRAS PARA CONTAGEM DE FUNÇÕES TRANSACIONAIS (FONTE: CPM 4.3.1)	29
TABELA 3: COMPLEXIDADE DAS FUNÇÕES DE DADOS (FONTE: CPM 4.3.1)	31
TABELA 4: COMPLEXIDADE FUNCIONAL DAS EES (FONTE: CPM 4.3.1)	31
TABELA 5: COMPLEXIDADE FUNCIONAL DAS CES E SES (FONTE: CPM 4.3.1)	31
TABELA 6: CONTRIBUIÇÃO FUNCIONAL DOS TIPOS FUNCIONAIS (FONTE: CPM 4.3.1)	32
FIGURA 2: PROCESSO DE CONTAGEM DE PONTOS DE FUNÇÃO COM ATIVIDADES EM DESTAQUE	36
TABELA 7: DIFERENÇA NA IDENTIFICAÇÃO DOS TIPOS FUNCIONAIS ENTRE OS MÉTODOS SFP E APF	39
TABELA 8: PONTUAÇÃO DOS COMPONENTES FUNCIONAIS BÁSICOS DO MÉTODO IFPUG SFP	40
TABELA 9: TIPOS E EXEMPLOS DE DADOS DE CÓDIGO CONFORME CPM 4.3.1	72
FIGURA 3: PROCESSO DE DERIVAÇÃO DE ESTIMATIVAS DE PROJETOS DE SOFTWARE	74
TABELA 10: DISTRIBUIÇÃO DO ESFORÇO UTILIZANDO PROCESSOS TÉCNICOS DA NORMA ISO/IEC 12207:2017	78
TABELA 11: EXEMPLO DE COMPOSIÇÃO MÍNIMA DE REFERÊNCIA PARA TIMES DE DESENVOLVIMENTO	80
TABELA 12: CARGA HORÁRIA MENSAL DO TIME DE REFERÊNCIA	82
TABELA 13: CÁLCULO DA ESTIMATIVA DE CUSTO MENSAL DO TIME DE REFERÊNCIA	86
TABELA 14: ESTIMATIVAS DO PROJETO ABC	88
FIGURA 4 – PROCESSOS DO CICLO DE VIDA DO SOFTWARE DA NBR ISO/IEC 12207	90
TABELA 15: PROCESSOS TÉCNICOS DA NORMA ISO/IEC 12207:2017 UTILIZADOS PARA DEFINIÇÃO DOS PERCENTUAIS DE IMPACTO	92
TABELA 16: MEMÓRIA DE CÁLCULO DE PONTO DE FUNÇÃO PARA O SISP – PARTE 1 (SUBITENS 6.1 A 6.9.3)	95
TABELA 17: MEMÓRIA DE CÁLCULO DE PONTO DE FUNÇÃO PARA O SISP – PARTE 2 (SUBITENS 6.10 A 6.17)	97
TABELA 18: PLANEJAMENTO DO BACKLOG DAS <i>SPRINTS</i> (N1, N2 E N3) DA <i>RELEASE</i> N	106
TABELA 19: CONTAGEM SFP DA <i>RELEASE</i> N	108
TABELA 20: CONTAGEM DE SPF DA <i>RELEASE</i> N PARA A BASELINE DA APLICAÇÃO	110

FIGURA 5: CICLO DE VIDA DO DESENVOLVIMENTO DE UM PROJETO DE IA (ADAPTADA SERPRO)	115
FIGURA 6: FUNÇÕES TRANSACIONAIS E DE DADOS DO PROCESSAMENTO EM LOTE (BATCH) DE IA (IMAGEM ADAPTADA SERPRO).....	116
TABELA 21: CONTAGEM PF DE DADOS DE ORIGEM	117
TABELA 22: CONTAGEM PF PARA O MACROPROCESSO EXPLORAR DADOS	119
TABELA 23: CONTAGEM PF PARA DADOS RESULTANTES	122
TABELA 24: CONTAGEM PF PARA DISPONIBILIZAR DADOS	122
FIGURA 7: FUNÇÕES TRANSACIONAIS DO PROCESSAMENTO INTERATIVO DE IA (IMAGEM SEPRO, ADAPTADA)	123
TABELA 25: CONTAGEM PF PARA PREPARAR DADOS EM EXECUÇÃO INTERATIVA DE IA	124
TABELA 26: CONTAGEM PF PARA CRIAR MODELOS EM EXECUÇÃO INTERATIVA DE IA.....	125
FIGURA 8: VISÃO SIMPLIFICADA DO FUNCIONAMENTO DE UM CHATBOT PARA CONTAGEM DE PF	127
TABELA 27: ARQUIVOS LÓGICOS DE UM BOT	128
TABELA 28: FUNÇÕES PARA CRIAÇÃO DE CAGA DE TABELAS DE PAINÉIS ANALÍTICOS.....	133
TABELA 29: FUNÇÃO PARA LEITURA DE DADOS DE OUTRAS APLICAÇÕES	133
TABELA 30: FUNÇÃO PARA APRESENTAÇÃO DE GRÁFICOS OU TABELAS DINÂMICAS	134
FIGURA 9: ELEMENTOS VISUAIS GEORREFERENCIADOS.....	136
TABELA 31: CONTAGEM DE FUNÇÕES TRANSACIONAIS E DE DADOS DE UM SISTEMA DE GEO	138
FIGURA 10: MODELO DE ARQUITETURA DE REFERÊNCIA DE PROJETO DE DW / DM.....	145
FIGURA 11: ENTIDADES DE UM ESQUEMA ESTRELA DE DATA WAREHOUSE	146
TABELA 8: PONTUAÇÃO DOS COMPONENTES FUNCIONAIS BÁSICOS DO MÉTODO IFPUG SFP	149
TABELA 32: RESUMO DA CONTAGEM ESTIMADA DE PROJETO DE DW, UTILIZANDO O MÉTODO SFP.....	149
TABELA 33: DIFERENÇA NA IDENTIFICAÇÃO DAS FUNÇÕES DE DADOS ENTRE OS MÉTODOS SFP E APF	151
FIGURA 12: VISÃO GERAL DA CONTAGEM DE PF COM OS MÉTODOS APF E SFP EM UM PROJETO DE DW/DM	152
FIGURA 13: VISÃO DA FRONTEIRA DW	153
TABELA 34: ENTIDADES RECONHECIDAS EM UM MODELO DE DADOS MULTIDIMENSIONAL.....	153
FIGURA 14: EXEMPLO DE MODELO DE DADOS MULTIDIMENSIONAL – ESQUEMA ESTRELA	154
TABELA 35: RESUMO DA CONTAGEM DE TABELAS DE AGREGAÇÃO PARA OS MÉTODOS SFP E APF	156
FIGURA 15: VISÃO DA FRONTEIRA DW	156
TABELA 36: RESUMO DA CONTAGEM DE FUNÇÕES DE DADOS EM UM PROCESSO ETL	157

TABELA 37: RESUMO DA CONTAGEM DE FUNÇÕES DE CARGA.....	159
TABELA 38: RESUMO DA CONTAGEM DE FUNCIONALIDADES DE GERAÇÃO DE RELATÓRIOS.....	160
FIGURA 16: CENÁRIO 1 – <i>FLAT FILES</i> SENDO UMA CÓPIA DE DADOS DE TABELA DO SISTEMA DE ORIGEM.....	162
FIGURA 17: CENÁRIO 2 – GERAÇÃO DE <i>FLAT FILES</i> NA FRONTEIRA DO SISTEMA DE ORIGEM	162
FIGURA 18: CENÁRIO 3 – INTERFACE DIRETA ENTRE A BASE DE DADOS DO SISTEMA DE ORIGEM E O DSA	164
FIGURA 19: CARGA DE DADOS DO SISTEMA DE ORIGEM NA DSA	164
FIGURA 20: CARGA DE DADOS DO SISTEMA DE ORIGEM NO ODS.....	165
TABELA 39: RESUMO DA CONTAGEM DE FUNÇÕES DE EXTRAÇÃO DE DADOS DO SISTEMA DE ORIGEM	166
TABELA 40: EXEMPLO DE CRIAÇÃO/ALTERAÇÃO DE CAMPOS EM TABELAS FATO E DIMENSÃO	168
TABELA 41: EXEMPLO DE CRIAÇÃO/ALTERAÇÃO DE CAMPOS EM TABELA DIMENSÃO COMPARTILHADA.....	168
FIGURA 21: ALINHAMENTO DE TERMOS DO BACKLOG DO PRODUTO	179
FIGURA 22: FLUXO DO PBI – DEFINIÇÃO DE PREPARADO E DEFINIÇÃO DE PRONTO.....	181
TABELA 42: EXEMPLOS DE MÉTODOS DE LEVANTAMENTO DE REQUISITOS.....	184
FIGURA 23: FLUXO DO PBI – DEFINIÇÃO DE PREPARADO E DEFINIÇÃO DE PRONTO.....	188
FIGURA 24: EXEMPLO DE DEFINIÇÃO DE PREPARADO PARA UM PBI.....	189
FIGURA 25: CARTAS DO PLANNING POKER	192
FIGURA 26: DETALHAMENTO DO BACKLOG DO PRODUTO.....	193
TABELA 43: EXEMPLOS DE FATORES A SEREM USADOS PELO TIME PARA AVALIAR A COMPLEXIDADE DE UMA HISTÓRIA DE USUÁRIO	194
FIGURA 27: PASSOS PARA CONTAGEM DE <i>STORY POINTS</i>	197
TABELA 44: DESCRIÇÃO DE CAMPOS DO CATÁLOGO DE SERVIÇOS TÉCNICOS	201
TABELA 45: CAMPOS ADICIONAIS DO CATÁLOGO DE SERVIÇOS TÉCNICOS QUE DEVEM CONSTAR NOS ESTUDOS TÉCNICOS PRELIMINARES.....	203
TABELA 46: CATÁLOGO DE SERVIÇOS TÉCNICOS MENSURADOS POR HST	204
TABELA 47: MAPA DE PESQUISA SALARIAL DE REFERÊNCIA (PORTARIA SGD/MGI Nº 750).....	216
TABELA 48: INDICADORES E METAS DE QUALIDADE DE CÓDIGO	229

1. INTRODUÇÃO

As métricas de software influenciam a tomada de decisões, tanto no planejamento como na execução do contrato. Portanto, nas contratações de serviços de desenvolvimento, manutenção e sustentação de software devem ser definidas métricas objetivas que permitam a gestão contratual, a mensuração e a devida remuneração dos serviços e produtos efetivamente entregues pela empresa contratada no contexto do processo de desenvolvimento de software adotado pelo órgão ou entidade.

A Portaria SGD/MGI nº 750, de 20 de março de 2023, que estabelece o modelo para a contratação de serviços de desenvolvimento, manutenção e sustentação de software no âmbito dos órgãos e entidades integrantes do SISP, orienta que se deve aferir a entrega de produtos por meio de métricas de software, independentemente da modalidade de remuneração adotada pela organização. Adicionalmente, no Anexo I, o modelo apresenta seção específica (12. MENSURAÇÃO DE SOFTWARE) para fortalecer a importância do uso de métricas de software nas contratações de serviços de desenvolvimento, manutenção e sustentação de software. Convém destacar o item 12.3, transcrito a seguir:

12.3. Independente da modalidade de contratação, deve-se aferir a entrega de produtos por meio de métricas de software, mantendo-se uma base histórica, a exemplo de:

a) Pontos de Função (IFPUG, NESMA, COSMIC, Simple Function Point - SFP);

b) Linhas de código implementadas;

c) Pontos de história (Story Point);

O roteiro de métricas do SISP apresenta novidades em sua versão 3.0, com o fim de apoiar os órgãos do SISP na utilização de métricas para aferição de produtos de software. Além do método de Análise de Ponto de Função (APF) tradicional, descrito no Manual de Práticas de Contagem de Pontos de Função (CPM 4.3.1), será apresentado o método de Ponto de Função Simples (Simple Function Point – SFP, descrito no SPM 2.1) do IFPUG que permite avaliar a medida de tamanho funcional de forma simplificada.

O método de APF tradicional é amplamente utilizado no âmbito dos órgãos do SISP. Ele apresenta muitos benefícios, dentre eles a independência da solução tecnológica. O método SFP usa as mesmas definições de processos elementares e arquivos lógicos da APF tradicional e apresenta muitos benefícios, a saber: independe da tecnologia adotada e oferece resultados confiáveis, repetíveis e objetivos. O grande diferencial é que o método SFP foi projetado para ser ágil, rápido, leve, previsível e de fácil utilização, acelerando consideravelmente o processo de dimensionamento funcional.

Para os métodos SFP e APF, foram adicionadas orientações para contagens de projetos de desenvolvimento e melhoria de IA, ChatBot, Geoprocessamento e Painéis Analíticos, além da Contagem de PF para Data Warehouse (DW) que foi incorporada a este roteiro de métricas.

Ainda, este roteiro apresenta a métrica de Hora de Serviço técnico (HST), que é baseada na quantidade de horas necessárias para se alcançar um resultado ou entregar um produto, por meio de atividades executadas por um ou mais perfis profissionais. Além da definição, forma de cálculo e outras orientações sobre o uso da métrica HST, esse roteiro disponibiliza catálogo de referência para serviços técnicos - contendo: descrição dos serviços, produtos ou resultados esperados, perfis profissionais, esforço estimado e outras informações - com o fim de auxiliar os órgãos do SISP na remuneração adequada de serviços técnicos eventuais e mitigando alguns dos riscos levantados no Acórdão nº 1508/2020 – TCU – Plenário, a saber:

- Antieconomicidade oriunda de superestimações de esforços, produtos ou perfis profissionais;
- Criação de unidades de medida de forma unilateral e sem padronização;
- Fixação e de disseminação de critérios subjetivos;
- Compartilhamento de metodologias e práticas sem a devida consistência e sem justificativas técnica e econômica;
- Cenários de incomparabilidade de preços, de heterogeneidade e de assimetria de informações entre a administração e o mercado;

Adicionalmente, este roteiro apresenta a mensuração por pontos de história (Story Points) e diretrizes de codificação segura.

2. OBJETIVO

O objetivo principal deste documento é apresentar um roteiro de métricas objetivas que permitam a gestão contratual, a mensuração e a devida remuneração dos serviços e produtos efetivamente entregues pela empresa contratada, independentemente da modalidade de remuneração adotada.

O roteiro está orientado a partir das seguintes bases:

- **Objetividade:** com a apresentação de métricas objetivas de mensuração de software;
- **Segurança na mensuração de software:** com a adoção e compartilhamento de métricas confiáveis, repetíveis e com metodologias consistentes;
- **Padronização:** facilitando a aplicação homogênea de boas práticas de mensuração de software, com a disponibilização de métodos de mensuração padronizados e a disseminação de critérios objetivos de mensuração de software;
- **Alinhamento:** com a adesão à Portaria SGD/MGI nº 750, de 20 de março de 2023 e às recomendações dispostas no Acórdão nº 2.037/2019-TCU-Plenário e no Acórdão nº 1.508/2020-TCU-Plenário.

3. TERMOS E DEFINIÇÕES

Para maior compreensão deste documento, aplicam-se os seguintes termos e definições:

Abordagem de desenvolvimento: o método usado para criar e desenvolver o produto, serviço ou resultado durante o ciclo de vida do projeto, tal como o método de previsão, iterativo, incremental, ágil ou um método híbrido.

Ambiente ágil: cultura organizacional, infraestrutura e metodologias que apoiam o desenvolvimento ágil. [FONTE: ISO/IEC/IEEE 26515:2018, 3.2]

Análise de decisão envolvendo critérios múltiplos: técnica que utiliza uma matriz de decisão para fornecer uma abordagem analítica sistemática para estabelecimento de critérios, tais como níveis de risco, incerteza e avaliação, para avaliar e classificar muitas ideias.

Análise de documentos: consiste em revisar e avaliar quaisquer informações documentadas relevantes. Na coleta de requisitos, a análise de documentos é usada para obter requisitos pela análise da documentação existente e a identificação das informações relevantes aos requisitos.

Análise de Ponto de Função: método de medida de tamanho funcional de software definido pela ISO/IEC 14143-1:2007, ISO/IEC 20926:2009, COSMIC (ISO/IEC 19761:2011), ou por métricas derivadas desses padrões internacionais como as contagens da *Netherlands software Metrics Association* (NESMA) ou *Simple Function Point* (SFP) do *International Function Point Users Group* (IFPUG).

Aplicação: é um conjunto coeso de dados e procedimentos automatizados que suportam um objetivo de negócio, podendo consistir em um ou mais componentes, módulos ou subsistemas.

Arquivo lógico: representa a funcionalidade fornecida ao usuário para atender requisitos de armazenamento de dados internos e externos.

Backlog do produto: representa tudo que é necessário para desenvolver e lançar um produto de valor agregado ao negócio. É uma lista priorizada de todos os requisitos (funcionais e não funcionais), funções, tecnologias, melhorias e correções de defeitos que constituem as mudanças que serão efetuadas no produto para versões futuras.

Baseline: corresponde ao tamanho funcional de uma aplicação. Este tamanho corresponde à medida de funções atuais que o aplicativo fornece ao usuário.

Benchmarking: envolve a comparação de produtos, processos e práticas reais ou planejadas com as de organizações similares para identificar as melhores práticas, gerar ideias para melhorias e fornecer uma base para medir o desempenho.

Brainstorming: técnica usada para identificar uma lista de ideias em um curto intervalo de tempo. É realizada em um ambiente de grupo e liderada por um facilitador. O *brainstorming* é composto por duas partes: geração e análise de ideias. O *brainstorming* é uma técnica usada para gerar e coletar múltiplas ideias relacionadas aos requisitos do projeto e do produto.

Código fonte: instruções de computador e definições de dados expressas em uma forma adequada para entrada em um montador, compilador ou outro tradutor.

Complexidade Funcional: Taxa de complexidade específica associada a uma função utilizando as regras definidas no CPM 4.2.3.

Componente Funcional Básico (CFB): Unidade elementar de Requisitos Funcionais do Usuário definido e utilizado pelo método Functional Size Measurement (FSM) para propósitos de medição (ISO/IEC 14143-1:2007). Arquivos lógicos Internos (ALIs), Arquivos de Interface Externa (AIEs), Entradas Externas (EEs), Saídas Externas (SEs), Consultas Externas (CEs) são os tipos de Componentes Funcionais Básicos (CFBs).

Critérios: Normas, regras ou testes pelos quais uma opinião ou decisão pode basear-se ou pelos quais um produto, serviço, resultado ou processo podem ser avaliados.

Critérios de aceitação (Acceptance Criteria): Um conjunto de condições que precisam ser atendidas antes das entregas serem aceitas.

Decisão autocrática: método de tomada de decisão em que uma pessoa assume a responsabilidade por tomar a decisão pelo grupo.

Desenvolvimento ágil: abordagem de desenvolvimento de *software* baseada em metodologias ágeis, nas quais os requisitos e as soluções evoluem por meio da colaboração em equipes multifuncionais e por meio de *feedback* contínuo dos *stakeholders*. Há diferentes métodos capazes de prover um desenvolvimento ágil de *software*, a exemplo de: *Scrum*, *Extreme Programming* (XP), *Kanban*, *Lean*, *Crystal Clear*, *Feature Driven Development* (FDD), entre outros.

Desenvolvimento orientado a testes: Método de desenvolvimento TDD onde os desenvolvedores escrevem testes unitários antes de implementar a funcionalidade relevante [FONTE: NEN NPR 5326:2019].

DevOps: conjunto de princípios e práticas que permitem uma melhor comunicação e colaboração entre as partes interessadas relevantes com o propósito de especificar, desenvolver e operar produtos e serviços de *software* e sistemas, e melhorias contínuas em todos os aspectos do ciclo de vida [FONTE: IEEE 2675-2021 3.1].

Diagrama de contexto: descreve visualmente o escopo do produto, mostrando um sistema de negócio (processo, equipamentos, sistema computacional etc.), e como as pessoas e outros sistemas (agentes) interagem com ele. Os diagramas de contexto mostram as entradas no sistema de negócio, o(s) agente(s) que fornecem a entrada, as saídas do sistema de negócio e o(s) agente(s) que recebem a saída.

Diagramas de afinidade: diagramas que permitem que grandes quantidades de ideias sejam classificadas em grupos para revisão e análise.

Design thinking (lit. pensar como um projetista): consiste em uma abordagem para solucionar problemas, que utiliza um conjunto de ferramentas e técnicas que orientam a pensar e criar soluções de forma ativa, criativa e colaborativa. O *design thinking* busca diversos ângulos e perspectivas para solução de problemas complexos, priorizando o trabalho colaborativo em equipes multidisciplinares em busca de soluções inovadoras.

Dívida Técnica: consiste em decisões de codificação que atendem o projeto a curto prazo, mas que podem comprometer ou encarecer mudanças futuras, ou até mesmo inviabilizá-las.

Equipe de desenvolvimento: equipe que desenvolve e/ou mantém software. As equipes de desenvolvimento podem ser divididas por função (por exemplo, uma equipe de designers, uma equipe de programadores, uma equipe de testadores) ou ser multidisciplinares (cada equipe possui, por exemplo, experiência em design, programação e teste).

Entrega: qualquer produto, resultado ou capacidade de realizar um serviço, que seja único e verificável, produzido para concluir um processo, fase ou projeto.

Entregas aceitas: Produtos, resultados ou recursos produzidos por um projeto e validados pelo cliente ou patrocinadores do projeto como tendo satisfeito seus critérios de aceitação.

Entrevistas: são usadas para obter informações sobre requisitos de alto nível, premissas ou restrições, critérios de aprovação e outras informações de partes interessadas conversando diretamente com elas. Ela normalmente é realizada através de perguntas preparadas ou espontâneas e do registro das respostas. As entrevistas são frequentemente conduzidas individualmente, entre um entrevistador e um entrevistado, mas podem envolver múltiplos entrevistadores e/ou entrevistados.

Etnografia: é uma técnica de observação que pode ser utilizada para compreender os requisitos sociais e organizacionais, ou seja, entender a política organizacional bem como a cultura de trabalho com objetivo de familiarizar-se com o sistema e sua história. O principal objetivo da etnografia é descobrir requisitos de sistema implícitos, que refletem os processos reais, em vez de os processos formais, onde as pessoas estão envolvidas.

Facilitação: é a capacidade de orientar eficazmente um evento de grupo para uma decisão, solução ou conclusão bem-sucedida. Um facilitador garante que haja participação efetiva, que os participantes alcancem uma compreensão mútua, que todas as contribuições sejam consideradas, que conclusões ou resultados sejam plenamente aceitos de acordo com o processo de decisão estabelecido para o projeto e que os acordos e ações alcançados sejam tratados de forma adequada posteriormente.

Fronteira da aplicação: pode ser entendida como a interface conceitual que delimita o *software* que será medido e seus usuários. A fronteira entre aplicações relacionadas está baseada nas áreas funcionais separadas conforme visão do usuário, não em considerações técnicas.

Funcionalidade de conversão: funções transacionais ou de dados fornecidas para converter dados e/ou fornecer outros requisitos de conversão especificados pelo usuário.

Grupos de discussão: Uma técnica de elicitação que reúne as partes interessadas pré-qualificadas e especialistas no assunto para entender suas expectativas e atitudes sobre um produto, serviço ou resultado proposto num formato mais de conversa do que entrevista frente a frente.

História de usuário: descrição em linguagem natural de um recurso de software, exigida por um usuário ou outras partes interessadas.

Horas de Serviço Técnico (HST): métrica baseada na quantidade de horas necessárias para se alcançar um resultado ou entregar um produto, por meio de atividades executadas por um ou mais perfis profissionais, e aferidas por meio de indicadores de níveis mínimos de serviço e critérios de aceitação previamente estabelecidos.

IFPUG: Grupo Internacional de Usuários de Ponto de Função.

Implantação: tornar o sistema ou o conjunto de funcionalidades disponível para os usuários, transferir dados dos *softwares* existentes e estabelecer comunicações com outros *softwares* no ambiente.

Implementação: processo que transforma requisitos, arquitetura e *design*, incluindo interfaces, em ações que criam um elemento ou componente de *software* de acordo com as práticas de codificação previamente estabelecidas, usando técnicas, especialidades ou disciplinas de desenvolvimento de software. Esse processo resulta em um elemento *software* que segue uma arquitetura e *design* estabelecidos.

Incremento de produto: versão de um produto que pode ser liberada no final de um período (*timebox*).

Independente: Nenhuma etapa de processamento anterior ou subsequente é necessária para iniciar ou concluir os Requisitos Funcionais do Usuário.

Informações: dados estruturados ou organizados, processados para uma finalidade específica para torná-los significativos, valiosos e úteis em contextos específicos.

Joint Application Design/Desenvolvimento (JAD): são sessões usadas no setor de desenvolvimento de *software*. São sessões facilitadas, focadas em reunir os especialistas em assuntos de negócio e a equipe de desenvolvimento para coletar requisitos e melhorar o processo de desenvolvimento de *software*.

Manutenção adaptativa (ISO/IEC 14764: 2006): A modificação de um produto de *software*, realizada após a entrega, para manter um produto de *software* utilizável em uma mudança ou mudança ambiente. A manutenção adaptativa fornece os aprimoramentos necessários para acomodar mudanças no ambiente em que um produto de *software* deve operar.

Manutenção corretiva (ISO/IEC 14764: 2006): A modificação reativa de um *software* produto realizado após a entrega para corrigir problemas detectados. A modificação repara o produto de software para atender aos requisitos.

Manutenção perfectiva (ISO/IEC 14764: 2006): Modificação de um produto de software após entrega para detectar e corrigir falhas latentes no produto de *software* antes que sejam manifestados como falhas. A manutenção perfectiva oferece melhorias para os usuários, melhoria da documentação do programa e recodificação para melhorar o *software* desempenho, manutenção ou outros atributos de *software*.

Manutenção preventiva (ISO/IEC 14764: 2006): a modificação de um produto de *software* após entrega para detectar e corrigir falhas latentes no produto de *software* antes que se tornem falhas efetivas.

Mapa de empatia: é uma ferramenta para sintetizar o perfil de um cliente, com questões sobre seus objetivos, comportamento e sua visão sobre o problema e a situação estudada. Assim, possibilidades futuras são analisadas, em conformidade com a percepção do contratante.

Mapeamento de histórias: confecção de uma representação bidimensional de (parte do) *Backlog*, que é um auxílio visual para o usuário cortar, agrupar e organizar histórias logicamente no *backlog*, o que dá uma visão geral das relações entre as histórias de usuário.

Mapeamento mental: no contexto da técnica de *brainstorming*, o mapeamento mental é uma técnica usada para consolidar as ideias criadas através de sessões individuais de *brainstorming* em um mapa único, a fim de refletir pontos em comum e diferenças de entendimento, e gerar novas ideias.

Medição: a tarefa de medir e seu resultado, atribuindo um valor a um atributo em de acordo com uma escala de referência.

Método Delphi: técnica de coleta de informações utilizada como forma de chegar ao consenso de especialistas sobre um assunto.

Método de dimensionamento de camisetas: método para fazer estimativas relativas comparando histórias de usuários e dividindo-as nas categorias extra pequeno, pequeno, médio, grande e extragrande. O tamanho da camiseta esclarece as inter-relações sem perder tempo com falsas precisão.

Método de medição: uma sequência lógica de operações realizadas para produzir medidas.

Metodologias ágeis: são um conjunto de práticas que visam a entrega rápida e de alta qualidade do produto ou serviço e que promovem um processo de gerenciamento de projetos que incentiva a inspeção e adaptação frequente, beneficiando a eficiência e efetividade dos gestores públicos no controle da prestação dos serviços de TI, haja vista que o foco passa a ser realmente nas atividades que entregam valor para as áreas de negócios.

Níveis mínimos de serviço: são regras objetivas e fixas que estipulam valores e/ou características mínimas de atendimento a uma meta a ser cumprida pela contratada na prestação dos serviços.

Observação/conversação: fornecem uma maneira direta de se examinar indivíduos em seu ambiente e como eles desempenham seu trabalho ou tarefas e executam processos. São particularmente úteis para processos detalhados,

quando as pessoas que usam o produto têm dificuldade ou relutam em expressar os seus requisitos. A observação é também conhecida como “*job shadowing*” (aprendizagem por observação). É normalmente feita externamente por um observador acompanhando um especialista de negócio na execução do seu trabalho. Também pode ser feita por um “observador participante”, que de fato realiza um processo ou procedimento para experimentar como o mesmo é feito e descobrir requisitos ocultos.

Parte interessada / Stakeholder: um indivíduo, grupo ou organização que possa afetar, ser afetado, ou sentir-se afetado por uma decisão, atividade, ou resultado de um projeto, programa ou portfólio.

Personas: é uma abordagem que utiliza uma pessoa fictícia para representar usuários de um produto. Personas é utilizado em projetos centrados no usuário, definindo os objetivos e desejos dos reais usuários, orientando decisões como: a interface, a navegação, os recursos e demais elementos do projeto de design.

Pesquisa exploratória: é uma técnica de pesquisa de campo que auxilia no entendimento do contexto a ser trabalhado e fornece insumos para a definição dos perfis de usuários, atores e ambientes ou momentos do ciclo de vida de um produto/serviço.

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

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

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

PF_CONVERSÃO ou SFP_CONVERSAO: 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 (PE para o método SFP / EE para o método APF) e relatórios associados à migração de dados, caso requisitado pelo usuário (PE para o método SFP / SE ou CE para o método APF). Observe que os dados carregados em um processo de migração não devem ser contados como Arquivos Lógicos da aplicação.

PF_REFINADO ou SFP_REFINADO: pontos de função associados a quaisquer mudanças ocorridas sobre uma função transacional ou de dados já previamente trabalhadas na release corrente, provocadas pelo aprofundamento, detalhamento e complementação de requisitos durante o processo de desenvolvimento.

Produto de Software ou Software: conjunto de programas, procedimentos, rotinas ou scripts, componentes, *Application Programming Interface* - API, *webservices*, incluindo os dados e documentação associada.

Produto com mínima viabilidade (MVP): versão de um produto de trabalho com recursos e requisitos suficientes para satisfazer os primeiros clientes e/ou fornecer *feedback* para desenvolvimento futuro.

Produto pronto: é uma descrição formal do estado do incremento, quando ele atende aos níveis de serviço exigidos para o produto; todo o time ágil deve estar em conformidade com a definição de pronto.

Projeto: Um esforço temporário empreendido para criar um produto, serviço ou resultado único.

Projeto ágil: projeto de desenvolvimento de *software* que utiliza abordagem de desenvolvimento ágil.

Proprietário/dono do produto (*product owner*): servidor e/ou representante da contratante que compartilha a visão do produto, incluindo funcionalidades necessárias e critérios de aceitação.

Protótipo: um método para se obter respostas iniciais sobre os requisitos através de um modelo funcional do produto esperado antes de efetivamente construí-lo.

Prototipagem: é um método para obter respostas iniciais sobre os requisitos através de um modelo do produto esperado, antes de efetivamente construí-lo. Exemplos de protótipos são produtos em menor escala, modelos 2D e 3D gerados em computador ou simulações. Os protótipos permitem que as partes interessadas façam experiências com um modelo do seu produto final, em vez de somente discutirem representações abstratas dos requisitos. Os protótipos suportam o conceito de elaboração progressiva em ciclos iterativos de criação de modelos, experimentação de usuário, geração de *feedbacks* e revisão do protótipo.

Qualidade de software: capacidade de um produto, serviço, sistema, componente ou processo de atender às necessidades, expectativas ou requisitos do cliente ou usuário. Para a qualidade de software e sistemas de TI, a ISO/IEC 25010 oferece uma divisão em aspectos, como: confiabilidade; segurança; usabilidade; adequação funcional; manutenibilidade; portabilidade; eficiência de desempenho; compatibilidade. [FONTE: ISO/IEC/IEEE 29119-1:2022]

Questionários e pesquisas: são conjuntos de perguntas escritas, projetadas para acumular rapidamente informações de um grande número de respondentes. Os questionários e/ou pesquisas são mais apropriados para audiências variadas, quando uma resposta rápida é necessária, quando os respondentes estão geograficamente espalhados, e quando uma análise estatística é apropriada.

Reconhecível pelo usuário: o termo “reconhecível pelo usuário” refere-se a requisitos para processos e / ou dados que são acordados e compreendidos pelo (s) usuário (s) e *software* desenvolvedor (es).

Refinamentos: são quaisquer mudanças ocorridas sobre uma função transacional ou de dados já previamente trabalhadas na *release* corrente (seja por meio de uma inclusão, alteração ou exclusão), provocadas pelo aprofundamento, detalhamento e complementação de requisitos durante o processo de desenvolvimento.

Release: distribuição/liberação de um incremento de produto para um cliente ou usuários. A quantidade de *sprints* por release deve ser definida previamente à execução dos serviços.

Requisito: uma condição ou capacidade que deve necessariamente estar presente em um produto, serviço ou resultado para atender a uma necessidade de negócio.

Requisitos das partes interessadas: descrevem as necessidades de uma parte interessada ou de um grupo de partes interessadas.

Requisitos da qualidade: capturam quaisquer condições ou critérios necessários para validar a conclusão bem-sucedida de uma entrega de projeto ou o cumprimento de outros requisitos do projeto. Exemplos incluem testes, certificações, validações etc.

Requisitos de negócio: descrevem as necessidades de nível mais alto da organização como um todo, tais como as questões ou as oportunidades de negócio e as razões por que um projeto foi empreendido.

Requisitos de projeto: descrevem ações, processos ou outras condições que devem ser cumpridas pelo projeto. Exemplos incluem datas de marcos, obrigações contratuais, restrições etc.

Requisitos de solução: descrevem os atributos, as funções e as características do produto, serviço ou resultado que atenderão aos requisitos do negócio e das partes interessadas. Os requisitos de solução são ainda agrupados em requisitos funcionais e não funcionais.

Requisitos do usuário: requisitos que descrevem o que o usuário está pedindo.

Requisitos funcionais: conjunto de requisitos do usuário que descrevem o que o software deve fazer em termos de tarefas e serviços. Os requisitos funcionais descrevem os comportamentos do produto. Exemplos incluem ações, processos, dados e interações que o produto deve executar.

Requisitos não funcionais: conjunto de requisitos relacionados a como deve ser construído ou mantido o *software*, como deve ser o desempenho em operação, aspectos relacionados às tecnologias, à qualidade do *software* e ao ambiente tecnológico que suporta o *software*. Os requisitos não funcionais podem ser descritos como atributos de qualidade, de desempenho, de segurança ou como uma restrição geral em um sistema. Não estão incluídos os aspectos relacionados às funções ou tarefas previstas no *software*.

Reunião diária: reunião diária curta, limitada a um período, usada para discutir o progresso, planos e quaisquer impedimentos com membros de um time ágil.

Roadmap ou Visão do produto: é um plano de ação de como um produto evoluirá ao longo do tempo. Esse plano apresenta uma linha do tempo com marcos de alto nível para um ciclo de vida do produto, particularmente o cronograma para implantação de funcionalidades do produto, com vistas a orientar o progresso em direção a uma meta definida.

Softwares de atividades-meio: aqueles que são utilizados para apoio de atividades de gestão ou administração operacional, como, por exemplo, *softwares* de recursos humanos, ponto eletrônico, portaria, biblioteca, gestão de patrimônio, controle de frotas, gestão eletrônica de documentos, e que não têm por objetivo o atendimento às áreas finalísticas para a consecução de políticas públicas ou programas temáticos.

Software pronto para uso: é aquele *software* disponibilizado (pago ou não) com um conjunto de funcionalidades pré-concebidas, também conhecido como *Ready to Use Software Product* (RUSP) ou comumente de “software de prateleira”.

Sprint: consiste em um ciclo de iteração por um período de até 4 semanas, em que um conjunto acordado de histórias de usuário ou funcionalidades são projetadas, desenvolvidas, testadas, aceitas e se tornam aptas para implantação.

Storyboarding: é uma técnica de prototipagem que exibe a sequência ou navegação por uma série de imagens ou ilustrações. *Storyboards* são usados em diversos projetos em muitos setores, como cinema, propaganda, criação de cursos e em projetos ágeis e de desenvolvimento de *software*. No desenvolvimento de *software*, os *storyboards* usam modelos para mostrar os caminhos de navegação pelas páginas web, telas ou outras interfaces de usuário.

Técnica de grupo nominal: no contexto de um *brainstorming*, a técnica de grupo nominal amplia o *brainstorming*, adicionando um processo de votação para ordenar as melhores ideias para *brainstorming* adicional ou priorização.

Time/Equipe ágil: pequeno grupo multifuncional de pessoas (entre 3 e 10 membros) que colaboram no desenvolvimento de um produto, dentro de uma metodologia ágil.

Timebox: período fixo, previamente estabelecido, durante o qual um indivíduo ou equipe trabalha constantemente para a conclusão de um objetivo acordado.

Simple Function Point - SFP: Pontos de função simplificado/simples.

SISP: Sistema de Administração dos Recursos de Tecnologia da Informação do Poder Executivo Federal.

SPM: Manual de práticas de contagem do método ponto de função simples/simplificado.

Tamanho funcional (ISO 14143-1: 2007 (R2019)): Tamanho do *software* derivado da quantificação os Requisitos Funcionais do Usuário.

Visão do usuário: são os requisitos funcionais percebidos pelo usuário.

Votação: é uma técnica de tomada de decisão coletiva e um processo de avaliação de várias alternativas com um resultado esperado na forma de ações futuras. Essas técnicas podem ser usadas para gerar, classificar e priorizar os requisitos do produto.

Workshop de cocriação: é um encontro organizado, onde são aplicadas uma série de atividades em grupo com o objetivo de estimular a criatividade e a colaboração, fomentando a criação de soluções inovadoras.

4. CONTAGEM DE PONTOS DE FUNÇÃO - MÉTODO APF

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. Tem como objetivo medir o que o software faz, por meio de uma avaliação padronizada dos requisitos de negócio do sistema.

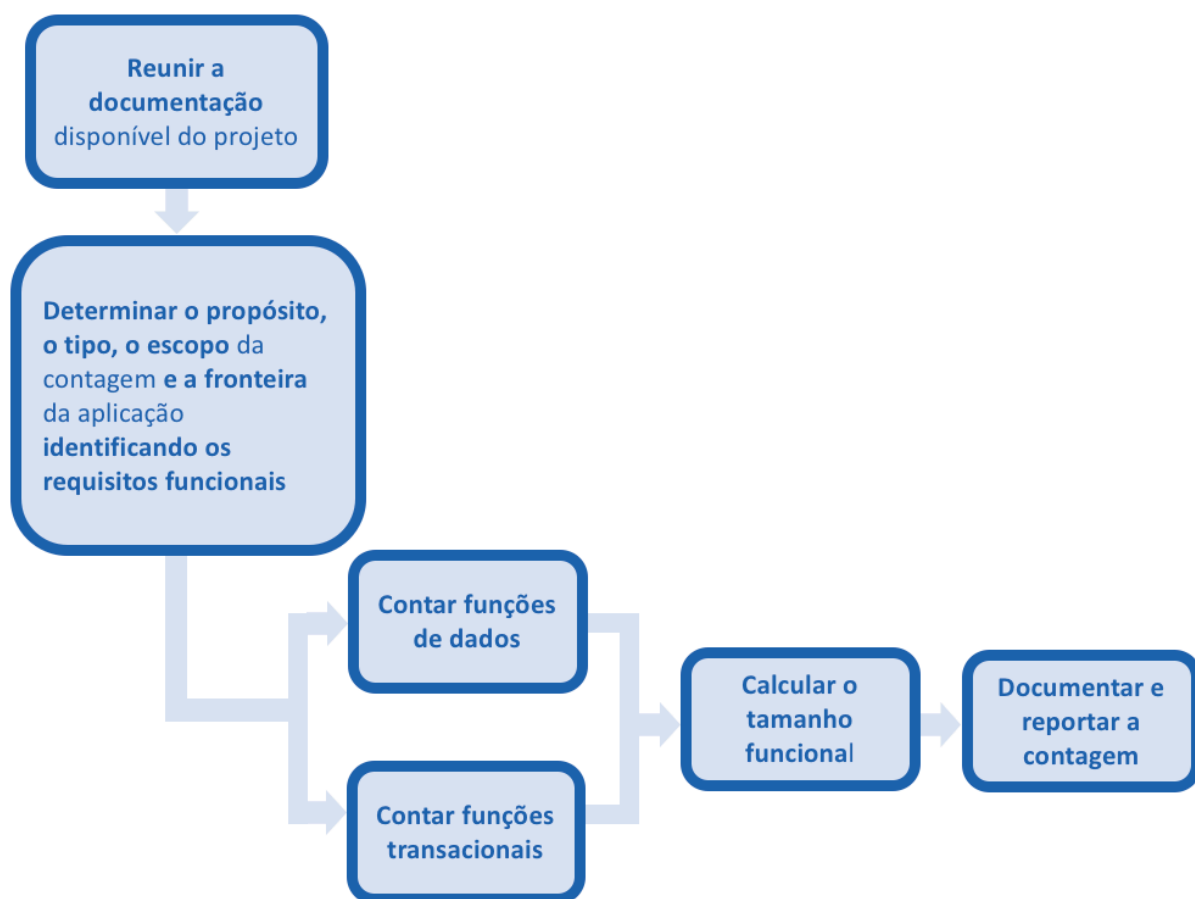
Além do método de Análise de Pontos de Função (APF) tradicional, que é um método amplamente utilizado pelos órgãos do SISP, esse roteiro traz também o método de Ponto de Função Simples (Método SFP, descrito no item 5 deste roteiro) como uma alternativa confiável, segura e capaz de acelerar consideravelmente o processo de contagem de PF.

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. O manual de contagem de pontos de função (CPM 4.3.1) destina-se a mensurar o tamanho funcional de projetos de software, não tendo por objetivo principal suportar contratos de prestação de serviços de desenvolvimento e manutenção de sistemas. Assim, torna-se necessário criar roteiros complementares, contemplando questões não abordadas pelos manuais do IFPUG, mas vivenciadas pelos órgãos e entidades do SISP.

4.1. PROCESSO DE CONTAGEM - MÉTODO APF

A **Figura 1** ilustra o procedimento de medição de pontos de função utilizando o método APF tradicional, descrito em nos subitens seguintes.

Figura 1: Processo de contagem de Pontos de Função conforme CPM 4.3.1



1. **Reunir a documentação disponível:** a contagem de pontos de função se inicia com a análise de toda a documentação disponível do projeto em questão, visando a identificação dos requisitos funcionais.
2. Determinar o propósito, o tipo e o escopo da contagem e a fronteira da aplicação:
 - **O propósito da contagem** fornece uma resposta para uma questão de negócio; sendo a questão do negócio a ser resolvida que determina o propósito. Como exemplo, o propósito da contagem poderia ser necessidade de dimensionar um projeto de um novo sistema para auxiliar no seu processo de contratação. Com base no propósito da contagem são definidos o escopo e o tipo de contagem.

- **O escopo da contagem** identifica quais funcionalidades serão incluídas na contagem de pontos de função;
- **O tipo de contagem** é determinado com base no propósito da contagem. Conforme CPM 4.3.1, existem três tipos de contagem:
 - **Contagem de pontos de função de projeto de desenvolvimento:** mede as funções entregues ao final da primeira release da aplicação e eventuais funções de conversão de dados;
 - **Contagem de pontos de função de projeto de melhoria:** mede as funções alteradas, incluídas e excluídas e eventuais funções de conversão de dados;
 - **Contagem de pontos de função de aplicação (ou baseline):** mede as funções de um software instalado.
- **A fronteira da aplicação** 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. O estabelecimento da fronteira da aplicação pode ser subjetivo, por exemplo, em uma aplicação com vários módulos, a fronteira pode ser estabelecida para cada módulo ou subsistema ou, ainda, pode-se considerar toda a aplicação, dependendo da visão do usuário. De fato, a definição da fronteira depende de processos de negócios, além disso, o posicionamento da fronteira influencia fortemente a contagem de pontos de função.

Conforme subitem 5.2.5.2 da Portaria SGD/MGI nº 750, de março de 2023:

Deve-se estabelecer, de forma clara, para fins de mensuração de pontos de função, as fronteiras das aplicações, considerando que:

a) a correta identificação da fronteira de uma aplicação é fundamental para o emprego consistente da métrica de análise de pontos de função, evitando-se contagens superdimensionadas ou subdimensionadas.

b) o posicionamento incorreto da fronteira pode alterar a perspectiva da medição de uma visão lógica (visão funcional) para uma visão física.

c) as principais consequências da não definição de fronteiras das aplicações são a contagem duplicada de transações e arquivos de dados, a contagem incorreta de funções de transferência de dados e dificuldade na contagem de arquivos.

d) uma fronteira de aplicação não pode ser subdividida por contextos gerenciais de desenvolvimento, por exemplo, interno e externo ao órgão, ou baseada em diferenças de plataformas ou tecnologias.

A seguir apresentam-se dicas para apoiar na identificação da fronteira da aplicação, conforme CPM 4.3.1:

- Utilize as especificações externas do sistema ou obtenha um fluxo do mesmo e desenhe a respectiva fronteira, destacando as partes internas e as externas à aplicação.
- Verifique como os grupos de dados estão sendo mantidos.
- Identifique as áreas funcionais, alocando certos tipos de objetos da análise (tais como entidades ou processos elementares) a uma área funcional.
- Observe dados de medição correlatos, tais como esforço, custo e defeitos. As fronteiras consideradas para os pontos de função e para os outros dados de medição devem ser as mesmas.
- Entreviste os especialistas no assunto para auxiliar na identificação da fronteira.

Complementarmente, vale informar que o Diagrama de Contexto é frequentemente utilizado para ilustrar o conceito de fronteira de aplicação, principalmente no início do desenvolvimento. É uma importante ferramenta para representar visualmente o limite lógico entre o sistema que está sendo medido, os usuários e outras aplicações.

Devido a subjetividade para o estabelecimento da fronteira da aplicação, recomenda-se que em editais para contratação de projetos de manutenção sejam definidas as fronteiras de todas as aplicações a serem contratadas. Os roteiros de contagem dos órgãos e entidades também devem definir as fronteiras das aplicações implantadas em um anexo e este deve ser atualizado sempre que for implantada uma nova aplicação.

3. **Contar funções de dados:** uma vez estabelecida a fronteira da contagem, é possível realizar o mapeamento dos requisitos de dados. A funcionalidade de dados satisfaz os requisitos funcionais do usuário referentes a armazenar e/ou referenciar dados. Toda função de dados, dentro do escopo da contagem, deve ser avaliada para identificar cada grupo lógico de dados. As regras de contagem, exemplos de identificação e como determinar a complexidade funcional de cada função de dados são encontradas no CPM 4.3.1.

No método APF temos as seguintes funções de dados:

- **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.

Como foi mencionado, as regras e exemplos de contagem de pontos de função encontram-se no CPM 4.3.1.

A seguir, apresentam-se regras e exemplos de contagem de DERs (Dado Elementar Referenciado), RLRs (Registro Lógico Referenciado) para funções de dados, extraídos do CPM 4.3.1, que não substituem a leitura e estudo do manual de contagem do método:

Tabela 1: Regras para contagem de Funções de Dados (Fonte: CPM 4.3.1)

Definição	Regras para contagem de funções de dados (AIEs e ALIs)	Exemplos
DER Um tipo de dado elementar. É um campo único, reconhecido pelo usuário e não repetido	1. Contar um DER para cada campo único, reconhecido pelo usuário e não repetido, mantido ou recuperado pela função de dados durante a execução de todos os processos elementares no escopo da contagem.	Por exemplo, o(s) resultado(s) do cálculo de um processo elementar, como o valor do imposto sobre uma venda, referente a um pedido de cliente mantido em um ALI é contado como um DER no ALI de pedido de cliente.
	2. Contar apenas os DERs que estão sendo usados pela aplicação que está sendo medida quando duas ou mais aplicações estiverem sendo mantidas e/ou referenciando a mesma função de dados.	Por exemplo, a Aplicação A pode identificar e utilizar um endereço como: rua, cidade, estado e CEP. A Aplicação B pode ver o endereço como um bloco de dados sem considerar os componentes individuais. A Aplicação A contaria quatro DERs; a Aplicação B contaria um DER.
	3. Contar um DER para cada parte de dado requisitada pelo usuário para estabelecer um relacionamento com outra função de dado.	Por exemplo, na Aplicação de RH, as informações de um funcionário são mantidas dentro de um ALI. O nome da função do funcionário é incluído como parte das informações do funcionário. Este DER é contado por que é necessário para relacionar um funcionário a uma função existente na organização. Este tipo de dado elementar é conhecido como chave estrangeira.

	<p>4. Revisar os atributos relacionados para determinar se eles estão agrupados e contados como um simples DER ou se são contados como DERs múltiplos; o agrupamento dependerá de como os processos elementares usam os atributos dentro da aplicação.</p>	<p>Por exemplo, um número de conta que é armazenado em vários campos é contado como um DER.</p> <p>Os atributos (primeiro nome, nome do meio, sobrenome) são agrupados e contados como:</p> <p>nome (primeiro nome, nome do meio, sobrenome) se esses atributos sempre forem utilizados juntos, primeiros nomes (primeiro nome e inicial do meio) e sobrenome se, além do que está acima, o sobrenome for utilizado independentemente, ou primeiro nome, inicial do meio e sobrenome, se os três puderem ser utilizados independentemente.</p>
<p>RLR</p> <p>Um Tipo de Registro Elementar (RLR) é um subgrupo de dados reconhecido pelo usuário dentro de uma função de dados</p>	<p>1. Contar um RLR para cada função de dados (por padrão cada função de dado tem um subgrupo de DERs para ser contado como um RLR).</p> <p>2. Contar um RLR adicional para cada subgrupo de DER lógico adicional (com a função de dados) que contém mais de um DER: entidade associativa com atributos não-chave subtipo (outro além do primeiro subtipo) e entidade atributiva, em um relacionamento que não seja obrigatório 1-1.</p>	<p>Por exemplo, em uma Aplicação de Recursos Humanos, a informação para um funcionário é adicionado através da adição de informações gerais. Além das informações gerais, o funcionário é um assalariado ou horista.</p> <p>O usuário determinou que um funcionário ou é assalariado ou é horista. Cada tipo de funcionário possui atributos próprios. Os dois tipos podem ter informações sobre dependentes. Neste exemplo, existem três subgrupos ou RLRs, como mostrado abaixo:</p> <p>Funcionário assalariado; incluindo informações gerais;</p> <p>Funcionário horista; incluindo informações gerais;</p>

2. **Contar funções transacionais:** uma função de transação é um processo elementar que provê funcionalidade para o usuário processar dados. Todas as funções transacionais, dentro do escopo da contagem, devem ser avaliadas a fim de identificar cada processo elementar único. As regras de contagem e exemplos para identificar e determinar a complexidade funcional de cada função de transação são encontradas no CPM 4.3.1. No método APF, cada função transacional corresponde a um (1) processo elementar, que pode ser uma entrada externa (EE), uma consulta externa (CE) ou uma saída externa (SE):

- **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. A intenção primária de uma consulta externa é apresentar dados ao usuário através de recuperação de dados ou informação de controle.
 - A lógica de processamento de uma Consulta Externa (CE) não contém fórmula matemática, nem cálculo, nem cria dados derivados.
 - Em uma Consulta Externa (CE), nenhum ALI é mantido durante o processamento, nem o comportamento do sistema é alterado.
- **Saída Externa (SE):** é um processo elementar que envia dados ou informação de controle para fora da fronteira da aplicação. e inclui processamento adicional além daquele existente em uma consulta externa. A intenção primária de uma saída externa é apresentar dados ao usuário através de lógica de processamento que não seja apenas recuperação de dados ou informação de controle. Diferentemente de uma CE:
 - A lógica de processamento de uma Saída Externa (SE) deve contar ao menos uma fórmula matemática ou cálculo, e/ou criar dados e/ou manter um ou mais ALIs, e/ou alterar o comportamento do sistema.

Como mencionado, as regras de contagem de pontos de função e exemplos encontram-se no CPM 4.3.1.

A seguir, apresentam-se regras e exemplos de contagem de ALRs (Arquivo Lógico Referenciado) e DERs (Dado Elementar Referenciado) para funções transacionais, extraídos do CPM 4.3.1, que não substituem a leitura e estudo do manual de contagem do método:

Tabela 2: Regras para contagem de Funções Transacionais (Fonte: CPM 4.3.1)

	Definição	Regras para contagem de funções de transação (EEs, CEs e SEs)	Exemplos/Notas
ALRs	Um tipo de arquivo referenciado (ALR) é uma função de dados lida e/ou mantida pela função de transação	Para cada função de transação, um ALR deve ser contado para cada função de dados única que for acessada (lida e/ou gravada) pela função de transação	Um tipo de arquivo referenciado inclui: um arquivo lógico interno (ALI) lido ou mantido por uma função de transação; ou um arquivo de interface externa (AIE) lido por uma função de transação.
DERs	Um tipo de dado elementar (DER) é um campo único, reconhecido pelo usuário e não repetido.	1. Revisar tudo o que atravessa (entre e/ou saia) a fronteira.	
		2. Contar um DER para cada atributo único, reconhecido pelo usuário e não repetido que atravessa (entre e/ou saia) a fronteira durante o processamento da função de transação.	DERs que atravessam a fronteira incluem: - Atributos que o usuário introduz por meio de uma tela, bem como aqueles apresentados em um relatório ou tela, — Atributos que entram pela fronteira da aplicação e que são necessários para especificar quando, o quê e/ou como os dados devem ser recuperados ou gerados pelo processo elementar, — Atributos fornecidos pelo usuário da função de transação, ou apresentados a ele, e — Atributos em um arquivo eletrônico que entrem ou saiam pela fronteira.
		3. Contar apenas um DER por função de transação para a habilidade de enviar uma mensagem de resposta da aplicação, mesmo que existam várias mensagens.	Se forem apresentadas diversas mensagens de erro/confirmação ao usuário, somente um DER será contado.

4. Contar apenas um DER por função de transação para a habilidade de iniciar ações, mesmo que existam diversas maneiras de fazer isso.

Se o usuário puder iniciar a geração de um relatório clicando no botão OK ou pressionando uma tecla de função, apenas um DER será contado.

5. Não contar os seguintes itens como

DERs:

- Constantes literais;
- Rótulos gerados pela aplicação;
- Variáveis de paginação, números de página e informações de posicionamento;
- Auxílios à navegação;
- Atributos gerados dentro da fronteira por uma função de transação e salvos em um ALI sem sair pela fronteira;
- Atributos recuperados ou referenciados de um ALI ou AIE para participarem do processamento sem que saiam pela fronteira.

Exemplos de constantes literais: títulos de relatórios, identificadores de telas ou painéis, cabeçalhos de colunas ou títulos de atributos;

Exemplos de rótulos gerados pela aplicação: atributos referentes a data e hora;

Exemplo de variáveis de paginação: “Linhas 37 a 54 de 211”;

Exemplo de auxílios à navegação: a habilidade de navegar em uma lista utilizando “anterior”, “próximo”, “primeiro”, “último” e seus equivalentes gráficos;



A contagem de pontos de função deve seguir rigorosamente as regras de contagem do manual de práticas de contagem do método escolhido (CPM 4.3.1 ou SPM 2.1) e as definições complementares do roteiro de métricas do órgão. Ademais, deve ser realizada por profissionais capacitados.

As regras de contagem e os exemplos para identificação e contagem das funções de dados (ALIs e AIEs) e transacionais (EEs, CEs e SEs) do método APF são encontrados no manual de práticas de contagem do método (CPM v.4.3.1).

3. **Avaliar a complexidade (Baixa, Média, Alta) e a contribuição funcional:** após a identificação dos tipos funcionais para cada requisito funcional, deve-se avaliar a complexidade (Baixa, Média, Alta) e a contribuição funcional observando as regras de contagem de pontos de função e exemplos descritos no CPM 4.3.1. A identificação e a avaliação das complexidades dos tipos funcionais não podem ser realizadas de maneira subjetiva:

- a. A complexidade funcional de cada função de dados (ALIs e AIEs) deve ser determinada com base na quantidade de tipos de dados elementares (DERs) e de tipos de registros elementares (RLRs) associados ao ALI ou AIE:

Tabela 3: Complexidade das Funções de Dados (Fonte: CPM 4.3.1)

		DERs		
		1 – 9	20 – 50	> 50
RLRs	1	Baixa	Baixa	Média
	2 - 5	Baixa	Média	Alta
	>5	Média	Alta	Alta

- b. A complexidade funcional a cada EE, SE e CE identificada, é determinada com base no número de tipos de arquivos referenciados (ALRs) e tipos de dados elementares (DERs):

Tabela 4: Complexidade Funcional das EEs (Fonte: CPM 4.3.1)

		DERs		
		1 – 4	5 - 15	> 15
ALRs	0 - 1	Baixa	Baixa	Média
	2	Baixa	Média	Alta
	>2	Média	Alta	Alta

Tabela 5: Complexidade Funcional das CEs e SEs (Fonte: CPM 4.3.1)

		DERs		
		1 – 5	6 - 19	> 19
ALRs	0 - 1	Baixa	Baixa	Média
	2 – 3	Baixa	Média	Alta
	>3	Média	Alta	Alta

Nota: 1 CE tem no mínimo 1 ALR

A tabela a seguir apresenta a contribuição dos tipos funcionais na contagem de PF utilizando o método tradicional:

Tabela 6: Contribuição Funcional dos Tipos Funcionais (Fonte: CPM 4.3.1)

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



A complexidade funcional de cada função de dados deve ser determinada utilizando o número de DERs e RLRs. E a complexidade funcional de cada função transacional será determinada utilizando o número de ALRs e DERs.

As atividades a serem executadas para contar DERs, RLRs, ALR, os exemplos de identificação das funções de dados e transacionais e as regras de contagem do método são encontradas no CPM v.4.3.1.

4. **Calcular Tamanho Funcional:** a penúltima atividade do processo de contagem é o cálculo do tamanho funcional. O objetivo e escopo da contagem deverão ser considerados na seleção e utilização da fórmula apropriada para calcular o tamanho funcional. O CPM 4.3.1 define dois tipos de projetos de software:

- **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 se consideram 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 se consideram as funcionalidades de conversão de dados, caso seja requisitado a migração ou carga inicial de dados no projeto de melhoria.

5. **Documentar e reportar a contagem:** é a última etapa do procedimento de contagem da figura 1.

- **Documentar:** uma contagem de pontos de função completamente documentada facilitará a rastreabilidade, usabilidade e manutenibilidade. Deve-se realizar a documentação da contagem com todas as premissas e decisões de medição tomadas, os padrões utilizados, as diretrizes adotadas e os links para a documentação do projeto.
- **Reportar:** Recomenda-se que as contagens a partir do Roteiro de Métricas de Software do SISP sejam reportadas conforme determinado pelo CPM, ou seja, S FP (IFPUG–IS–c), indicando que o resultado da contagem de pontos de função não mantém conformidade plena com o CPM e o padrão internacional de contagem de PF (ISO/IEC 20926:200x) e se mantém conformidade com uma customização, neste caso, o Roteiro de Métricas de Software do SISP.

Assim: S FP (IFPUG–IS–c)

Onde:

- **S** é o resultado da contagem de pontos de função;
- **FP** (*Function Point*) é a unidade de tamanho do método FSM (*Functional Size Measurement*) do IFPUG; PF em português.
- **IS** (*International Standard*) é o padrão internacional (ISO/IEC 20926:200x);
- **c** representa um ou mais caracteres indicando que o resultado não mantém conformidade plena com o padrão internacional.

Exemplo: 250 PF* (IFPUG–ISO/IEC 20926:200x–sisp); PF* na versão em português.

5. CONTAGEM DE PONTOS DE FUNÇÃO - MÉTODO SFP

O Ponto de Função Simples (Simple Function Point - SFP) é também um método de medição funcional compatível e consistente com o IFPUG APF, descrito no CPM v.4.3.1, em terminologia e definições. O método IFPUG SFP adota a suposição de que o valor funcional de um software é proporcional apenas ao número de transações lógicas e ao número de arquivos lógicos necessários.

Ele é útil no início do processo de desenvolvimento, independe da metodologia e tecnologia adotada, oferece resultados confiáveis, repetíveis e objetivos. Foi projetado para ser ágil, rápido, leve e de fácil utilização, conforme descrito no Manual de Práticas de Contagem do Método - SPM v.2.1.

A decisão de usar o método IFPUG SFP pode ser influenciada por vários fatores como: a fase do ciclo de vida de desenvolvimento de software, as restrições de cronograma, o uso dos resultados, as informações disponíveis, a disponibilidade de especialistas no assunto etc. Visto que a medida é orientada ao produto de software, pode ser derivada em qualquer estágio do ciclo de vida de desenvolvimento do software, dadas as informações apropriadas.

Pode ser utilizado para:

- Apoiar análises de qualidade e produtividade;
- Complementar métricas de equipe;
- Estimar o tamanho, custo e os recursos necessários para o desenvolvimento, evoluções e manutenções;
- Estimar o tamanho funcional de software a ser desenvolvido;
- Contagem de pontos de função de software em desenvolvimento;
- Contagem de pontos de função de projetos de melhoria;
- Contagem de pontos de função da baseline da aplicação;

O que torna o método SFP ágil, simples e leve? Como ele pode acelerar consideravelmente o processo de dimensionamento funcional se utiliza os mesmos conceitos da APF tradicional?

A medição com o método SFP não exige qualquer detalhe adicional à própria identificação da funcionalidade. Ele não diferencia os tipos de processo elementar (EE, CE ou SE) e tipos de arquivo lógico (AIE ou ALI), tampouco a sua complexidade.

Não é necessária a identificação de DERs, ALRs ou RLRs, nem a identificação de "intenção primária". Apenas dois componentes funcionais básicos: Processo Elementar e Arquivo Lógico.

As pontuações a serem atribuídas para os dois componentes funcionais básicos são: 7,0 SFP (para Arquivo Lógico) e 4,6 SFP (para Processo elementar).

Outro ponto muito importante é a previsibilidade. Não há diferença de complexidade entre as funcionalidades, então a contribuição funcional de cada processo elementar ou função de dados não será alterada ao longo do processo de desenvolvimento ou de melhoria do software.

O método tem correspondência de conceitos, tipos de medição e fórmulas de cálculo do método APF tradicional, o que leva a identificação das mesmas funções a serem medidas. A definição de um processo elementar é a mesma. Cada função transacional corresponde a um (1) Processo elementar no método SFP e cada função de dados corresponde a um (1) arquivo lógico no método SFP.

O contexto em que os órgãos do SISP estão inseridos exige métodos de medição rápidos, ágeis, com baixo impacto nos processos produtivos, que não exijam muita especialização, que seja confiável nos resultados, que não dependa de tecnologia e que esteja correlacionado ao trabalho, custo e duração de um projeto.

Esse roteiro traz esse método de medição que é leve, ágil, consistente com a estrutura da família de padrões ISO 14143, e totalmente compatível com o método IFPUG APF. É um método confiável, menos sujeito a subjetividade de interpretações, mais fácil, simples de aplicar e mais fácil de manter. Pode ser aplicado no início do processo de desenvolvimento do *software*. A sua utilização pode reduzir custos, tempo e discussões.

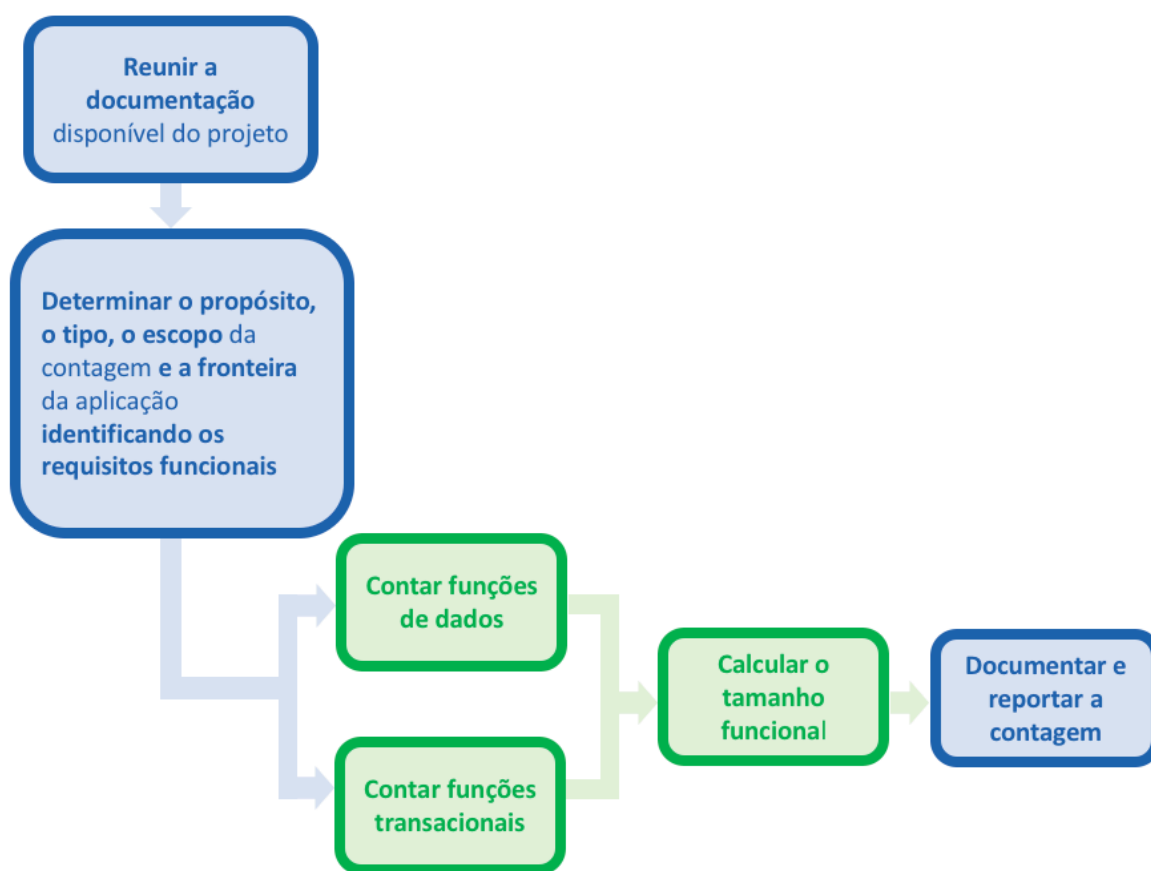
A seguir, descreve-se todo o processo de contagem do método SFP e a fórmula para conversão do método IFPUG APF em IFPUG SFP.

5.1. PROCESSO DE CONTAGEM DO MÉTODO SFP

O processo de mensuração do método SFP é idêntico ao processo de medição do método APF. Observe que a figura a seguir é a mesma utilizada para representar o processo de contagem do método APF, descrito no item 4.1 deste roteiro.

A diferença principal entre os dois métodos está na execução das atividades destacadas (em verde) na figura a seguir, descritas nos itens subsequentes:

Figura 2: Processo de contagem de Pontos de Função com atividades em destaque



Convém esclarecer que todas as regras de contagem do Ponto de Função Simples são encontradas no manual de contagem do método SFP (SPM v.2.1).

A seguir, o detalhamento de cada etapa do procedimento de contagem de pontos de função usando o método SFP:

1. **Reunir a documentação disponível para o projeto:** etapa que envolve a coleta de todas as informações necessárias para uma medição funcional confiável. Assim como o método APF, o método SFP independe de metodologia ou tecnologia adotada, entretanto o analista deve localizar todos os documentos do projeto e as pessoas que podem ser úteis nas etapas seguintes.

2. Determinar:

- **O propósito da contagem** fornece uma resposta para uma questão de negócio a ser resolvida. Com base no propósito da contagem são definidos o escopo e o tipo de contagem.
- **O tipo de contagem** é determinado com base no propósito da contagem. Conforme CPM 4.3.1 e SPM 2.1, existem três tipos de contagem:

a) **Contagem de pontos de função de projeto de desenvolvimento:** mede as funções entregues ao final da primeira *release* da aplicação e eventuais funções de conversão de dados;

b) **Contagem de pontos de função de projeto de melhoria:** mede as funções alteradas, incluídas e excluídas e eventuais funções de conversão de dados;

c) **Contagem de pontos de função de aplicação (ou baseline):** mede as funções de um *software* instalado.

- **O escopo da contagem** é determinado pelo propósito da atividade de medição. Nessa etapa define-se um (sub)conjunto de *software* a ser medido. O escopo pode incluir mais de um aplicativo de *software*.
- **A fronteira da aplicação** é determinada pela identificação das aplicações e seus usuários e também entre outras aplicações. Os limites são orientados por princípios lógicos, não técnicos, focados no ponto de vista do usuário (em qualquer nível de abstração). Vale destacar que, assim como no método de APF, no método SFP o posicionamento da fronteira influencia fortemente a contagem de pontos de função, portanto, em editais para contratação de projetos de manutenção é recomendada a definição das fronteiras de todas as aplicações a serem contratadas.

TOME NOTA

Toda contagem ou estimativa de pontos de função é realizada em uma fronteira da aplicação. Como a definição de fronteira da aplicação é baseada em processos de negócio, esta pode ser subjetiva. Por exemplo, um sistema de capacitação de empregados faz parte ou não da fronteira de um sistema de recursos humanos?

Em algumas organizações, um sistema de capacitação pode ser visto como parte do processo de negócio de gestão recursos humanos. Neste caso, trata-se de uma fronteira única de aplicação de recursos humanos, abrangendo o módulo de capacitação.

Em outras organizações, a capacitação de empregados pode ser tratada como um processo de negócio independente. Neste caso, tem-se duas fronteiras: o sistema de recursos humanos e o sistema de capacitação.

Ao ampliar a fronteira da aplicação para abranger módulos considerados independentes em outras situações, pode-se ter como efeito a redução da quantidade de AIEs, EEs, SEs e CEs. Por outro lado, o desmembramento de uma fronteira para considerar a hipótese de módulos independentes pode ter efeito inverso e gerar situações em que um componente (rotinas de validação de CPF, por exemplo) tenha de ser reproduzido em vários módulos, e consequentemente fazer parte de mais de uma fronteira.

Nesse contexto, pode-se afirmar que tanto a ampliação da fronteira da aplicação quanto a introdução de novas fronteiras podem ocasionar alterações no número de transações da aplicação e possivelmente alterar o quantitativo de arquivos de dados. Essas situações precisam ser consideradas na delimitação das fronteiras, devido ao impacto que têm na contagem dos pontos de função.

Conforme o item 6.17 (Componente Reusável) deste roteiro, componentes que são utilizados por mais de uma funcionalidade, estando ela definida em uma nova fronteira ou não, devem gerar apenas uma contagem de pontos de função.

A adoção de várias fronteiras (segurança, domínio, contexto, interface de usuário etc.), por sua vez, pode gerar múltiplas instâncias da aplicação (por exemplo: web, móvel, API), com repercussão na quantidade de AIEs, EEs, SEs e CEs, potencialização da duplicidade de componentes e impacto significativo no custo final do produto. Múltiplas instâncias estão tratadas no item 7.1 (Contagem com Múltiplas Mídias) deste roteiro de métricas.

Nesse sentido, é fundamental estabelecer e documentar as fronteiras das aplicações. Recomenda-se que as fronteiras dos sistemas a serem mantidos estejam documentadas no edital de contratação e também no roteiro de métricas do órgão.

É importante definir também quais serão as fronteiras das novas aplicações a serem contratadas, que devem estar em conformidade com o Plano Diretor de TI do órgão contratante.

3. **Contar funções de dados:** nesta etapa identificam-se os arquivos lógicos, que são conjuntos lógicos de dados usados pelos processos elementares. Os dois métodos (APF e SFP) levam a identificação das mesmas funções de dados, entretanto neste método não é feita a diferenciação entre os arquivos lógicos internos (ALI, consultados e que armazenam dados) e os arquivos de interface externa (AIE, usados apenas para consulta de dados). No método SFP cada função de dados corresponde a um (1) arquivo lógico:
 - **Arquivo lógico:** representa a funcionalidade fornecida ao usuário para atender aos requisitos de armazenamento de dados internos e externos.
4. **Contar funções transacionais:** nesta etapa os processos elementares devem ser identificados. As regras e exemplos de identificação dos processos elementares são encontradas no manual do método (SPM v.2.1). No método SFP cada função transacional corresponde a um (1) processo elementar:
 - **Processo Elementar:** a menor unidade de atividade significativa para o usuário, que constitui uma transação completa. É independente e deixa o aplicativo que está sendo medido em um estado consistente.

Tabela 7: Diferença na identificação dos tipos funcionais entre os métodos SFP e APF

O método SFP identifica:		O método APF identifica:
Funções de Dados	Arquivo Lógico (AL)	Arquivo Lógico Interno (ALI)
		Arquivo de Interface Externa (AIE)
Funções transacionais	Processo Elementar (PE)	Entrada Externa (EE)
		Consulta Externa (CE)
		Saída Externa (SE)

5. **Calcular o tamanho funcional:** o cálculo do tamanho funcional para o método SFP requer apenas a identificação de dois componentes funcionais básicos: (1) Processo Elementar e (2) Arquivo Lógico. Para o método IFPUG SFP não são diferenciados os tipos de processo elementar e os tipos de arquivo lógico, tampouco as suas complexidades. Não é necessária a identificação de DERs, ALRs ou RLRs, tampouco a identificação da “intenção primária”. Ainda assim, convém realçar que as regras de contagem de pontos de função simples - descritas no

SPM 2.1 - devem ser observadas e seguidas. As pontuações a serem atribuídas para os dois componentes funcionais básicos do método SFP são:

Tabela 8: Pontuação dos Componentes funcionais básicos do método IFPUG SFP

Componentes funcionais básicos do IFPUG SFP	Pontuação
Arquivo Lógico (AL)	7,0 SFP
Processo Elementar (PE)	4,6 SFP

6. **Documentar e reportar a contagem:** a última etapa do processo de medição é realizar a documentação da contagem e reportar a contagem. Todas as premissas e decisões de medição tomadas, os padrões utilizados, as diretrizes adotadas e os links para a documentação do projeto devem ser documentados e reportados. No SPM 2.1 é apresentado um conjunto mínimo de informações necessárias para essa tarefa.

Caso a organização já utilize o método APF tradicional, pode prosseguir com o seu uso ou realizar conversão para o método SFP, descrita no item seguinte.

O método SFP tem correspondência de conceitos, tipos de medição e fórmulas de cálculo do método padrão IFPUG. Os tipos de projetos apresentados no item 6 deste roteiro aplicam-se aos dois métodos de contagem de PF (APF e SFP) descritos neste roteiro de métricas.

5.2. CORRESPONDÊNCIA E CONVERSIBILIDADE ENTRE OS MÉTODOS

Conforme manual do método SFP (SPM 2.1), a análise da correspondência teórica entre os elementos do método IFPUG FP e o método IFPUG SFP encontrou as seguintes evidências:

- correspondência dos conceitos de aplicação, escopo, limite e finalidade;
- correspondência de tipos de medição;

- correspondência dos componentes funcionais básicos;
- correspondência de fórmulas de cálculo: as fórmulas são semelhantes; há diferença apenas na fórmula usada para atualizar as linhas de base após aprimoramento funcional, uma vez que não há mudança na complexidade.

Não é possível derivar algoritmicamente uma medida PF de uma medida SFP. Tal impossibilidade se dá pelo fato de que para medir o tamanho de um projeto, utilizando o método SFP, é preciso que sejam identificados apenas processos elementares e arquivos lógicos. Para derivar dos processos elementares os EEs, CEs ou SEs e dos arquivos lógicos os ALLs e AIEs com suas respectivas complexidades é necessário voltar aos requisitos funcionais para buscar as informações detalhadas necessárias para obter uma medida de ponto de função pelo método APF.

Para derivar um valor SFP de um valor PF, deve-se utilizar a fórmula a seguir:



$$\text{SFP} = (\text{QtdEES} + \text{QtdCEs} + \text{QtdSEs}) \times 4,6 + (\text{QtdALLs} + \text{QtdAIEs}) \times 7$$

Onde:

- **QtdEES:** quantidade de Entradas Externas (EES) contadas com o método de APF tradicional;
- **QtdCEs:** quantidade de Consultas Externas (CEs) contadas com o método de APF tradicional;
- **QtdSEs:** quantidade de Saídas Externas (SEs) contadas com o método de APF tradicional;
- **QtdALLs:** quantidade de Arquivos Lógicos Internos (ALLs) contados com o método de APF tradicional;
- **QtdAIEs:** quantidade de Arquivos de Interface Externa (AIEs) contados com o método de APF tradicional.

5.3. EXEMPLO DE APLICAÇÃO DA CONVERSIBILIDADE

Para exemplificar a aplicação da conversibilidade entre os métodos, suponha que uma aplicação possua:

- Quantidade e tipos de funções transacionais: 10 EEs, 5 CEs, 10 SEs;
- Quantidade e tipos funções de dados: 5 ALLs e 2 AIEs.

Aplicando a fórmula para conversibilidade de PF para SFP, temos:



$$\text{SFP} = (\text{QtdEEs} + \text{QtdCEs} + \text{QtdSEs}) \times 4,6 + (\text{QtdALIs} + \text{QtdAIEs}) \times 7$$

$$\text{SFP} = (10 + 5 + 10) \times 4,6 + (5 + 2) \times 7$$

$$\text{SFP} = (25) \times 4,6 + (7) \times 7$$

Em síntese, o método SFP:

- Requer apenas a identificação de dois componentes funcionais básicos: Processo Elementar (PE) e Arquivo Lógico (AL);
- Não exige a identificação da “intenção primária”;
- Não requer a diferenciação dos tipos de processo elementar (EE, CE, SE) e dos tipos de arquivo lógico (ALI, AIE), tampouco a sua complexidade;
- Não demanda a identificação de DERs, ALRs ou RLRs;
- Estabelece que a pontuação funcional do processo elementar (PE) é de 4,6 SFP e do arquivo lógico (AL) é de 7,0 SFP.

6. CÁLCULO DE PONTOS DE FUNÇÃO PARA O SISP

Este capítulo tem como propósito descrever os diversos tipos de projetos de software e definir métricas para seu dimensionamento baseadas nas regras de contagem do CPM 4.3.1 e SPM 2.1.



Como foi dito em outras partes deste documento, o método SFP tem correspondência de conceitos, tipos de medição e fórmulas de cálculo do método padrão IFPUG.

Os tipos de projetos apresentados nos itens a seguir aplicam-se aos dois métodos de contagem de PF expostos neste roteiro: APF e SFP.

Nos itens seguintes serão apresentadas fórmulas para dimensionamento de projetos de software – considerando os métodos SFP e APF - que utilizarão os seguintes termos técnicos:

- **PF_INCLUÍDO ou SFP_INCLUÍDO:** pontos de função associados às novas funcionalidades que farão parte da aplicação após um projeto de desenvolvimento ou de manutenção.
- **PF_ALTERADO ou SFP_ALTERADO:** pontos de função associados às funcionalidades existentes na aplicação que serão alteradas no projeto de manutenção.
- **PF_EXCLUÍDO ou SFP_EXCLUÍDO:** pontos de função associados às funcionalidades existentes na aplicação que serão excluídas no projeto de manutenção.
- **PF_CONVERSÃO ou SFP_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 (Processo Elementar para o método SFP / Entrada Externa para o método APF) e relatórios associados à migração de dados, caso requisitado pelo usuário (Processo Elementar para o método SFP / SE ou CE para o método APF). Observe que os dados carregados em um processo de migração não devem ser contados como Arquivos Lógicos da aplicação.
- **PF_NÃO_AJUSTADO ou SFP_NÃO_AJUSTADO:** corresponde a contagem de Pontos de Função, aplicando as regras do método de medição de tamanho funcional (FSM) do IFPUG, sem a aplicação de fatores de ajuste.

Os pontos a seguir também devem ser observados pela organização:

- **Supressão do PF_CONVERSÃO/SFP_CONVERSÃO:** este roteiro recomenda a supressão do PF_CONVERSÃO/SFP_CONVERSÃO das fórmulas de contagem de pontos de função de projetos de desenvolvimento e de melhoria nos casos específicos onde for caracterizado um esforço relativamente maior dessa atividade. Por exemplo, os projetos que envolvem a migração de dados de banco de dados e o tratamento de funções complexas de migração de dados. Nesses casos, recomenda-se tratá-los como projetos separados de migração de dados.
- **Documentação de projetos de desenvolvimento ou de manutenção:** deve-se registrar a solicitação e documentar os requisitos do projeto de forma detalhada, visando apoiar a contagem de pontos de função da demanda, independentemente do método de contagem de PF a ser utilizado (APF ou SFP).
- **Fator de Impacto conforme processos técnicos da ISO/IEC 12207:** Os percentuais de FI utilizados em algumas fórmulas de contagem observam a Tabela 15 (item 9.1), que utiliza os processos técnicos e atividades de referência do ciclo de vida do software, padronizados pela ABNT NBR ISO/IEC 12207. Em alguns casos, o órgão contratante pode não ter necessidade de contratar todas as fases do ciclo de vida do *software*. Dessa forma, para a remuneração da contratada, o órgão deve considerar as fases contratadas e a composição do time de referência para desenvolvimento das fases a serem contratadas.

Convém destacar que a métrica Ponto de Função é usada para dimensionamento do tamanho funcional, ou seja, dimensiona projetos de software com base nos requisitos funcionais da aplicação, não contemplando diretamente os requisitos não-funcionais do projeto.

6.1.PROJETO DE DESENVOLVIMENTO

É o projeto para desenvolver e entregar a primeira versão de uma aplicação de *software*. Ao criar um novo software, dois componentes devem ser considerados: as funções incluídas (PF_INCLUÍDO ou SFP_INCLUÍDO) e as funções de conversão (PF_CONVERSÃO ou SFP_CONVERSÃO) que apoiam o início do uso do *software*, como a população de arquivos lógicos por exemplo.

Segue a fórmula de cálculo utilizada no dimensionamento de projetos de desenvolvimento de *software*, que é idêntica para os métodos APF e SFP:

$$PF_DESENVOLVIMENTO = PF_INCLUÍDO + PF_CONVERSÃO$$

$$SFP_DESENVOLVIMENTO = SFP_INCLUÍDO + SFP_CONVERSÃO$$



Para o caso exposto, as funções de conversão farão parte apenas da medição do projeto de desenvolvimento, mas não da medição de baseline na aplicação após o desenvolvimento.

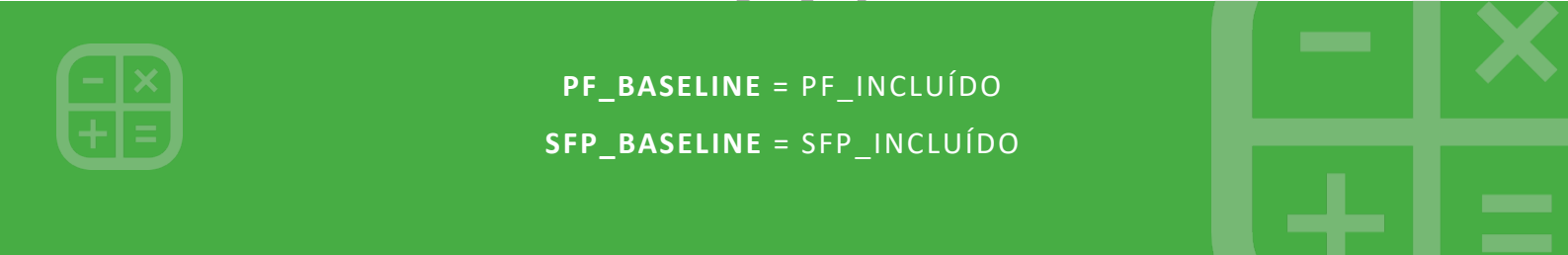
Este roteiro recomenda a supressão do PF_CONVERSÃO / SFP_CONVERSÃO das fórmulas de contagem de projetos de desenvolvimento quando for caracterizado um esforço relativamente maior dessa atividade. Nesses casos, recomenda-se tratá-los como projetos separados de migração de dados, conforme item 6.6.

6.2. LINHA DE BASE APÓS DESENVOLVIMENTO INICIAL

Ao final de um projeto de desenvolvimento inicial, a medição da linha de base do *software* vai considerar apenas as funções de transação e as funções de dados incluídas (PF_INCLUÍDO ou SFP_INCLUÍDO); não vai considerar as funções de conversão (PF_CONVERSÃO ou SFP_CONVERSÃO).

Caso sejam consideradas, as funções de conversão farão parte apenas da medição do projeto de desenvolvimento inicial, mas não da medição da baseline da aplicação.

Segue a fórmula de cálculo utilizada no dimensionamento da linha de base após desenvolvimento inicial, que é a mesma para os métodos APF e SFP:


$$\begin{aligned} \text{PF_BASELINE} &= \text{PF_INCLUÍDO} \\ \text{SFP_BASELINE} &= \text{SFP_INCLUÍDO} \end{aligned}$$

6.3. 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 implantadas.

Conforme CPM 4.3.1, mudanças nas funcionalidades podem ocorrer a partir de novos requisitos, revisão de requisitos do usuário, mudanças legais/regulamentares ou novos usuários. A ênfase deve ser nos requisitos de negócio, com a contagem de PF do projeto de melhoria refletindo a intenção dos requisitos do usuário:

- Uma função de dados (ALI ou AIE/ Arquivo Lógico) é considerada alterada, quando a alteração contemplar mudanças de tipos de dados, inclusão ou exclusão de tipos de dados, ou mudança de tamanho (número de

posições) ou tipo de campo (por exemplo: mudança de numérico ou alfanumérico), sendo que esta ocorre por mudança de regra de negócio do usuário.

- Uma função transacional (EE, CE ou SE/ Processo Elementar) é considerada alterada, quando a alteração contemplar:
 - Mudança de tipos de dados em uma função existente;
 - Mudança de arquivos referenciados; e
 - Mudança de lógica de processamento, segundo as ações das lógicas e processamento do 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).

O dimensionamento de um projeto de melhoria consiste na soma de funcionalidades incluídas, alteradas, excluídas e funcionalidades de conversão de dados.

A fórmula de cálculo utilizada no dimensionamento de um projeto de melhoria para os métodos APF e SFP é:



$$\begin{aligned} \text{PF_MELHORIA} &= \text{PF_INCLUÍDO} + (\text{FI}^* \times \text{PF_ALTERADO}) + \\ &\quad (0,32 \times \text{PF_EXCLUÍDO}) + \text{PF_CONVERSÃO} \\ \text{SFP_MELHORIA} &= \text{SFP_INCLUÍDO} + (\text{FI}^* \times \text{SFP_ALTERADO}) + \\ &\quad (0,32 \times \text{SFP_EXCLUÍDO}) + \text{SFP_CONVERSÃO} \end{aligned}$$



Onde o **FI*** (**Fator de Impacto**) pode variar conforme condições abaixo:

- **FI = 47%** para funcionalidade de sistema desenvolvida ou mantida por meio de um projeto de melhoria pela empresa contratada.

- **FI = 55%** para funcionalidade de sistema não desenvolvida ou mantida por meio de um projeto de melhoria pela empresa contratada:
 - A contratada deve documentar ou atualizar a documentação da funcionalidade mantida, gerando um registro completo da funcionalidade, aderente ao processo de software da contratante, mesmo que a empresa contratada não tenha desenvolvido ou realizado manutenção prévia na funcionalidade que está demandando o projeto de melhoria.
 - Se houver uma nova demanda de projeto de melhoria da funcionalidade em questão, será considerado que a contratada já desenvolveu a funcionalidade.

Os percentuais de FI (fator de impacto) acima observam a Tabela 15 (item 9.1) que utiliza processos técnicos de referencia do ciclo de vida do software, padronizados pela ABNT NBR ISO/IEC 12207.

Este roteiro recomenda a supressão do PF_CONVERSÃO/SFP_CONVERSÃO das fórmulas de contagem de pontos de função de projetos de melhoria quando for caracterizado um esforço relativamente maior dessa atividade. Nesses casos, recomenda-se tratá-los como projetos separados de migração de dados, conforme item 6.6 deste roteiro.

6.4. BASELINE APÓS MELHORIA - MÉTODO SFP

Conforme exposto no item 5.2, as fórmulas dos dois métodos de contagem de PF são semelhantes (APF e SFP); a diferença está apenas na fórmula usada para atualizar a baseline após um projeto de melhoria, uma vez que não há mudança na complexidade e pontuação dos componentes funcionais básicos (Arquivo Lógico – AL e Processo Elementar - PE) no método SFP.

Após um projeto de melhoria, a medição da baseline do aplicativo lançado (SFP_BASELINE) será a medição da baseline anterior ao projeto de melhoria, mais as novas funções (SFP_INCLUÍDO) e menos as funções removidas (SFP_EXCLUÍDO):



$$\text{SFP_BASELINE_NOVA} = \text{SFP_BASELINE_ANTERIOR} + \text{SFP_INCLUÍDO} - \text{SFP_EXCLUÍDO}$$



Para o método SFP, na contagem da baseline da aplicação não devem constar as funcionalidades alteradas em um projeto de melhoria, visto que estas já foram incluídas a partir de outro projeto e, portanto, já estão contempladas na

baseline; ademais, o método SFP não determina a complexidade das funções, não gerando assim alteração em sua pontuação.

6.5.BASELINE APÓS MELHORIA – MÉTODO APF

Conforme CPM 4.3.1, a combinação de funcionalidades incluídas, o efeito das mudanças feitas nas funcionalidades existentes e as funcionalidades excluídas devem ser utilizados para medir não apenas o projeto de melhoria, mas também para atualizar a contagem de PF da aplicação.

Para o método APF, quando uma funcionalidade é alterada por um projeto de melhoria pode ter sua pontuação de complexidade alterada. Neste caso o analista deve:

- Determinar as complexidades das funções antes da alteração (a partir da documentação da última medição ou medindo como existiam antes da mudança);
- Determinar as complexidades das funções depois da alteração;
- Atualizar na baseline da aplicação, contemplando a alteração da complexidade das funções existentes após o projeto de melhoria, se for o caso.

A fórmula para contagem da baseline após um projeto de melhoria no método APF, considera a atualização da baseline, a adição das funções incluídas (PF_INCLUÍDO) e a subtração das funções excluídas (PF_EXCLUÍDO), conforme fórmula a seguir:


$$PF_BASELINE = (PF_BASELINE^*) + PF_INCLUÍDO - PF_EXCLUÍDO$$

Onde:

- PF_BASELINE* corresponde a atualização da pontuação das funções existentes, caso haja alteração em sua complexidade após a execução do projeto de melhoria.

Para o método APF, na contagem da baseline da aplicação não devem constar as funcionalidades de conversão e nem as excluídas, visto que a baseline representa o tamanho funcional da aplicação que está sendo medida.

6.6. PROJETOS DE MIGRAÇÃO DE DADOS

Este roteiro recomenda a supressão do (PF_CONVERSÃO ou SFP_CONVERSÃO) das fórmulas de contagem de pontos de função de projetos de desenvolvimento e de melhoria nos casos específicos onde for caracterizado um esforço relativamente maior dessa atividade, tais como nos casos de migração de dados de banco de dados hierárquico para relacional e no tratamento de funções de migração de dados.

Nesses casos, recomenda-se tratar esse serviço como projeto separado de migração de dados. Os projetos de migração de dados devem seguir a fórmula abaixo, idêntica para os métodos APF e SFP:


$$PF_CONVERSÃO = PF_INCLUÍDO \times 0,47$$

$$SFP_CONVERSÃO = SFP_INCLUÍDO \times 0.47$$


Um projeto de migração deve contemplar minimamente:

- Para o método APF: os ALIs mantidos pela migração, as Entradas Externas (considerando as cargas de dados nos ALIs) e, caso seja solicitado pelo usuário, os relatórios gerenciais das cargas que serão contados como consultas externas ou saídas externas;
- Para o método SFP: os arquivos lógicos (AL) mantidos pela migração, os processos elementares (PE) considerando as cargas de dados nos arquivos lógicos e, caso seja solicitado pelo usuário, os relatórios gerenciais das cargas (relatórios de erros, de exceções, de conversão e/ou de controle) que serão contados como processos elementares.

O percentual de ajuste utilizado na fórmula acima observa a Tabela 15 (item 9.1), que utiliza os processos técnicos de referência do ciclo de vida do software, padronizados pela ABNT NBR ISO/IEC 12207.

Todas as contagens devem ser realizadas com base nas funcionalidades requisitadas e recebidas pelo usuário. Tanto as funções de dados como as funções transacionais devem ser incluídas na medição do projeto de migração de dados, mas

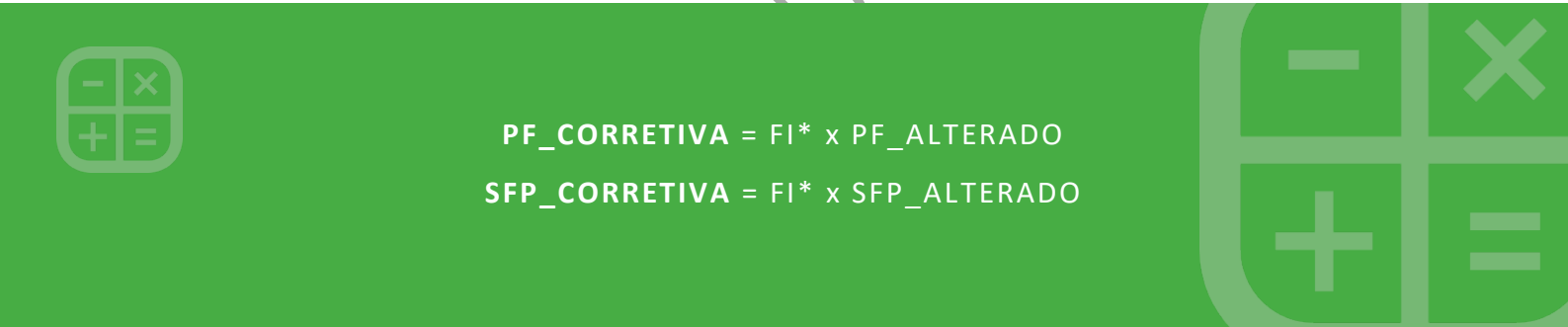
não devem fazer parte do tamanho funcional do software, porque não integram as funcionalidades que estarão disponíveis para o usuário após a implantação do sistema.

6.7.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. Caso não exista cláusula contratual de garantia, deve ser considerada a garantia preconizada por lei (Código do Consumidor).

Quando o sistema estiver fora da garantia ou não tiver 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 /SFP_ALTERADO:


$$\begin{aligned} \text{PF_CORRETIVA} &= \text{FI} * \text{PF_ALTERADO} \\ \text{SFP_CORRETIVA} &= \text{FI} * \text{SFP_ALTERADO} \end{aligned}$$

Fator de Impacto (FI):

- **FI = 47%** quando estiver fora da garantia e a correção for feita pela mesma empresa que desenvolveu a funcionalidade.
- **FI = 55%** quando estiver fora da garantia e a correção for feita por empresa diferente daquela que desenvolveu a funcionalidade.
 - A contratada deve documentar ou atualizar a documentação da funcionalidade mantida, gerando um registro completo da funcionalidade, aderente ao processo de software da contratante, mesmo que a empresa contratada não tenha desenvolvido ou realizado manutenção prévia na funcionalidade que está demandando um projeto de melhoria.

- Se houver uma nova demanda correção da funcionalidade em questão, será considerado que a contratada já desenvolveu a funcionalidade.

Os fatores de impacto acima observam a Tabela 15 (item 9.1) que utiliza processos técnicos de referência do ciclo de vida do software, padronizados pela ABNT NBR ISO/IEC 12207.

6.8.MUDANÇA DE PLATAFORMA

São considerados nesta categoria, os projetos que precisam ser migrados para outra plataforma. Por exemplo, um sistema legado em ASP.Net que necessita ser redesenvolvido em Lavarel; o banco de dados de um sistema legado que precisa ser migrado para PostgreSQL.

Recomenda-se enfaticamente a 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; neste caso pode-se utilizar apenas o percentual relativo aos processos de teste, apresentados na **Tabela 15 (item 9.1)** que utiliza processos técnicos de referencia do ciclo de vida do software, padronizados pela ABNT NBR ISO/IEC 12207.

6.8.1. Mudança de Plataforma - Linguagem

Nesta categoria encontram-se as demandas de redesenvolvimento de sistemas em outra linguagem de programação.

Para estes casos, a aferição do tamanho em pontos de função deve considerar um fator de impacto (FI) sobre o PF_INCLUÍDO /SFP_INCLUÍDO:

- **FI = 84%**, caso o projeto legado não possua documentação;
- **FI = 73%**, caso o projeto já possua documentação de requisitos.

E ainda, deve-considerar um FI de 47% para as funções de conversão, conforme fórmulas a seguir:



$$\text{PF_REDESENV_LINGUAGEM} = (\text{PF_INCLUÍDO} \times \text{FI}^*) + (\text{PF_CONVERSÃO} \times 0,47)$$
$$\text{SFP_REDESENV_LINGUAGEM} = (\text{SFP_INCLUÍDO} \times \text{FI}^*) + (\text{SFP_CONVERSÃO} \times 0,47)$$

Caso não existam mudanças nas funções de dados, ou seja, o banco de dados da aplicação seja mantido, as funções de dados não devem ser contabilizadas para fins de pagamento (caso o contrato utilize a modalidade de remuneração por Ponto de Função). Contudo, recomenda-se que as funções de dados sejam registradas a fim de compor a baseline do projeto.

Ressalta-se que os percentuais de FI se baseiam na Tabela 15 (item 9.1), que utiliza processos técnicos de referência do ciclo de vida do *software*, padronizados pela ABNT NBR ISO/IEC 12207.

Este roteiro recomenda a supressão do PF_CONVERSÃO ou SFP_CONVERSÃO da fórmula de contagem de projetos de redensenvolvimento quando for caracterizado um esforço relativamente maior dessa atividade. Nesses casos, recomenda-se tratá-los como projetos separados de migração de dados, conforme item 6.6.

6.8.2. Mudança de Plataforma – Banco de Dados


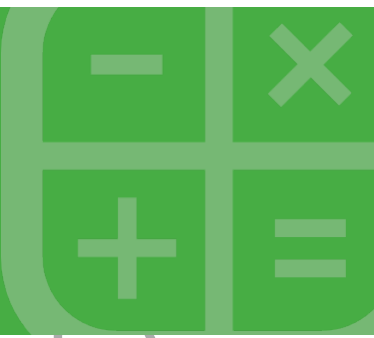
Nesta categoria encontram-se as demandas de redensenvolvimento de sistemas para utilizar um outro sistema gerenciador de banco de dados.

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 contadas. No entanto, nesse caso, deve ser realizada a contagem das funções de dados a fim de compor a documentação da contagem final do projeto.

Em casos de mudança de banco hierárquico para relacional, a aferição do tamanho em pontos de função deve considerar um fator de impacto (FI) sobre o PF_INCLUÍDO /SFP_INCLUÍDO e PF_CONVERSÃO/ SFP_CONVERSÃO:

- **FI = 84%**, caso o sistema que não possua documentação;
- **FI = 73%**, caso o sistema já possua documentação de requisitos; e
- **FI = 47%** para as funções de conversão.

Assim, será utilizada a fórmula abaixo:


$$\begin{aligned}\text{PF_REDESENV_BD_HIERÁRQUICO} &= \\ &(\text{PF_INCLUÍDO} \times \text{FI}^*) + (\text{PF_CONVERSÃO} \times 0,47) \\ \text{SFP_REDESENV_BD_HIERÁRQUICO} &= \\ &(\text{SFP_INCLUÍDO} \times \text{FI}^*) + (\text{SFP_CONVERSÃO} \times 0,47)\end{aligned}$$




Nos projetos de redesenvolvimento de banco de dados hierárquico para relacional, recomenda-se a supressão do PF_CONVERSÃO/SFP_CONVERSÃO da fórmula acima. Nesses casos, recomenda-se tratá-los como projetos separados de migração de dados, conforme item 6.6.



Se o projeto já possui documentação de requisitos, o percentual referente aos processos de definição de requisitos apresentados na tabela 15 (item 9.1) deve ser desconsiderado da contagem.

O órgão deve considerar apenas os percentuais referentes aos processos do ciclo de vida a serem executados.

Caso a demanda de redesenvolvimento seja de um sistema gerenciador de banco de dados relacional para outro relacional, deve ser utilizada a seguinte fórmula:


$$\begin{aligned}\text{PF_REDESENV_BD_RELACIONAL} &= \\ &(\text{PF_ALTERADO} \times 0,39) + (\text{PF_CONVERSÃO} \times 0,47) \\ \text{SFP_REDESENV_BD_RELACIONAL} &= \\ &(\text{SFP_ALTERADO} \times 0,39) + (\text{SFP_CONVERSÃO} \times 0,47)\end{aligned}$$


A definição dos percentuais dos fatores de impacto (FI) apresentados nas fórmulas acima, baseiam-se nos processos técnicos de referência do ciclo de vida do software, padronizados pela ABNT NBR ISO/IEC 12207 e descritos na Tabela 15 (item 9.1).

Convém destacar que nas contagens de redesenvolvimento de banco de dados relacional para outro banco de dados relacional:

- O PF_ALTERADO ou SFP_ALTERADO deve considerar apenas as funcionalidades impactadas;
- As funcionalidades que possuem apenas demandas de testes, devem ser contadas usando o percentual referente aos processos de teste, conforme tabela 15, item 9.1;
- Em mudanças de banco relacional para relacional, geralmente a estrutura de dados não é alterada, desta forma não se contam as funções de dados;
- Recomenda-se tratar o PF_CONVERSÃO ou SFP_CONVERSÃO dentro do mesmo projeto.

6.9. 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*) 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. Um ponto a ser observado é a classificação, em alguns casos, dessas demandas como componente interno reusável (seção 6.17).

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. Para esses casos, deve-se utilizar um percentual contemplando apenas os processos técnicos de teste, conforme tabela 15 do item 9.1 deste roteiro.

Caso os testes apontem a necessidade de atualizar alguma função transacional, deve-se utilizar as fórmulas apresentadas nas subseções seguintes.

Ressalta-se que a definição dos fatores de impacto (FI) apresentados nas fórmulas a seguir, baseiam-se nos processos técnicos de referência do ciclo de vida do software, padronizados pela ABNT NBR ISO/IEC 12207 e descritos na Tabela 15 (item 9.1).

6.9.1. Atualização de Versão - Linguagem

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 a seguir:



$$\begin{aligned} \text{PF_ATUALIZAÇÃO_VERSÃO_LINGUAGEM} &= \text{PF_ALTERADO} \times 0,39 \\ \text{SFP_ATUALIZAÇÃO_VERSÃO_LINGUAGEM} &= \text{SFP_ALTERADO} \times 0,39 \end{aligned}$$

O PF_ALTERADO/SFP_ALTERADO deve considerar apenas as funcionalidades impactadas.

As funcionalidades que possuem apenas demandas de testes, devem ser contadas usando o percentual correspondente, conforme Tabela 15 (item 9.1).

6.9.2. 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.

Essas demandas devem ser dimensionadas de acordo com a fórmula abaixo:





$$\begin{aligned} \text{PF_ATUALIZAÇÃO_VERSÃO_BROWSER} &= \text{PF_ALTERADO} \times 0,39 \\ \text{SFP_ATUALIZAÇÃO_VERSÃO_BROWSER} &= \text{SFP_ALTERADO} \times 0,39 \end{aligned}$$

O PF_ALTERADO/SFP_ALTERADO deve considerar apenas as funcionalidades impactadas. As funcionalidades que possuem apenas demandas de testes, devem ser contadas utilizando o percentual relativo aos processos técnicos de testes da Tabela 15 (item 9.1).

Essas atualizações podem implicar em manutenções em componentes comuns da plataforma utilizada. Nesses casos, a demanda deve ser contada como componente interno reusável, descrita na seção 6.17 deste roteiro.

6.9.3. 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 a seguir, idêntica para os dois métodos:


$$\begin{aligned} \text{PF_ATUALIZAÇÃO_VERSÃO_BD} &= \text{PF_ALTERADO} \times 0,39 \\ \text{SFP_ATUALIZAÇÃO_VERSÃO_BD} &= \text{SFP_ALTERADO} \times 0,39 \end{aligned}$$


O PF_ALTERADO/SFP_ALTERADO deve considerar apenas as funcionalidades impactadas. As funcionalidades que possuem apenas demandas de testes, devem ser contadas usando o percentual relativo aos processos técnicos de testes da Tabela 15 (item 9.1).

6.10. 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.

Neste tipo de manutenção não são contadas funções de dados.



$$PF_INTERFACE = PF_ALTERADO \times 0,39$$

$$SFP_INTERFACE = SFP_ALTERADO \times 0,39$$



6.11. ADAPTAÇÃO 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 estar navegando no sistema.

A demanda deve ser contada como manutenção adaptativa considerando as funcionalidades impactadas. Observe que se trata de mudança em validação com regra de negócio não funcional. Para estes casos, na contagem de uma ou mais funcionalidades que sofreram impacto, deve-se considerar um fator de impacto (FI) sobre o PF_ALTERADO/SFP_ALTERADO (apresentados na seção 6.3), conforme fórmula a seguir, igual para os dois métodos:

$$\begin{aligned} \text{PF_ADAPTATIVA} &= \text{FI}^* \times \text{PF_ALTERADO} \\ \text{SFP_ADAPTATIVA} &= \text{FI}^* \times \text{SFP_ALTERADO} \end{aligned}$$

Onde o FI* (Fator de Impacto) pode variar conforme condições abaixo:

- **FI = 47%** para funcionalidade de sistema desenvolvida ou mantida por meio de um projeto de melhoria pela empresa contratada.
- **FI = 55%** para funcionalidade de sistema não desenvolvida ou mantida por meio de um projeto de melhoria pela empresa contratada:
 - Se houver uma nova demanda de projeto de manutenção adaptativa na funcionalidade em questão, será considerado que a contratada já desenvolveu a funcionalidade.

Deve-se destacar que além da adequação das funcionalidades em questão, a atualização da documentação do projeto de manutenção adaptativa deve ser realizada.

6.12. APURAÇÃO ESPECIAL

Consideram-se projetos de apuração especial, funcionalidades desenvolvidas e 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 6.12.1;

- Gerar um relatório específico ou arquivo para o usuário por meio de recuperação de informações nas bases da aplicação, detalhados na subseção 6.12.2.

Caso a apuração seja de correção de dados devido a erros de funcionalidades de aplicações desenvolvidas pela contratada, deve-se observar as cláusulas contratuais com relação a garantias e prazos de correção.

Existem também os casos de:

- ALTERAÇÃO de uma apuração especial, tratados na subseção 6.12.3; e
- REEXECUÇÃO de uma apuração especial, tratados na subseção 6.12.4.



Recomenda-se ao órgão contratante **SEMPRE SOLICITAR FORMALMENTE** para a empresa contratada **O ARMAZENAMENTO DO SCRIPT** para permitir posterior alteração e/ou reexecução.

Cumpra-se destacar que apesar das demandas de apuração especial serem solicitadas pelo usuário requisitante, elas não farão parte do *software* instalado e portanto, não devem ser contabilizadas na *Baseline*.

6.12.1. 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.

Destaca-se 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; como exemplo temos a inclusão de valor “sim” ou “não” no campo “indicador de matriz” referente ao CNPJ.

Normalmente, nesse tipo de atualização são afetados múltiplos registros. Para estes casos, considera-se apenas a contagem das funcionalidades desenvolvidas. Para o método APF, geralmente essas funcionalidades são classificadas como Entradas Externas. Para esse cenário, 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.

Nesse sentido, foram identificados três tipos de Apuração Especial - Base de Dados, cujas fórmulas de cálculo apresentadas a seguir utilizam fatores de impacto (FI) baseados nos processos técnicos de referência do ciclo de vida do software, padronizados pela ABNT NBR ISO/IEC 12207 e descritos na Tabela 15 (item 9.1) deste roteiro.

6.12.1.a. Atualização de dados sem consulta prévia

Existem casos de Apuração Especial – Base de Dados, em que o usuário não solicita uma consulta prévia das informações. Nesses casos, deve ser utilizada a fórmula a seguir, análoga para os dois métodos:


$$PF_APURAÇÃO_BD = PF_INCLUÍDO \times 0,39$$

$$SFP_APURAÇÃO_BD = SFP_INCLUÍDO \times 0,39$$


6.12.1.b. Consulta prévia sem atualização

Em outros 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.

Essa é uma prática interessante com o fim de evitar futuras atualizações errôneas na base de produção dos sistemas.

A consulta prévia não é definida pela empresa contratada. Deve ser solicitada obrigatoriamente pelo órgão contratante para a avaliação da viabilidade de implementar a Apuração Especial - Base de Dados, conforme fórmula abaixo:


$$PF_CONSULTA_PRÉVIA = PF_INCLUÍDO \times 0,39$$

$$SFP_CONSULTA_PRÉVIA = SFP_INCLUÍDO \times 0,39$$


Para o método APF, essa consulta prévia é classificada como CE ou SE e deve ser dimensionada considerando sua complexidade funcional.

Para o método SFP, essa consulta corresponde a um processo elementar cuja pontuação funcional é 4,6 SFP, visto que neste método não é necessário diferenciar os tipos de processo elementar, tampouco a sua complexidade.

6.12.1.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 impacto na fórmula de contagem da Apuração Especial - Base de Dados, seguindo a fórmula abaixo, equivalente para os dois métodos:


$$\begin{aligned} \text{PF_APURAÇÃO_BD_PÓS_CONSULTA_PRÉVIA} &= \text{PF_INCLUÍDO} \times 0,28 \\ \text{SFP_APURAÇÃO_BD_PÓS_CONSULTA_PRÉVIA} &= \text{SFP_INCLUÍDO} \times 0,28 \end{aligned}$$


6.12.2. Apuração especial – Geração de Relatórios

Trata-se de um projeto de apuração especial 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.

Vale destacar que essas funções são executadas apenas uma vez, não fazendo parte da aplicação. Nesses casos, considera-se a contagem das funcionalidades desenvolvidas:

- Para o método APF frequentemente essas funcionalidades são classificadas como Saídas Externas; mas podem ser classificadas como Consultas Externas, caso não possuam cálculos ou criação de dados derivados;
- Para o método SFP essas funcionalidades são classificadas como Processo Elementar com pontuação funcional igual a 4,6 SFP, visto que neste método não é necessário diferenciar os tipos de processo elementar tampouco a sua complexidade.

Deve-se utilizar a fórmula abaixo, igual para os dois métodos:


$$\begin{aligned} \text{PF_APURAÇÃO_RELATÓRIOS} &= \text{PF_INCLUÍDO} \times 0,39 \\ \text{SFP_APURAÇÃO_RELATÓRIOS} &= \text{SFP_INCLUÍDO} \times 0,39 \end{aligned}$$


É importante ressaltar que - para os dois métodos (APF e SFP) - as funções de dados associadas aos dados atualizados não devem ser contadas, visto que não houve mudanças nas estruturas dos Arquivos Lógicos.

6.12.3. Apuração especial – Alteração

Recomenda-se sempre solicitar formalmente à contratada o armazenamento do script para permitir posterior alteração. Uma funcionalidade solicitada como apuração especial pode ser considerada alterada, quando a alteração contemplar:

- Mudança de tipos de dados em uma apuração especial já executada;
- Mudança de arquivos referenciados; e
- Mudança de lógica de processamento, segundo as ações das lógicas e processamento do CPM 4.3.1.

Nesse sentido, se for solicitada a alteração de uma apuração especial - conforme cenários descritos nos subitens 6.12.1 ou 6.12.2 - esta deve ser dimensionada com a aplicação de um fator de impacto de 26%, aplicando a fórmula a seguir:


$$\begin{aligned} \text{PF_ALTERAÇÃO_APURAÇÃO} &= \text{PF_ALTERADO} \times 0,26 \\ \text{SFP_ALTERAÇÃO_APURAÇÃO} &= \text{SFP_ALTERADO} \times 0,26 \end{aligned}$$

6.12.4. Apuração especial – Reexecução

Como já foi mencionado no início do item 6.12, recomenda-se sempre solicitar formalmente para a empresa contratada o armazenamento do script para permitir posterior alteração e/ou reexecução. Desta forma, quando for solicitada a reexecução de uma apuração especial, esta deve ser dimensionada com a aplicação de um fator redutor de 15% na contagem da apuração especial em questão, da seguinte maneira:



$$PF_REEXECUÇÃO_APURAÇÃO = PF_NÃO_AJUSTADO \times 0,15$$

$$SFP_REEXECUÇÃO_APURAÇÃO = SFP_NÃO_AJUSTADO \times 0,15$$

A contagem de pontos de função não-ajustada corresponde a contagem de Pontos de Função, utilizando as regras do método de medição de tamanho funcional (FSM) do IFPUG, sem a aplicação de fatores de ajuste.

6.13. DESENVOLVIMENTO, MANUTENÇÃO E PUBLICAÇÃO DE PÁGINAS ESTÁTICAS

Nesta seção são tratados desenvolvimentos e manutenções específicas em páginas estáticas de portais, *intranets* ou *websites*. As demandas desta seção abrangem a publicação de páginas *Web* com conteúdo estático. Por exemplo: criação de página HTML, atualização de menu estático, atualização de texto ou banner estáticos em páginas HTML existentes.

Caso o desenvolvimento de páginas estáticas esteja contido em um projeto de desenvolvimento, então elas serão contabilizadas no projeto de desenvolvimento e não devem ser mensuradas em separado. Ou seja, esta seção se aplica quando ocorrer a demanda exclusivamente para o desenvolvimento ou manutenção de páginas estáticas:



$$PF_PUBLICAÇÃO = PF_ALTERADO \times 0,31$$

$$SFP_PUBLICAÇÃO = SFP_ALTERADO \times 0,31$$

As demandas de criação de logomarcas ou identidade visual, além de outras demandas de criação de arte, associadas à área de Comunicação Social, não são enquadradas nessa categoria, mas podem ser mensuradas com a métrica HST (Horas de Serviço Técnico) quando forem complementares ao desenvolvimento e manutenção do software, seguindo as regras descritas na seção 15 deste roteiro.

6.14. MANUTENÇÃO DE DOCUMENTAÇÃO DE SISTEMAS LEGADOS

Nesta seção são tratadas demandas de documentação ou atualização de documentação de sistemas legados. Nessas demandas geralmente o analista realiza uma engenharia reversa da aplicação para gerar a documentação. Assim, foi definido o fator de impacto considerando as atividades executadas para atendimento da demanda, conforme processos técnicos padronizados pela ABNT NBR ISO/IEC 12207 e descritos na Tabela 15 (item 9.1):


$$PF_DOCUMENTAÇÃO = PF_NÃO_AJUSTADO \times 0,18$$

$$SFP_DOCUMENTAÇÃO = SFP_NÃO_AJUSTADO \times 0,18$$


Caso a demanda seja geração de artefatos de documentação de outras etapas do processo de desenvolvimento, deve-se utilizar outro fator de impacto baseado nos processos técnicos correspondentes e os artefatos a serem gerados. As premissas utilizadas devem ser definidas nas cláusulas contratuais e registradas no documento de estimativas do projeto.

6.15. VERIFICAÇÃO DE ERROS

As verificações de erro ou a análise e solução de problemas são as demandas – apontadas pelo órgão - referentes a todo comportamento anormal ou indevido do sistema.

Nesses casos, 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 6.7).

Se não for 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:



$$\text{PF_VERIFICAÇÃO} = \text{PF_Funcionalidade_Reportada_Com_Erro} \times 0,23$$
$$\text{SFP_VERIFICAÇÃO} = \text{SFP_Funcionalidade_Reportada_Com_Erro} \times 0,23$$

Ressalta-se 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.

6.16. 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].

Nesses casos, o tamanho das funções a serem apenas testadas deve ser aferido em Pontos de Função de Teste (PFT). Na contagem de Pontos de Função de Teste, não devem ser consideradas as funcionalidades incluídas, alteradas ou excluídas do projeto de manutenção.

Para os dois métodos (APF e SFP) a contagem de PFT deve corresponder ao somatório da pontuação funcional de todas as funções transacionais envolvidas no teste:



$$\text{PFT} = \text{Somatório dos Tamanhos das Funções Transacionais Testadas}$$

A conversão do PFT com o método tradicional ou com o método de ponto de função simplificado deve ser feita de acordo com a fórmula abaixo:



$$PF_TESTES = PFT \times 0,15$$

$$SFP_TESTES = PFT \times 0,15$$



- Uma função só pode ser contada uma vez, mesmo quando é testada várias vezes com cenários diferentes;
- As funcionalidades da aplicação que necessitem de teste devem ser requisitadas pela contratante;
- As funções testadas, consideradas no PFT, devem ser documentadas pela contratada considerando a documentação de testes definida no processo de desenvolvimento da contratante.

6.17. COMPONENTE REUSÁVEL

Em alguns casos são demandadas manutenções em componentes, que implementam regras de negócio, específicos de uma aplicação e estes são reusados por várias funcionalidades da aplicação. Por exemplo, uma mudança em uma rotina de validação de um CPF usada em várias funcionalidades de cadastro.

Se considerarmos o método de contagem de projetos de melhoria do CPM, seriam contadas todas as funcionalidades impactadas por essa mudança. No entanto, este roteiro propõe que o componente, o qual deverá ser testado, seja considerado como um processo elementar independente e sua alteração seja contada aplicando-se um fator de impacto (FI) sobre o PF_ALTERADO ou SFP_ALTERADO, seguindo os conceitos apresentados na seção 6.3 - Projeto de Melhoria.



$$PF_COMPONENTE = FI* \times PF_ALTERADO$$

$$SFP_COMPONENTE = FI* \times SFP_ALTERADO$$



Além disso, as funcionalidades da aplicação que necessitem de teste devem ser requisitadas pela contratante e dimensionadas conforme proposto na seção 6.16.

Um exemplo de manutenção de componentes é a mudança em tópico de um menu de um sistema que aparece em todas as telas da aplicação. Neste caso, a contagem pode ser realizada considerando o componente “Apresentar Menu”.

Existem também casos em que são realizadas manutenções de valores de elementos internos de configuração que afetam 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 e arquivos de internacionalização:



$$\text{PF_COMPONENTE_ARQUIVO} = \text{PF_ALTERADO} \times 0,47$$
$$\text{SFP_COMPONENTE_ARQUIVO} = \text{SFP_ALTERADO} \times 0,47$$



VERSÃO CONSULTA

7. ORIENTAÇÕES COMPLEMENTARES

Este capítulo apresenta diretrizes complementares aos Manuais de Práticas de Contagem do IFPUG (CPM 4.3.1 e SPM 2.1) e reforça pontos sensíveis nas contratações que podem impactar significativamente o resultado de uma contagem em caso de falhas.

7.1. CONTAGEM 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 as abordagens, *single instance* e *multiple instance*, para a aplicação das regras definidas no CPM. A determinação da contagem de PF seguindo a abordagem *multiple instance* ou *single instance* depende da avaliação da equipe de métricas da instituição.

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, 2010a]:

- **Canal:** também se refere 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, uma 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 específica 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.

7.1.1. 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, sugere-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.

7.1.2. CENÁRIO 2: Dados de saída idênticos

Uma aplicação grava dados em um arquivo de saída e imprime um relatório com informações idênticas às gravadas no arquivo.

Nesses casos, sugere-se a utilização da abordagem *single instance* considerando que os dados impressos e os dados apresentados no arquivo de saída sejam idênticos e que a ferramenta de desenvolvimento apoie a geração dessas múltiplas saídas. Assim, apenas uma funcionalidade será incluída na contagem de pontos de função.

7.1.3. CENÁRIO 3: Dados de saída distintos

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*.

7.1.4. CENÁRIO 4: Dados de entrada batch e on-line idênticos

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.

Para esses casos, sugere-se a utilização da 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*.

7.2. ABAS E IDENTIFICAÇÃO DE PROCESSO ELEMENTAR

Um Processo Elementar é a menor unidade de atividade que é significativa para o(s) usuário(s). O Processo Elementar deve ser autocontido e deixar o negócio da aplicação que está sendo contada em um estado consistente.

Um processo elementar com múltiplas etapas de processamento lógico não deve ser dividido em múltiplos processos elementares. Se um processo elementar é subdividido inapropriadamente, o mesmo não reúne os critérios de um processo elementar.

Como exemplo pode-se destacar o recurso de abas para a construção de interfaces em sistemas computacionais. Em geral, cada aba de uma tela não representa uma transação completa e desta forma não deve ser considerada um processo elementar isolado. O adequado para este tipo de situação é identificar um processo elementar para um conjunto de abas que se relacionam em vez de considerar um processo elementar para cada aba.

Ressalta-se a importância do atendimento a todos os critérios listados no Manual de Práticas de Contagem do IFPUG e da observação dos seus exemplos para a correta identificação de um processo elementar evitando, por exemplo, relacionar processos elementares com telas ou abas de uma transação, considerar fluxos alternativos como processos elementares distintos ou considerar transações que atualizem dados de código (descrito no item 7.3.1.).

7.3. PRINCIPAIS FALHAS DE CONTAGEM IDENTIFICADAS

O Relatório por Área de Gestão nº 5 da CGU evidenciou algumas falhas frequentes em contagens de pontos de função, com o método APF tradicional, em órgãos do SISP que podem ocasionar em divergências significativas no quantitativo de pontos de função mensurados. A listagem abaixo apresenta situações que constituem falhas de contagem:

- a) Identificação de um ALI para armazenamento de dados de código;
- b) Considerar transações que atualizam dados de código;
- c) Relacionar processos elementares com telas ou abas de uma transação não levando em consideração as regras de identificação e unicidade do processo elementar;
- d) Considerar fluxos alternativos como processos elementares distintos não levando em consideração as regras de identificação e unidade do processo elementar;
- e) Contar DER para variáveis de paginação, número de páginas e informação de posicionamento;
- f) Contar DER para ajudas de navegação como habilidade de navegar com uma lista utilizando “anterior”, “próximo”, “primeiro”, “último” e seus equivalentes gráficos;
- g) Contar mais de um DER por função de transação para a habilidade de iniciar ações quando há múltiplos meios para isso;
- h) Contar mais de um DER quando a função de transação tem capacidade de enviar várias mensagens.



Convém destacar que boa parte das falhas de contagem expostas neste item podem ser dirimidas com a utilização do método SFP, posto que este método requer apenas a identificação de dois componentes funcionais básicos:

- **Processo Elementar:** a menor unidade de atividade significativa para o usuário, que constitui uma transação completa. Corresponde a uma (1) função transacional; e
- **Arquivo Lógico:** representa a funcionalidade fornecida ao usuário para atender aos requisitos de armazenamento de dados internos e externos. Corresponde a uma (1) função de dados.

No método IFPUG SFP não são diferenciados as funções transacionais e as funções de dados, tampouco as suas complexidades. Também não é necessária a identificação de DERs, ALRs ou RLRs, tampouco a identificação da “intenção primária”.

7.3.1. Dados de Código

Dentre as principais falhas de contagem identificadas, destacamos que a contagem de Dados de Código não deve ser contabilizada na contagem funcional da aplicação.

Os Dados de Código são uma implementação de requisitos não-funcionais do usuário. Como consequência, os Dados de Código podem influenciar o tamanho não-funcional do produto de software, mas não o tamanho funcional do mesmo [“A Framework for Functional Sizing”, IFPUG 2003].

Conforme CPM 4.3.1, o impacto dos dados de código pertencerem à dimensão não-funcional é que nem os Dados de Código, nem as transações que os mantem devem ser contadas.

Nesse sentido, a contagem de dados de código provoca graves deformidades na contagem funcional, bem como na estimativa de esforço, prazo e custo, visto que o esforço para seu desenvolvimento geralmente é menor em relação a requisitos funcionais.

Abaixo destacamos os diferentes tipos de dados de código, descritos no CPM4.3.1:

Tabela 9: Tipos e Exemplos de Dados de Código conforme CPM 4.3.1

TIPOS	Descrição	Exemplos
Substituição	<p>Esse tipo de dado pode servir como um meio para tornar mais ágil a entrada de dados; pode ser implementado para economizar espaço de armazenamento ou ser um resultado de normalização.</p> <p>Este tipo de Dado de Código contém um código e um nome explicativo ou descrição.</p>	<p>Estados: Código do Estado, Nome do Estado;</p> <p>Cores: Código da Cor, Descrição da Cor.</p>
Uma ocorrência	<p>Esse tipo de dado possui uma e apenas uma ocorrência independentemente da quantidade de atributos. Os atributos são relativamente constantes; podem mudar, mas muito raramente.</p>	<p>Uma entidade com dados sobre uma organização em particular; por exemplo, nome e endereço.</p>
Estáticos e Constantes	<p>Dados estáticos</p> <p>Contém dados que são estáticos. A quantidade de instâncias e o conteúdo de uma instância raramente muda.</p>	<p>Uma entidade elementos químicos: símbolo, número atômico, descrição.</p>
Valores Default	<p>Este tipo de Dado de Código contém valores default para (alguns atributos</p>	

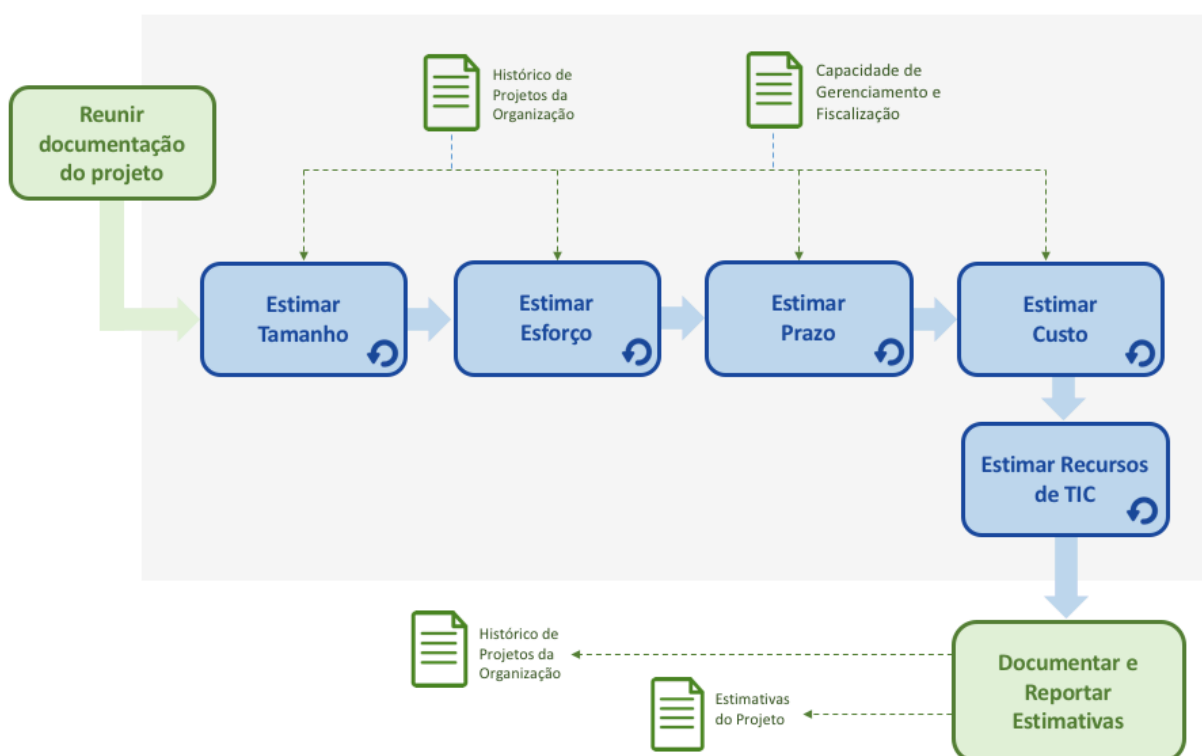
		em) novas instâncias de um objeto de negócio.	
Valores Válidos	Valores Válidos	Fornecer uma lista de valores válidos para um atributo de um ou mais tipos de objetos de negócio. É normalmente usado para listar valores disponíveis para seleção pelo usuário e/ou validar a entrada fornecida pelo mesmo.	Nome do estado: Contém todos os valores válidos para o atributo nome do estado. Código do estado: Contém todos os valores válidos para o atributo código do estado.
	Faixas de Valores	Este tipo de Dado de Código contém dados basicamente estáticos.	Faixa de Números de Telefone Permissíveis: menor número de telefone, maior número de telefone. Faixa de temperatura térmica.

Complementarmente, vale destacar que o CPM 4.3.1 (parte 3, capítulo 1) apresenta seção específica para auxiliar na identificação de Dados de Código e também para identificar o que NÃO SÃO DADOS DE CÓDIGO.

8. ORIENTAÇÕES PARA REALIZAR ESTIMATIVAS DE PROJETOS

Este capítulo apresenta orientações para uma etapa importante do planejamento projetos: a derivação de estimativas de tamanho, esforço, prazo e custo para projetos de desenvolvimento e melhoria de *software*.

Figura 3: Processo de Derivação de Estimativas de Projetos de Software



Com o fim de facilitar o entendimento, nos tópicos seguintes descrevem-se atividades do processo de derivação de estimativas retratado na imagem acima; para cada tipo de estimativa será apresentado um exemplo mostrando a sua aplicação. As diretrizes descritas a seguir serão apresentadas com auxílio do método de ponto de função simplificado (IFPUG SFP) e outras técnicas alinhadas à Portaria SGD/MGI nº 750, de 20 de março de 2023.

Além da documentação disponível do projeto, ao realizar as estimativas é importante analisar os dados históricos de projetos concluídos da organização, assim como a capacidade de gerenciamento e fiscalização de contratos da organização.

Mesmo em projetos ágeis, cabe ressaltar que outras análises, premissas e suposições utilizadas na geração das estimativas também devem ser documentadas e reportadas.

As estimativas devem ser atualizadas no decorrer do processo de desenvolvimento, conforme a necessidade.

Quando o projeto é concluído, deve-se documentar e reportar as estimativas de tamanho, prazo, custo, esforço e recursos utilizados, assim como outros atributos relevantes do projeto.

O histórico de projetos da organização e as estimativas devem ser atualizadas, visto que apoiam a organização e as equipes ágeis na tomada decisões, adaptações, inovação e entregar valor. Dentre outros benefícios, a documentação do projeto atualizada promove o alinhamento das expectativas e a transparência para o time ágil e para a organização.

8.1. REUNIR A DOCUMENTAÇÃO DISPONÍVEL

As estimativas normalmente são realizadas no início do processo de desenvolvimento ou melhoria do *software*. A primeira atividade a ser feita em um processo de derivação de estimativas é reunir e analisar toda a documentação disponível do projeto.

A documentação inicial de um projeto pode envolver: documento(s) de requisitos, documento de visão, amostras de relatórios, telas ou outras interfaces, demonstração da operação de um aplicativo, guia de usuário, documentação de design do sistema, especificação funcional e não funcional, roadmap do produto, backlog do produto, histórias de usuário, casos de uso, diagramas, modelos ou outro documento de formalização dos requisitos.

O estimador deve analisar cuidadosamente esses e/ou outros documentos relacionados ao projeto para garantir a qualidade da estimativa.

A realização das estimativas por um analista de métricas que não atue na equipe do projeto, constitui uma boa-prática. O analista de métricas deve analisar também a consistência da documentação utilizada na estimativa.

8.2. ESTIMATIVA DE TAMANHO COM SFP

Apesar de existirem outros métodos para realizar a estimativa de pontos de função de um projeto, este roteiro recomenda a utilização do método SFP. Vale reiterar que o método SFP é útil no início do processo de desenvolvimento, oferece resultados confiáveis, repetíveis e objetivos; foi projetado para ser ágil, rápido, leve e de fácil utilização.

A estimativa de tamanho com o método SFP fornece uma avaliação muito aproximada do tamanho funcional final de um software, visto que a pontuação funcional das funções de dados e funções transacionais não sofre alteração ao longo do processo de desenvolvimento.

Para realizar a contagem estimada de um software com o método SFP, deve-se realizar todas as etapas do procedimento de contagem exposto no item 5.1 deste roteiro. Ademais, deve-se realizar a leitura de toda a documentação disponível buscando informações relevantes para identificação dos processos elementares e arquivos lógicos.

O método SFP requer apenas a identificação de dois componentes funcionais básicos: Processo Elementar (PE) e Arquivo Lógico (AL); não é necessário diferenciar os tipos de processo elementar (EE, CE, SE) e os tipos de arquivo lógico (ALI, AIE), tampouco a sua complexidade. Assim, após identificação dos processos elementares e arquivos lógicos, é atribuída a pontuação de 7 SFP para os Arquivos Lógicos e 4,6 SFP para os processos elementares identificados.

A estimativa de um projeto com o método SFP considera as funcionalidades que farão parte da aplicação (SFP_INCLUIDO) e as funcionalidades de conversão de dados (SFP_CONVERSÃO) que apoiam o início do uso do software, conforme fórmula a seguir:



$$\text{ESTIMATIVA DE TAMANHO (SFP)} = \text{SFP_INCLUIDO} + \text{SFP_CONVERSÃO}$$

A definição dos termos SFP_INCLUÍDO e SFP_CONVERSÃO pode ser encontrada com mais detalhes no item 6 deste roteiro.

8.2.1. Exemplo de Estimativa de Tamanho com SFP

Considere um projeto ABC em que foram realizadas todas as etapas do procedimento de contagem exposto no item 5.1 deste roteiro para estimar o tamanho do software a ser desenvolvido.

Foram identificados 95 processos elementares e 20 arquivos lógicos. Foi identificado também a necessidade de realizar 5 cargas de dados para popular o novo sistema a ser desenvolvido.

Para os tais, foi atribuída a pontuação funcional determinada pelo método SFP (PE=4,6 SFP / AL=7 SFP). O tamanho estimado do Projeto ABC foi obtido usando a fórmula apresentada no item 8.2, conforme a seguir:



ESTIMATIVA DE TAMANHO (SFP) = SFP_INCLUIDO + SFP_CONVERSÃO

ESTIMATIVA DE TAMANHO (SFP) = [(95 x 4,6) + (20 x 7)] + (5 x 4,6)

ESTIMATIVA DE TAMANHO (SFP) = 437 + 140 + 23

ESTIMATIVA DE TAMANHO (SFP) = 600 SFP

8.3. ESTIMATIVA DE ESFORÇO DE PROJETOS DE SOFTWARE

Uma vez que o tamanho do projeto foi estimado com o método SFP, pode-se estimar o esforço de desenvolvimento do projeto conforme a fórmula a seguir:



ESTIMATIVA DE ESFORÇO (HORAS) = Tamanho (SFP) x Índice de Produtividade (H/SPF)

Em geral, a produtividade considerada para projetos ágeis de TI é de 10 horas por Ponto de Função (10H/PF ou 10H/SFP). Entretanto, caso a organização possua sua própria tabela de produtividade, considerando dados históricos dos projetos já realizados, deve utilizá-la para estabelecer seu índice de produtividade.

8.3.1. Exemplo de Estimativa de Esforço

Para o cálculo do esforço, vamos considerar o projeto ABC descrito no subitem 8.2.1, cuja contagem estimada é de 600 SFP. Aplicando a fórmula para estimativa de esforço e utilizando índice de produtividade de 10 H/SPF temos:



ESTIMATIVA DE ESFORÇO (HORAS) = Tamanho (SPF) x Índice de Produtividade (H/SPF)

ESTIMATIVA DE ESFORÇO (HORAS) = 600 x 10

ESTIMATIVA DE ESFORÇO (HORAS) = 6000 horas



Destaca-se novamente que o método SFP tem a característica de medir os requisitos funcionais do usuário com a mesma precisão do método IFPUG padrão, além de ser totalmente compatível com ele em termos de resultados.

8.3.2. Distribuição de Esforço por fase de projeto

Considerando a diversidade de realidades existentes nos órgãos da administração pública federal, buscou-se utilizar uma documentação de referência cujos processos se adaptassem aos mais diversos ciclos de vida de software das organizações do SISF, e assim apoiar os órgãos na definição de fatores de impacto (FI) para contagem dos projetos de desenvolvimento, manutenção e/ou sustentação de SW; para tanto, serão utilizados os processos técnicos da NBR ISO/IEC 12207 conforme exposto no item 9.1 deste roteiro.

A seguir, apresenta-se a tabela com os processos técnicos da norma ISO/IEC 12207:2017 detalhada no item 9.1 deste roteiro, com a quantidade de atividades relacionadas a cada processo e o percentual de esforço por processo. Ressaltamos que o órgão pode definir outros macroprocessos e subdividi-los para melhor aderência à sua metodologia e aos marcos de entrega.

A distribuição do esforço por processos deve estar documentada na metodologia do órgão (especificada contratualmente) ou formalizada diretamente no contrato.

Tabela 10: Distribuição do Esforço utilizando processos técnicos da norma ISO/IEC 12207:2017

Processos técnicos da ISO/IEC 12207:2017		Quantidade de Atividades do processo	Percentual por processo
P1	Processo de definição das Necessidades e Requisitos das partes interessadas;	6	16 %

P2	Processo de definição de Requisitos de Sistema/Software;	4	11 %
P3	Processo de definição de Arquitetura;	6	16 %
P4	Processo de definição do Projeto/design;	4	11 %
P5	Processo de Análise do Sistema;	3	8 %
P6	Processo de Implementação;	3	8 %
P7	Processo de Integração;	3	8 %
P8	Processo de Verificação;	3	8 %
P9	Processo de Transição;	3	8 %
P10	Processo de Validação;	3	8 %
	Total de Atividades	38	100%

8.4. ESTIMATIVA DE PRAZO DE PROJETOS DE SOFTWARE

Sabe-se que as estimativas de prazo não são lineares com o tamanho do projeto. Ademais, diversos fatores contribuem para que os prazos estimados não sejam cumpridos. Entretanto, não há como planejar ou mesmo acompanhar adequadamente o desenvolvimento de um projeto de software quando não se tem uma ideia estimada de quanto tempo vai durar, quanto tempo será necessário alocar recursos e quanto vai custar.

Para derivar a estimativa de prazo de um projeto, este roteiro recomenda que sejam realizados os passos a seguir:

- **Passo 1:** Definir a composição do time de referência;
- **Passo 2:** Estimar a produtividade mínima esperada para time de referência;
- **Passo 3:** Estimar o prazo do Projeto.

8.4.1. Passo 1: Definir a composição do time de referência

Esse roteiro recomenda que seja prevista a composição mínima do time de referência, incluindo a identificação dos perfis profissionais, senioridade, alocação e quantidades, pois entende que um time ágil qualificado é unidade fundamental para um projeto de desenvolvimento de software que utiliza metodologias ágeis.

Sabe-se que um time ágil deve ser multifuncional e ter todas as competências necessárias para realizar seu trabalho com dependências mínimas; diante disso, para definir a composição de um time ágil a organização deve fazer as verificações a seguir, conforme as características, complexidade e necessidades do projeto:

- Quais os perfis profissionais necessários para o desenvolvimento do projeto?
- Qual a senioridade necessária para cada perfil profissional?
- Qual a quantidade de profissionais de cada perfil?
- Qual o percentual de alocação de cada profissional no projeto?

Essas questões podem ser respondidas observando as composições que deram certo na organização - conforme base histórica do órgão - ou benchmarking com outros órgãos do governo que utilizam metodologias ágeis e regras para composição mínima dos seus times.

Com o fim de responder aquelas questões, suponha que a composição exposta na tabela a seguir seja a composição mínima ideal para o time que vai atuar no desenvolvimento do projeto ABC:

Tabela 11: Exemplo de composição mínima de referência para times de desenvolvimento

Perfis Profissionais	Qtd estimada de profissionais por time	% Alocação	Capacidade máxima de compartilhamento (no de projetos simultâneos)
Scrum Master	1	33%	Até 3 projetos
Desenvolvedor Pleno	1	50%	Até 2 projetos
Desenvolvedor Sênior	1	100%	Não pode ser compartilhado entre projetos
Arquiteto Sênior	1	33%	Até 3 projetos
Analista de Negócios/Requisitos Pleno	1	50%	Até 2 projetos
Analista de Negócios/Requisitos Sênior	1	50%	Até 2 projetos
Administrador de Dados Sênior	1	25%	Até 4 projetos
Analista de Testes/Qualidade Sênior	1	25%	Até 4 projetos

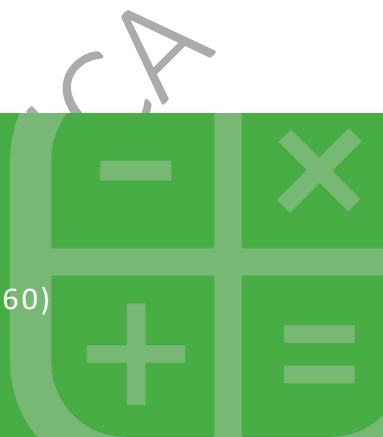
8.4.2. Passo 2: Estimar a produtividade do time de referência

Estimar a produtividade do time de referência também contribui para a determinar a estimativa de prazo do projeto. Considerando que o projeto terá um time de profissionais com experiência e qualificação definidos, é possível estimar a produtividade mínima com base na sua carga horária mensal.

Inicialmente estimamos a capacidade produtiva do time a partir da soma da carga horária mensal de todos os profissionais que vão compor o time, de acordo com a sua taxa de alocação:



$$\text{CARGA HORÁRIA DO TIME (H/MÊS)} = (P1 \times \text{Aloc} \times 160) + (P2 \times \text{Aloc} \times 160) + \dots + (Pn \times \text{Aloc} \times 160)$$



Onde:

- **P1, P2 e PN** corresponde a quantidade de profissionais por perfil, com compõem o time de referência;
- **Aloc** é o percentual de alocação de cada profissional;
- **160** é a carga horária mensal (20 dias x 8h = 160h).

Na sequência, é possível chegar à produtividade mensal esperada para o time utilizando a seguinte fórmula:



$$\text{PRODUTIVIDADE ESPERADA (SFP/MÊS)} = \text{Carga horária do time (h/mês)} / \text{IPe}$$



Onde:

- **IPe** é o índice de produtividade.

Para facilitar o entendimento, vamos aplicar as fórmulas acima e estimar a produtividade do time de referência do projeto ABC, descrito no subitem 8.2.1.

A tabela a seguir apresenta o cálculo da carga horária mensal de cada perfil profissional do time de referência (ver subitem 8.4.1), considerando:

- a quantidade estimada de profissionais por perfil;
- o percentual de alocação; e
- a carga horária mensal de trabalho.

Na última linha da tabela soma-se a carga horária mensal de todos os perfis a serem alocados no projeto e chega-se à carga horária mensal do time:

Tabela 12: Carga horária mensal do time de referência

Perfis Profissionais	Qtd estimada de profissionais por time (A)	% Alocação (B)	Carga Horária mensal do perfil = A*B*160
Scrum Master	1	33%	53 H
Desenvolvedor Pleno	1	50%	80 H
Desenvolvedor Sênior	1	100%	160 H
Arquiteto Sênior	1	33%	53 H
Analista de Negócios/Requisitos Pleno	1	50%	80 H
Analista de Negócios/Requisitos Sênior	1	50%	80 H
Administrador/Projetista de Dados Sênior	1	25%	40 H

Analista de
Testes/Qualidade Sênior

1

25%

40 H

CARGA HORÁRIA MENSAL DO TIME: 586 H

Após obter a carga horária mensal do time, é possível estimar a sua capacidade produtiva mensal, conforme exemplo a seguir:



PRODUTIVIDADE ESPERADA (SFP/MÊS) = Carga horária do time (h/mês) / IPe

PRODUTIVIDADE ESPERADA (SFP/MÊS) = 586 / 10

PRODUTIVIDADE ESPERADA (SFP/MÊS) = 58,6 SFP/MÊS

A produtividade de um time ágil está relacionada à capacidade de entregar funcionalidades de valor para o cliente, em um curto período de tempo.

No exemplo foi utilizado índice de produtividade igual a 10H/SPF, que é a produtividade normalmente considerada para projetos ágeis de TI (10H/PF ou 10H/SFP). Contudo, caso a organização possua sua própria tabela de produtividade, considerando dados históricos dos projetos já realizados, deve utilizar o seu índice de produtividade, conforme as características do projeto que está sendo estimado.

8.4.3. Passo 3: Estimar o prazo do projeto

Após realizar a contagem estimada do projeto com o método SFP, o esforço em horas necessário para seu desenvolvimento e a produtividade mínima esperada para um time de referência, é possível estimar a sua duração.

Aqui apresentam-se 2 fórmulas para a estimativa de prazo:

- Fórmula 1: estimativa de prazo a partir da estimativa de esforço e carga horária mensal do time.
- Fórmula 2: estimativa de prazo a partir do tamanho em SFP e produtividade mínima esperada para o time.

Ambas utilizam conceitos descritos nos itens anteriores:



(Fórmula 1) ESTIMATIVA DE PRAZO (MESES) =
ESFORÇO (H) / CARGA HORÁRIA MENSAL DO TIME (H)

(Fórmula 2) ESTIMATIVA DE PRAZO (MESES) =
TAMANHO (SFP) / PRODUTIVIDADE MÍNIMA PARA O MÊS (SPF/MÊS)

Caso a organização deseje estimar o prazo em Sprints basta dividir a estimativa de tamanho em SFP pela produtividade mínima esperada por sprint:



ESTIMATIVA DE PRAZO (SPRINTS) =
TAMANHO (SFP) / PRODUTIVIDADE MÍNIMA POR SPRINT

Observe que a estimativa de prazo pode mudar, para mais ou para menos, de acordo com a produtividade mínima do time.

Para exemplificar a aplicação das fórmulas de estimativa de prazo, vamos utilizar os dados colhidos para o projeto ABC:

- Estimativa de tamanho: 600 SPF;
- Estimativa de esforço: 6000 Horas;
- Carga horária mensal do Time de referência: 586 horas;
- Produtividade mínima esperada para o time de referência: 58,5 SFP/Mês;
- *Timebox da Sprint*: 2 semanas (10 dias úteis);
- Produtividade mínima esperada para a Sprint: 29,3 SFP/Sprint.



ESTIMATIVA DE PRAZO (MESES) = ESFORÇO (H) / CARGA HORÁRIA MENSAL DO TIME (H)

$$\text{ESTIMATIVA DE PRAZO (MESES)} = 6000 / 586 \cong 10 \text{ MESES}$$

ESTIMATIVA DE PRAZO (MESES) = TAMANHO (SFP) / PRODUTIVIDADE MÍNIMA PARA O MÊS (SFP/MÊS)

$$\text{ESTIMATIVA DE PRAZO (MESES)} = 600 / 58,6 \cong 10 \text{ MESES}$$

ESTIMATIVA DE PRAZO (SPRINTS) = TAMANHO (SFP) / PRODUTIVIDADE MÍNIMA POR SPRINT

$$\text{ESTIMATIVA DE PRAZO (SPRINTS)} = 600 / 29,3 \cong 20 \text{ SPRINTS}$$

8.5. ESTIMATIVA DE CUSTO DE PROJETOS

Esse roteiro preconiza que a estimativa de custo do projeto deve ser calculada a partir do custo da composição mínima do time ágil de referência para o projeto. As orientações para definir a composição mínima do time de referência foram descritas no item 8.4.1 deste roteiro.

A Portaria SGD/MGI nº 750, de 20 de março de 2023 orienta que na modalidade baseada no pagamento por Ponto de Função, o cálculo do patamar mínimo do valor do Ponto de Função deve considerar os seguintes parâmetros:

- A composição mínima da equipe ágil, em termos dos perfis profissionais e suas respectivas taxas de alocação;
- A média dos salários de referência dos perfis que integram a composição mínima da equipe ágil;
- A produtividade mínima considerada para projetos ágeis de TI (em geral, tem-se 10 horas por Ponto de Função);
- A duração máxima da sprint;
- O custo mensal médio estimado do time ágil.

Nessa perspectiva, o custo mensal do time de referência determinará o (1) custo do ponto de função e o (2) custo do projeto, conforme fórmulas a seguir:



$$\text{ESTIMATIVA DE CUSTO DO PF (EcPF)} = E_{cm} / P_e$$

$$\text{ESTIMATIVA DE CUSTO DO PROJETO (EcP)} = \text{Tamanho SPF} \times E_{cPF}$$

Onde:

- **E_{cm}** corresponde a estimativa de custo mensal do Time Ágil de Referência;
- **P_e** corresponde a produtividade esperada para o mês.
- **E_{cPF}** corresponde a estimativa de custo do PF.

No subitem 8.2 foi utilizado o método SFP para calcular a estimativa de tamanho do projeto e essa estimativa foi utilizada para calcular o esforço e prazo do projeto. Note que na fórmula apresentada acima utiliza o termo ESTIMATIVA DE CUSTO DO PF (E_{cPF}). Vale lembrar que SPF é um método de contagem de Pontos de Função. Mesmo utilizando o método SFP para calcular o tamanho do projeto, a métrica continua sendo o PF.

No item seguinte será apresentada uma tabela para exemplificar o cálculo do custo mensal do time de referência e a aplicação das fórmulas para derivar as estimativas de custos do PF e do projeto.

8.5.1. Exemplo de Estimativa de Custo

A tabela a seguir apresenta memória de cálculo para definir o custo do PF, utilizando o custo mensal do time referência do projeto ABC:

Tabela 13: Cálculo da estimativa de custo mensal do time de referência

Identificação do Perfil	Salário (\$)	Custo do Perfil (Cp = S x Fator-K)	Taxa de Alocação (Ta)	Alocação em horas (A = Ta x 160)	Qtd de profissionais por perfil (Q)	Custo por hora (Ch=Cp/160)	Custo mensal do perfil (Cm = A x Q x Ch)
-------------------------	--------------	------------------------------------	-----------------------	----------------------------------	-------------------------------------	----------------------------	--

Scrum Master	R\$11.732,20	R\$22.760,47	33%	52,8	1	R\$142,25	R\$7.510,95
Desenvolvedor Pleno	R\$10.677,45	R\$20.714,25	50%	80	1	R\$129,46	R\$10.357,13
Desenvolvedor Sênior	R\$14.016,77	R\$27.192,53	100%	160	1	R\$169,95	R\$27.192,53
Arquiteto Sênior	R\$18.084,53	R\$35.083,99	33%	52,8	1	R\$219,27	R\$11.577,72
Analista de Negócios/ Requisitos Pleno	R\$8.744,98	R\$16.965,26	50%	80	1	R\$106,03	R\$8.482,63
Analista de Negócios/ Requisitos Sênior	R\$11.227,93	R\$21.782,18	50%	80	1	R\$136,14	R\$10.891,09
Administrador de Dados Sênior	R\$12.115,48	R\$23.504,03	25%	40	1	R\$146,90	R\$5.876,01
Analista de Testes/Qualidade – Sênior	R\$11.081,16	R\$21.497,45	25%	40	1	R\$134,36	R\$5.374,36
CUSTO MENSAL DO TIME							R\$87.262,42

A tabela acima utilizou os valores do mapa de pesquisa salarial (Anexo II - MAPA DE PESQUISA SALARIAL DE REFERÊNCIA PARA SERVIÇOS DE DESENVOLVIMENTO E SUSTENTAÇÃO DE SOFTWARE) e fator-k de 1,94 presentes na Portaria SGD/MGI nº 750, de 20 de março de 2023.

Na última linha da tabela acima somou-se o custo mensal de todos os perfis profissionais para obter o custo mensal do time de referência.

A tabela a seguir apresenta os valores obtidos nas estimativas anteriores (tamanho, esforço e prazo) que vão apoiar nas estimativas de custo do projeto ABC:

Tabela 14: Estimativas do projeto ABC

Estimativa de Tamanho do Projeto em SFP	600 SFP
Índice de Produtividade Mínima	10 H/SFP
Estimativa de Esforço do Projeto	6000 Horas
Total de Horas do Time por mês	586 Horas
Produtividade esperada para o mês	58,6 SFP/mês
Estimativa de Prazo em meses	10 meses
Estimativa de Custo mensal do Time (Ecm*)	
<i>*Ecm corresponde a soma do custo mensal dos profissionais do time, conforme a sua taxa de alocação</i>	
	R\$87.262,42

A estimativa do custo do PF corresponde ao custo mensal do time (Ecm) dividido pela produtividade mínima esperada para o mês (Pe):

$$\text{ESTIMATIVA DE CUSTO DO PF (EcPF)} = \text{Ecm} / \text{Pe}$$

$$\text{ESTIMATIVA DE CUSTO DO PF (EcPF)} = 87.262,42 / 58,6$$

$$\text{ESTIMATIVA DE CUSTO DO PF (EcPF)} = \text{R\$1.489,11}$$

Sabendo o custo estimado do ponto de função, é possível estimar o custo do projeto, que corresponde ao produto da estimativa de tamanho do projeto pela estimativa de custo do ponto de função (EcPF):



ESTIMATIVA DE CUSTO DO PROJETO = Tamanho PF x EcPF

ESTIMATIVA DE CUSTO DO PROJETO = 600 x R\$1.489,11

ESTIMATIVA DE CUSTO DO PROJETO = R\$ 893.466,00

8.6. Estimativa de recursos de TIC

A estimativa de recursos de TIC também deve ser considerada, pois constitui um componente importante para as estimativas de custo dos projetos. Suponha que para o desenvolvimento de um projeto de Software é necessário adquirir um hardware específico. O custo deste recurso afetará o custo do projeto. Sendo assim, esse custo precisa ser considerado e registrado na documentação de estimativas.

Devem ser registradas as seguintes informações associadas aos recursos computacionais críticos:

- Nome do Recurso de TIC [considere exclusivamente hardware: micro, periférico, expansão de memória, área em disco, banda de rede etc.]
- Descrição: [definição das características do recurso necessárias ao atendimento ao projeto]
- Responsável pela Disponibilização: [defina quem é o responsável pela disponibilização do recurso para o projeto]
- Data Limite: [informe a data limite para disponibilização do recurso]
- Parâmetros: [características do recurso: quantidade, perfil, configuração etc.]
- Tipo do Recurso: [D: recurso para ambiente de Desenvolvimento; P: recurso para ambiente de Produção; H: recurso para ambiente de Homologação]
- Custo (Opcional): [Custo do recurso de TIC. Não considerar custos de processamento ou custos operacionais de produção]

Caso o projeto a ser desenvolvido não possua nenhum recurso de TIC crítico, isto deve ser registrado no documento de estimativas.

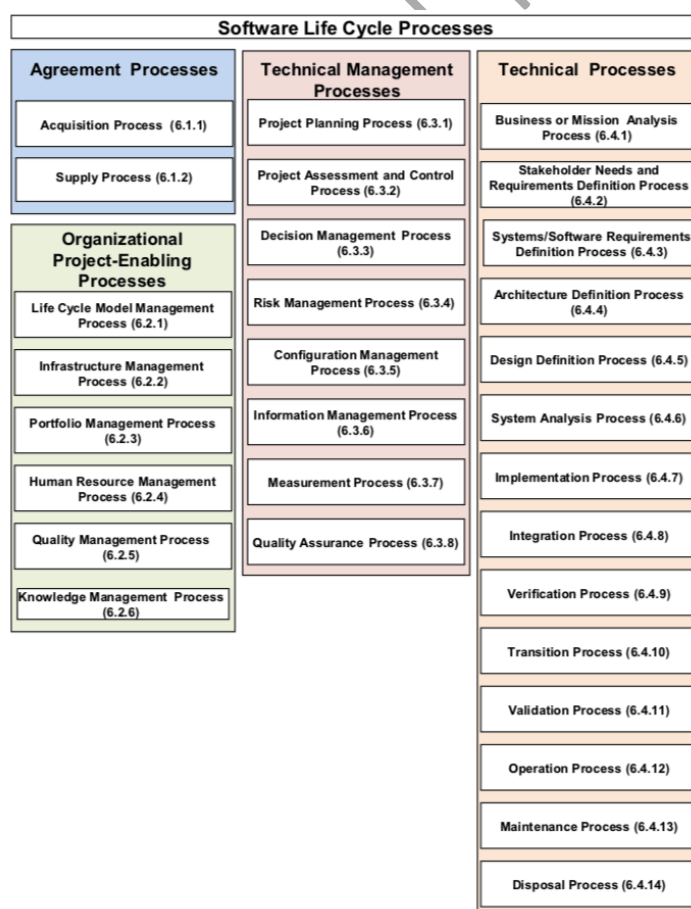
9. DIRETRIZES PARA DEFINIÇÃO DE FATORES DE IMPACTO

Considerando a diversidade de realidades existentes nos órgãos da administração pública federal, buscou-se utilizar uma documentação de referência cujos processos se adaptassem aos mais diversos ciclos de vida de software das organizações do SISP.

Nesse sentido, foram utilizados os processos técnicos da NBR ISO/IEC 12207 para a definição e fundamentação dos fatores de impacto utilizados nas fórmulas de contagem dos projetos descritos no item 6 (CÁLCULO DE PONTOS DE FUNÇÃO PARA O SISP) e do percentual de esforço para cada fase ou macroprocessos do projeto.

A ABNT NBR ISO/IEC 12207 é baseada nos princípios gerais de engenharia de sistemas e engenharia de software e estabelece uma estrutura de referência para os processos do ciclo de vida do software. Nela, as atividades que podem ser executadas durante o ciclo de vida de um sistema de software são agrupadas em quatro grupos de processos: (1) Processos contratuais, (2) Processos de Habilitação de Projetos Organizacionais, (2) Processos Técnicos de Gestão e (3) Processos Técnicos, conforme imagem a seguir:

Figura 4 – Processos do ciclo de Vida do Software da NBR ISO/IEC 12207 (Fonte: NBR ISO/IEC 12207)



O escopo deste roteiro abrange apenas os processos técnicos (*technical processes*) da ISO/IEC 12207, que transformam as necessidades das partes interessadas em um produto ou serviço. Conforme essa norma, os processos técnicos são usados para definir os requisitos de um sistema de software, para transformar os requisitos em um produto eficaz, para permitir a reprodução consistente do produto quando necessário, para usar o produto para fornecer os serviços necessários, para sustentar a prestação desses serviços e para descartar o produto quando ele for retirado do serviço.

Cabe ressaltar que esse roteiro não tem o propósito de prescrever um modelo de ciclo de vida de software específico ou metodologia de desenvolvimento a serem utilizados pelos órgãos do SISP. Além disso, não é objetivo deste roteiro prescrever uma sequência específica de processos dentro do modelo de ciclo de vida; em vez disso, ele apresenta um conjunto de processos de referência que foram utilizados como memória de cálculo na definição dos fatores de impacto relacionados a contagem de pontos de função.

A seguir, os Processos TÉCNICOS de referência da norma ISO/IEC 12207:

- Processo de Análise de Negócio ou Missão;
- Processo de definição das necessidades e requisitos das partes interessadas;
- Processo de definição de requisitos de sistema/software;
- Processo de definição de arquitetura;
- Processo de definição do projeto/design;
- Processo de análise do Sistema;
- Processo de implementação;
- Processo de integração;
- Processo de verificação;
- Processo de transição;
- Processo de validação;
- Processo operacional;
- Processo de manutenção; e
- Processo de descarte.

Para o cálculo dos percentuais de ajuste que serão aplicados nas fórmulas de contagem de PF, foram considerados apenas os processos relacionados diretamente ao desenvolvimento em si, do SW. Não foram considerados os processos de Análise de Negócio ou Missão, Operação, Manutenção e Descarte.

No que diz respeito a conformidade com a ISO/IEC 12207, este roteiro se qualifica com a aplicação de conformidade personalizada, visto que utilizará apenas alguns dos seus processos técnicos como base e referência na definição de fatores de impacto a serem aplicados nas fórmulas de contagem de pontos de função do SISP.

9.1.DEFINIÇÃO DE FATOR DE IMPACTO

Para a definição dos fatores de impacto (FI) utilizados nas fórmulas de contagens dos projetos, descritos no item 6 (CÁLCULO DE PONTOS DE FUNÇÃO PARA O SISP) deste roteiro, foram considerados dez processos técnicos da ABNT NBR ISO/IEC 12207 e suas atividades de referência.

Ressalta-se novamente esse roteiro não tem o objetivo de prescrever um modelo de ciclo de vida específico para os órgãos do SISP, tampouco uma sequência específica de processos dentro do modelo de ciclo de vida. Nesse sentido, recomenda-se que os processos praticados no ciclo de vida dos softwares da organização observem a metodologia ou processo de desenvolvimento de software do órgão.

A tabela a seguir apresenta a descrição de cada processo técnico utilizado, em termos dos seguintes atributos:

- a) **Título do Processo:** transmite o escopo do processo como um todo.
- b) **Finalidade do Processo:** descreve também o objetivo de realizar o processo.
- d) **Atividades:** são um conjunto de tarefas coesas de um processo.

Tabela 15: Processos técnicos da norma ISO/IEC 12207:2017 utilizados para definição dos percentuais de impacto

Processos Técnicos (ISO 12207:2017)		Finalidade	Atividades	
P1	Processo de definição das necessidades e	O objetivo do processo de definição das necessidades e requisitos das partes interessadas é definir os	P1.1	Preparar-se para a definição das necessidades e requisitos das partes interessadas

	requisitos das partes interessadas	requisitos das partes interessadas para um sistema que possa fornecer os recursos necessários aos usuários e outras partes interessadas em um ambiente definido.	P1.2	Definir as necessidades das partes interessadas.
			P1.3	Desenvolver o conceito operacional e outros conceitos de ciclo de vida
			P1.4	Transformar as necessidades das partes interessadas em requisitos das partes interessadas
			P1.5	Analisar os requisitos das partes interessadas
			P1.6	Gerenciar a definição de necessidades e requisitos das partes interessadas
P2	Processo de definição de requisitos de sistema/software	O objetivo é transformar a visão das capacidades desejadas em uma visão técnica de uma solução que atenda às necessidades operacionais das partes interessadas.	P2.1	Preparar-se para a definição dos requisitos do sistema/software
			P2.2	Definir requisitos de sistema/software
			P2.3	Analisar requisitos de sistema/software
			P2.4	Gerenciar requisitos de sistema/software
P3	Processo de definição de arquitetura	O objetivo é gerar alternativas de arquitetura de sistema, selecionar uma ou mais alternativas que enquadrem as preocupações das partes interessadas e atender aos requisitos do sistema, e expressar isso em um conjunto de visões consistentes.	P3.1	Preparar-se para a definição da arquitetura.
			P3.2	Desenvolver pontos de vista de arquitetura
			P3.3	Desenvolver modelos e visões de arquiteturas candidatas.
			P3.4	Relacionar a arquitetura com o design
			P3.5	Avaliar arquiteturas candidatas
			P3.6	Gerenciar a arquitetura selecionada
P4	Processo de definição do projeto	O objetivo é fornecer dados e informações detalhadas suficientes sobre o sistema e seus elementos para permitir a implementação	P4.1	Preparar-se para a definição do projeto do sistema de software
			P4.2	Estabelecer projetos relacionados a cada elemento do sistema de software

		consistente com as entidades arquitetônicas definidas nos modelos e visualizações da arquitetura do sistema.	P4.3	Avaliar alternativas para obtenção de elementos de sistemas de software
			P4.4	Gerenciar o projeto
P5	Processo de Análise do Sistema	O objetivo do processo de Análise do Sistema é fornecer uma base rigorosa de dados e informações para a compreensão técnica para auxiliar na tomada de decisões ao longo do ciclo de vida.	P5.1	Definir a estratégia de análise do sistema e preparar-se para a análise do sistema
			P5.2	Realizar a análise do sistema
			P5.3	Gerenciar a análise do sistema
P6	Processo de implementação	Este processo transforma requisitos, arquitetura e design, incluindo interfaces, em ações que criam um elemento do sistema de acordo com as práticas da tecnologia de implementação selecionada, usando especialidades ou disciplinas técnicas apropriadas.	P6.1	Preparar-se para a implementação
			P6.2	Realizar a implementação
			P6.3	Gerenciar os resultados da implementação
P7	Processo de integração	O objetivo do processo de integração é sintetizar um conjunto de elementos do sistema em um sistema realizado (produto ou serviço) que satisfaça os requisitos, a arquitetura e o design do sistema/software.	P7.1	Preparar-se para a integração
			P7.2	Realizar a integração
			P7.3	Gerenciar resultados de integração
P8	Processo de verificação	O objetivo do processo de verificação é fornecer evidências objetivas de que um sistema ou elemento do sistema atende aos seus requisitos e características especificados.	P8.1	Preparar-se para verificação
			P8.2	Realizar a verificação
			P8.3	Gerenciar os resultados da verificação
P9	Processo de transição	O objetivo do processo de Transição é estabelecer a capacidade de um sistema fornecer serviços especificados pelos requisitos das	P9.1	Preparar-se para a transição do sistema de software
			P9.2	Executar a transição

		partes interessadas no ambiente operacional.	P9.3	Gerenciar os resultados da transição
P10	Processo de validação	O objetivo do processo de validação é fornecer evidências objetivas de que o sistema, quando em uso, cumpre seus objetivos de negócio ou missão e os requisitos das partes interessadas, alcançando o uso pretendido no ambiente operacional pretendido.	P10.1	Preparar-se para validação
			P10.2	Realizar a validação
			P10.3	Gerenciar resultados de validação

Cumprir informar que as tarefas relacionadas a cada atividade dos processos técnicos acima relacionados, bem como os resultados esperados podem ser encontradas na documentação da norma ISO/IEC 12207:2017.

A seguir, apresenta-se a memória de cálculo para a definição dos fatores de impacto utilizados no item 6 deste roteiro:

- **Na tabela 16** apresenta-se a memória de cálculo de Ponto de Função para o SISP para os projetos descritos nos subitens 6.1 a 6.9.3 deste roteiro;
- **Na tabela 17** apresenta-se a memória de cálculo de Ponto de Função para o SISP para os projetos descritos nos subitens 6.10 a 6.17 deste roteiro.

Tabela 16: Memória de Cálculo de Ponto de Função para o SISP – Parte 1 (subitens 6.1 a 6.9.3)

CÁLCULO de PF para o SISP (item 6)	Projeto de Desenvolvimento	Projeto de Melhoria / Funcionalidade desenvolvida	Projeto de Melhoria / Funcionalidade <u>não desenvolvida</u> COM redocumentação	PF Excluído	Projeto de Migração de dados	Manutenção Corretiva	Mudança de Plataforma - Linguagem (SEM REQUISITOS)	Mudança de Plataforma - Linguagem (COM REQUISITOS)	Mudança de Plataforma - Banco de Dados (SEM REQUISITOS)	Mudança de Plataforma - Banco de Dados (COM REQUISITOS)	Mudança de Plataforma - Banco de Dados (RELACIONAL P/ RELACIONAL)	Atualização de Versão - Linguagem	Atualização de Versão - Browser	Atualização de Versão - Banco de Dados
--	----------------------------	---	--	-------------	------------------------------	----------------------	--	--	---	---	---	-----------------------------------	---------------------------------	--

Processo Técnicos e Atividades (ISO 12207:2017)		6.1	6.3	6.3		6.6	6.7	6.8.1	6.8.1	6.8.2	6.8.2	6.8.2	6.9.1	6.9.2	6.9.3
P1	1.1	1													
	1.2	1													
	1.3	1													
	1.4	1													
	1.5	1													
	1.6	1													
P2	2.1	1						1		1					
	2.2	1	1	1			1	1		1					
	2.3	1	1	1	1		1	1		1					
	2.4	1	1	1	1		1	1		1					
P3	3.1	1						1	1	1	1				
	3.2	1						1	1	1	1				
	3.3	1						1	1	1	1				
	3.4	1						1	1	1	1				
	3.5	1						1	1	1	1				
	3.6	1						1	1	1	1				
P4	4.1	1						1	1	1	1				
	4.2	1						1	1	1	1				
	4.3	1						1	1	1	1				
	4.4	1						1	1	1	1				
P5	5.1	1		1		1		1	1	1	1				
	5.2	1		1		1		1	1	1	1				
	5.3	1		1		1		1	1	1	1				
P6	6.1	1	1	1		1	1	1	1	1	1	1	1	1	1
	6.2	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	6.3	1	1	1	1	1	1	1	1	1	1	1	1	1	1
P7	7.1	1	1	1		1	1	1	1	1	1	1	1	1	1
	7.2	1	1	1		1	1	1	1	1	1	1	1	1	1
	7.3	1	1	1		1	1	1	1	1	1	1	1	1	1
P8	8.1	1	1	1		1	1	1	1	1	1	1	1	1	1
	8.2	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	8.3	1	1	1	1	1	1	1	1	1	1	1	1	1	1
P9	9.1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	9.2	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	9.3	1	1	1	1	1	1	1	1	1	1	1	1	1	1

P10	10.1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	10.2	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	10.3	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Qtd Atividades		38	18	21	12	18	18	32	28	32	28	15	15	15	15
FI		1	0,47	0,55	0,32	0,47	0,47	0,84	0,73	0,84	0,73	0,39	0,39	0,39	0,39

Tabela 17: Memória de Cálculo de Ponto de Função para o SISP – Parte 2 (subitens 6.10 a 6.17)

CÁLCULO de PF para o SISP (item 6)		Manutenção em Interface	Adaptação sem alteração de requisitos funcionais	Apuração Especial - Base de Dados (Atualização de dados sem consulta prévia)	Apuração Especial - Base de Dados (Consulta prévia sem atualização)	Apuração Especial - Base de Dados (Atualização com consulta prévia)	Apuração especial – Geração de Relatórios	Apuração especial – Alteração	Apuração especial – Reexecução	Desenvolvimento, Manutenção e Publicação de Páginas Estáticas	Manutenção de Documentação de Sistemas Legados	Verificação de Erros	Pontos de Função de Teste	Componente Reusável
Processo Técnicos e Atividades (ISO 12207:2017)		6.10	6.11	6.12.1a	6.12.1b	6.12.1c	6.12.2	6.12.3	6.12.4	6.13	6.14	6.15	6.16	6.17
P1	1.1													
	1.2													
	1.3													
	1.4													
	1.5													
	1.6													
P2	2.1										1			
	2.2		1	1	1		1				1			
	2.3		1	1	1		1	1			1			
	2.4		1	1	1		1	1			1			
P3	3.1													
	3.2													
	3.3													

	3.4													
	3.5													
	3.6													
P4	4.1													
	4.2													
	4.3													
	4.4													
P5	5.1										1	1		1
	5.2										1	1		1
	5.3										1	1		1
P6	6.1	1	1	1	1		1							1
	6.2	1	1	1	1	1	1	1		1				1
	6.3	1	1	1	1	1	1	1		1				1
P7	7.1	1	1							1				1
	7.2	1	1							1				1
	7.3	1	1							1				1
P8	8.1	1	1	1	1	1	1					1	1	1
	8.2	1	1	1	1	1	1	1	1	1		1	1	1
	8.3	1	1	1	1	1	1	1	1	1		1	1	1
P9	9.1	1	1	1	1	1	1			1				1
	9.2	1	1	1	1	1	1	1	1	1				1
	9.3	1	1	1	1	1	1	1	1	1				1
P10	10.1	1	1	1	1	1	1					1	1	1
	10.2	1	1	1	1	1	1	1	1	1		1	1	1
	10.3	1	1	1	1	1	1	1	1	1		1	1	1
Qtd Atividades		15	18	15	15	11	15	10	6	12	7	9	6	18
FI		0,39	0,47	0,39	0,39	0,28	0,39	0,26	0,15	0,31	0,18	0,23	0,15	0,47

Nas duas últimas linhas das tabelas acima temos:

- **Qtd Atividades:** corresponde a estimativa da quantidade (soma) de atividades a serem executadas em cada tipo de projeto abordado no **item 6** deste roteiro;
- **FI:** corresponde ao cálculo do Fator de impacto (FI) para cada tipo de projeto abordado no **item 6** deste roteiro. Para o cálculo do fator de impacto (FI) de cada projeto utilizou-se a seguinte fórmula:



$$FI = \text{Qtd. Atividades} / 38$$



Onde:

- **Qtd. Atividades:** corresponde a quantidade (soma) de atividades estimadas para cada tipo de projeto abordado no **item 6** deste roteiro (penúltima linha das tabelas 16 e 17);
- **38:** é a quantidade total de atividades dos 10 Processos Técnicos da ISO 12207:2017, considerados para o cálculo do FI.

VERSÃO CONSULTA PÚBLICA

10. CONTAGEM E ACOMPANHAMENTO DE PROJETOS ÁGEIS

Este item descreve orientações sobre a medição da contagem de PF de projetos de desenvolvimento e manutenção de *software* que utilizam metodologias ágeis, utilizando o método *Simple Function Point* (SFP), com o fim de subsidiar a gestão, planejamento, medição e contratação desses serviços na Administração Pública Federal (APF).

Uma das principais dificuldades e desafios na adoção de métodos ágeis em contratação de desenvolvimento de *software* é definir um modelo de remuneração que seja equilibrado, remunerando o produto de *software* efetivamente entregue conforme critérios de qualidade previamente estabelecidos.

Devido às características inerentes ao processo ágil, entende-se que os refinamentos e as mudanças em funcionalidades são mais constantes e recorrentes nesse cenário de desenvolvimento de *software*, entretanto o processo ágil de desenvolvimento de *software* em contratações não deve comprometer os princípios de economicidade e efetividade dos resultados previstos e entregues com a garantia da exequibilidade do projeto.

A seguir, destacam-se duas premissas - citadas no item 5.1.4 da Portaria SGD/MGI nº 750, de 20 de março de 2023- que devem ser observadas na construção do termo de referência, independentemente da modalidade de remuneração adotada em uma contratação de serviços de desenvolvimento, manutenção e sustentação de software:

5.1.4. São premissas que devem ser observadas na construção do Termo de Referência, independentemente da modalidade adotada:

...

f) utilização, preferencialmente, de metodologia ágil para a prestação dos serviços;

j) uso preferencial de métricas de software orientadas a entregas de produtos de software;

...

As metodologias ágeis, indicadas como preferenciais para o desenvolvimento de software na Portaria SGD/MGI nº 750, possuem um manifesto público que indica 12 princípios recomendados em sua adoção. A seguir, destacam-se o primeiro, o terceiro e o sétimo:

1 - Nossa maior prioridade é satisfazer o cliente através da entrega contínua e adiantada de software com valor agregado.

...

3 - Entregar frequentemente software funcionando, de poucas semanas a poucos meses, com preferência à menor escala de tempo.

...

7 - Software funcionando é a medida primária de progresso.

...

É importante destacar aqui que entrega de valor não se confunde simplesmente com a prestação de serviços. Entrega de valor corresponde ao atendimento das expectativas e necessidades dos usuários dos serviços públicos prestados pela organização, os quais são apoiados por softwares que devem estar em funcionamento com qualidade, segurança e usabilidade adequadas às suas finalidades.

Os subitens 12.1, 12.3 e 12.4 da Portaria SGD/MGI nº 750 informam ainda que devem ser definidas métricas objetivas nas contratações de serviços de desenvolvimento, manutenção e sustentação de software, independentemente da modalidade de remuneração adotada:

12.1. ... *devem ser definidas métricas objetivas que permitam a gestão contratual, a mensuração e a devida remuneração dos serviços e produtos efetivamente entregues pela empresa contratada no contexto do processo de desenvolvimento de software adotado pelo órgão ou entidade.*

...

12.3. *Independente da modalidade de contratação, deve-se aferir a entrega de produtos por meio de métricas de software, mantendo-se uma base histórica, a exemplo de:*

- a) Pontos de Função (FPUG, NESMA, COSMIC, Simple Function Point - SFP);*
- b) Linhas de código implementadas;*
- c) Pontos de história (Story Point);*

12.4. *A métrica de software deve estar prevista no processo de desenvolvimento de software da organização. Deve-se descrever no instrumento convocatório ou no processo de software da organização as regras de uso, a forma de mensuração, o mecanismo de cálculo, o escopo de aplicação e eventuais recursos ou procedimentos padronizados para realização das medições.*

Nesse sentido, e com o fim de trazer agilidade conjuntamente na mensuração do *software*, os subitens a seguir apresentarão orientações e exemplos de como utilizar o método SFP na medição funcional de um projeto de desenvolvimento e manutenção de *software* que utiliza metodologias ágeis.



A Portaria SGD/MGI nº 750 orienta que **independente da modalidade de contratação** (remuneração por ponto de função complementar por HST, alocação de profissionais de TI ou por Sprint), **a organização deve aferir a entrega de produtos por meio de métricas de software, mantendo-se uma base histórica.**

Este roteiro recomenda o uso do método SFP por ser ágil, de fácil utilização, previsível, independe da metodologia e tecnologia adotada, oferece resultados confiáveis, repetíveis e objetivos.

10.1. ORIENTAÇÕES GERAIS

O desenvolvimento de *software* utilizando métodos ágeis deve respeitar uma abordagem específica que considere as características dos métodos ágeis, tanto no desenvolvimento quanto na gestão de projetos.

Nesse sentido, o processo de desenvolvimento de *software* deve ser segmentado em iterações curtas, entregas frequentes e projetos com escopos delimitados, referenciando-o no instrumento convocatório.

Essas características podem requerer adaptações para o contexto de contratações de *software*. A Portaria SGD/MGI nº 750, de 20 de março de 2023, apresenta orientações quanto a adoção de metodologias ágeis, dentre as quais destacamos o subitem 4.5.10:

4.5.10. Para cada projeto, devem ser definidos parâmetros para a execução das sprints, tais como:

- a) configuração mínima do time que irá executar o conjunto de sprints, indicando perfis profissionais mínimos e nível de compartilhamento aceitável para determinados perfis, conforme exemplo constante do Anexo IV;*
- b) duração máxima da sprint;*
- c) meta de velocidade da sprint, como a quantidade de histórias de usuário e pontos de função;*
- d) meta de escopo planejado x realizado, que indica o percentual realizado a cada sprint em comparação ao escopo planejado; e*
- e) meta de itens de backlog planejados x não planejados, que mapeia se o esforço, a cada sprint, está sendo gasto com novas funcionalidades planejadas ou com refatorações de código, dívidas técnicas e correções de falhas.*

Alinhados à Portaria SGD/MGI nº 750 de 2023, destacam-se pontos que devem ser observados em projetos de desenvolvimento de *software* que utilizam metodologias ágeis:

- O ciclo da *sprint* (iteração) deve ser de 2 até 4 semanas;
- O projeto de desenvolvimento ou manutenção deve ser dividido em *releases*;
- O ciclo da *release* não deve ser igual ao ciclo da *sprint*, ou seja, uma *release* formada por apenas uma *sprint* não permite a adoção das orientações trazidas neste roteiro;
- O ciclo da *release* deve, sempre, promover o aumento do percentual de completude do sistema (entrega de valor agregado ao negócio);
- Deve-se realizar a contagem estimada do projeto a fim de estimar o tamanho final da *release* e viabilizar o planejamento do projeto e a geração das ordens de serviço de desenvolvimento ou manutenção de *software*;
- O pagamento deve estar vinculado à entrega de produto de *software* com qualidade;
- O pagamento deve estar atrelado a uma ordem de serviço.

10.2. MUDANÇAS EM FUNCIONALIDADES

As mudanças em funcionalidades podem ocorrer dentro da *release* ou em *releases* diferentes, conforme detalhado a seguir:

- **Dentro de uma *release*:** as mudanças em funcionalidades desenvolvidas previamente na mesma *release* não são contadas e remuneradas durante o projeto, pois são absorvidas pela contratada como parte do processo de desenvolvimento ágil.
- **Em *releases* diferentes:** sugere-se remunerar conforme os itens de manutenção abordados neste roteiro, tal como, a manutenção evolutiva aplicando-se o fator de impacto sobre o tamanho da funcionalidade impactada, conforme sugerido no item Projeto de Melhoria deste roteiro (item 6.3).

Conforme definição apresentada no item 3 deste roteiro, refinamentos são quaisquer mudanças ocorridas sobre uma função transacional ou de dados já previamente trabalhadas na *release* corrente (seja por meio de uma inclusão, alteração ou exclusão), provocadas pelo aprofundamento, detalhamento e complementação de requisitos durante o processo de desenvolvimento.

As mudanças em funcionalidades já desenvolvidas dentro da mesma *release* devem ser registradas e atendidas pelo contrato como **SFP_REFINADO**, mas sem remuneração adicional ao total de pontos de função da contagem final da *release*, pois se entende que são relativas à evolução de requisitos do processo de desenvolvimento adotado no projeto.

Portanto, na contagem final da *release* não deve haver nem acréscimo de ponto de função nem de qualquer outra natureza:

- A contagem de encerramento da *release* deve conter todas as funcionalidades que foram entregues;
- Mudanças em funcionalidades ocorridas dentro da mesma *release* não são contadas, mas devem ser registradas com o fim de compor um histórico sobre os refinamentos e esforço empregado em seu atendimento.

Complementarmente, vale destacar que é fundamental que o instrumento convocatório de licitação especifique o máximo de fatores, características e aspectos relevantes do projeto que podem influenciar no volume de mudanças em funcionalidades em um projeto de desenvolvimento com métodos ágeis para que as empresas candidatas ao certame avaliem adequadamente as possibilidades de atendimento do contrato, fornecendo profissionais qualificados e preço de ponto de função exequível para o contrato.

É importante destacar também a necessidade de o órgão contratante avaliar e controlar a sua gestão de riscos pela adoção de um contrato de desenvolvimento de *software* com métodos ágeis. O risco poderá se mostrar inversamente proporcional ao detalhamento dos fatores, características e aspectos do projeto expostos no edital de contratação que possam interferir no desenvolvimento e no sucesso do projeto.

Este roteiro recomenda a utilização do método SFP para a contagem de PF de projetos de desenvolvimento e manutenção que utilizam metodologias ágeis:

- A contagem de encerramento da *release* deve conter todas as funcionalidades que foram entregues;
- Mudanças em funcionalidades ocorridas dentro da mesma *release* não são contadas, mas devem ser registradas com o fim de compor um histórico sobre os refinamentos e esforço empregado em seu atendimento.

10.2.1. Fatores que influenciam o número de mudanças

A flexibilidade no desenvolvimento de projetos é necessária dentro da metodologia ágil. Destaca-se aqui um dos princípios do manifesto ágil:

“Aceitar mudanças de requisitos, mesmo no fim do desenvolvimento. Processos ágeis se adequam a mudanças, para que o cliente possa tirar vantagens competitivas.”

Nesse sentido, alguns fatores devem ser considerados e avaliados para a estimativa do volume de mudanças em funcionalidades de um projeto de desenvolvimento com métodos ágeis:

- Maturidade dos requisitos do projeto;

- Conhecimento do negócio pelo *product owner*;
- Disponibilidade e experiência com métodos ágeis da área de negócio (*product owner*);
- Maturidade do processo ágil implantado no órgão (nível de aderência às práticas ágeis);
- Nível de experiência com métodos ágeis da equipe da contratante;
- Nível de experiência requerido para a equipe de desenvolvimento da contratada;
- Tamanho da *sprint* e da *release*;
- Volume de mudanças em funcionalidades de projetos similares já executados.

10.3. REGISTRO E CONTROLE

As contagens de ponto de função devem estar devidamente registradas, a fim de possibilitar o controle de *baselines* de contagens por sistema e de fronteiras de aplicações, com vistas a mitigar o risco de contagem duplicada. Este roteiro recomenda o uso de ferramentas especializadas para a manutenção e atualização da baseline.

Deve-se estabelecer de forma clara as fronteiras das aplicações, visto que o posicionamento incorreto da fronteira pode alterar a perspectiva da medição de uma visão lógica (visão funcional) para uma visão física. Vale destacar que uma fronteira de aplicação não pode ser subdividida por contextos gerenciais de desenvolvimento ou baseada em diferenças de plataformas ou tecnologias.

Para efeito de gestão das mudanças e geração de indicadores, recomenda-se que as mudanças em funcionalidades sejam registradas em planilha separada da contagem do projeto de desenvolvimento. Nessa planilha de mudanças devem ser registradas todas as funcionalidades incluídas, alteradas e excluídas.

10.3.1. Exemplo de aplicação - Contagem Estimada da *Release N*

Para exemplificar a aplicação da contagem com o método SFP em projetos de desenvolvimento e manutenção de SW que utilizam métodos ágeis, suponha o planejamento de 3 iterações (*Sprint N1*, *Sprint N2* e *Sprint N3*) da *Release N*.

Na contagem da *Release N* com o método SFP, apresentada na tabela a seguir, tem-se:

- Quatro funções a serem desenvolvidas (incluídas):

- Beneficiário (AL);
 - Incluir Beneficiário (PE);
 - Alterar Beneficiário (PE);
 - Emitir Relatório de Beneficiários (PE).
- Três funções prontas em releases anteriores para serem alteradas:
 - Benefício (AL);
 - Incluir Benefício (PE);
 - Alterar Benefício (PE).

Como este exemplo utiliza o método de contagem SFP, vale destacar que:

- A medição em SFP não exige qualquer detalhe adicional à própria identificação da funcionalidade. O método identifica apenas 2 componentes funcionais básicos: processos elementares (PE) e arquivos lógicos (AL);
- Diferentemente no método de contagem de PF tradicional, o método SFP não diferencia os tipos de processo elementar (EE, CE ou SE) e tipos de arquivo lógico (AIE ou ALI), tampouco a sua complexidade;
- Na tabela exemplificativa não há uma coluna para COMPLEXIDADE, já que os processos elementares e os arquivos lógicos têm contribuição funcional padronizada no método SFP;
- As funções transacionais (PE) incluídas apresentam contribuição funcional de 4,6 SFP;
- A função de dados (AL) incluída apresenta contribuição funcional de 7 SFP;
- Nas 3 *sprints* foram realizadas alterações em funcionalidades que já haviam sido entregues em *release* anterior (RN -1) pela empresa contratada. Dessa forma, as alterações foram caracterizadas como projeto de melhoria, conforme item 6.3 deste roteiro;
- O tamanho estimado do *backlog* das *sprints* que fazem parte da *Release* N é de 28,41 SFP, que corresponde a soma da contribuição funcional de todas as funcionalidades (transacionais e de dados) a serem desenvolvidas e alteradas na *Release* N.

Tabela 18: Planejamento do *Backlog* das *Sprints* (N1, N2 e N3) da *Release* N

<i>Release N</i>	Nome da Função	Categoria (Inc, Alt, Exc, Refin, etc)	Tipo (PE, AL)	SFP	Observação
<i>Sprint N1</i>	Incluir Beneficiário	Inc	PE	4,6	
	Beneficiário	Inc	AL	7	
	Benefício	Alt	AL	3,29	Alteração caracterizada como Projeto de Melhoria (AL “Benefício” desenvolvido e pronto na <i>Release N-1</i>). Aplicado FI de 47%, conforme item 6.3 ($7 \text{ SFP} * 0,47 = 3,29 \text{ SFP}$)
	Alterar Beneficiário	Inc	PE	4,6	
<i>Sprint N2</i>	Incluir Benefício	Alt	PE	2,16	Alteração caracterizada como Projeto de Melhoria (PE desenvolvido e pronto na <i>Release N-1</i>). Aplicado FI de 47%, conforme item 6.3 ($4,6 \text{ SFP} * 0,47 = 2,16 \text{ SFP}$)
	Emitir Relatório de Beneficiários	Inc	PE	4,6	
<i>Sprint N3</i>	Alterar Benefício	Alt	PE	2,16	Alteração caracterizada como Projeto de Melhoria (PE desenvolvido e pronto na <i>Release N-1</i>). Aplicado FI de 47%, conforme item 6.3 ($4,6 \text{ SFP} * 0,47 = 2,16 \text{ SFP}$)
Total SFP da Release				28,41 SFP	

10.3.2. Exemplo de aplicação - Contagem Final da *Release*

Apresenta-se na tabela a seguir a contagem final da *Release* N. Para o método SFP não há necessidade de uma contagem detalhada, visto que:

- Não é necessária a identificação de DERs, ALRs ou RLRs, nem a identificação de "intenção primária";
- São identificados apenas dois componentes funcionais básicos, com contribuição funcional padronizada: Processo Elementar (PE = 4,6 SFP) e Arquivo Lógico (AL = 7 SFP);
- Não há necessidade de contagem detalhada dos DER, ALR;
- Não há alteração da complexidade funções transacionais e de dados.

A PREVISIBILIDADE é um ponto muito importante a ser destacado quando se utiliza a contagem com o método SFP. Como não há diferença de complexidade entre as funcionalidades, a contribuição funcional de cada processo elementar ou arquivo lógico não vai ser alterado ao longo do processo de desenvolvimento.

Na tabela exemplificativa a seguir vale destacar que

- O campo "Categoria" mostra a opção "Refin" (Refinamento) para representar as mudanças em funcionalidades desenvolvidas na mesma *release*;
- As funções transacionais refinadas (Categoria "Refin") foram absorvidas pela contratada e, portanto, não houve remuneração adicional ao total de pontos de função da *Release* N (células destacadas em amarelo);
- Apesar de não ter remuneração para os refinamentos das funcionalidades já desenvolvidas na *release*, o registro é importante para controle da equipe do projeto, da gestão de mudanças e da gestão do projeto;
- As alterações em funcionalidades já desenvolvidas em *releases* anteriores foram caracterizadas como projeto de melhoria, conforme item 6.3 deste roteiro, atendendo ao que já havia sido previsto no planejamento da *release*.

O objetivo principal desse exemplo é destacar a necessidade de REGISTRAR as funcionalidades incluídas, alteradas, excluídas e refinamentos (mudanças em funcionalidades desenvolvidas na mesma *release*) durante a *release*, independente da identificação da iteração (*sprint*) onde elas ocorreram. Nesse sentido, é facultativo o registro das contagens por *sprint* desde que a contagem da *Release* registre as novas funcionalidades desenvolvidas, bem como, as mudanças em funcionalidades.

Tabela 19: Contagem SFP da *Release* N

Release N	Nome da Função	Categoria (Inc, Alt, Exc, Refin, etc)	Tipo (PE, AL)	SFP	Observação
-----------	----------------	---	------------------	-----	------------

Sprint N1	Incluir Beneficiário	Inc	PE	4,6	Alteração caracterizada como Projeto de Melhoria (AL “Benefício” desenvolvido e pronto na Release N-1). Aplicado FI de 47%, conforme item 6.3 (7 SFP *0,47=3,29 SFP)
	Beneficiário	Inc	AL	7	
	Benefício	Alt	AL	3,29	
Sprint N2	Alterar Beneficiário	Inc	PE	4,6	Mudança caracterizada como refinamento, pois o arquivo lógico (AL) foi incluído na Sprint N1 da mesma <i>release</i> . Sem custo SFP.
	Beneficiário	Refin	AL	-	
	Incluir Benefício	Alt	PE	2,16	
Sprint N3	Emitir Relatório de Beneficiários	Inc	PE	4,6	Mudança caracterizada como refinamento. Apesar de ter sido desenvolvida na <i>Release</i> N-1, a funcionalidade foi alterada na <i>Sprint</i> N2. Sem custo SFP.
	Incluir Benefício	Refin	PE	-	
	Incluir Beneficiário	Refin	PE	-	

Alterar Beneficiário	Refin	PE	-	Mudança caracterizada como refinamento. Sem custo SFP. A funcionalidade foi desenvolvida em <i>sprint</i> da mesma <i>release</i> (<i>Sprint</i> N2).
Alterar Benefício	Alt	PE	2,16	Alteração caracterizada como Projeto de Melhoria (PE desenvolvido e pronto na <i>Release</i> N-1). Aplicado FI de 47%, conforme item 6.3 ($4,6 \text{ SFP} * 0,47 = 2,16 \text{ SFP}$)
Total SFP da <i>Release</i> N:				28,41 SFP

10.3.3. Exemplo de aplicação - Contagem para a *Baseline* da Aplicação

Como o exemplo utiliza o método SFP para a contagem de PF, a *baseline* será atualizada apenas com os processos elementares (PE) e arquivos lógicos (AL) incluídos. Apesar de terem ocorrido alterações em funcionalidades, a contribuição funcional destas não é alterada, conforme pressupõe o método SFP.

Sendo assim, a *baseline* da aplicação deverá ser atualizada com as quatro funções transacionais incluídas:

- Beneficiário (AL);
- Incluir Beneficiário (PE);
- Alterar Beneficiário (PE);
- Emitir Relatório de Beneficiários (PE).

A seguir, observa-se que não foram listadas as três funções desenvolvidas em *releases* anteriores, considerando que já foram contabilizadas na *baseline*, em *release* anterior. Como a contribuição funcional do método SFP é padronizada, não há alteração na contribuição funcional dessas funções na *baseline* da aplicação.

Tabela 20: Contagem de SFP da *Release* N para a *Baseline* da Aplicação

	Nome da Função	Tipo	SFP	Observação
Contagem SFP da Release N Para a Baseline	Beneficiário	AL	7	Na contagem da <i>Release</i> para a Baseline da aplicação, não devem constar as funcionalidades alteradas, excluídas e refinamentos.
	Incluir Beneficiário	PE	4,6	
	Alterar Beneficiário	PE	4,6	
	Emitir Relatório de Alunos por Disciplina	PE	4,6	
	Total de SFPs da Release N		20,8	

10.4. RECOMENDAÇÕES PARA O USO DO FATOR ÁGIL

O Fator Evolutivo Ágil é útil para comportar a dinâmica ágil de forma a incorporar as mudanças entre as *Sprints*. Corresponde ao cálculo do volume de evoluções inerentes ao processo ágil de uma *release*. Sua adoção consiste na substituição da contabilização de exclusões e alterações de processos elementares entre as *sprints* de uma mesma *release*, pela aplicação de um percentual sobre o tamanho contabilizado da *release*.

No contexto de projetos que utilizam metodologias ágeis, os refinamentos dentro das *releases* são inerentes ao processo ágil e não são contabilizadas na *release* corrente. Entretanto, a organização pode optar pela aplicação de um fator de impacto para absorção do esforço de construção dos refinamentos.

O item 5.2.4.2 da Portaria SGD/MGI nº 750 traz um alerta: “admite-se a aplicação de um fator ágil a ser definido no Termo de Referência seguindo as diretrizes constantes do roteiro de métricas do SISP, não sendo superior a 30%”.

O percentual definido no Termo de Referência será aplicado sobre o tamanho PF contabilizado na *release*; ou seja, a contratada será remunerada pelo produto entre o quantitativo PF obtido na *release* e o fator evolutivo ágil definido no TR:



$$\text{Remuneração da Contratada} = (\text{SFP_Release}) * (1 + \text{Fator Evolutivo Ágil}) * \text{Custo do PF}$$



Outros pontos importantes a serem observados:

- O Fator ágil deve ser aplicado em *releases* com quantidade de *sprints* estabelecidas. Por exemplo: para *Releases* com até 4 *sprints*, com *sprints* de até 4 semanas;
- O fator ágil não é aplicável para projetos executados em apenas 1 (uma) *sprint*.

10.4.1. Exemplo de Remuneração da Contratada

Para aplicação da fórmula de remuneração da contratada, o exemplo a seguir utiliza a quantidade de SFP da *Release N* (28,41 SFP) , Fator evolutivo ágil igual a 25% e custo do PF igual a R\$1.000,00.

Assim, aplicando a fórmula do item anterior para a *Release N*, temos:



$$\text{Remuneração da Contratada} = (\text{SFP_Release}) * (1 + \text{Fator Evolutivo Ágil}) * \text{Custo do PF}$$

$$\text{Remuneração da Contratada} = (28,41) * (1 + 0,25) * 1.000$$

$$\text{Remuneração da Contratada} = \text{R\$35.512,50}$$



No exemplo acima foi utilizado o Total de SFPs apurados na *Release N* (exemplo descrito no item 10.3.2), contudo vale destacar que a fórmula para a remuneração da contratada pode ser aplicada para a contagem de pontos de função com o método SFP ou com o método de APF tradicional.

10.5. CANCELAMENTO DE PROJETOS ÁGEIS

A contagem de PF de um projeto ágil cancelado deve levar em consideração as contagens de PF das *Releases* e/ou *Sprints* concluídas e a contagem da *Sprint* em andamento. No caso da *Sprint* em andamento, deve-se aplicar um percentual conforme os processos e atividades executados na *Sprint*, considerando o ciclo de desenvolvimento da *Sprint* em questão.

VERSÃO CONSULTA PÚBLICA

11. CONTAGEM DE PF EM CONTEXTOS ESPECÍFICOS

11.1. CONTAGEM DE IA (Inteligência Artificial)

A inteligência artificial (IA) é uma área da ciência da computação que busca resolver problemas utilizando métodos inspirados na inteligência humana; busca desenvolver algoritmos e sistemas que possam realizar tarefas que normalmente requerem inteligência humana, como reconhecimento de padrões, aprendizado e tomada de decisão. Isso é alcançado por meio da utilização de lógica, árvores de decisão e algoritmos de aprendizado de máquina, incluindo aprendizado profundo.

Sabendo-se do seu potencial para transformar, inovar e melhorar significativamente a vida das pessoas, essa tecnologia está sendo utilizada também nas organizações públicas para tomada de decisão, além de aumentar a eficácia e a eficiência do poder público.

Em caráter propositivo, para confecção deste instrumento, salienta-se que inteligência artificial tem vertentes amplas de aplicações e esse roteiro tem característica de refinamento sucessivo, de acordo com a evolução de técnicas e aplicações da IA.

Existe a IA baseada em regras, também conhecida como IA simbólica, que segue regras pré-definidas para realizar uma tarefa; e a IA baseada em aprendizado, também conhecida como IA sub-simbólica, que é treinada usando dados para aprender como realizar uma tarefa, sem seguir regras específicas.

11.1.1. Determinar a Fronteira da Aplicação

Os projetos de IA que têm impacto em diferentes fronteiras deverão ter uma contagem para cada fronteira impactada. É importante destacar que o conceito de fronteiras para as soluções de IA deve observar a visão do usuário da aplicação e não as tecnologias utilizadas.

Nesse sentido, o analista de métricas deve estar alerta para não contar a mesma funcionalidade mais de uma vez, a exemplo de uma funcionalidade que serve de SoT (*source of truth*) para mais de uma fronteira.

Da mesma forma, a existência de várias fronteiras (segurança, domínio, contexto, interface de usuário etc.) pode gerar múltiplas instâncias da aplicação (por exemplo: web, móvel, API etc.), com repercussão na quantidade de funções transacionais e de dados (SFP: ALs e PEs /APF: AIEs, EEs, SEs e CE), potencialização da duplicidade de componentes e impacto no custo final do produto.

Com vistas a minimizar as possibilidades de incorreção ou imprecisão da contagem de PF em projetos com impacto em mais de uma fronteira da aplicação, convém destacar que este roteiro contém orientações sobre como tratar componentes que são utilizados por mais de uma funcionalidade (item 6.17- Componente Reusável) e sobre a contagem de PF de funcionalidades entregues em mais de uma mídia (item 7.1- Contagem com Múltiplas Mídias).



Com o fim de evitar o superdimensionamento, incorreção e/ou imprecisão da contagem de PF em projetos com impacto em mais de uma fronteira da aplicação, este roteiro contém orientações sobre como tratar :

- Componentes que são utilizados por mais de uma funcionalidade (item 6.176.17- Componente Reusável)
- Funcionalidades entregues em mais de uma mídia (item 7.1- Contagem com Múltiplas Mídias).

11.1.2. Escopo da contagem de IA

O Ciclo de Vida do Desenvolvimento de um projeto de IA envolve várias atividades. De forma geral, o desenvolvimento de uma solução de IA segue o ciclo de vida ilustrado pela figura a seguir:

Figura 5: Ciclo de Vida do Desenvolvimento de um projeto de IA (adaptada SERPRO, 2024)



Vale salientar que o procedimento de mensuração de pontos de função tem seu escopo centrado na entrega para o usuário final. Dessa forma, atividades de descoberta no ciclo de vida da IA não são contempladas na contagem de PF, devendo ter seus produtos e serviços mensurados através de outra métrica, a exemplo da métrica HST (Hora de Serviço Técnico), descrita no **item 15** deste roteiro.

De forma análoga ao desenvolvimento de sistemas, as soluções de IA passam por dois momentos: (I) desenvolvimento; e (II) produção, mas em modelos de IA é comum chamar a produção de operacionalização.

Para efeito deste roteiro, convencionou-se aplicar regras de contagem de PF com foco na operacionalização da solução, ou seja, em como a aplicação será consumida após a fase de implantação. Nesse sentido, nos projetos de IA, a fase de operacionalização – que tem relação com as entradas e saídas da fronteira da aplicação – será utilizada para inferir a contagem de PF.

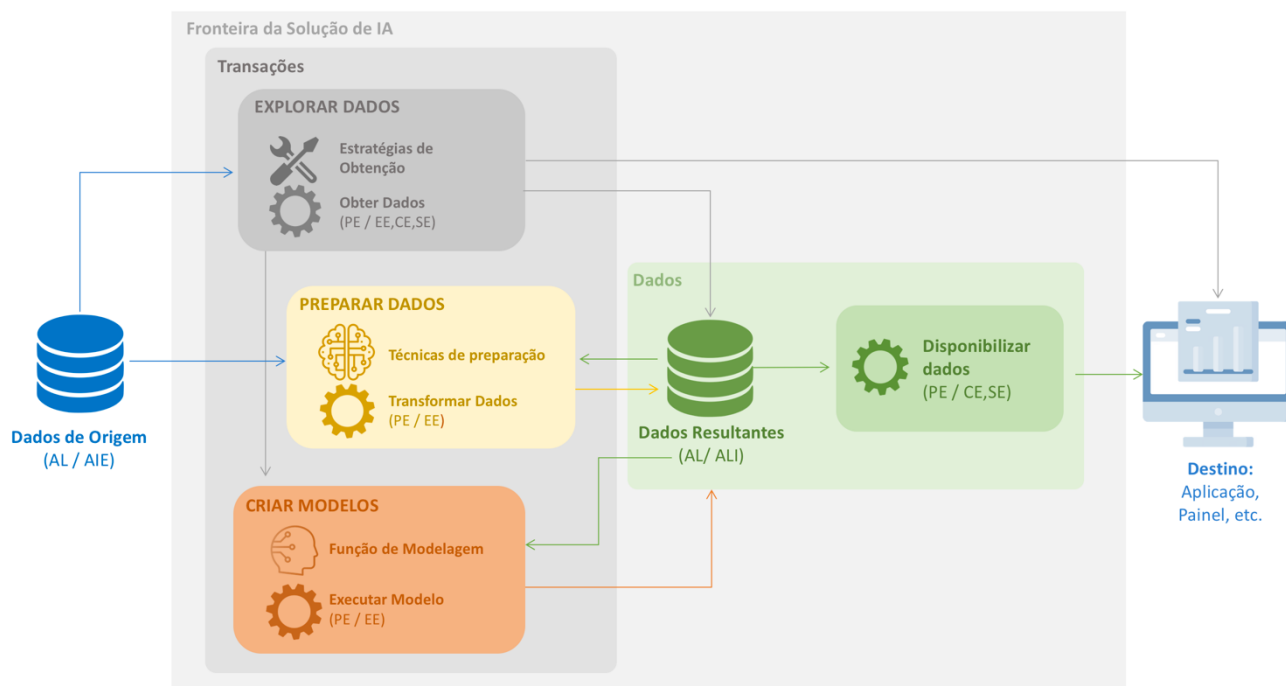
Sistemas ou modelos de IA podem ser operacionalizados seguindo arquiteturas de processamento em lote (batch) ou interativo. Dessa forma, este roteiro foi orientado por essas variações arquiteturais, que estão relacionadas com o fluxo de informação dentro da aplicação, conforme explicado nas próximas subseções.

11.1.3. Contagem para Processamento Batch de IA

No contexto da IA, o Processamento em Lote (Batch), se refere à execução usualmente automatizada, em horário previamente agendado, de inferências sobre novos conjuntos de dados.

A figura a seguir ilustra o fluxo de dados de uma solução de processamento que utiliza IA, onde os dados são inseridos na fronteira por meio de processamento em lote (batch):

Figura 6: Funções Transacionais e de Dados do Processamento em Lote (Batch) de IA (adaptada SERPRO, 2024)



A imagem ilustra o fluxo de dados de uma solução de processamento que utiliza IA, sendo iniciado com os Dados de Origem, fora da fronteira de IA, seguindo com a execução de macroprocessos para Explorar Dados, Preparar Dados e Criar Modelos. Os dados resultantes, tratados dentro da fronteira de IA são disponibilizados a partir da função transacional “Disponibilizar dados”. O fluxo termina com a exposição dos dados em aplicações, painéis etc., fora da fronteira de IA.

Nas alíneas seguintes serão detalhados os macroprocessos (expostos na imagem acima) , as funções transacionais e as funções de dados relacionadas, no contexto da contagem de PF, utilizando os métodos APF e SFP.

a) DADOS DE ORIGEM

Considera-se para efeitos de contagem, a situação em que o dado está na origem, havendo a necessidade de sua ingestão para dentro da fronteira da solução de IA.

Ressalta-se que este roteiro de métricas não leva em consideração a tecnologia e o formato dos dados de origem.

Para efeitos de contagem nos métodos APF tradicional e SFP, temos:

Tabela 21: Contagem PF de Dados de Origem

	Método APF identifica:	Método SFP identifica:	OBS
Para cada conjunto de dados lidos de outra aplicação e que tem sentido para usuário	Arquivo de Interface Externa (AIE)	Arquivo Lógico (AL)	Para caracterizar um AIE no método APF, é necessário que a função Obter Dados acarrete acesso à entidade do arquivo lógico pertencente a outra fronteira.
Processo de ingestão de dados	EE (Entrada Externa)	PE (Processo Elementar)	
Arquivos lógicos lidos para apresentação de dados em painéis ou geração de consultas	AIE	PE	

b) EXPLORAR DADOS

Esse processo tem como principal objetivo a obtenção dos dados e sua exploração através de gráficos, tabelas, estatísticas, dentre outros. Permite compreender melhor os dados, avaliar a sua qualidade e documentar resultados preliminares que ajudarão a subsidiar ações em etapas seguintes do processo.

Convém ressaltar que estruturas de dados temporárias ou arquivos transitórios (tais como: visões de bancos de dados – *views*, arquivos texto transitórios, arquivos com extensão .CSV não permanentes, dentre outros) criados a partir dos dados de origem em forma passageira não serão contados (semelhante a buffers ou *data staging area* – DSA, que não são considerados para contagem), conforme orientações para contagem de PF para DW, descritas no item 12 deste roteiro.

Para efeito de contagem será considerada apenas a tarefa **“Obter Dados”**, conforme exposto na figura 6. Contudo, no contexto da exploração de dados, independentemente da intenção primária, nesse tipo de projeto, haverá, na prática, a obtenção de dados de uma aplicação de fora da fronteira.

Sendo assim, admite-se a contabilização da função de obtenção de dados (PE/EE) seguida da função de exploração dos dados (PE/ CE ou SE).

Nesse sentido, para contagem de PF com os métodos APF tradicional e SFP temos:

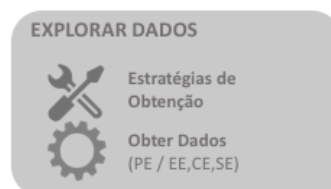


Tabela 22: Contagem PF para o macroprocesso explorar dados

	Método APF identifica:	Método SFP identifica:	OBS
Para funções transacionais com a intenção primária de obtenção de dados de uma aplicação	EE	PE	
Para funções transacionais com intenção primária de exibir dados em outra aplicação/painel, nos casos que não exigem nenhuma forma de tratamento dos dados, ou seja, a simples extração, não sendo aplicada nenhuma transformação ou cálculo sobre o dado, e se não houver gravação dos dados em arquivo lógico	CE	PE	Para o método SFP não há necessidade da identificação da intenção primária. Nesse sentido, as funções transacionais para obtenção e exibição de dados são contabilizadas como PE (processo elementar) no método SFP, com contribuição funcional igual a 4,6 SFP.
Para funções transacionais com intenção primária de exibir dados em outra aplicação/painel, nos casos em que houver a implementação de tratamento com transformação ou cálculo sobre o dado na extração e/ou gravação do dado em arquivo lógico	SE	PE	



ATENÇÃO! A análise exploratória não terá pontos de função contados.

Porém, se houver disponibilização dos dados para construção de gráficos e painéis, deve ser considerada a tarefa “Disponibilizar dados” (alínea f).

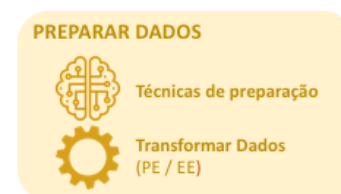
c) PREPARAR DADOS

Esta atividade realiza as ações necessárias para disponibilizar os dados no formato e nos padrões exigidos pela modelagem analítica sucessora.

Para efeito de contagem, será considerada apenas a tarefa “Transformar Dados”, quando houver a ingestão de dados de Origem.

Independentemente da quantidade de técnicas distintas aplicadas na preparação de dados, será considerada apenas a tarefa de Transformação de Dados para a contagem de PF. Desta forma, a função transacional de transformação dos dados:

- a) Será considera uma EE, no método APF tradicional;
- b) Será considerado um Processo elementar (PE), no método SFP, com pontuação igual a 4,6 SFPs.



Independentemente da quantidade de técnicas distintas aplicadas na preparação de dados, será considerada apenas a Transformação de Dados para a contagem de PF.

AS TÉCNICAS DE PREPARAÇÃO NÃO DEVEM SER CONTABILIZADAS NA CONTAGEM DE PF, mesmo aquelas que transformarem o dado de maneira que seu formato seja necessário para ser aceito em uma função de modelagem analítica.

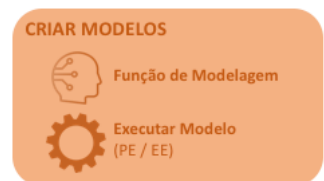
Para exemplificar algumas das técnicas de preparação de dados que habitualmente podem ser encontradas em soluções de IA, destaca-se o rol abaixo, sem exaurir demais técnicas que sejam necessárias:

- a) Concatenar campos;
- b) Anonimizar campo;
- c) Realizar transformação ou imputação de dado;
- d) Converter para minúsculas ou maiúsculas;
- e) Realizar estemização;
- f) Realizar tokenização qualificada;

- g) Remover caracteres especiais;
- h) Remover palavras muito curtas ou longas;
- i) Remover stopwords ou palavras proibitivas;
- j) Tratar dados ausentes;
- k) Remover palavras muito ou pouco frequentes;
- l) Normalização;
- m) Discretização;
- n) Remoção de ruídos em imagens.

d) CRIAR MODELOS

A atividade Criar Modelos em Inteligência Artificial engloba o emprego de uma ou mais técnicas de representação do conhecimento e de modelagem analítica, compreendendo a aplicação de algoritmos ou estratégias matemáticas na solução de problemas que envolvam descoberta de padrões em dados. Trata-se de processo iterativo e incremental, no sentido de refinamentos sucessivos dos modelos desenvolvidos.



Para efeitos de contagem de Pontos de Função, qualquer modelo a ser implementado em torno da resolução de um problema será representado por uma função que englobará suas técnicas de modelagem, estratégias de representação das entradas do modelo e sua efetiva sensibilização como resultado do processo.

No macroprocesso Criar Modelos será considerado, para efeitos de contagem, apenas a tarefa “**Executar Modelo**”. Nesse sentido:

- No método APF, independentemente da quantidade de técnicas avaliadas, aplicadas e/ou testadas, o processo de execução do modelo será contabilizado como entrada externa (EE);
- No método SFP, independentemente da quantidade de técnicas avaliadas, aplicadas e/ou testadas, o processo de execução do modelo será contabilizado como processo elementar (PE);

Ressalta-se que quando houver entidade que guarde dados de treinamento e estes forem utilizados para geração de relatórios para o cliente, deverá ser considerado o respectivo arquivo lógico (AL/ALI) e as funções transacionais geradas (PE/ CE ou SE).

e) DADOS RESULTANTES

Quanto aos dados resultantes, para efeito de contagem nos métodos APF tradicional e SFP, temos:

Tabela 23: Contagem PF para Dados Resultantes

	Método APF identifica:	Método SFP identifica:	OBS
Para cada origem lida e armazenada, de acordo com a percepção de valor de negócio pelo usuário	ALI e EE	AL e PE	Deve-se contabilizar o Arquivo lógico (ALI/AL) e o processo elementar de carregamento da entidade lógica (EE/PE)
Se houver gravação em entidade lógica de dados diferente da já existente	ALI	AL	Exemplo: os dados resultantes da execução e dados estatísticos do modelo

Para os casos em que o dado gravado seguir a estrutura lógica de um AL/ALI já criado e por uma decisão de projeto se optar pela criação de uma nova tabela, não será contado um novo arquivo lógico, mas melhoria alterada da entidade existente. Por exemplo, se já houver uma tabela com o dado original e for decidido a criação de uma nova tabela com dado tratado, neste caso será considerada uma melhoria alterada da entidade existente.

f) DISPONIBILIZAR DADOS

Em alguns casos o requisitante solicita a apresentação dos dados resultantes como uma visualização externa útil.

Observe na **figura 6** que pode haver leitura tanto de arquivos lógicos internos, como de arquivos lógicos externos para alimentar painéis/aplicações externas.

Nestes casos, para efeito de contagem de PF nos métodos APF tradicional e SFP, temos:

Tabela 24: Contagem PF para Disponibilizar Dados

	Método APF identifica:	Método SFP identifica:
Para cada objeto visual independente de dados (gráfico, relatório) com sentido completo apresentado ao usuário por meio de: gráficos, painéis, relatórios, serviços, dentre outros	CE ou SE	PE

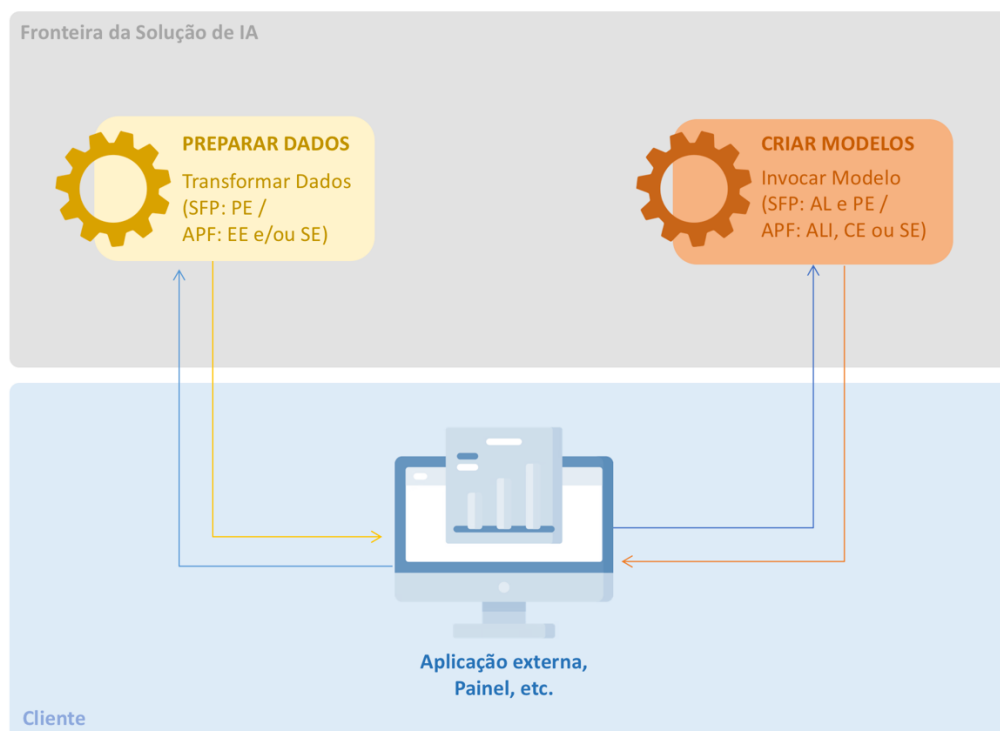
g) DESTINO

Tratam-se de aplicações, painéis, etc, fora da fronteira de IA, que podem utilizar os dados processados na solução de IA podendo ter contagem própria.

11.1.4. Contagem para a Execução Interativa de IA

O processamento interativo, no contexto da IA, se refere a inferências realizadas sob demanda, em geral fornecendo os resultados de maneira síncrona. A Figura a seguir detalha o cenário da execução interativa, contemplando as atividades envolvidas:

Figura 7: Funções Transacionais do Processamento Interativo de IA (adaptada SERPRO, 2024)



a) PREPARAR DADOS

Esta atividade realiza as ações necessárias para disponibilizar os dados no formato e nos padrões exigidos pela modelagem analítica sucessora.

Na atividade Preparar Dados será considerado para efeitos de contagem a tarefa “Transformar Dados”.

Nesse sentido, para os métodos APF tradicional e SFP, temos:



Tabela 25: Contagem PF para Preparar Dados em Execução Interativa de IA

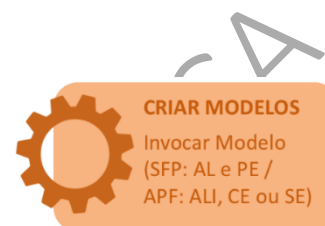
	Método APF identifica:	Método SFP identifica:
Para cada entrada de dados na fronteira da solução de IA, que resultou em gravação na base	EE	PE
Para cada transformação devolvida para a aplicação, fora da fronteira, que acionou a solução IA	SE	PE

Exemplos de técnicas de preparação de dados para execução interativa são similares as técnicas de execução em lote; algumas foram citados na alínea “c)” da seção 11.1.3.

Ressalta-se que a identificação de tipos de dados seguirá com as mesmas regras da identificação de atributos reconhecidos a partir do ponto de vista do usuário.

b) CRIAR MODELOS

Para o macroprocesso Criar Modelos de uma execução interativa de IA, será considerado para efeitos de contagem de PF apenas o processo “Invocar Modelo”.



Dessa forma, para os métodos APF e SFP, temos:

Tabela 26: Contagem PF para Criar Modelos em Execução Interativa de IA

	Método APF identifica:	Método SFP identifica:
Para cada modelo executado, independentemente da quantidade de técnicas avaliadas, aplicadas e/ou testadas	SE	PE
Quando houver entidade que guarda dados de treinamento nos resultados do modelo e estes forem utilizados para geração de relatórios para o cliente, deverá ser computado o respectivo arquivo lógico e as funções transacionais geradas	ALI, SE ou CE	AL e PE

Convém destacar que, independentemente do método de contagem de PF utilizado, toda função de dados deverá ser computada uma única vez na contagem.

c) CLIENTE

Tratam-se de aplicações, fora da fronteira, que podem acionar a solução de IA devendo ter contagem própria. Convém destacar que esse roteiro não leva em conta a tecnologia e o formato das aplicações externas.

11.1.5. Manutenções de Soluções de IA

Posteriormente à implantação da solução de IA, poderá ocorrer a necessidade de manutenções nas funções de dados e/ou transacionais, que deverão ser tratadas seguindo as definições do item 6.3 deste roteiro.

11.2. CONTAGEM DE PF EM *CHATBOT*

Este item apresenta diretrizes de Contagem de Pontos de Função de soluções de *ChatBot*. Antes de detalhar como realizar a contagem de PF, convém entender o que é e como funciona essa solução.

Podemos dizer que “Bot” é um diminutivo de “Robot” (robô). Assim, ChatBot se auto explica como “Robô de conversas”. Essa solução é capaz de simular uma conversa com um usuário em linguagem natural NLP (*Natural-Language Processing*) por meio de aplicativos de mensagens, sites e outras plataformas digitais (*WhatsApp, Instagram, Sites, Twiter, etc*).

Os ChatBots são sistemas que utilizam uma interface conversacional para entregar um produto, serviço ou experiência e podem ser orientados por regras e/ou por Inteligência Artificial (IA)

Essa tecnologia automatiza conversas e o seu uso nas instituições governamentais auxilia na aproximação com o público através de chats que respondem com rapidez e eficiência, permitindo o atendimento e disponibilização de informações para o cidadão 24h. Funciona também como uma espécie de FAQ interativo e personalizado. O usuário chega ao ponto que deseja sem precisar percorrer uma página extensa, dando apenas comandos através de palavras-chave.

A contagem de PF vai abranger apenas atividades relacionadas aos projetos de desenvolvimento e manutenção para a solução de *ChatBot*. Outras atividades como Descoberta e Curadoria devem considerar outras formas de dimensionamento, a exemplo da métrica HST.

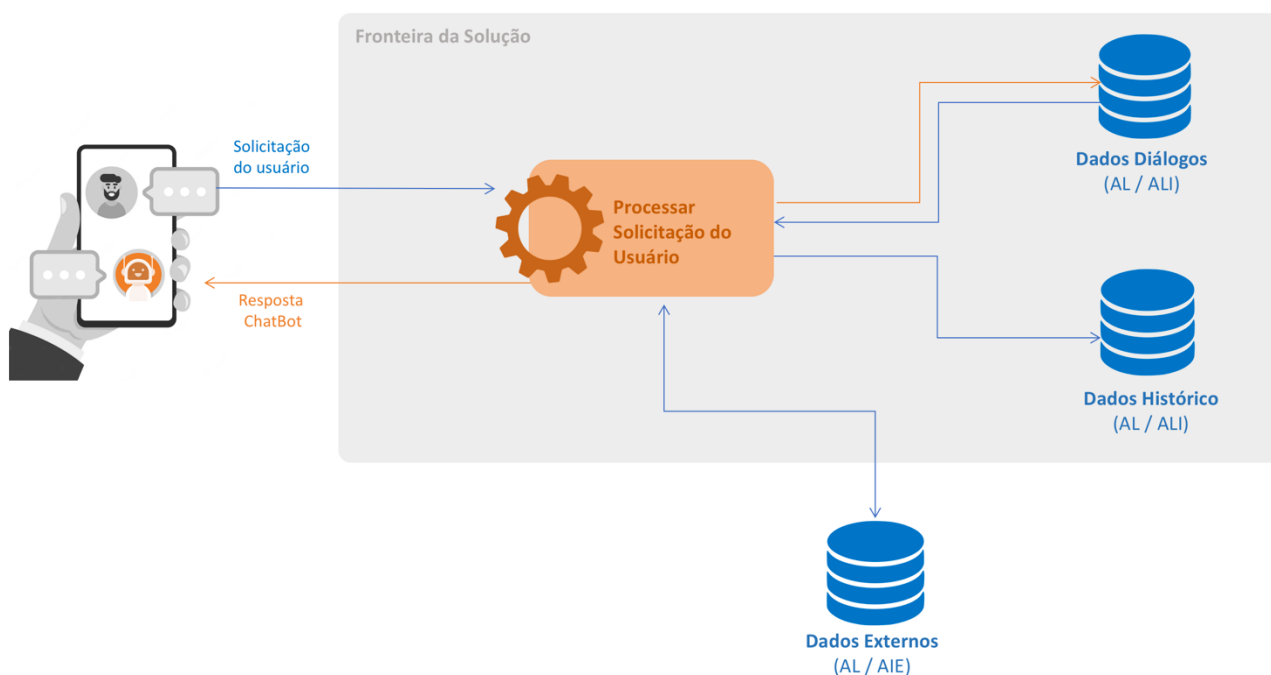
Convém destacar que a métrica PF considera a visão do usuário e é independente da metodologia e tecnologia utilizadas.

11.2.1. Visão *ChatBot* para contagem PF

A imagem a seguir reproduz, de forma simplificada, o funcionamento de ChatBots baseados em regras ou fluxos estruturados (que utilizam, principalmente, árvores de decisão) que após recebimento de uma solicitação do usuário processa vários dados para então enviar a melhor resposta a solicitação do usuário.

Contudo, convém destacar que há uma gama enorme de tipos de ChatBots não cobertos pela imagem a seguir, que utilizam modelos estatísticos, redes neurais e aprendizado profundo para compreender e responder às entradas dos usuários, como o ChatGPT, por exemplo.

Figura 8: Visão simplificada do funcionamento de um ChatBot para contagem de PF (adaptada SERPRO, 2024)



De forma simplificada, a imagem mostra que após a solicitação do usuário, é iniciado o processamento da sua solicitação; para isso, o sistema verifica os dados referentes as configurações de nós de diálogo do ChatBot, podendo utilizar dados de sistemas externos. O sistema envia resposta para o usuário, atendendo sua solicitação e registra a interação entre o usuário e o ChatBot.

Como pode ser observado na imagem acima, o usuário utiliza uma interface de conversação para obter informações. Existem vários canais possíveis, a saber: web, aplicativos, *WhatsApp*, *FB Messenger*, *Twitter DM*, etc.

Para enviar uma mensagem de resposta para o usuário, os ChatBots podem ser orientados por regras e/ou por Inteligência Artificial (IA):

- o conhecimento dos bots orientados a regras é limitado às regras pré-definidas, ou seja, ele vai conversar sobre aquilo que lhe for orientado nas árvores de decisão do fluxo criado. Eles realizam tarefas objetivas e o seu vocabulário é limitado ao que foi definido.
- os bots que utilizam recursos, técnicas e algoritmos de IA têm uma base de conhecimento externa ampla e regularmente treinada, o que aumenta sua capacidade de conversa. Quanto mais treino, maior conhecimento; quanto maior o conhecimento, maior e melhor pode ser a conversa. Ele aprende a linguagem e procura soluções.

- existem também bots híbridos que são orientados a regras e também recebem treinamento com técnicas de IA. Nestes casos, são implementados os dois cenários em um mesmo ChatBot.

11.2.2. Escopo da contagem PF para *ChatBot*

O escopo da contagem abrange geralmente o Projeto de Desenvolvimento e/ou de Melhoria de aplicações que se integram ao ChatBot e a evolução da solução de IA.

O projeto de desenvolvimento ou de melhoria realizado nas aplicações de integração, que exibirão as informações resultantes das soluções de ChatBot devem ter a contagem realizada separadamente em cada fronteira.

Nesse sentido, cumpre alertar sobre a especificidade na contagem de PFs de componentes utilizados por mais de uma funcionalidade e de funcionalidades entregues em mais de uma mídia, que podem refletir na quantidade de funções transacionais e de dados, potencialização da duplicidade e impacto no custo final do produto.

Com o fim de minimizar as possibilidades de incorreção ou imprecisão da contagem de PF nos projetos de desenvolvimento ou de melhoria realizados nas aplicações de integração, que exibirão as informações resultantes das soluções de ChatBot, destaca-se novamente que este roteiro contém orientações sobre como tratar componentes que são utilizados por mais de uma funcionalidade (item 6.17- Componente Reusável) e sobre a contagem de PF de funcionalidades entregues em mais de uma mídia (item 7.1- Contagem com Múltiplas Mídias).

11.2.3. Contagem de funções de dados para *ChatBot*

Considerando a figura 8, que apresenta de forma simplificada o funcionamento de ChatBots baseados em regras ou fluxos estruturados, é possível identificar:

- Arquivos lógicos que mantêm dados referentes à configuração dos nós de diálogo da aplicação, independentemente das características da ferramenta tecnológica utilizada na criação do bot; e
- Entidades apenas lidas de outras fronteiras, associadas a integração com sistemas externos), API ou qualquer outra forma de comunicação existente.

Nesse contexto, o quadro a seguir apresenta a estrutura básica dos arquivos lógicos de um bot (para os métodos APF e SFP), mas vale ressaltar que não estão limitadas às listadas, podendo haver variação na visão do usuário dependendo das características específicas do negócio:

Tabela 27: Arquivos lógicos de um Bot

Funções de Dados	Método APF identifica:	Método SFP identifica:	Tipos de dados básicos
Dados de Diálogo: dados referentes às configurações dos nós de diálogo do chatbot;	ALI	AL	Condição de entrada, Resposta Textual, Resposta Textual de Opções, Nome do componente, Tipo do Componente
Dados de Histórico: registro das interações entre usuário e chatbot;	ALI	AL	Data/Hora, ID de conversação, Canal, Entrada do usuário, Intenção, Resposta, Acurácia, Tipo de Componente, Nome do Componente
Dados externos: eventuais integrações com sistemas externos	AIE	AL	Campos percebidos pelo usuário que são retornados do sistema externo em questão

Vale observar que os registros lógicos a seguir devem ser subgrupos da entidade Diálogo: *onboarding*, inatividade, repetição, avaliação, *feedback*, termos Políticos, impropriedades, transbordo, sugestões, canal (*Webchat*, *WhatsApp*, *Twitter*, *Facebook Messenger* etc.), motores de -conversação (*Watson*, *Rivescript*, etc), etc.

A alteração de funções de dados poderá ocorrer em projetos de manutenção que demandem a mudança da estrutura das entidades e, nesse caso, deverão ser tratadas seguindo as definições do item 6 deste roteiro.

11.2.4. Contagem de funções transacionais pra *Chatbot*

Considerando ainda a figura 8, que representa de forma simplificada o funcionamento de *ChatBots* baseados em regras ou fluxos estruturados, para projetos de desenvolvimento ou melhoria convencionou-se computar (1) as intenções do usuário e (2) os tipos de nós de diálogos criados para serem apresentados ao usuário a partir de cada ação realizada por ele.

“Exibir mensagem” corresponde a mensagem de retorno do ChatBot conforme nó configurado, envolvendo itens como: *onboarding*, repetição, *feedback*, termos políticos, impropriedades, transbordo, avaliação, etc.

Cada tipo de nó de diálogo que envia dados para fora da fronteira deve ser contado como uma função transacional (APF: SE/ SFP: PE), tendo em vista que as informações retornadas são resultantes de um processamento que deriva a resposta mais adequada de diálogo para a intenção do usuário, a partir da base de conhecimento previamente curada.

A primeira configuração de cada tipo de nó de diálogo será contabilizada como uma função transacional (APF: SE/ SFP: PE).

A seguir, uma lista básica sugerida de tipos de nós de diálogo, representadas por funções de transação:

- Exibir a mensagem *Onboarding*;
- Exibir a mensagem Inatividade;
- Exibir mensagem Repetição;
- Exibir mensagem Avaliação Positiva;
- Exibir mensagem Avaliação Negativa;
- Exibir a mensagem *Feedback*;
- Exibir mensagem Termos Políticos;
- Exibir mensagem Impropropriedades;
- Exibir mensagem Transbordo;
- Exibir mensagem Sugestões.

Esse rol de funções transacionais é uma referência não exaustiva e pode ser ajustada de acordo com as particularidades de cada projeto. Para cada novo tipo nó de diálogo identificado para o bot do negócio será computado um processo elementar adicional.

11.2.5. Manutenção em soluções de *Chatbot*

Convém destacar novamente que a figura 8 e as orientações para contagem de PF apresentadas nos subitens anteriores deste roteiro, referem-se a contagem PF para ChatBots baseados em regras ou fluxos estruturados.

Nesse contexto, a alteração de funções de dados poderá ocorrer em projetos de manutenção que demandem a mudança da estrutura das entidades e, nesse caso, deverão ser tratadas seguindo as definições do item 6 deste roteiro.

Em projetos de manutenção, admite-se considerar adicionalmente às funções transacionais identificadas como alteradas, uma função transacional (PE/EE) para a configuração interna realizada em cada Nó de Diálogo.

11.2.6. Itens sem contagem de PF em soluções de *Chatbot*

A realização de curadoria e as atividades relacionadas à experiência do usuário – UX realizadas durante o processo de desenvolvimento ou manutenção de ChatBot não estão contempladas na contagem de PF; sugere-se que o seu dimensionamento seja feito através da métrica de HST (horas de serviço técnico), seguindo as regras descritas no **item 15** deste roteiro.

VERSÃO CONSULTA PÚBLICA

11.3. CONTAGEM PF DE PAINÉIS ANALÍTICOS

Os Painéis Analíticos têm como propósito a apresentação de informações gerenciais para o usuário em *Dashboards* (gráficos) e Tabelas Dinâmicas.

Em alguns casos, há necessidade de implementação dos Painéis em mais de uma plataforma de desenvolvimento, por exemplo *Tableau* e *Qlik*. Neste caso, as funções implementadas nas duas plataformas devem seguir a abordagem *Multiple Instance*, no conceito de múltiplas mídias, sendo contadas como funções independentes, considerando-se tratar de implementações distintas em plataformas distintas para atendimento de necessidades de negócio do cliente.

Com vistas a minimizar as possibilidades de incorreção ou imprecisão da contagem de PF em projetos com impacto em mais de uma fronteira da aplicação, o roteiro de métricas contém orientações sobre como tratar componentes que são utilizados por mais de uma funcionalidade (item 6.17- Componente Reusável) e sobre a contagem de PF de funcionalidades entregues em mais de uma mídia (item 7.1- Contagem com Múltiplas Mídias).

Nos tópicos a seguir apresentam-se diretrizes para contagem de PF de projetos de desenvolvimento e manutenção de Painéis Analíticos utilizando o método APF ou SFP, além das funções de dados e transacionais geralmente identificadas nesses projetos.

11.3.1. Criação e carga de tabelas

Os principais componentes de um painel analítico incluem gráficos, tabelas, indicadores chave de desempenho (KPIs) e filtros interativos.

Com relação às tabelas, destaca-se aqui que nem toda tabela pode ser contada como arquivo lógico; ou seja: a contagem de PF não deve contabilizar um arquivo lógico para cada tabela criada.

Um grupo lógico tem caráter orgânico, funcional, podendo ser implementado por uma ou mais tabelas e mantido por vários processos elementares diferentes. Tabelas possuem regras específicas para criação, não comportando todo e qualquer grupo lógico de dados. Nesse sentido, deve-se identificar no escopo da contagem todos os dados e informações de controle logicamente relacionados e reconhecidos pelo usuário.

O processo de carga de dados é contado como um processo elementar. Cada processo de carga de dados deve ser contado como um processo elementar independente.

Caso o Painel tenha carga inicial (*Full*) e carga incremental (*Delta*), então deve-se contar um processo elementar para cada processo de carga.

Na tabela a seguir, apresentam-se as funções transacionais e de dados a serem contadas, para os métodos APF tradicional ou SFP:

Tabela 28: Funções para criação de carga de tabelas de painéis analíticos

	APF	SFP	OBSERVAÇÕES
Grupos lógicos de Dados	ALI	AL	Atenção: nem toda tabela pode ser contada como arquivo lógico; ou seja: a contagem de PF não deve contabilizar um arquivo lógico para cada tabela criada.
Carga de dados	EE	PE	Cada processo de carga de dados é contado como uma Entrada Externa independente
Carga Inicial (<i>Full</i>)	EE	PE	Deve-se contar um processo elementar para cada processo de carga.
Carga Incremental (Delta)	EE	PE	Deve-se contar um processo elementar para cada processo de carga.



ATENÇÃO! Na contagem de PF, independente do método utilizado (APF ou SFP), o termo arquivo lógico não significa arquivo físico ou tabela.

Na contagem de PF, “arquivo” se refere a um grupo de dados logicamente relacionados e não à implementação física desses grupos de dados. Nesse sentido, os arquivos lógicos contabilizados na contagem de PF não se confundem com tabelas. Assim, dependendo da implementação, um arquivo lógico pode ter “n” tabelas.

11.3.2. Leitura de dados de outras aplicações

Os grupos de dados lógicos lidos de outras aplicações para validação de informações no processo de carga de dados ou apresentação de informações nos gráficos são contados como arquivos lógicos.

Os grupos lógicos usados na geração dos gráficos dos painéis podem ser lidas de outros sistemas, devendo ser consideradas na contagem de pontos de função.

Tabela 29: Função para leitura de dados de outras aplicações

	APF	SFP	OBSERVAÇÕES
Dados de outras aplicações	AIE	AL	São contados os grupos de dados lógicos lidos de outras aplicações para validação de informações no processo de carga de dados ou apresentação de informações nos gráficos.

11.3.3. Gráficos e tabelas dinâmicas

A função de geração e apresentação do gráfico ou tabela dinâmica deve ser contada como processo elementar. Geralmente, os gráficos e tabelas apresentadas nos Painéis possuem dados gerenciais calculados.

Um Painel pode apresentar vários gráficos independentes, assim a geração de cada gráfico deve ser contada como um processo elementar distinto.

Tabela 30: Função para apresentação de gráficos ou tabelas dinâmicas

	APF	SFP	OBSERVAÇÕES
Apresentação de Gráficos ou Tabelas dinâmicas	SE	PE	<p>Geralmente, os gráficos e tabelas apresentadas nos Painéis possuem dados gerenciais calculados, por isso geralmente são contados como SE no método APF;</p> <p>Para o método APF, os Arquivos Referenciados (ARs) nos Gráficos ou Tabelas são os dados lidos dos ALIs e/ou AIEs da aplicação. Os Tipos de Dados (TDs) são os dados apresentados nos gráficos ou tabelas e os filtros utilizados para geração do gráfico ou tabelas.</p>

11.4. CONTAGEM PF PARA PROJETOS BASEADOS EM GEOTECNOLOGIAS

Sistemas baseados em geotecnologias são ferramentas utilizadas para análise de informações geográficas que usam funções de dados geométricos ligados a tabelas de atributos alfanuméricos e numéricos, a exemplo do S2 do Google e H3 da Uber, usam atributos numéricos de 64 bits.

Neste item serão apresentadas diretrizes para medição de Pontos de Função, utilizando os métodos APF ou SFP, de sistemas georreferenciados, considerando diversos cenários de identificação funções de dados e transacionais, não tratados diretamente pelo CPM 4.3.1 e SPM 2.1.

11.4.1. Medição de camadas georreferenciadas

Esta seção se refere a camadas de georreferenciamento, tais como “camadas *overlay*”, “*baselayer*”, entre outras. Existem características comumente apresentadas em sistemas de georreferenciamento, a saber:

- cada camada é geralmente obtida de outra aplicação, distinta das demais camadas;
- cada camada é individualmente reconhecida pelo usuário da aplicação, fazendo parte da área comercial da aplicação;
- cada camada constitui uma estrutura de dados bem heterogênea e distinta das demais;
- estas camadas são grupos de dados independentes entre si, não constituindo entidades dependentes de qualquer tipo.

Interpretando essas características sob a ótica de pontos de função, cada camada de georreferenciamento será considerada um arquivo lógico independente.

Observa-se que os Tipos de Dados (TDs) de cada camada, via de regra, são distintos e incompatíveis, não fazendo sentido, por exemplo, agrupar “todas as camadas de *overlay* do sistema” em um único arquivo lógico “Camada *Overlay*”, nem agrupar “todas as camadas *baselayer*” em um arquivo lógico “Camada *Baselayer*”.

Desta forma, cada camada georreferenciada deverá ser contada como um Arquivo lógico distinto.

11.4.2. Apresentação de dados em formatos diferentes

Esta seção trata o cenário em que um mesmo processo elementar é apresentado em mídias/formatos diferentes, conforme o conceito “*multiple instance*”, descrito no item 7.1 deste roteiro de métricas.

Considere a necessidade de uma consulta ‘C’ realizando a leitura em um grupo de dados georeferenciados. E ainda, considere que o sistema possua as seguintes funcionalidades:

- apresentação destes dados da consulta ‘C’ em forma de gráfico;

- apresentação desta mesma consulta 'C' em forma de tabela;
- apresentação desta mesma consulta 'C' em forma de mapa.

Tais funcionalidades são independentes e constituem três (3) processos elementares na visão do usuário, considerando a abordagem *multiple instance*. Desta forma, a apresentação de uma mesma consulta de dados em formatos distintos, por exemplo: gráfico, tabela e mapa deve ser contada separadamente.

Dessa forma, para o cenário de representação de grupos georreferenciados idênticos em 2 ou mais formatos, devem ser contados 2 ou mais processos elementares distintos, em aderência a abordagem *Multiple Instance*. Ou seja, conta-se um Processo Elementar para cada formato de apresentação. No método APF, em geral, estas consultas gerenciais são contadas como Saídas Externas.

11.4.3. Apresentação simultânea de vários elementos visuais

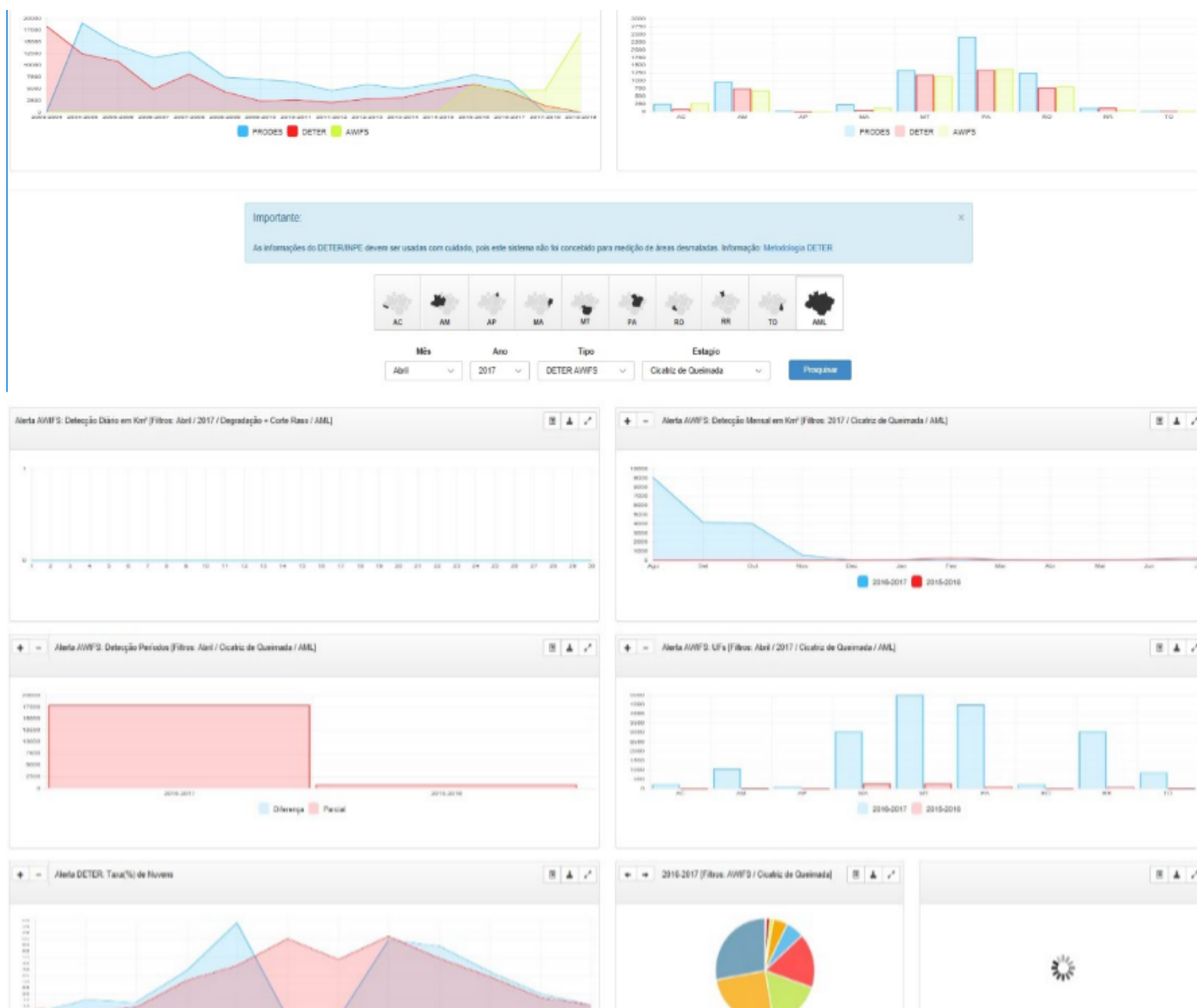
Esta seção trata cenários em que vários “elementos visuais georreferenciados” são exibidos simultaneamente, em uma mesma tela.

Deve-se contar um (1) processo elementar para cada “elemento visual georreferenciado”, mesmo quando eles forem exibidos simultaneamente em uma mesma tela, considerando a visão do usuário.

Entenda por “elemento visual georreferenciado” uma consulta feita e exibida em um formato visual (e.g. em gráfico, em tabela ou em mapa). Por exemplo, um gráfico representa a consulta de dados em um “elemento visual georreferenciado”. Note que esta mesma consulta poderia ser representada em forma de tabela, mapa, etc).

Como exemplo, observe a figura a seguir que apresenta oito (8) “elementos visuais georreferenciados”: oito consultas exibidas em forma visual de gráficos, atendendo aos requisitos de negócio do usuário.

Figura 9: Elementos visuais georreferenciados



Dessa forma:

- Para o método SFP, o desenvolvimento de um “elemento visual georreferenciado” deve ser considerado um processo elementar (PE);
- Para o método APF, a apresentação de cada “elemento visual georreferenciado” deve ser contado como uma Saída Externa (SE).

11.4.4. Funcionalidades complexas de georreferenciamento

A ocorrência de cada função transacional listada a seguir (e outras funcionalidades similares) deverá ser contada como um processo elementar distinto, considerando que na visão do usuário tratam-se de requisitos independentes:

a) Funcionalidades que disponibilizam a “edição de desenhos de elementos em mapas dentro do próprio sistema”, tais como:

- Inclusão de elementos (polígono, ponto, reta...) no mapa – efetuada via desenho em mapa;
- Inclusão de elementos (polígono, ponto, reta...) no mapa – efetuada via coordenadas do elemento;
- Alteração de elementos (polígono, ponto, reta...) no mapa – efetuada via desenho em mapa;
- Alteração de elementos (polígono, ponto, reta...) no mapa – efetuada via coordenadas do elemento;
- Exclusão de elementos (polígono, ponto, reta...) no mapa – efetuada via desenho em mapa;
- Exclusão de elementos (polígono, ponto, reta...) no mapa – efetuada via coordenadas do elemento.

b) Funcionalidades que disponibilizam a manipulação de dados geo por arquivos shapefile, tais como:

- Funcionalidade para gerar, armazenar e disponibilizar em um “arquivo *shapefile*” as informações geoespaciais obtidas de uma fonte de dados (e.g. uma “camada geo”, uma representação visual de mapa contendo vários elementos e/ou camadas geo, um conjunto de dados geoespaciais armazenados em banco, etc.);
- Funcionalidade para efetuar download de arquivo *shapefile*;
- Funcionalidade para efetuar upload de arquivo *shapefile* (incluindo validações negociais dos atributos do arquivo);

c) Funcionalidades para visualização mapas na tela (com ou sem filtros de pesquisa), junto aos seus Arquivos Lógicos Referenciados (camadas, elementos, etc).

d) Funcionalidades onde o usuário solicita um zoom no mapa para uma localização / área específica.

Como exemplo, apresenta-se a tabela a seguir com exemplos de funções transacionais e de dados descritas nos subitens anteriores, contabilizadas a partir dos métodos APF e SFP:

Tabela 31: Contagem de funções transacionais e de dados de um sistema de Geo

APF	SFP	OBSERVAÇÕES
-----	-----	-------------

Cada camada georreferenciada	ALI ou AIE	AL	Ex: “camadas <i>overlayer</i> ”, “ <i>baselayer</i> ”, entre outras
Exibir dados de consulta em forma de gráfico	CE ou SE	PE	Para o cenário de representação de grupos georreferenciados idênticos em 2 ou mais formatos, devem ser contados 2 ou mais processos elementares distintos, em aderência a abordagem Multiple Instance, descrita no item 7.1 deste roteiro de métricas.
Exibir dados de consulta em forma de tabela	CE ou SE	PE	
Exibir dados de consulta em forma de mapa	SE	PE	No método APE, estas consultas gerenciais normalmente são contadas como SE.
Cada elemento visual georreferenciado	SE	PE	Exemplo no item 11.4.3.
Cada funcionalidade para INCLUIR elementos (polígono, ponto, reta) no mapa, via DESENHO em mapa	EE	PE	
Cada funcionalidade para INCLUIR elementos (polígono, ponto, reta) no mapa, via COORDENADAS do elemento	EE	PE	
Cada funcionalidade para ALTERAR elementos (polígono, ponto, reta) no mapa, via DESENHO em mapa	EE	PE	
Cada funcionalidade para ALTERAR elementos (polígono, ponto, reta) no mapa, via COORDENADAS do elemento	EE	PE	
Cada funcionalidade para EXCLUIR elementos (polígono, ponto, reta) no mapa, via DESENHO em mapa	EE	PE	

Cada funcionalidade para EXCLUIR elementos (polígono, ponto, reta) no mapa, via COORDENADAS do elemento	EE	PE
Cada funcionalidade que disponibiliza a manipulação de dados Geo por arquivos shapefile	EE	PE
Cada funcionalidade para visualização de mapas na tela (com ou sem filtros de pesquisa)	SE	PE
Cada funcionalidade de zoom no mapa para uma localização/área específica	SE	PE

11.5. CONTAGEM DE DESENVOLVIMENTO MULTIPLATAFORMA

O desenvolvimento multiplataforma tem expandido consideravelmente. Existem diversos frameworks para desenvolvimento que permitem criar produtos em multiplataformas e muitas vezes são necessários testes, homologação e implantação diferenciados para cada plataforma. Essas atividades podem exigir inclusive profissionais especializados para cada plataforma.

Diante disso, observando a abordagem de multiple instance, reconhecendo que a plataforma para entrega constitui uma característica de diferenciação na identificação da unicidade da função transacional, este roteiro sugere a utilização de fatores de impacto diferenciados para o desenvolvimento multiplataforma, levando em conta se a plataforma permite o desenvolvimento de aplicativos apenas para uma plataforma de produção (Nativa) ou se permite o desenvolvimento para mais de uma plataforma de produção (Híbrida).

Além do tipo da plataforma de desenvolvimento, os fatores de impacto são diferenciados pelo tipo da plataforma (Base e Adicional) de instalação do aplicativo:

- A plataforma Base é plataforma a de referência escolhida, podendo ser qualquer uma das quais o aplicativo será disponibilizado, para esta serão aplicados os fatores de impacto já utilizados neste roteiro de métricas;
- A plataforma adicional é qualquer outra plataforma - que não seja a Base - e terá a aplicação de fatores de impacto que busquem contemplar na contagem o esforço empreendido em atividades dos processos de Integração, Verificação, Transição e Validação, conforme disposto no item 9.1 deste roteiro.

As fórmulas descritas nos subitens seguintes, aplicam-se para os métodos APF e SFP.

11.5.1. Plataforma de desenvolvimento híbrida

11.5.1.a. Contagem de Projetos de Desenvolvimento

Para a plataforma de desenvolvimento híbrida, sugere-se a aplicação de fatores de impacto diferenciados apenas para a contagem de PF da Plataforma adicional. A seguir, apresentam-se as fórmulas a serem utilizadas para projetos de desenvolvimento em plataformas de base e adicional:



$$\begin{aligned} \text{PF_DESENV_HIBRIDO_PLAT_BASE} &= \text{PF_DESENVOLVIMENTO} \\ \text{PF_DESENV_HIBRIDO_PLAT_ADICIONAL} &= \text{PF_DESENVOLVIMENTO} * 0,31 \\ \text{PF_DESENV_HIBRIDO_TOTAL} &= \text{PF_DESENV_HIBRIDO_PLAT_BASE} + \\ &\quad \text{PF_DESENV_HIBRIDO_PLAT_ADICIONAL} \end{aligned}$$

Para a plataforma adicional, utilizou-se o fator de impacto de 0,31%, que corresponde ao esforço empreendido em atividades dos processos de Integração, Verificação, Transição e Validação, conforme tabela 15 (item 9.1) deste roteiro.

Convém destacar que:

- As fórmulas descritas acima aplicam-se para os métodos APF e SFP.
- Devem ser contabilizadas apenas as funções transacionais na contagem das plataformas adicionais. As funções de dados não devem ser contabilizadas para plataforma adicional, considerando que a estrutura dos Arquivos Lógicos é a mesma para as duas plataformas (base e adicional).

11.5.1.b. Contagem de Projetos de Melhoria

Para projetos de melhoria em plataformas de desenvolvimento híbrida, sugere-se também a aplicação de fatores de impacto diferenciados apenas para a contagem de PF da plataforma de desenvolvimento adicional:



$$\begin{aligned} \text{PF_MELHORIA_HIBRIDO_PLAT_BASE} &= \text{PF_MELHORIA} \\ \text{PF_MELHORIA_HIBRIDO_PLAT_ADICIONAL} &= \text{PF_MELHORIA} * 0,31 \\ \text{PF_MELHORIA_HIBRIDO_TOTAL} &= \text{PF_MELHORIA_HIBRIDO_PLAT_BASE} + \\ &\quad \text{PF_MELHORIA_HIBRIDO_PLAT_ADICIONAL} \end{aligned}$$

As fórmulas descritas acima aplicam-se aos métodos APF e SFP.

11.5.2. Plataforma de desenvolvimento nativa

11.5.2.a. Contagem de Projetos de Desenvolvimento

Para a plataforma de desenvolvimento nativa, sugere-se também a aplicação de fator de impacto diferenciado, apenas para a contagem de PF da Plataforma adicional.

Para a fórmula de contagem da plataforma de desenvolvimento nativa adicional, utilizou-se um fator de impacto de 0,39%, que corresponde ao esforço empreendido nos processos de Implementação, Integração, Verificação, Transição e Validação, conforme tabela 15 (item 9.1) deste roteiro.



$$\begin{aligned} \text{PF_DESENV_NATIVO_PLAT_BASE} &= \text{PF_DESENVOLVIMENTO} \\ \text{PF_DESENV_NATIVO_PLAT_ADICIONAL} &= \text{PF_DESENVOLVIMENTO} * 0,39 \\ \text{PF_DESENV_NATIVO_TOTAL} &= \text{PF_DESENV_PLAT_NATIVO_BASE} + \\ &\quad \text{PF_DESENV_NATIVO_PLAT_ADICIONAL} \end{aligned}$$

Vale ressaltar que:

- As fórmulas descritas acima aplicam-se para os métodos APF e SFP.
- Devem ser contabilizadas apenas as funções transacionais na contagem das plataformas adicionais. As funções de dados não devem ser contabilizadas para plataforma adicional, considerando que a estrutura dos Arquivos Lógicos é a mesma para as duas plataformas (base e adicional).

11.5.2.b. Contagem de Projetos de Melhoria

Para projetos de melhoria em plataformas de desenvolvimento nativo, sugere-se também a aplicação de fator de impacto diferenciado apenas para a contagem de PF da plataforma de desenvolvimento adicional:



$$\begin{aligned} \text{PF_MELHORIA_NATIVO_PLAT_BASE} &= \text{PF_MELHORIA} \\ \text{PF_MELHORIA_NATIVO_PLAT_ADICIONAL} &= \text{PF_MELHORIA} * 0,39 \\ \text{PF_MELHORIA_NATIVO_TOTAL} &= \text{PF_MELHORIA_NATIVO_PLAT_BASE} + \\ &\quad \text{PF_MELHORIA_NATIVO_PLAT_ADICIONAL} \end{aligned}$$

As fórmulas descritas acima aplicam-se para os métodos APF e SFP.

VERSÃO CONSULTA

12. CONTAGEM PF DE PROJETOS DE DW

Esta sessão orienta a contagem de projetos de desenvolvimento e manutenção de Data Warehouse (DW), contemplando os componentes e requisitos de uma arquitetura de DW e suas ferramentas, considerando não apenas o projeto de desenvolvimento, mas também a estimativa de tamanho, o projeto de melhoria e um conjunto de atividades de manutenção em DW que, nas contratações, devem ser dimensionados por uma métrica de software.

Essa sessão foi elaborada a partir da primeira versão do Guia de Contagem de Pontos de Função do SISP para Projetos Data Warehouse, que se baseou:

- no estudo e análise das publicações IFPUG (2007) e NESMA (2014) que abordam a contagem em Pontos de Função de projetos de DW; e
- em guias de contagem de projetos de DW já disponíveis em instituições públicas, como os do SERPRO (2014), DATAPREV (2014), INFRAERO (2011), STN (2013) e INCRA (2013).

Adicionalmente, este roteiro apresentará orientações para contagem utilizando o método SFP (Simple Function Point / Ponto de Função Simplificado) com o fim de trazer agilidade e previsibilidade para o processo de contagem de pontos de função para Projetos de DW. Como já explanado neste documento, o método SFP é um método de medição funcional compatível e consistente com o IFPUG APF, descrito no CPM v.4.3.1, em terminologia e definições.

12.1. OBJETIVO

Este roteiro apresenta diretrizes e regras de contagem de projetos de DW utilizando os métodos APF e SFP, com o fim de apoiar os órgãos integrantes do Sistema de Administração dos Recursos de Tecnologia da Informação (SISP) na contratação e dimensionamento de serviços de desenvolvimento e manutenção de projetos de DW.

Em projetos de DW, a modelagem da base de dados (modelo multidimensional) tem um papel fundamental e exerce grande influência no desenvolvimento das funcionalidades que buscam atender aos requisitos do usuário em relação às consultas e relatórios, à análise de grandes volumes de dados e à obtenção de informações estratégicas para a tomada de decisão. Além disso, o processo de contagem de projetos de DW, usando a métrica de Pontos de Função, baseia-se fortemente no modelo de dados multidimensional do DW, considerando as tabelas Fato e Dimensão definidas nesse modelo.

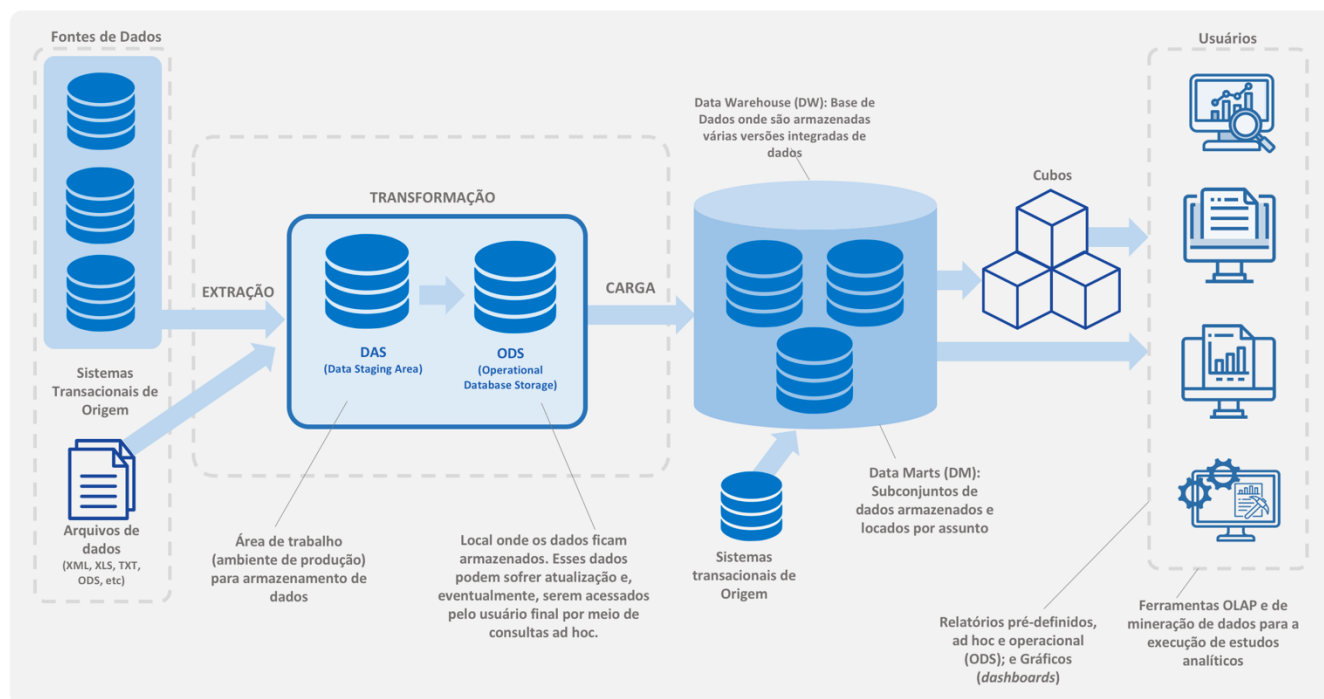
Em um projeto de DW observa-se que o esforço de desenvolvimento para as cargas de dados através do processo de ETL (Extração, Transformação e Carga) é maior (e mais complexo) do que o esforço para realização de outros serviços de desenvolvimento em um projeto de DW, como por exemplo, destaca-se a geração de consultas e relatórios no OLAP (*On-line Analytical Processing*), além da especificidade de profissionais de mercado que atendem essas demandas.

12.2. MODELO DE ARQUITETURA DE REFERÊNCIA DE DATA WAREHOUSE/DATA MART

A figura a seguir representa o modelo de arquitetura de referência de projeto de DW dividido em quatro camadas com seus principais componentes. Esses componentes constituem os elementos a serem considerados na contagem de PF de projeto de DW.

Existem duas abordagens de desenvolvimento de um DW: Kimball e Inmon. Segundo NESMA (2014), as diretrizes de contagem são aplicáveis tanto no modelo Kimball quanto no Inmon. Inclusive, observa-se muito mais discussão e trabalhos sobre a medição funcional de Fatos e Dimensões em um Esquema Estrela (abordagem Kimball) do que sobre a medição de *data warehouses* relacionais (abordagem Inmon). Além disso, no desenvolvimento de um DW usando a filosofia de Inmon, as orientações sobre a medição de Dimensões e Fatos são igualmente relevantes e aplicadas quando ocorre a criação de data marts (numa fase posterior ao DW). Este roteiro adota a filosofia de Kimball.

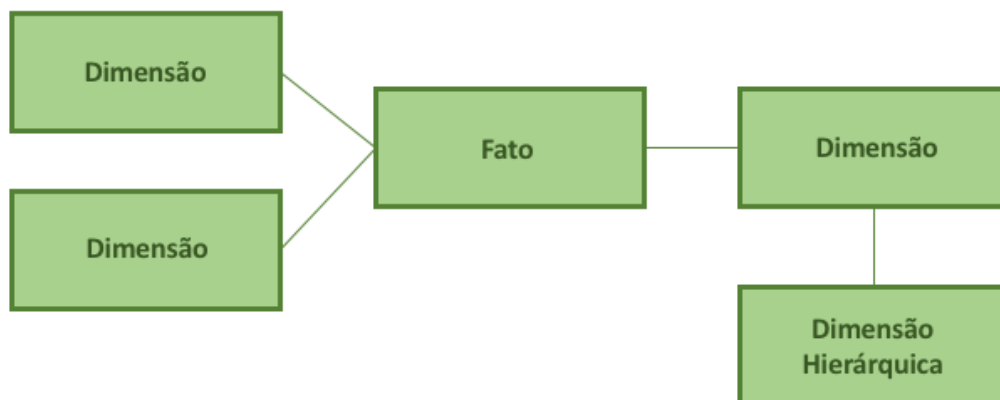
Figura 10: Modelo de arquitetura de referência de projeto de DW / DM



As camadas e componentes da arquitetura de referência de projetos DW/DM são:

- **(1) Camada de Origem de Dados:** esta camada representa as fontes de dados que alimentam a base de dados do DW. Seu principal componente são os Sistemas Transacionais de Origem que representam todas as aplicações que fornecem dados para o DW, além de fontes como arquivo de dados nos formatos XML, XLS, TXT, ODS entre outros. Esta camada é usada, ainda, para a leitura de dados de sistemas de origem para a validação de dados no DW e para a geração de cubos.
- **(2) Camada de Dados:** esta camada é subdividida em três subcamadas: *Data Staging Area (DAS)*, *Data Warehouse (DW)* e *Data Marts (DM)*, responsáveis por armazenar os dados dos sistemas de origem e disponibilizá-los para consultas e relatórios. O processo de disponibilização dos dados para consultas e relatórios envolve, principalmente, alguns componentes adicionais como o ETL (Extração, Transformação e Carga) e ODS (*Operational Database Storage*).
 - **Data Staging Area (DSA):** representa uma base de dados temporária que armazena os dados extraídos dos sistemas transacionais de origem para realizar as transformações necessárias antes de carregar esses dados nas tabelas do DW. Em outras palavras, os dados são importados da aplicação de origem para a DSA e então, em outro processo de integração, os dados da DSA são importados para as tabelas Fato e Dimensão do DW.
 - **Operational Database Storage (ODS):** similar ao componente DSA descrito acima, o ODS é, também, uma base de dados que armazena os dados extraídos dos sistemas transacionais de origem antes de carregar esses dados nas tabelas do DW. Entretanto, os dados da ODS podem sofrer atualização e, eventualmente, podem ser acessados pelo usuário final por meio de consultas ad hoc.
 - **Extração, Transformação e Carga (ETL):** é o processo que compreende as três etapas (extração, transformação e carga de dados) para a disponibilização dos dados no DW a partir dos sistemas transacionais de origem. Os diferentes cenários de extração de dados dos sistemas transacionais de origem são apresentados e discutidos no **item 12.7.3** (Contagem de funções de extração de dados do Sistema Transacional de Origem).
 - **Data Warehouse (DW):** representa a base de dados que armazena várias versões integradas dos dados recebidos de diferentes fontes de origem, após o processo de ETL. Em um modelo de dados multidimensional do tipo Esquema Estrela, são reconhecidos dois tipos de entidades: Tabelas Fato e Tabelas Dimensão. Sendo que as tabelas Dimensão podem ser classificadas em três tipos: Dimensão, Dimensão Estática e Dimensão Hierarquia. A Figura a seguir apresenta um modelo de dados multidimensional do tipo Esquema Estrela de um DW:

Figura 11: Entidades de um Esquema Estrela de Data Warehouse



- **Tabela Fato:** principal tabela em um modelo multidimensional. Sua finalidade é armazenar medidas do negócio. Cada tabela Fato pode ou não se conectar com outras tabelas Dimensão ou outras tabelas Fato;
- **Tabela Dimensão:** as tabelas Dimensão detalham as informações contidas numa tabela Fato. Elas descrevem os fatos. Uma tabela Dimensão pode se conectar a mais de uma tabela Fato;
- **Tabela Dimensão Estática:** as tabelas Dimensão Estática são tabelas criadas dentro do contexto do DW que detalham as informações contidas em uma tabela Fato mas que não são extraídas a partir de fonte de dados original. As tabelas Dimensão Estática podem ser carregadas manualmente ou por meio de um processo computacional (procedure). São exemplos de Dimensão Estática: códigos de estado e dimensão data e hora. A tabela Dimensão Estática implementa o conceito de Dados de Código (IFPUG, 2010);
- **Tabela Dimensão Hierárquica:** É uma tabela Dimensão que não se relaciona diretamente com uma Tabela Fato. A tabela Dimensão Hierárquica é ligada a outra tabela Dimensão e representa um conjunto de dados que descrevem essa Dimensão;
- **Tabela Dimensão Compartilhada:** é uma tabela Dimensão que atende a várias tabelas Fatos ou vários DW/DM e é usada para a validação de dados no processo ETL e para a geração de Cubos. As dimensões compartilhadas podem ser mantidas por outro DW ou por uma equipe específica. Devem ser atualizadas constantemente para atender as necessidades de todos os temas que as utilizam. A Dimensão Compartilhada é conhecida, também, como Dimensão Global ou Corporativa;

- **Tabela de Agregação:** a característica dos DW é possuir uma grande quantidade de dados em grande nível de detalhe. Quando o tempo de resposta na recuperação de dados no DW trazem impacto ao negócio, aumentando o tempo de resposta das consultas, são utilizadas tabelas sumarizadas. Elas agregam dados armazenados no DW em níveis mais elevados gerando melhor performance às consultas.
- **Data Marts (DM):** representa um subconjunto de dados armazenados e locados por assunto. Usado para agrupar os dados por departamento ou área de negócio de uma organização. Em um DM podem ocorrer agregações entre tabelas Fato e/ou Dimensão;
- **Cubos:** a geração do Cubo, também, conhecido como Universo ou Contexto de Análise, representa a disponibilização das possíveis combinações, totalizações e medidas estatísticas básicas dos dados do DW em uma nova tabela, para acesso pela ferramenta OLAP, atendendo aos requisitos de negócio. De uma forma geral, Kimball e Ross (2002) definem como sendo o nome de uma estrutura dimensional em uma plataforma de banco de dados de processamento analítico on-line (OLAP) ou multidimensional, originalmente referindo-se ao caso simples de três dimensões: produto, mercado e hora. Para complementar o entendimento, ainda em Kimball e Ross (2002), é dito que a modelagem dimensional se aplica a bancos de dados relacionais e multidimensionais. Embora ambos possuam um design lógico comum com dimensões que podem ser reconhecidas, a implementação física é diferente.
- **(3) Camada de Aplicação:** contém os relatórios pré-definidos pelo usuário e disponíveis para consulta dos dados armazenados no DW/DM.
- **(4) Camada de Apresentação:** é a interface do usuário final com o projeto de DW e, onde estão disponíveis várias funcionalidades como relatórios pré-definidos, relatórios ad hoc, relatórios operacionais (ODS) e mineração de dados.

12.3. ESTIMATIVA DE PROJETO DE DW

A contagem estimada de um projeto de DW consiste em identificar as funções de dados e funções transacionais a partir dos requisitos iniciais do sistema. Assim sendo, além do documento de requisitos funcionais do sistema, o esboço do modelo de dados multidimensional é um artefato importante a ser considerado para a estimativa do projeto de DW, pois ajudará na estimativa dos arquivos lógicos do projeto (tabelas Fato e Dimensão do modelo).

O Guia de Contagem de Pontos de Função do SISP para Projetos Data Warehouse 1.0 utilizou o método de Contagem Estimada de Pontos de Função da NESMA (2004) para a contagem estimada de projetos de DW, aplicando-se a complexidade baixa para as funções de dados e a complexidade média para as funções transacionais.

Este roteiro recomenda o uso do método SFP (Simple Function Point /Ponto de Função Simplificado) para a contagem estimada de projetos DW, com o fim de trazer agilidade e previsibilidade para a estimativa.

- O processo de contagem de pontos de função utilizando o método SFP (Simple Function Point) está descrito no item 5.1 (Processo de Contagem do Método SFP) deste roteiro; e
- As orientações para realizar a contagem estimada com o método SFP estão descritas no item 8.2 (Estimativa de Tamanho com SFP) deste roteiro.

Convém relembrar que o método SFP tem correspondência de conceitos, tipos de medição, fórmulas de cálculo do método APF tradicional, o que leva a identificação das mesmas funções a serem medidas.

A definição de um processo elementar e dos arquivos lógicos é a mesma para os métodos APF e SFP. Cada função transacional corresponde a um (1) Processo elementar no SFP e cada função de dados corresponde a um (1) arquivo lógico no método SFP. O dimensionamento com o método SFP requer apenas a identificação de dois componentes funcionais básicos: Processo Elementar (PE) e Arquivo Lógico (AL).

Para o método IFPUG SFP não são diferenciados os tipos de processo elementar e os tipos de arquivo lógico, tampouco as suas complexidades, não é necessária a identificação de DERs, ALRs ou RLRs, tampouco a identificação da “intenção primária”. A pontuação funcional é única para cada arquivo lógico e cada processo elementar identificado:

Tabela 8: Pontuação dos Componentes funcionais básicos do método IFPUG SFP

Componentes funcionais básicos do IFPUG SFP	Pontuação
Arquivo Lógico (AL)	7,0 SFP
Processo Elementar (PE)	4,6 SFP

A Tabela a seguir apresenta um resumo do que deve ser medido como função de dados e função transacional em um projeto de DW, utilizando o método SFP:

Tabela 32: Resumo da contagem estimada de projeto de DW, utilizando o método SFP

Componentes funcionais básicos do IFPUG SFP	Contribuição Funcional	Contagem no DW com SFP
Arquivo Lógico (AL)	7 SFP	1 AL para cada Tabela Dimensão
		1 AL para cada Tabela Fato
		1 AL para cada tabela do sistema de origem usada na validação de dados do processo de ETL
		1 AL para cada tabela Dimensão Compartilhada usada na validação de dados do processo ETL ou geração do Cubo
Processo Elementar (PE)	4,6 SFP	1 PE para a carga de dados em cada AL de tabela Dimensão
		1 PE para a carga de dados em cada AL de tabela Fato
		1 PE para cada Cubo (ou Universo ou Contexto de Análise)
		1 PE para cada gráfico do <i>dashboard</i> solicitado
		1 PE para cada relatório solicitado (não <i>ad hoc</i>)
		1 PE para extração de dados de cada sistema de origem (quando houver requisito de desenvolvimento). *Se não houver requisito relacionado, não deve ser contabilizado na contagem.

12.4. VISÃO GERAL DA CONTAGEM DE PROJETO DW

Este item descreve a visão geral da contagem de um processo de desenvolvimento e melhoria de DW e os itens seguintes descrevem parte do processo de contagem de PF de um projeto DW/DM, utilizando os métodos APF e SFP (*Simple Function Point*):

- a definição do escopo da contagem e da fronteira de aplicação, descrita no subitem 12.5;
- a contagem de funções de dados: descrita no subitem 12.6.
- a contagem de funções transacionais: descrita no subitem 12.7.

Destaca-se novamente que os dois métodos (APF e SFP) levam a identificação das mesmas funções de dados e das mesmas funções transacionais, entretanto no método SFP não é feita a diferenciação entre os arquivos lógicos internos (ALI, consultados e que armazenam dados) e os arquivos de interface externa (AIE, usados apenas para consulta de dados), nem a diferenciação de funções transacionais (EE, CE e SE). O processo de contagem para o método SFP está descrito no **item 5** deste roteiro:

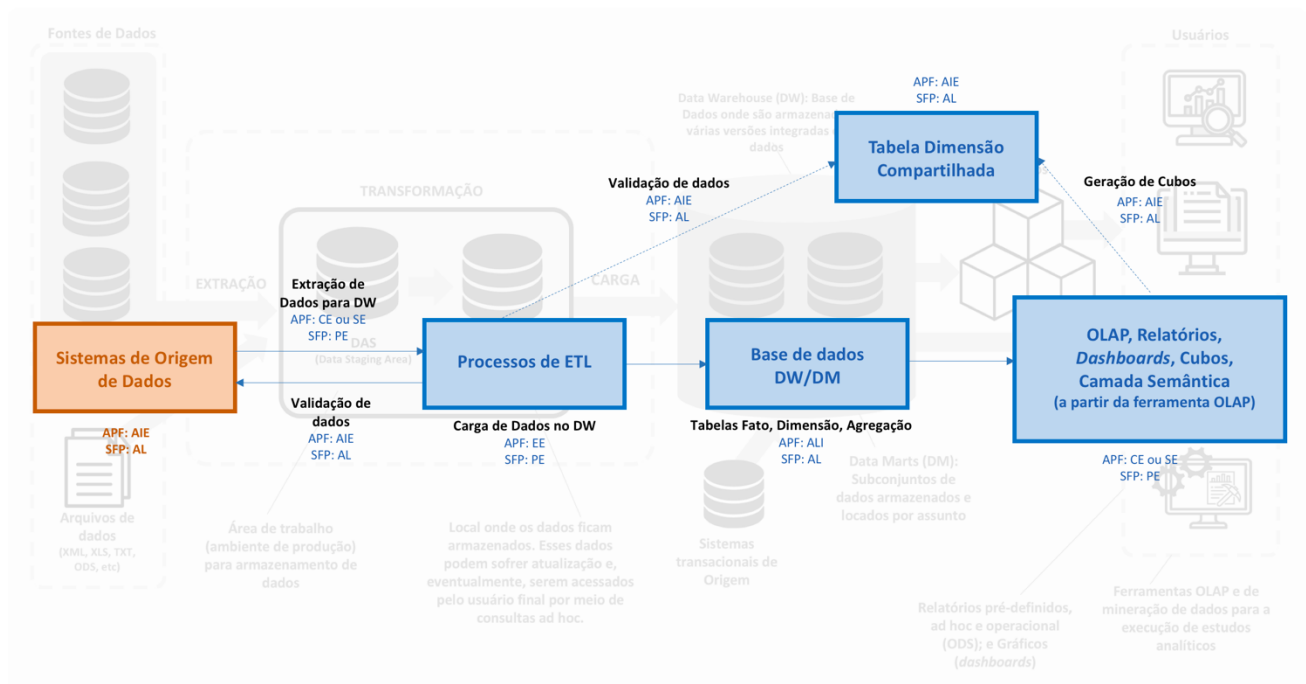
Tabela 33: Diferença na identificação das funções de dados entre os métodos SFP e APF

O método SFP identifica:		O método APF identifica:
Funções de Dados	Arquivo Lógico (AL)	Arquivo Lógico Interno (ALI)
		Arquivo de Interface Externa (AIE)
Funções transacionais	Processo Elementar (PE)	Entrada Externa (EE)
		Consulta Externa (CE)
		Saída Externa (SE)

A figura a seguir mostra uma visão geral da contagem de PF em um projeto de desenvolvimento de DW/DM, contemplando os componentes funcionais básicos dos métodos SFP e APF e utilizando como plano de fundo as camadas

definidas no modelo de arquitetura de referência de projeto de DW / DM (camadas de origem, dados, aplicação e apresentação) descritas no subitem 12.2:

Figura 12: Visão geral da contagem de PF com os métodos APF e SFP em um projeto de DW/DM



12.5. ESCOPO DA CONTAGEM E FRONTEIRA DA APLICAÇÃO

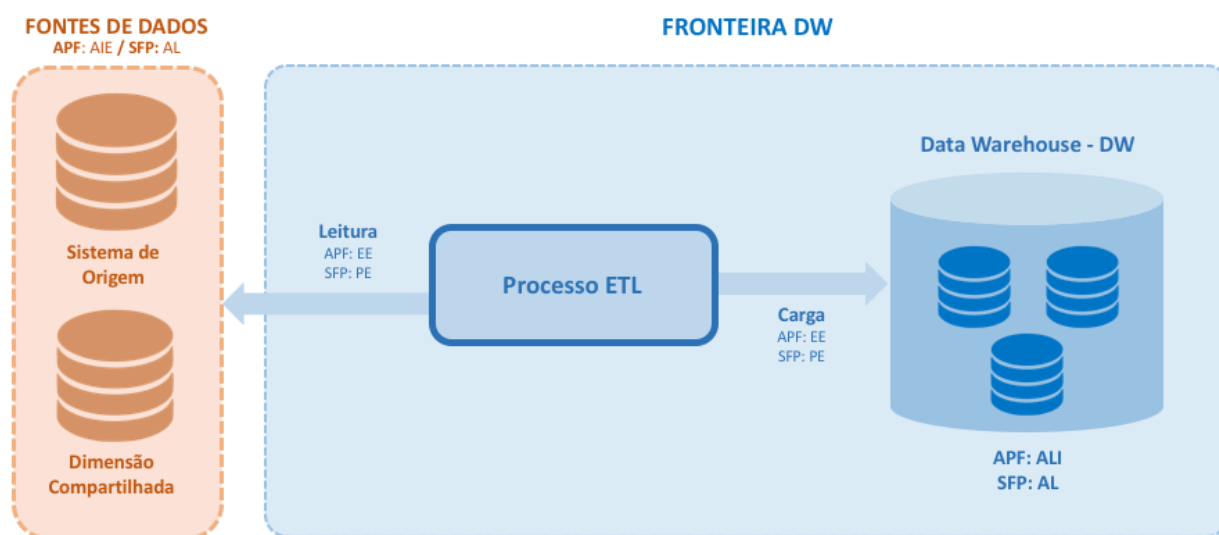
O escopo da contagem define quais são as funcionalidades objeto de determinada contagem. O escopo da contagem é determinado pelo propósito da contagem e identifica os sistemas, as aplicações ou seus componentes que serão dimensionados. Um escopo de contagem pode conter mais de uma aplicação.

A definição da fronteira de cada aplicação contida no escopo da contagem é baseada na visão de negócio do usuário e não em considerações técnicas de projeto.

A contagem de Pontos de Função é realizada separadamente, considerando cada fronteira de aplicação. No contexto de contagem de PF de *Data Warehouse* (DW), o escopo da contagem abrange, normalmente, o Projeto de Desenvolvimento ou Melhoria da fronteira do DW e o Projeto de Melhoria realizado nas aplicações de origem para que estas forneçam informações para o DW.

Assim, o *Data Warehouse* (DW) é definido como uma fronteira de aplicação distinta dos Sistemas de Origem, sendo necessária a contagem separadamente destas fronteiras.

Figura 13: Visão da Fronteira DW



12.6. CONTAGEM DE FUNÇÕES DE DADOS

Os subitens a seguir apoiam na identificação das funções de dados em um projeto de DW, utilizando os métodos APF e SFP.

12.6.1. Contagem funções de dados internas do DW

Em um modelo de dados multidimensional (Esquema Estrela) são reconhecidos dois tipos de entidades: tabelas Fato e tabelas Dimensão.

Dessa forma, as tabelas Fato e Dimensão são contadas conforme a seguir:

Tabela 34: Entidades reconhecidas em um modelo de dados multidimensional

		O método SFP identifica:	O método APF identifica:
Tabelas Fato	Mantida por 1 ou mais processos ETL	1 Arquivo Lógico (AL)	1 Arquivo Lógico Interno (ALI)
Tabelas Dimensão	Mantida por 1 ou mais processos ETL	1 Arquivo Lógico (AL)	1 Arquivo Lógico Interno (ALI)
Tabelas Dimensão Estática	Se assemelha ao conceito de tabelas <i>Code Data</i> do CPM, armazenando dados estáticos	não é contabilizado	não é contabilizado
Tabelas Dimensão Hierárquica	<p>Ocorrem quando algumas tabelas Dimensão não estão ligadas diretamente na tabela Fato. São considerados níveis hierárquicos de uma outra tabela Dimensão.</p> <p>* Na Figura 14, o “Histórico Estado Civil” é um exemplo de dimensão hierárquica.</p>	Não é contabilizado, visto que para o método SFP, na contagem de arquivos lógicos, não é necessária a identificação de DERs e RLRs.	<p>Contar 1 Registro Lógico para cada Dimensão Hierárquica ligada.</p> <p>Caso não existam níveis hierárquicos ou subgrupos de dados dentro da Dimensão, deve-se considerar apenas 1 Registro Lógico.</p>

Podem existir também as **tabelas de agregação** geradas a partir da combinação de tabelas Fato e Dimensão e com o objetivo de apoiar a geração de consultas aos dados do DW:

- Para o método APF: podem ser contadas como 1 ALI e sua carga de dados como 1 EE;
- Para o método SFP: podem ser contadas como 1 AL e sua carga de dados como 1 PE.

Figura 14: Exemplo de modelo de dados multidimensional – Esquema Estrela

Tabela 35: Resumo da contagem de tabelas de agregação para os métodos SFP e APF

		O método SFP identifica:	O método APF identifica:
Tabelas de agregação	Tabelas geradas a partir da combinação entre tabelas do DW com o objetivo de apoiar a geração de consultas aos dados do DW, sem objetivo de desempenho.	Tabelas de Agregação são contadas como 1 AL e sua carga de dados como 1 PE.	Tabelas de Agregação são contadas como 1 ALI e sua carga de dados como 1 EE

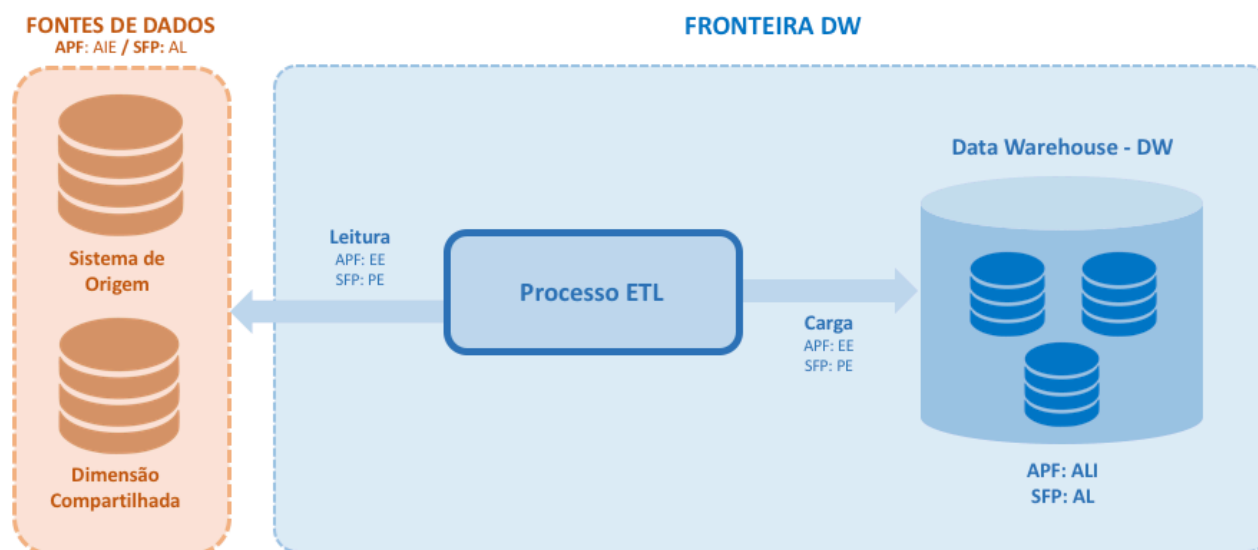
12.6.2. Contagem de funções de dados do processo ETL

No processo de ETL, a validação de dados para a carga na tabela do DW pode requerer uma consulta à base de dados das aplicações de origem ou às tabelas Dimensão Compartilhada.

Observe que, nesses casos não há transferência de dados, e conta-se apenas a função transacional referente ao processo de ETL, onde a validação de dados é parte desse processo. Assim, conta-se uma função de dados (AL/AIE) para cada tabela de dados consultada do sistema de origem ou da tabela Dimensão Compartilhada para a validação de dados do processo de ETL.

As tabelas Dimensões Compartilhadas são consideradas funções de dados (AL/ALI) de outro DW que as mantém e, são contadas como função de dados (AL/AIE) no DW que as utiliza para validação de dados no processo ETL ou na geração de um Cubo, conforme imagem a seguir:

Figura 15: Visão da Fronteira DW



A seguir, apresenta-se um resumo da contagem de funções de dados em um processo ETL, considerando os métodos SFP e APF:

Tabela 36: Resumo da contagem de funções de dados em um processo ETL

		O método SFP identifica:	O método APF identifica:
Sistema de Origem	Para cada arquivo lógico do sistema de origem, usado na validação de dados do processo de ETL	1 AL	1 AIE
Tabela dimensão compartilhada	Para cada tabela dimensão compartilhada, usada na validação de dados do processo ETL ou na geração de cubos	1 AL para cada tabela Dimensão compartilhada	1 AIE para cada tabela dimensão compartilhada

12.6.3. Considerações finais sobre funções de dados em projetos DW

É importante observar alguns cenários (pontos de atenção) na contagem das funções de dados em projetos DW. São eles:

- Não conte o mesmo grupo de dados mais de uma vez. Por exemplo, contar o mesmo grupo de dados como ALI e AIE, usando o método APF;
- Não conte duas vezes o mesmo grupo de dados, por exemplo uma tabela Dimensão que esteja modelada em duas Estrelas do DW;
- As hierarquias das tabelas Dimensão devem ser contadas no método APF como Registros Lógicos e não como um arquivo lógico interno (ALI). Por exemplo: Departamento e Setor.

12.7. CONTAGEM DE FUNÇÕES TRANSACIONAIS

Nesta seção apresentamos como identificar as funções transacionais de um projeto de DW.

Ressalta-se, novamente, que este roteiro apresenta a contagem de projetos de DW utilizando os métodos APF e SFP (Simple Function Point).

Como já foi dito em diversas partes desse roteiro, os dois métodos (APF e SFP) levam a identificação das mesmas funções de dados e das mesmas funções transacionais, contudo, no método SFP não é feita a diferenciação de funções transacionais.

Para o método SFP, cada função transacional corresponde a um (1) processo elementar com pontuação igual a 4,6 SFP. Para o método IFPUG SFP não são diferenciados os tipos de processo elementar e os tipos de arquivo lógico e tampouco as suas complexidades; não é necessária a identificação de DERs, ALRs ou RLRs, tampouco a identificação da “intenção primária”.

12.7.1. Contagem de funções de carga de dados

Em projetos de Data Warehouse, o processo de ETL é considerado um processo elementar único e representa a extração, transformação e carga de dados (IFPUG, 2007) nas tabelas Fato e Dimensão de um modelo multidimensional.

O processo de ETL será denominado, neste guia, de funcionalidade de carga de dados de uma tabela do DW. A carga de dados periódica em uma tabela do DW pode ser do tipo carga completa (ou carga *full*) ou carga incremental (ou carga delta):

- A carga completa representa a exclusão dos dados da tabela para a inserção dos dados da extração;
- A carga incremental representa a atualização dos dados da tabela a partir dos dados da extração.

Frequentemente, tem-se apenas uma funcionalidade de carga de dados requisitada para cada tabela Fato e Dimensão do DW, podendo ser do tipo carga completa ou carga incremental. Entretanto, se houver requisito de negócio para o desenvolvimento das duas cargas (completa e incremental) para uma mesma tabela do DW, então, neste caso, contam-se duas (2) funções transacionais (1 carga completa e 1 carga incremental).

Os dados de uma mesma tabela Fato ou Dimensão podem ser carregados de vários sistemas transacionais de origem. Conta-se uma função transacional para a carga de dados de cada sistema transacional de origem distinto, observando o critério de unicidade de um processo elementar.

Considere a funcionalidade de carga de dados, também, para cada Registro Lógico da tabela Dimensão, onde os processos de manter os dados do Registro Lógico e da respectiva tabela Dimensão, são distintos e independentes.

Segundo NESMA (2014), geralmente, não existe funcionalidade de alteração de dados em projetos de DW e a exclusão de dados como requisito funcional, raramente ocorre. Entretanto, quando ocorrer, o requisito de usuário para a exclusão de dados pode ser tratado como uma funcionalidade de limpeza de dados, conforme descrito no item 12.9.8 - Exclusão de Dados Antigos do DW.

As cargas de dados em tabelas de Agregação do DW, conforme descrito 12.6.1 (Contagem de Funções de Dados do DW), também são contadas como uma função transacional de carga.

A seguir, um resumo, aplicando a contagem de funções transacionais de carga para os métodos SFP e APF:

Tabela 37: Resumo da contagem de funções de carga

	O método SFP identifica:	O método APF identifica:
Para cada Tabela Fato identificada como função de dados	1 PE	1 EE
Para cada Tabela Dimensão identificada como função de dados (exceto Dimensão Estática)	1 PE	1 EE
Para cada Tabela Agregação identificada como função de dados conforme item 12.6	1 PE	1 EE

Para cada Sistema de Origem, quando identificado o critério de unicidade de processo elementar (IFPUG, 2010) para a carga em cada tabela Fato ou Dimensão	1 PE	1 EE
Para cada Registro Lógico (ou Dimensão Hierárquica) da tabela Dimensão (mantido independentemente da Dimensão)	1 PE	1 EE

12.7.2. Contagem de funções de geração de relatórios ou gráficos

Em aplicações de *Data Warehouse* existem funcionalidades de geração de relatórios (ou consulta de relatórios pré-definidos) ou gráficos (painéis, *dashboards*) usando as ferramentas OLAP.

Cada relatório ou gráfico requisitado pelo usuário e implementado pela equipe de desenvolvimento são contados como funções transacionais.

Para o método APF, quando a geração do relatório não possuir lógicas de processamento de cálculos ou criação de dados derivados, são funções transacionais do tipo Consulta Externa (CE) devendo-se, ainda, verificar se atendem os critérios de determinação da unicidade de processo elementar do CPM. Para o método SFP, não há distinção entre as funções transacionais, todas são contadas como PE (processo elementar).

Os relatórios gerados pelo usuário por meio da funcionalidade de consultas *ad-hoc* ou personalizadas, disponível na ferramenta OLAP, não são contados porque não constituem um requisito do usuário para a equipe de desenvolvimento.

A geração do Cubo (também denominado Universo ou Contexto de Análise) deve ser contada como uma função transacional. Esse tipo de tabela, normalmente, é utilizada para consumo por outras aplicações ou pelo próprio *Datamart*. Para o método SFP conta-se como PE (processo elementar) e para o método APF, conta-se uma SE para a geração do Cubo por cada Estrela modelada.

Além dos Cubos por Estrela (uma tabela Fato e suas respectivas Dimensões), podem existir Cubos com mais de uma Estrela. Para estes casos, as tabelas Fatos e Dimensões envolvidas em cada Estrela devem ser consideradas como arquivo lógico referenciado, bem como as tabelas Dimensões Compartilhadas, sendo respeitadas as regras descritas no item 12.6.

Tabela 38: Resumo da contagem de funcionalidades de geração de relatórios

	O método SFP identifica:	O método APF identifica:

Cada relatório ou gráfico requisitado pelo usuário e implementado pela equipe de desenvolvimento	1 PE	1 CE ou 1 SE
Para cada relatório que não possuir lógicas de processamento de cálculos ou criação de dados derivados	1 PE	1 CE
Para cada geração do Cubo por cada Estrela modelada	1 PE	1 SE

12.7.3. Contagem de funções de extração de dados do Sistema Transacional de Origem

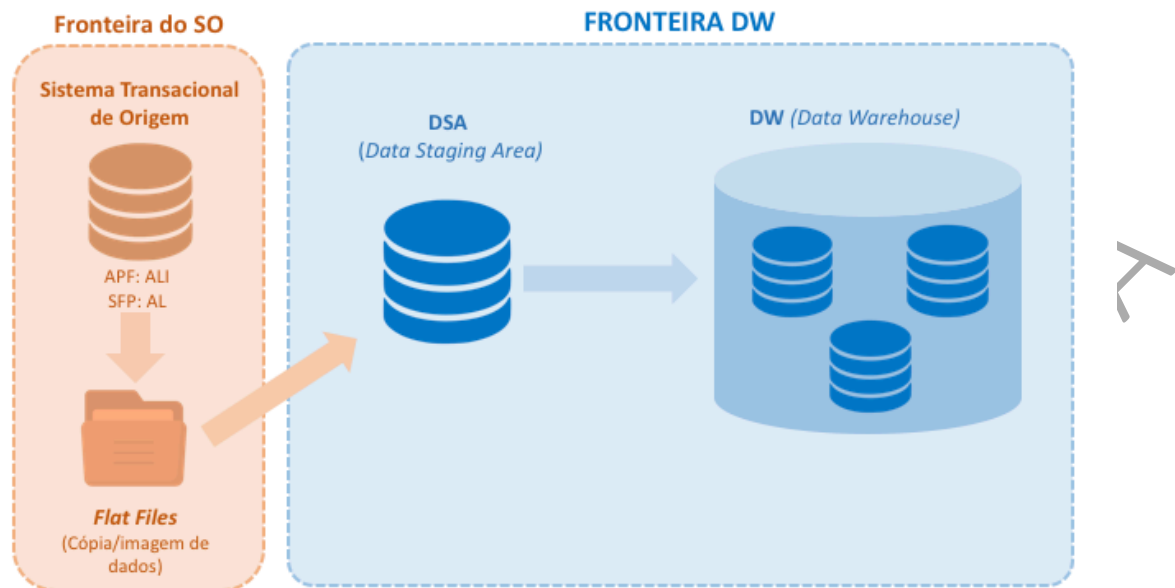
Segundo NESMA (2014), a extração de dados de um sistema transacional de origem para a carga nas tabelas do DW através do processo de ETL, acontece por dois cenários:

- **Geração de *flat files* na fronteira do sistema de origem:** Conta-se uma função transacional para a geração do *flat file* no sistema de origem. Posteriormente, no processo ETL, os dados do *flat file* são carregados no DSA para o processamento das transformações e limpeza de dados antes de serem, efetivamente, carregados nas tabelas do DW; e,
- **Interface direta entre a base de dados do sistema de origem e o DSA:** neste caso, não ocorre a transferência de dados da base do sistema de origem para o DW, mas esses dados ficam disponíveis para o DW.

Neste roteiro, a partir da análise das regras de contagem de “dados compartilhados” (IFPUG, 2010) estendemos esses cenários de extração de dados apresentados em NESMA (2014) para três cenários, conforme ilustrado nas figuras a seguir, e descritos abaixo:

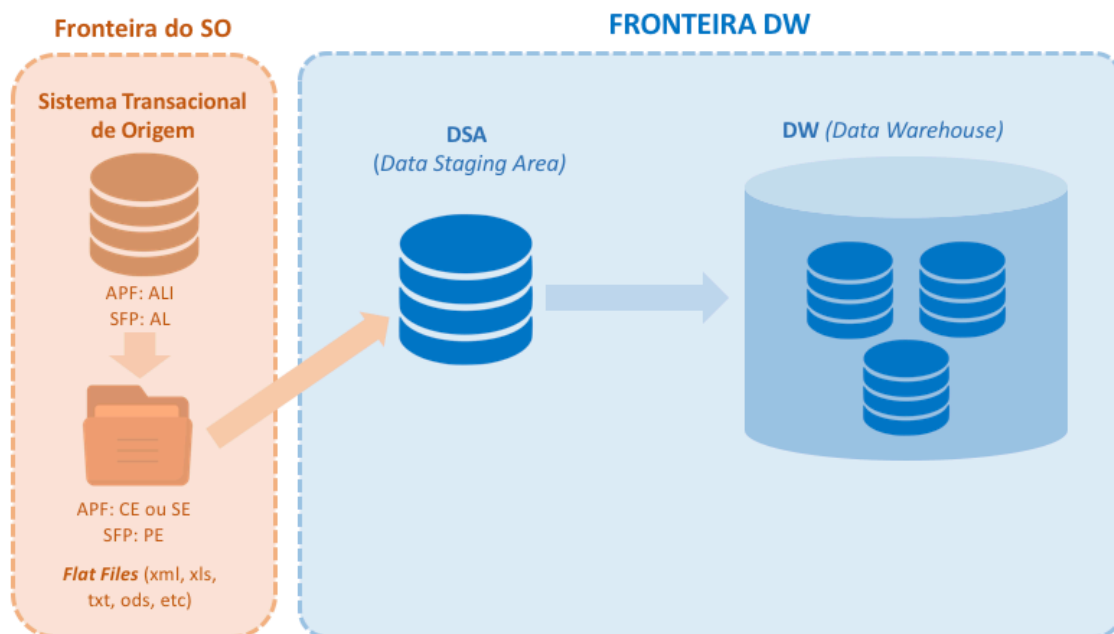
- **Cenário 1: flat files sendo uma cópia de dados de tabela do sistema de origem.** Neste caso, o *flat file* é uma cópia ou imagem de um arquivo lógico (APF: ALI/ SFP: AL) do sistema de origem, sem nenhum processamento adicional para a sua geração.
 - **Para o método APF,** considera-se apenas, no processo ETL, os dados do sistema de origem como DERs (Dado Elementar Referenciado) da função EE de carga no DW.
 - **Para o método SFP,** não há alteração da contribuição funcional do PE (processo elementar) de Carga no DW, já que o método SFP não requer a identificação de DERs e ARLs e a contribuição funcional do processo elementar é sempre 4,6 SFP.

Figura 16: Cenário 1 – *flat files* sendo uma cópia de dados de tabela do sistema de origem



- **Cenário 2: geração de *flat files* na fronteira do sistema de origem.** Ocorre quando existe uma necessidade de negócio para a transferência de dados do sistema de origem para o DW e o grupo de DER (Dado Elementar Referenciado) é diferente nos dois sistemas. Conta-se, apenas, uma função transacional (APF: CE ou SE / SFP: PE) para a geração do *flat file* no sistema de origem. Posteriormente, no processo ETL, os dados do *flat file* são carregados no DSA para o processamento das transformações e limpeza de dados antes de serem, efetivamente, carregados nas tabelas do DW,

Figura 17: Cenário 2 – geração de *flat files* na fronteira do sistema de origem

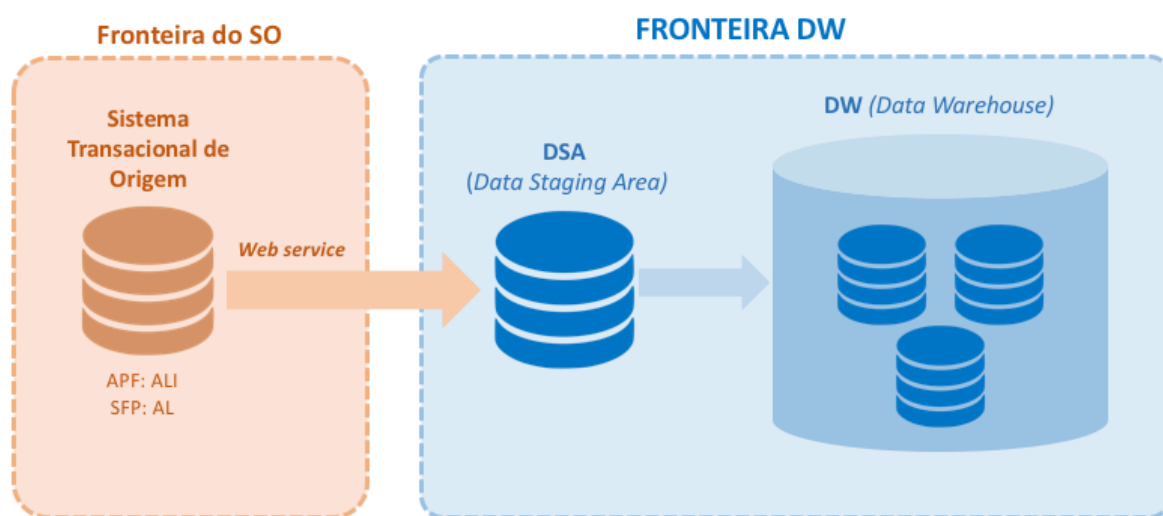


- **Cenário 3: interface direta entre a base de dados do sistema de origem e o DSA.** Neste caso, não ocorre a transferência de dados da base do sistema de origem para o DW, mas esses dados ficam disponíveis para leitura pelo DW. A consulta por *WebService* para a extração de dados dos sistemas de origem é considerada um exemplo deste Cenário 3.

Com relação à contagem de pontos de função:

- Para o método APF, considera-se o mesmo tratamento descrito para o Cenário 1, ou seja, os dados do sistema de origem disponibilizados por *WebService*, são DERs da função EE de carga no DW.
- Para o método SFP, não há alteração da contribuição funcional do PE (processo elementar) de Carga no DW, já que o método SFP não requer a identificação de DERs e ARLs e a contribuição funcional do processo elementar é sempre 4,6 SFP.

Figura 18: Cenário 3 – interface direta entre a base de dados do sistema de origem e o DSA



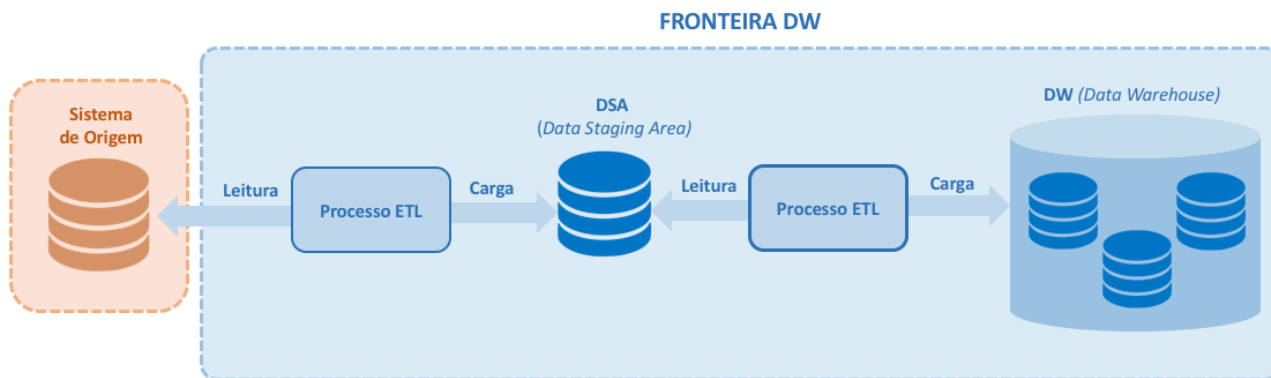
A funcionalidade de extração de dados de uma aplicação de origem, quando houver requisito de desenvolvimento, deve ser contada como uma função transacional (APF: CE ou SE/ SFP: PE) dentro da fronteira da aplicação de origem.

Consequentemente, essa funcionalidade faz parte da contagem da *baseline* da aplicação de origem. Entretanto, caso não exista contrato que suporte o desenvolvimento/manutenção do sistema de origem, então, sugere-se incluir a funcionalidade de extração de seus dados na contagem do projeto de DW.

Geralmente, os dados do DW provenientes das aplicações de origem, são armazenados em uma base de dados temporária, denominada *Data Staging Area* (DSA). Assim, os dados são extraídos do sistema de origem para a DSA e, então, em outro processo de integração, a partir da DSA, os dados são carregados para as tabelas Fato e Dimensão do DW.

Observe que, a utilização da DSA é uma solução técnica, portanto não tem contagem de Pontos de Função, conforme imagem a seguir.

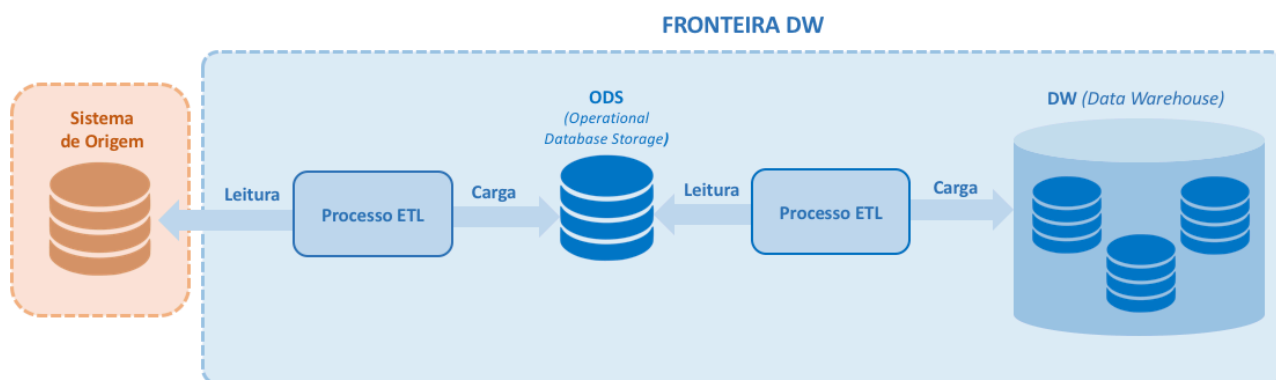
Figura 19: Carga de dados do sistema de origem na DSA



Em alguns casos, o usuário com receio de perder dados das aplicações de origem, requisita que os dados dos sistemas de origem sejam copiados para uma área de armazenamento de dados operacional (*Operational Data Store* – ODS) do DW. Nestes casos:

- **Para o método APF:** os dados são copiados do sistema transacional de origem para o ODS e esses dados serão contados como DERs na função EE de carga no ODS (Figura 20). Caso os dados copiados sejam transformados, os dados do ODS serão contados como ALI e as cargas desses dados serão contadas como EE. Entretanto, as consultas e relatórios sobre esses dados do ODS são *ad hoc* e não serão contados como CE ou SE. Posteriormente, esses dados são armazenados em um ALI do DW (tabela Fato ou tabela Dimensão).
- **Para o método SFP:** não há alteração da contribuição funcional do PE (processo elementar) de Carga no ODS. Caso os dados copiados sejam transformados, os dados do ODS serão contados como AL (arquivo lógico) e as cargas desses dados serão contadas como PE (processo elementar). Já as consultas e relatórios sobre esses dados do ODS são *ad hoc* e não serão contados como PE (processo elementar). Posteriormente, esses dados são armazenados em um AL (arquivo lógico) do DW (tabela Fato ou tabela Dimensão).

Figura 20: Carga de dados do sistema de origem no ODS



A seguir, um resumo das contagens de funções de extração de dados do Sistema Transacional de Origem com os métodos SFP e APF:

Tabela 39: Resumo da contagem de funções de extração de dados do Sistema de Origem

	O método SFP identifica:	O método APF identifica:
Para extração de dados da fronteira do sistema de origem, se houver desenvolvimento da funcionalidade (Cenário 2)	1 PE	1 CE ou 1 SE
Para cada relatório ou gráfico (painel, dashboard) requisitado pelo usuário e implementado pela equipe de desenvolvimento;	1 PE	1 CE ou 1 SE
Para cada geração do Cubo	1 PE	1 SE

12.8. PROJETO DE MELHORIA DE DW

Um projeto de melhoria consiste em demandas de inclusão, alteração e exclusão de funcionalidades (grupos de dados ou processos elementares) em uma aplicação de DW implantada em produção.

Este roteiro de contagem de pontos de função para projetos de DW 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).

O projeto de melhoria de DW deve ser medido conforme o item 6.3 (PROJETO DE MELHORIA) deste roteiro, que apresenta a contagem de PF utilizando os métodos SFP e APF.

12.9. DEMANDAS TÍPICAS EM PROJETOS DE DW

Neste item apresentam-se algumas demandas (serviços) relacionadas ao projeto de manutenção de DW para as quais pode ser utilizada a contagem de PF como forma de medida para a remuneração do esforço envolvido na sua execução, principalmente, no cenário de contratação.

Para as demais demandas que não estão contempladas na contagem de projetos de desenvolvimento e de melhoria, propriamente ditos, nem neste item (Demandas Típicas em Projetos de DW), deve-se utilizar o item 6. (CÁLCULO DE PONTOS DE FUNÇÃO PARA O SISP) deste roteiro, que define a contagem de PF, utilizando os métodos APF e SFP, para diversos tipos de manutenção: corretiva, cosmética, adaptação em funcionalidades sem alteração de requisitos funcionais, apuração especial, atualização de base de dados, atualização de versão, etc.

12.9.1. Alteração de Dados de Dimensões Estáticas

Em projetos de manutenção, a inclusão ou alteração de dados nas tabelas do tipo Dimensão Estática representam, normalmente, demandas de alteração da descrição associada ao Código na tabela Dimensão Estática. Por exemplo: 1-homem, 2-mulher. Alterar a descrição dos códigos 1 e 2 para 1-masculino, 2-feminino.

Para esses casos, faz-se analogia ao item 6.12.1 Apuração Especial – Base de Dados, deste roteiro de métricas, utilizando a fórmula a seguir, igual para os dois métodos:



PF_DIMENSÃO ESTÁTICA_BD = PF_INCLUÍDO x 0,39 por Dimensão alterada

SFP_DIMENSÃO ESTÁTICA_BD = SFP_INCLUÍDO x 0,39 por Dimensão alterada

12.9.2. Criação/Alteração de Campos em Tabelas Fato e Dimensão

É importante ressaltar que caso seja solicitada a criação de novos campos em tabelas Fato e Dimensão ou a alteração em campos (tipo, tamanho) existentes, deve-se considerar a contagem como PF_ALTERADO e aplicar o Fator de Impacto (FI) conforme definido no Projeto de Melhoria de DW (item 12.8).

A contagem de PF deve considerar, além das funções de dados alteradas (tabelas Fato e Dimensão), todas as funções transacionais impactadas pela alteração nas funções de dados, conforme exemplos apresentados nas tabelas a seguir:

Tabela 40: Exemplo de Criação/Alteração de Campos em tabelas fato e dimensão

	O método SFP identifica:	O método APF identifica:
Extração de dados do sistema de origem (lembre-se, essa função transacional, se existir, está fora da fronteira da aplicação de DW)	1 PE	1 CE ou 1 SE
Carga de dados nas tabelas Fato e Dimensão impactadas na alteração	1 PE	1 EE
Geração do Cubo	1 PE	1 SE

Tabela 41: Exemplo de Criação/Alteração de Campos em tabela dimensão Compartilhada

	O método SFP identifica:	O método APF identifica:
Tabela Dimensão Compartilhada referenciada como arquivo lógico externo (SFP: AL /APF: AIE) no DW	AL	AIE
Carga de dados de tabelas do DW que utilizam a tabela impactada para a validação de dados	1 PE	1 EE
Geração do Cubo	1 PE	1 SE

12.9.3. Alteração de Dados em Tabelas Fato e Dimensão

A demanda de alteração de dados em tabelas Fato e Dimensão não é comum em projetos de DW, mas, se houver, deve seguir o item 6.12.1 (Apuração Especial – Base de dados) deste roteiro de métricas.

12.9.4. Extração de Dados em Tabelas Fato e Dimensão

A extração ou recuperação de dados de tabelas Fato e Dimensão não é uma demanda comum de ocorrer em projetos de DW, sendo requisitada pelo usuário, pois a ferramenta OLAP já fornece diversas formas e possibilidades de consulta aos dados do DW. Entretanto, se apesar disso, ainda houver requisito para o desenvolvimento de extração de dados, deve-se seguir o item 6.12.2 (Apuração Especial – Geração de Relatórios) deste roteiro de métricas. Um exemplo desse tipo de demanda é a extração de dados de um DW para a carga em outro DW.

12.9.5. Criação, Configuração e Disponibilização de um Filtro de Relatório

A criação de um filtro de relatório para o DW é interpretada de forma semelhante ao item 6.17 (Componente Interno Reusável) deste roteiro, de forma a não considerar a contagem de todos os relatórios/consultas impactados pelo filtro. Um filtro de relatório pode ser gerado pela combinação de uma ou mais tabelas Dimensão.

12.9.6. Criação de Métricas

As métricas (fórmulas), criadas a partir de um requisito de usuário, são atributos lógicos associados a tabela Fato e são implementadas como DERs (Dado Elementar Referenciado) nas tabelas Fato. O processo de carga das tabelas Fato populam essas métricas que são totalizações ou cálculos a serem apresentados em relatórios gerados pelo usuário.

Caso o usuário solicite a criação ou alteração de uma métrica (por exemplo, sumarizações, médias, máximas e percentagens), deve-se avaliar o impacto nas funções transacionais que referenciam essa métrica.

Essa solicitação pode envolver a criação de novo arquivo para armazenar resultado de sumarizações, médias, máximas e percentagens de tabelas do DW. A contagem de PF deve considerar, além da função de dados alterada (no caso, a tabela Fato), todas as funções transacionais impactadas pela nova métrica, aplicando-se o Projeto de Melhoria de DW.

É importante ressaltar que, caso seja solicitada alteração em campos ou criação de campos em tabelas Fato ou Dimensão, a contagem deve avaliar o impacto nas funções transacionais que referenciam a tabela alterada para atender a criação de uma nova métrica. Nesse caso, a contagem final da demanda deve contemplar, ainda, a contagem das funções de dados e transacionais conforme descrito no item 12.9.2 (Criação/ Alteração de Campos em Tabelas Fato e Dimensão) deste roteiro de métricas.

Essas métricas também podem ser criadas a partir de um requisito tecnológico quanto à melhoria de performance ou outro requisito não-funcional. Esses casos podem ser tratados conforme os itens 6.10 (Manutenção em Interface), 6.11 (Adaptação sem Alteração de Requisitos Funcionais) ou 6.17 (Componente Reusável) deste roteiro de métricas.

12.9.7. Reorganização de Bancada

A reorganização de bancada ou reposicionamento de itens é uma demanda similar a manutenção em interface (*bancada* é uma interface para o usuário). Os itens da bancada são objetos da interface como, por exemplo: métricas, atributos, filtros, gráficos, tabelas e etc. Uma demanda deste tipo implica na necessidade de trocar a ordem dos itens/campos em relatórios e *dashboard* da bancada por solicitação do usuário.

Para esses casos, sugere-se seguir o item 6.10 (Manutenção em Interface) deste roteiro de métricas e considerar o respectivo FI para cada item reposicionado/reorganizado da bancada no projeto de DW.

12.9.8. Exclusão de Dados Antigos do DW

As funcionalidades de limpeza de dados do DW estão, geralmente, associadas à administração do ambiente de DW, por exemplo, dados históricos com mais de 60 (sessenta) meses devem ser excluídos da base do DW.

Caso exista o requisito de exclusão de dados antigos (ou históricos) do DW, considerando, principalmente, o escopo de tabelas Fato, deve ser contado uma função transacional (APF, como EE /SFP, como PE). Importante observar que, não necessariamente, identifica-se uma função transacional para cada função de dados (APF: ALI / SFP: AL) do DW.

12.9.9. Metadados Relacionados ao Controle do Processo de ETL

Os metadados relacionados ao controle do processo de carga de dados no DW correspondem aos dados usados para gerenciar o DW e podem ser, por exemplo: dados para controlar a adição de novos dados às tabelas do DW; o número de registros adicionados, alterados ou rejeitados; ou parâmetros usados para o processamento do ETL no DW.

Normalmente, a ferramenta de ETL já possui e disponibiliza esses controles (metadados) do processo de carga de dados no DW. Nesse caso, não há contribuição ao tamanho funcional. Senão, e caso essas funcionalidades sejam requeridas e desenvolvidas para apresentar ou editar esses dados de controle de forma diferente do disponível na ferramenta de ETL, então sugere-se a sua contagem.

De forma geral, sempre que as ferramentas e o ambiente de DW já disponibilizarem as funcionalidades e dados requisitados pelo usuário ou administrador do DW, então não há contagem de Pf.

12.9.10. Migração de Banco de Dados

Essa demanda representa a necessidade de alteração do banco de dados (ou sua versão) que atende ao projeto de DW. Por exemplo, a migração do banco Oracle para o banco SQL Server.

Sugere-se tratar esse serviço conforme o item 6.8.2 (Mudança de Plataforma – Banco de Dados) ou 6.9.3 (Atualização de Versão – Banco de Dados) deste roteiro.

12.9.11. Migração de ferramenta ETL

A demanda de migração de ferramenta ETL implica em refazer um processo de ETL e, nesse caso, deve ser contado conforme item 6.8.1 (Mudança de Plataforma - Linguagem de Programação) ou 6.9.1 (Atualização de Versão - Linguagem de Programação) deste roteiro.

13. ATIVIDADES SEM CONTAGEM DE PF

Deve-se ressaltar que no processo de desenvolvimento de um projeto de software há atividades que devem ser consideradas como complementares ou pré-requisitos ao processo de desenvolvimento, de modo que os esforços e produtos entregues devem ser contratados e remunerados em itens distintos do desenvolvimento por não se tratarem de atividades de desenvolvimento do software ou inerentes ao processo desenvolvimento do software.

Considerando um projeto de desenvolvimento de software que utiliza métodos ágeis, ao utilizar a modalidade de remuneração por Pontos de Função complementados por HST padronizada pela Portaria SGD/MGI nº 750 ou posterior:

- as atividades relacionadas a execução de uma Sprint, indispensáveis para a entrega de um incremento de software, são remuneradas pela métrica de ponto de função;
- enquanto as atividades complementares ao desenvolvimento de um software e não são mensuráveis pela métrica de PF, devem ser remuneradas pela métrica de Horas de Serviço Técnico – HST, descrita no item 15 deste roteiro.

No item 15.2 deste roteiro é apresentado um catálogo de serviços técnicos que são complementares ao desenvolvimento e manutenção do software e não mensuráveis em Pontos de Função. Os serviços técnicos relacionados no catálogo de serviços utilizam a métrica HST e refletem o custo médio do esforço especializado para a realização de uma atividade técnica complementar ao desenvolvimento, manutenção ou sustentação do SW.

O catálogo apresenta serviços técnicos relacionados a elaboração de manual de usuário, execução de testes avançados e outras atividades que são necessárias, ao longo do processo de desenvolvimento de SW, mas não são mensuráveis em ponto de função.

Nos subitens a seguir apresentam-se:

- Atividades diversas, não remuneradas pela métrica de PF e não contempladas no catálogo de serviços remunerados pela métrica HST; e
- Atividades sem contagem de PF em projetos de DW.

13.1. ATIVIDADES SEM CONTAGEM DE PF EM PROJETOS DE DESENVOLVIMENTO DE SW

A seguir apresentam-se atividades não remuneradas pela métrica de PF e não contempladas no catálogo de serviços, remunerados pela métrica HST:

- **Definição de Processo de Desenvolvimento de Soluções:** são as demandas para definição de Processos de Software que devem estar definidos antes da contratação de serviços de desenvolvimento de software.
- **Desenvolvimento de Cursos para EaD:** são as demandas de elaboração de conteúdo e montagem de material para 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. Essa é uma atividade que deve ser realizada antes da abertura do projeto de desenvolvimento de software.
- **Treinamentos em Tecnologia da Informação em geral:** são as demandas de treinamentos em linguagens de programação, ferramentas de gestão, processos, modelos da qualidade, métricas etc.
- **Curadorias em Geral:** UI, UX, IA, *Agile Coach*, etc.

Existem também atividades presentes em um processo de software (inerentes ao processo de desenvolvimento de software) que devem ser gerenciadas dentro do projeto de desenvolvimento, não devendo ser mensuradas separadamente. São elas:

- **Acompanhamento de Projetos:** é a atividade que a contratada faz internamente de modo a se organizar e planejar o atendimento dos cronogramas e outras demandas recebidas da contratante, cuja natureza é intrínseca ao desenvolvimento e manutenção de sistemas. Ou seja, ao desenvolver e manter sistemas, a tarefa de acompanhar e gerir o projeto por parte da contratada figuram como seus deveres contratuais, não cabendo pagamento por atividades que dizem respeito à sua própria gestão interna;
- **Correção de erros:** erros e bugs que venham a se manifestar em ambiente de produção dentro do período de garantia contratado.
- **Especificação de Requisitos:** em metodologias ágeis, o levantamento de requisitos é inerente ao processo de desenvolvimento de software, não devendo ser mensurado e remunerado separadamente. Em outras metodologias, caso o órgão opte por realizar o levantamento de requisitos separadamente do processo de desenvolvimento de software, esse deve ser remunerado por horas de consultoria.
- **Projeto e desenvolvimento de Banco de Dados:** as atividades de banco de dados associadas ao projeto de desenvolvimento, modelagem dos bancos seguindo as políticas de dados da contratante, preparação de ambiente (testes, homologação, implantação), desempenhadas pela contratada já devem ser consideradas dentro do projeto de software, não cabendo cobrança adicional.

- **Treinamento para Implantação:** são demandas de treinamentos sobre utilização do sistema desenvolvido pela contratada a ser implantado, para os gestores de solução do cliente e usuários e devem ser tratadas no escopo da fase de transferência do conhecimento para a contratante.

Finalmente, tendo em vista que já foram identificados casos concretos do uso incorreto do Ponto de Função, cabe reforçar que atividades cuja natureza difere totalmente do objeto contratado (serviços de desenvolvimento e manutenção de software) não podem ser remuneradas por pontos por função, são exemplos:

- **Deslocamentos e viagens** de integrantes da contratada para prestação dos serviços em diferentes localidades;
- **Suporte ao Usuário e à Rede no uso do software desenvolvido**, principalmente quando englobando atividades como instalação de microcomputadores e demais periféricos.

13.2. ATIVIDADES SEM CONTAGEM DE PF EM PROJETOS DE DW

Aqui destacam-se algumas atividades SEM contagem de Pontos de Função que estão relacionadas ao ambiente produtivo do *Data Warehouse (DW)*, como administração de usuários, segurança, customização de ferramentas OLAP e ETL para acesso de usuários etc.

13.2.1. Tabelas Sem Contagem de Ponto de Função

Algumas tabelas do DW não devem ser contadas como funções de dados em projetos de DW, a saber:

- Tabelas de repositório das ferramentas OLAP ou ETL não são contadas, mesmo que o usuário tenha necessidade de visualizá-las, pois, são desenvolvidas e mantidas pelas ferramentas, portanto, não são desenvolvidas pela equipe de *Data Warehouse*;
- Tabelas criadas para auxiliar o processo de transformação dos dados (tabelas da DSA ou temporárias), não são contadas, pois tratam-se de soluções técnicas;
- Tabelas Dimensão Estática;
- Tabelas de Agregação criadas para atender requisitos não funcionais, como desempenho.

13.2.2. Customização de Ferramenta OLAP

Corresponde a customizações feitas em ferramentas OLAP para personalizações de forma a atender as necessidades do cliente. Por exemplo, alterações de uma página web da ferramenta OLAP para a substituição do logo do fornecedor pelo logo do cliente; disponibilização de mensagens, dentre outras. Geralmente, essas customizações são semelhantes as customizações realizadas em desenvolvimento de software.

13.2.3. Integração da Ferramenta OLAP ao AD/LDAP

A maioria das ferramentas OLAP permitem autenticação de usuário integrada com o servidor AD/LDAP da organização. Para isso é necessário realizar algumas configurações na ferramenta OLAP para obter este tipo de integração, onde o DW torna-se cliente do servidor AD/LDAP para validar o acesso de seus usuários. Essa atividade está associada a ações administrativas do DW e permissão de uso na ferramenta OLAP, independentemente do tipo de acesso. Nesses casos, não há contagem de Ponto de Função, pois a atividade está associada a infraestrutura de segurança do órgão.

13.2.4. Customização da Ferramenta ETL

De forma similar ao item 13.2.2 (Customização da Ferramenta OLAP), a customização da ferramenta de ETL corresponde a customizações/personalizações feitas em ferramentas ETL de forma a atender as necessidades do cliente. Por exemplo, alterações de uma página web da ferramenta ETL para a substituição do logo do fornecedor pelo logo do cliente; disponibilização de mensagens, dentre outras necessidades.

13.2.5. Criação de usuários e perfis de acesso às ferramentas OLAP e ETL

A criação de usuários e perfis de acesso às ferramentas OLAP e ETL são atividades que fazem parte da administração do ambiente de DW e, normalmente, estão sob a responsabilidade da área de infraestrutura e produção dos sistemas do órgão. Portanto, esses serviços de criação de usuários e perfis de acesso às ferramentas OLAP e ETL não devem ser contabilizados na contagem de PF.

13.2.5.a. Automação do Processo de ETL

Essa atividade representa a implantação de um processo de ETL no DW. Constitui a criação dos *jobs* ou processos/*scripts* que fazem o encadeamento/automatização de todos os processos de carga das tabelas Fato e Dimensão. Sugere-se que a automação do processo de ETL seja tratada como uma atividade do ambiente produtivo do DW e, portanto, sem contagem de PF.

VERSÃO CONSULTA PÚBLICA

14. STORY POINTS

Story points é um método para medir o esforço em uma escala relativa; é a estimativa relativa ao esforço para realizar uma atividade comparado com outra atividade no projeto.

A métrica *story points* (pontos por história) é uma técnica usada para planejamento e estimativa e é alternativa a mensuração por hora. É uma métrica simples que pode auxiliar no planejamento de itens no *Backlog*, estimando as histórias e suas complexidades.

Ressalta-se que não é uma métrica adequada para fins de pagamento e sua utilização não é indispensável para projetos que utilizam metodologias ágeis.

Antes de detalhar como funciona a técnica de contagem por *Story points*, vamos reforçar alguns conceitos essenciais para utilização dessa métrica, além de outras orientações quanto ao levantamento e construção do *backlog* do produto e das histórias de usuário.



A métrica *Story Points* é uma técnica usada para planejamento e estimativa dos itens de *backlog*.

**ESTE ROTEIRO NÃO RECOMENDA O USO DA MÉTRICA
STORY POINTS PARA FINS DE PAGAMENTO.**

14.1. ORIENTAÇÕES GERAIS

O desenvolvimento ágil tem sido muito utilizado pelos órgãos da APF para o desenvolvimento de *software*. Em projetos grandes ou pequenos, diversos métodos ágeis podem ser adotados em momentos diferentes do ciclo de vida.

Em geral, os times e organizações utilizam uma combinação de métodos ágeis e termos, de acordo com o seu contexto e cultura organizacional.

Há diferentes métodos capazes de prover um desenvolvimento ágil de software, a exemplo do *Scrum*, *Extreme Programming (XP)*, *Kanban*, *Lean*, *Feature Driven Development*, entre outros.

O desenvolvimento de software com *Scrum* avança por meio de uma série de iterações chamadas de *Sprints*, que duram entre 2 e 4 semanas. Conforme *Scrum guide* (versão 2020), o Scrum possui apenas 3 artefatos:

- o *Backlog* do produto (*Product backlog*), que é uma lista emergente e ordenada do que é necessário para desenvolver o produto;
- o *Backlog da sprint* (*sprint backlog*), é o conjunto de itens do backlog do produto selecionados para a sprint. Conforme *scrum guide*, o *Sprint Backlog* é um plano DE e PARA os desenvolvedores. É uma imagem altamente visível e em tempo real do trabalho a ser realizado durante o *Sprint*;
- o Incremento do produto, que é um pedaço do produto que será desenvolvido. Cada incremento é adicionado aos incrementos previamente entregues e cuidadosamente verificados assegurando que todos os incrementos funcionem em conjunto. Todos e cada um dos incrementos deve ser utilizável. No momento em que um item do *Backlog* do produto satisfaz a definição de pronto (*definition of Done*), nasce um Incremento.

Esses artefatos possibilitam a inspeção, maximizam a transparência, promovem o foco e o alinhamento entre os membros do time, contudo o *Scrum* não orienta quanto a sua construção.

A partir deles chegaremos às histórias de usuários, que são peças-chaves para utilização da métrica *Story Points*. As histórias do usuário (*user stories*) surgiram no *Extreme Programming* (XP) e o seu principal objetivo era apresentar o desejo do usuário por alguma funcionalidade, devendo ser de fácil priorização.

Neste sentido, vamos ao entendimento e construção de alguns destes artefatos, viabilizando assim a contagem por *Story points*. Destaca-se que neste roteiro não vamos detalhar a fundo os artefatos do *scrum*, técnicas e etc. O objetivo é detalhar os elementos mínimos a serem observados e expor a métrica de contagem por *Story points*.

14.2. BACKLOG DO PRODUTO

A atividade de levantamento do *backlog* do produto é uma maneira de reunir e priorizar requisitos, restrições, objetivos e itens de trabalho.

O *backlog* do produto é um documento vivo e em contínua evolução. Dentre os itens que devem ser considerados na construção do *backlog* do produto temos: requisitos funcionais, requisitos não funcionais, requisitos de *interface*, requisitos de interação, requisitos de domínio, itens de trabalho, problemas, dificuldades, preocupações das partes interessadas, tarefas técnicas, processos e resultados relacionados, cenários e contexto, restrições, e assim por diante.

É importante que o profissional responsável pela construção do *backlog* do produto tenha conhecimento e experiência em práticas ágeis. Sugere-se que a primeira versão do backlog seja construída com a colaboração de todas as pessoas do time.

O *backlog* do produto é utilizado por todo o time para acompanhar os projetos em andamento durante a sua construção. Ele facilita o planejamento, a alocação de recursos, ajuda o time a gerenciar as expectativas, além de manter todos alinhados. Ele é uma ponte entre a organização e o time ágil.

A organização deve definir qual a ferramenta especializada mais adequada para que todos tenham acesso aos itens do *backlog* do produto (PBI, do inglês: *Product Backlog Item*).

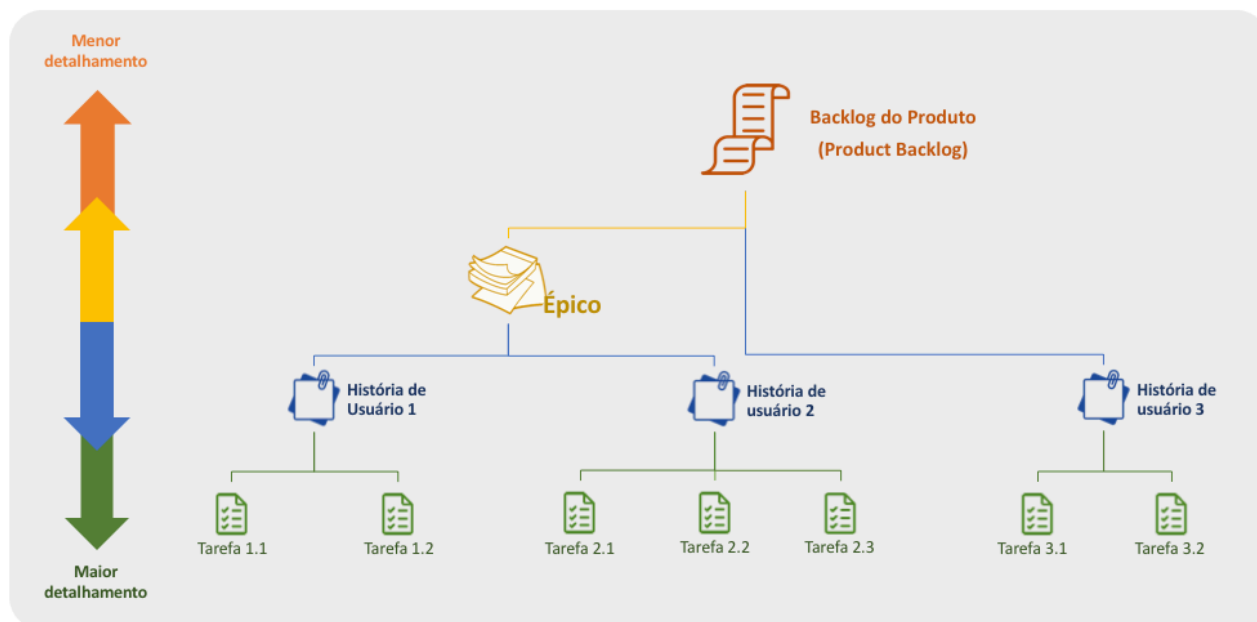
Como resultado do levantamento bem-sucedido do *backlog* do produto, são compreendidos os aspectos do domínio do negócio, as partes interessadas, as prioridades de negócio, as prioridades conflitantes etc.

O tipo mais comum de item do *backlog* do produto é uma história de usuário, embora outros tipos de itens do *backlog* incluam requisitos funcionais, não funcionais, técnicos e de teste.

O PO (*Product Owner*) geralmente prioriza os itens do *backlog* de acordo com o valor de cada história de usuário para o negócio/cliente, embora também possam considerar os riscos do projeto ou do produto identificados durante o planejamento.

Antes de iniciar a construção do *backlog* do produto, é importante observar se todos estão alinhados em relação aos termos utilizados. Aqui vamos utilizar os termos *Backlog* do Produto, Épico, História de Usuário e Tarefa. A imagem a seguir apresenta uma visão simplificada desses termos:

Figura 21: Alinhamento de termos do backlog do produto



A história do usuário é uma breve descrição de uma funcionalidade necessária, descrevendo as funções, as metas, os benefícios e a motivação das partes interessadas.

Histórias de usuários maiores, que estão em um estágio anterior de refinamento do *backlog* do produto são chamadas de épicos. Uma história do usuário não precisa estar necessariamente associada a um épico. O *backlog* do produto pode ter nenhum ou muitos épicos.

Já as tarefas (*tasks*) são pedaços ainda menores do trabalho a ser feito numa história do usuário. Elas identificam algo que precisa ser feito; algo que é necessário para uma história. A tarefa não será necessariamente independente e não irá demonstrar o valor do negócio.

Ao listar as tarefas necessárias para construir uma história, o time de desenvolvimento entra em detalhes técnicos de como os pedaços menores serão implementados.

14.2.1. Refinamento contínuo do backlog do produto

Como já foi dito, o *backlog* do produto é vivo. Portanto, o seu refinamento contínuo é peça fundamental para o trabalho dos times.

O objetivo do refinamento contínuo do *backlog* do produto é depurar, limpar, revisar, esclarecer e priorizar itens do *backlog* o produto para o escopo de trabalho atual.

Com a prática do refinamento contínuo, funcionalidades podem ser quebradas e informações adicionais são incluídas, resultando em um *backlog* do produto com itens (PBIs) menores e mais precisos.

O time deve considerar os *insights* e *feedbacks* das partes interessadas a cada incremento do produto. Com isso, os novos itens podem ser acrescentados e itens existentes podem ser reordenados e/ou refinados, praticando assim o refinamento contínuo.

Nem todos os itens de *Backlog* tem o mesmo tamanho ou nível de detalhe. Normalmente, os itens mais detalhados do *Product Backlog* (PBIs) são alocados às *Sprints* seguintes.

O *Product Owner* (PO) deve estar sempre trabalhando nos itens candidatos da próxima *sprint*. Assim, é interessante que o refinamento seja feito apenas para os PBIs seguintes, para atender de uma a 3 sprints e evitar perda de foco e retrabalho futuro. Sugere-se que os itens que serão trabalhados em um futuro distante não sejam refinados antecipadamente.

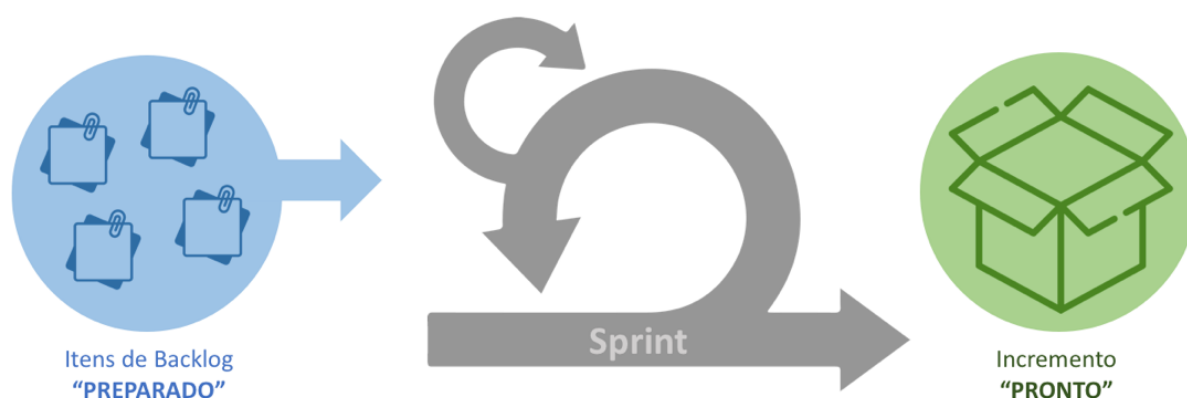
14.3. BACKLOG DA SPRINT

Depois que os itens de *backlog* estão definidos (PBIs), refinados, organizados e priorizados, o time parte para definir o *backlog* da *Sprint*, geralmente em uma reunião de Planejamento da *Sprint*.

O *backlog* da *sprint*/iteração é um subconjunto do *backlog* do produto, escolhido para ser desenvolvido na iteração atual. A qualidade de um item de *backlog* deve ser avaliada:

- Na entrada: na **Definição de Preparado** (*definition of ready*), que é elaborada em comum acordo entre o PO e *time* de desenvolvimento sobre o estado em que um item do *backlog* do produto deve estar para ser considerado "preparado" para entrar no *backlog* da *sprint*.
- Na saída: se o item estiver pronto de acordo com a **Definição de Pronto** (*definition of done*), ao final da *Sprint*, ele é aceito na reunião de Revisão da *Sprint* e sai da *Sprint* como parte do incremento do produto gerado.

Figura 22: Fluxo do PBI – Definição de preparado e definição de pronto



A **Definição de Preparado**, assim como a Definição de Pronto, é criada antes do início do desenvolvimento do produto, geralmente antes mesmo da primeira *Sprint*. Entretanto, ela pode ser modificada e evoluir de forma a acomodar novas necessidades identificadas ao longo do projeto.

Nos itens seguintes serão expostos mais detalhes sobre a construção das histórias de usuário e das definições de preparado e pronto, que devem ser verificados para atestar a qualidade de um item de *Backlog* que será implementado na *Sprint* atual.

14.4. HISTÓRIAS DE USUÁRIO (*user stories*)

Uma história de usuário fornece a identificação e a elicitación dos requisitos, na perspectiva do usuário final. É uma breve descrição da funcionalidade necessária descrevendo as funções, metas, benefícios e motivação das partes interessadas.

As histórias de usuários mudam o foco dos requisitos funcionais pesados para a compreensão e o atendimento às necessidades do usuário, permitindo que as equipes de produto forneçam soluções bem próximas das reais necessidades do usuário e que aumentam o valor do negócio.

Numa história de usuário devem ser consideradas as necessidades, os desejos do usuário, o valor para o negócio, os requisitos de qualidade do software, os requisitos funcionais e não funcionais, os riscos, as declarações de problemas, as dificuldades, as restrições, etc.

Numa história de usuário bem escrita os requisitos são claramente indicados e bem compreendidos. A interação frequente com o PO fornece à equipe ágil histórias de usuários refinadas e *feedback* regular sobre seu produto, melhorando a capacidade do time ágil de atender à visão do PO.

Nesse sentido, recomenda-se que as histórias de usuário:

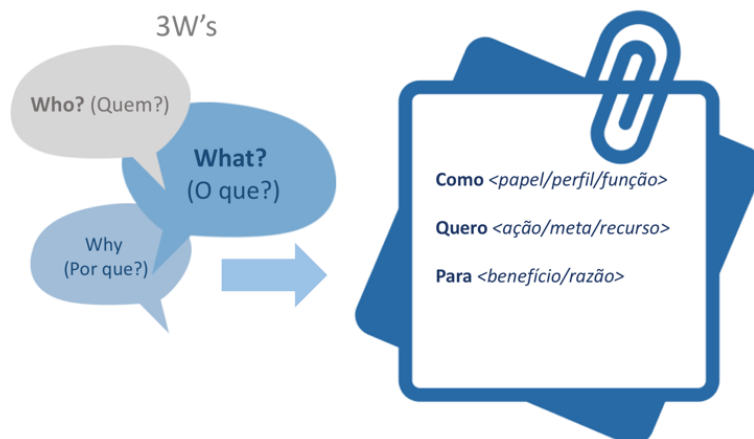
- Apresentem descrição em linguagem simples, com requisitos claros e bem compreendidos;
- Atendam aos requisitos do usuário e destaquem o benefício associado;
- Apresentem critérios de aceitação claros e vinculados aos requisitos do usuário;
- Gerem valor para o negócio;
- Sejam refinadas progressivamente, aplicando padrões de divisão eficazes.

14.4.1. Formato das Histórias do usuário

A História de usuário é um formato textual para a descrição de um requisito. Ela busca responder três perguntas do acrônimo conhecido como 3Ws: *Who* – Quem? *What* – O quê? E *Why* – Por quê?

O formato de uma história do usuário é o seguinte:

- Como (papel/perfil/função)
- Quero (ação/meta/recurso)
- Para (benefício/razão)



Como exemplo temos:

- Como Técnico do Programa, quero incluir o endereço de um beneficiário na tela “Detalhes do Beneficiário” para enviar correspondência para o beneficiário.

Sugere-se que ao escrever histórias do usuário, seja utilizado um design participativo onde o cliente e o time participam da sua construção.

O acrônimo "INVEST" pode ser usado para revisar cada história de usuário antes de entrar em uma determinada iteração:

- *Independent* (independente): cada história de usuário deve ser autocontida (ou seja, não depende de nenhuma outra história de usuário) para que cada uma possa ser projetada, desenvolvida, testada e aceita dentro de uma iteração;
- *Negotiable* (negociável): em uma história de usuário as conversas e negociações são bem-vindas; deve existir colaboração com as partes interessadas e evolução da história à medida que se aproxima da implementação;
- *Valuable* (valiosa): uma história descreve e entrega valor *stakeholders*, apoiando a priorização eficaz;
- *Estimável* (estimável): a história deve fornecer informações para o time realizar uma estimativa;
- *Small* (pequena) / *Sized-Appropriately* (sob medida): Uma boa história deve ser relativamente pequena em tamanho para ser concluída no menor tempo possível e caber em uma iteração, considerando o contexto do time;
- *Testável* (testável): uma boa história deve ser clara, permitindo que os testes sejam definidos para ela.



Esse acrônimo ajuda a escrever boas histórias de usuário, focando ainda mais na entrega de valor ao cliente. Cabe ressaltar que o acrônimo INVEST serve como guia de características ideais para escrever boas histórias de usuário e, portanto, nem toda história vai apresentar todas as características obrigatoriamente.

14.4.2. Métodos adicionais para levantamento das Histórias de Usuário

Escrever uma história em linguagem simples, mas que seja atraente, independente, clara, estimável, testável e que gere valor para o produto, não é uma tarefa simples.

Em tese, qualquer um pode escrever uma história do usuário, mas o dono do produto (PO) é o responsável pelo *Backlog* do produto, por criar, comunicar e ordenar os itens do *Backlog* do produto.

O acrônimo 3W's (quem? o que? por quê?) promove a colaboração das partes interessadas, mas deve-se evitar que esse modelo substitua as perguntas certas.

As histórias de usuários podem ser escritas: em revisões de *Sprint*, sempre que tiver uma nova ideia, durante o refinamento do *Backlog* do produto, em *workshops* de mapeamento de histórias, etc.

Não existe uma técnica padrão para o processo de levantamento ou mapeamento de histórias de usuário. Nesse sentido, para alcançar um levantamento de requisitos mais preciso e de alta qualidade é importante que o profissional tenha conhecimento de diversas técnicas, para então definir qual a técnica mais adequada a ser aplicada em cada situação.

A seguir, alguns dos muitos exemplos de métodos de levantamento de requisitos que podem ser usados no levantamento ou mapeamento de histórias de usuários:

Tabela 42: Exemplos de métodos de levantamento de requisitos

Técnicas/abordagens		Exemplos
Técnicas tradicionais de coleta de dados	Técnicas de coletas de dados primários, utilizadas em várias áreas de conhecimento para obter dados relevantes e necessários.	Questionários, entrevistas, observação, pesquisas, análise de documentos, <i>benchmarking</i> , etc.

Técnicas de elicitação em grupo, ideação	Técnicas de coletas de dados que tem por objetivo compreender melhor o pensamento e comportamento dos grupos, as necessidades dos usuários e gerar ideias inovadoras.	Brainstorming, Design Thinking, Joint Application Development (JAD), Workshop de cocriação, grupos de discussão, etc.
Prototipação	A prototipação pode ser utilizada para elicitar requisitos quando há um alto grau de incerteza ou quando é necessário um rápido <i>feedback</i> dos usuários. O uso de protótipo auxilia na elicitação e validação dos requisitos de sistema.	Protótipos em papel, protótipos digitais, <i>storyboard</i> , diagramas, desenhos, <i>wireframes</i> , etc.
Técnicas contextuais	Técnicas que auxiliam no entendimento do contexto a ser estudado e/ou trabalhado. É uma alternativa para as técnicas tradicionais e cognitivas.	Etnografia, análise social, pesquisas exploratórias etc.
Técnicas de análise, síntese e representação de dados	Técnicas utilizadas para representação, organização, análise e/ou síntese de dados	Diagrama de afinidade, cartões de <i>insight</i> , mapas conceituais, mapeamento mental, matriz de posicionamento, etc.
Técnicas para identificação dos usuários	Podem ser utilizadas em várias fases do processo para alinhar informações dos usuários com todas as pessoas envolvidas e para gerar um melhor entendimento do público-alvo, mas são especialmente úteis na geração e validação de ideias.	Personas, Mapa de empatia, <i>workshops</i> de modelagem de papéis do usuário etc.

14.4.3. Critérios de Aceitação

Cada história de usuário deve possuir critérios de aceitação (ACs, do inglês, *Acceptance Criteria*) que permitem validar uma história de usuário a partir de um conjunto de requisitos predefinidos a serem atendidos com o fim de comprovar que a história está concluída, completa e funcionando.

Os critérios de aceitação podem ser obtidos por meio de conversas com as partes interessadas, *workshops* com clientes, refinamento do *backlog* do produto e testes de usabilidade, dentre outros.

Eles são únicos para cada história de usuário e sincronizam as visões do cliente e da equipe de desenvolvimento. Cada critério de aceitação deve ser uma declaração clara, concisa e verificável.

Os critérios de aceitação podem ser escritos em diferentes formatos. Os dois principais são:

- **Formato orientado a regras:** apresenta uma lista de regras que definem o comportamento do sistema. Como exemplo temos:
 - Cada beneficiário deve estar cadastrado no Cadastro Único (CadÚnico);
 - Cada beneficiário deve apresentar endereço residencial válido cadastrado;
 - O beneficiário com saldo suficiente consegue sacar dinheiro da sua conta.
- **Formato orientado a cenários:** comumente utiliza a sintaxe *Gherkin*, que segue o formato “Dado/Quando/Então” para estruturar textos simples, os quais descrevem o cenário, a ação do usuário e o resultado esperado da ação.
 - Dado: Qual é o contexto inicial do teste? Quais as pré-condições explícitas que devem ser mantidas antes que o evento ou operação ocorra?
 - Quando: Qual é o evento ou operação? Qual a ação inicial?
 - Então: Quais são os resultados esperados? Qual o resultado pretendido ou as pós-condições após o evento ou operação?

Utilizando como exemplo a história de usuário apresentada no item anterior, podemos escrever alguns critérios de aceitação utilizando a sintaxe *Gherkin*:

- Critério de Aceitação 1: Dado que um endereço válido é inserido, quando o Técnico do programa aciona “Salvar”, então o endereço é salvo no banco de dados.
- Critério de Aceitação 2: Dado que um endereço inválido é inserido, quando o Técnico do programa aciona “Salvar”, então o sistema apresentará uma mensagem de erro indicando que o endereço é inválido.

É importante observar que os cenários, utilizando a sintaxe *Gherkin*, devem descrever o cenário, a ação do usuário e não ser um passo a passo. Dessa forma, os critérios de aceitação são escritos de forma simples, permitindo que qualquer pessoa possa ler e compreender.

É importante que os critérios de aceitação sejam o mais específico possível, garantindo que cada critério seja aprovado ou reprovado. A aprovação parcial não deve ser permitida. Se os critérios de aceitação não estão claros, provavelmente a história de usuário necessita de maior refinamento.

14.4.4. Interface de usuário (ou UI, do inglês *User Interface*)

Uma interface de usuário pode ser descrita de diversas formas:

- Baixa fidelidade: esboços simples;
- Média fidelidade: *wireframes*;
- Alta fidelidade: *mockups* ou protótipos.

A forma de escrever varia conforme o time, a organização, o tempo disponível para descrevê-la, etc.

O time geralmente acorda a forma que a história deve estar descrita para ser considerada preparada, do ponto de vista da UI. O importante é que o time esteja alinhado com o trabalho a ser realizado.

Vale destacar que nem todo item de *Backlog* ou história de usuário precisa estar necessariamente associado a uma *interface* de usuário, entretanto, quando isso ocorre, é importante apresentar e associar à história do usuário.

14.4.5. Tarefas (*tasks*)

Como já foi dito, as tarefas são pedaços ainda menores do trabalho a ser feito numa história do usuário. Cada tarefa identifica algo que precisa ser feito, algo necessário para uma história.

Com elas, o time de desenvolvimento entra em detalhes técnicos sobre como as histórias de usuário serão implementadas. As tarefas são mais diretas e apresentam linguagem técnica; normalmente não seguem um formato textual definido. Elas não são necessariamente independentes e não demonstram o valor do negócio.

Como exemplo de tarefas temos:

- Criar UI para cadastro de endereço do beneficiário;
- Criar dados de teste para verificar se o endereço é válido;
- Automatizar os scripts de geração de dados;

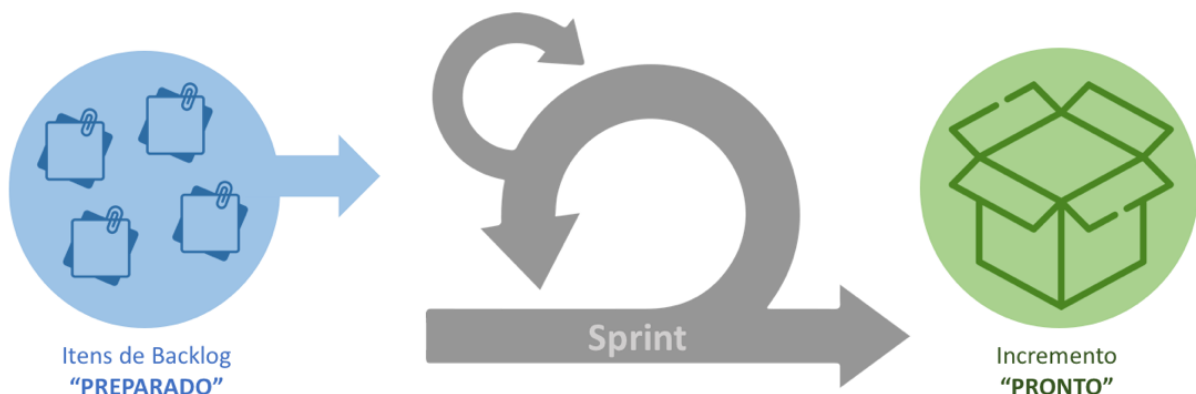
- Etc.

14.4.6. Definição de Preparado

Antes da história do usuário entrar em uma iteração, ela deve ser revisada em relação a uma série de critérios que determinarão se ela pode (em teoria) ser desenvolvida, testada e entregue em uma iteração.

Um item de backlog (PBI) que não está preparado para entrar em uma sprint, mas é incluído sem a correta verificação da qualidade, produz retrabalhos e atrasos ao projeto. A Definição de Preparado, assim como a Definição de Pronto, é criada antes do início do desenvolvimento do produto, geralmente antes mesmo da primeira Sprint. Contudo, ela pode ser modificada e evoluir de forma a acomodar novas necessidades identificadas ao longo do projeto.

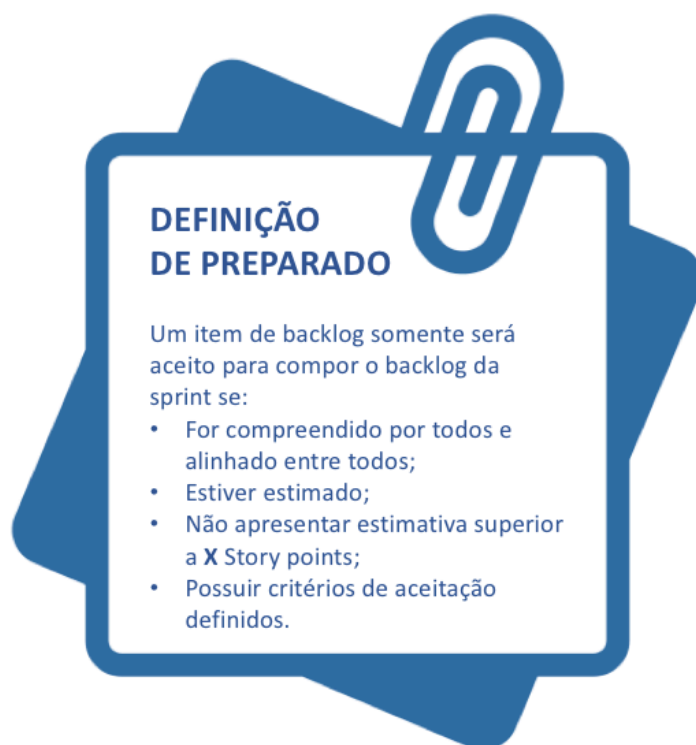
Figura 23: Fluxo do PBI – Definição de preparado e definição de pronto



A **Definição de Preparado** tem geralmente o formato de uma lista de critérios, condições, conforme exemplo a seguir:

- Deve ter havido um alinhamento sobre o item de backlog, de forma a gerar uma compreensão compartilhada sobre o que ele representa;
- O item deve possuir critérios de aceitação discutidos, compreendidos e acordados entre PO e o time de desenvolvimento;
- O item deve ter sido estimado pelo Time de Desenvolvimento;
- O item deve ser pequeno o suficiente, de acordo com algum critério estabelecido pelo time de desenvolvimento (um valor máximo para sua estimativa, por exemplo).

Figura 24: Exemplo de Definição de Preparado para um PBI



Outros critérios podem ser adicionados à definição de preparado. Essa lista pode ser modificada e evoluir de forma a acomodar novas necessidades identificadas ao longo do projeto.

Assim como a Definição de Pronto, a Definição de Preparado é criada, compreendida e compartilhada por todos os membros do time, devendo ser mantida visível para todo o time.

14.5. INCREMENTO DO PRODUTO

O incremento do produto é um pedaço do produto que será desenvolvido. Cada incremento é adicionado aos incrementos previamente entregues e cuidadosamente verificados assegurando que todos os incrementos funcionem em conjunto. Todos e cada um dos incrementos deve ser utilizável.

14.5.1. Definição de Pronto

A definição de pronto é uma descrição formal do estado do incremento, quando este cumpre as medidas de qualidade exigidas para o produto. Quando um item do *Backlog* do produto satisfaz a Definição de Pronto, nasce um incremento.

A definição de pronto cria transparência; proporciona a todos uma compreensão do trabalho que foi concluído como parte do Incremento. Todo o time ágil deve estar em conformidade com a definição de pronto.

Se um item de *Product Backlog* não cumpre a definição de pronto, não pode ser lançado ou mesmo apresentado na Revisão da *Sprint*. Em vez disso, volta ao Backlog do produto para consideração futura.

A seguir, alguns exemplos de critérios para aceitação dos produtos, que podem ser adaptados à realidade operacional da organização contratante:

Para admissibilidade do produto:

- a. Código-fonte submetido ao controle de versões;
- b. Existência de testes unitários e do Relatório de Testes;
- c. Existência de scripts de banco de dados com dicionário de dados embutido nos metadados (ausência apenas quando não houver mudança no modelo de dados);
- d. Existência de arquivo para geração de Build;
- e. Disponibilização de processos prontos para execução na ferramenta de CI/CD adotada, juntamente com a entrega e configuração de containers configurados pela ferramenta orquestração adotada;
- f. Existência de manual de implantação, conforme modelo disponibilizado pela Contratante;
- g. Existência documentação concluída, de acordo com os padrões de qualidade definidos pela Contratante e validadas pelo demandante.
- h. Resultado da execução de teste SAST indicando ausência de vulnerabilidades de nível HIGH ou CRITICAL, ou equivalente.

Para aceitação da demanda: após realizar a inspeção do produto quanto à sua admissibilidade, a empresa contratada deverá:

- a. Executar testes funcionais automatizados que tenham sido solicitados e, consequentemente, verificar se estão corretamente implementados ou mesmo se existem, além de observar os resultados da execução;
- b. Executar testes unitários ou verificar relatórios de execução destes que possam envolver porções críticas do produto;

- c. Realizar alguns testes funcionais, pelo menos nos principais fluxos do produto entregue.

14.6. CONTAGEM POR STORY POINTS

Story Points é uma estimativa relativa do esforço para desenvolver uma atividade comparado com o esforço para desenvolver outra atividade no projeto. Essa métrica difere das estimativas de tempo: ela discute o esforço e a complexidade do item.

Vale destacar novamente que apesar de ser uma métrica muito utilizada entre times ágeis, não é uma métrica obrigatória para as organizações e times que utilizam metodologias ágeis, visto que existem outras métricas que podem ser utilizadas para estimar o trabalho a ser desenvolvido por um time ágil.

A métrica *story points* utiliza números abstratos que dão a ideia de proporcionalidade entre os requisitos. Os times que utilizam *Story Points* usam uma escala exponencial, quadrática, logarítmica ou Fibonacci.

Os *Story Points* podem medir a complexidade de algo dentro do time. Podem ser histórias de usuário ou até mesmo tarefas que o time realiza, por isso devem ser utilizados pelas pessoas que fazem parte do time.

Os *story points* são medidas internas para apoiar o planejamento da sprint. Não há tamanho definido para uma história de usuário. Histórias de usuário, mesmo com objetivos/propósitos semelhantes, podem ter tamanhos diferentes.

É importante que os desenvolvedores tenham conhecimento e clareza do fluxo de trabalho e considerem questões como: O projeto é similar a algum outro trabalho anteriormente? Em caso afirmativo, qual foi o esforço para concluí-lo? Existe algum impedimento que possa causar atrasos? Dependemos de terceiros para a conclusão?

Existem vários métodos de dimensionamento de histórias de usuário, mas este roteiro detalha brevemente apenas o *Planning Poker*, que é o método de dimensionamento mais utilizado pelas equipes ágeis.

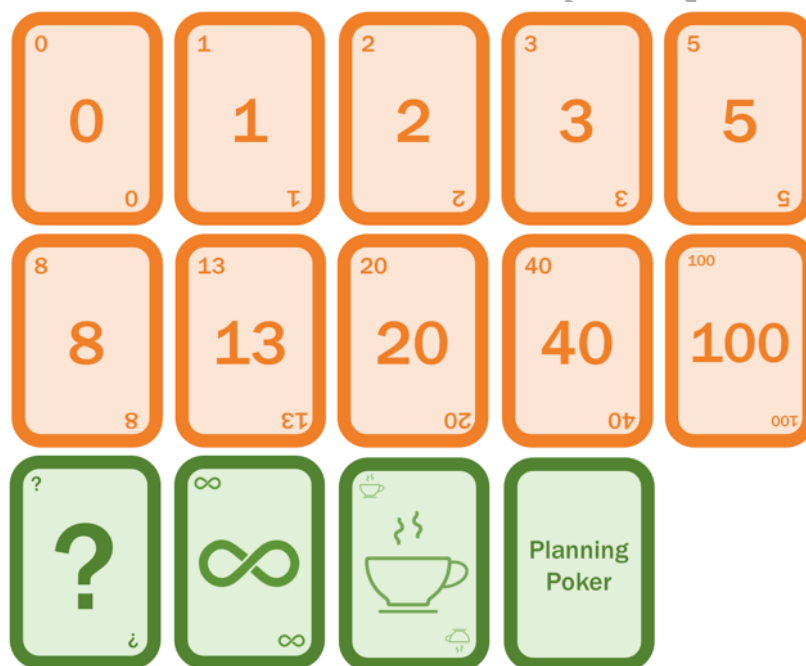
O *planning poker* é uma técnica de planejamento e estimativa em que as estimativas relativas do esforço necessário para desenvolver e testar cada história de usuário são definidas por consenso.

O pôquer do planejamento, também conhecido como pôquer da estimativa, constrói estimativas com base no domínio e na opinião de especialistas técnicos do time. Todos os membros da equipe contribuem durante o jogo, mas apenas aqueles que fazem o trabalho fornecem uma estimativa:

- **Preparo:** O facilitador garante que cada estimador tenha um baralho completo, com as cartas do jogo, e a compreensão do jogo e das histórias de referência;

- **Discussão:** O PO apresenta a história do usuário e os estimadores fazem perguntas para compreender completamente o trabalho a ser realizado;
- **Estimativa:** Cada estimador seleciona secretamente um cartão que reflete a sua compreensão do trabalho necessário para completar a história do usuário;
- **Votação:** Todas as cartas são reveladas em ao mesmo tempo. Quando existirem valores muito diferentes, os estimadores com as estimativas mais baixas e mais altas discutem seu raciocínio e então os jogadores votam novamente;
- **Registro:** Deve-se chegar a um consenso sobre uma única estimativa para cada história, registrar o valor acordado e seguir em frente. Três rodadas de votação sem consenso sinalizam que é necessário um maior refinamento da história.

Figura 25: Cartas do Planning Poker



Além da sequência de Fibonacci modificada, o *planning poker* geralmente inclui três cartas extras:

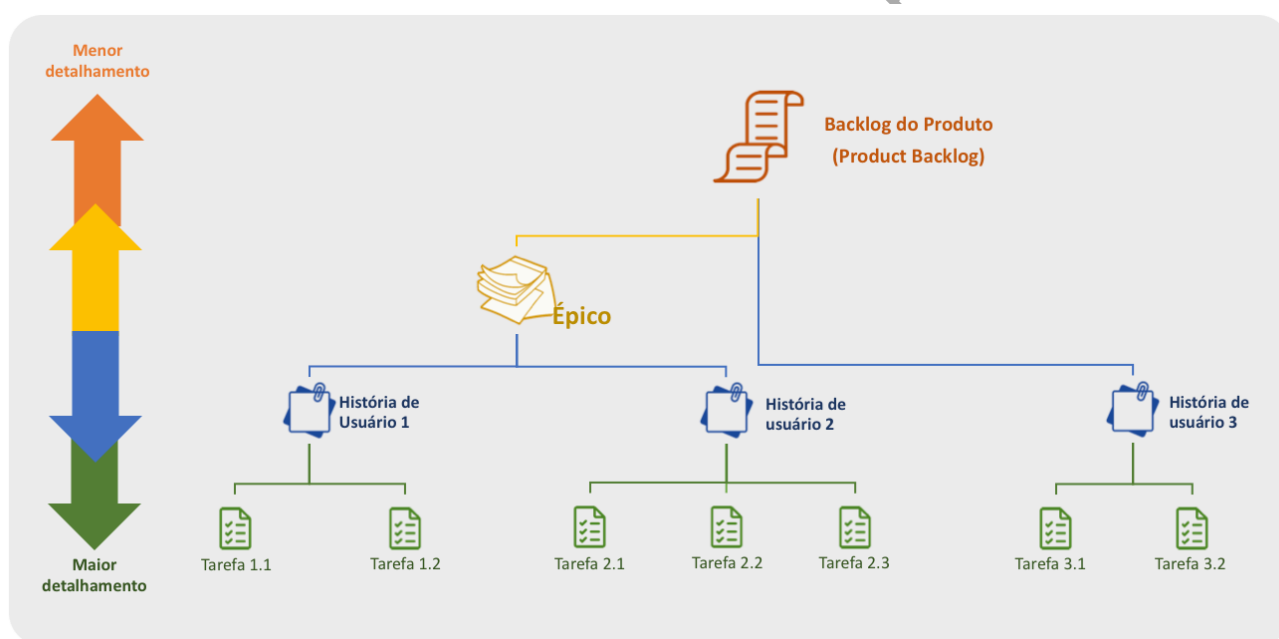
- O cartão com o ponto de interrogação indica que o estimador não tem certeza da estimativa;
- O cartão com o símbolo do infinito é usado quando o estimador acredita que a história é muito grande e requer divisão antes da estimativa;
- E, finalmente, o cartão da xicara de café para indicar que o estimador precisa fazer uma pausa.

O *Planning Poker* pode expor situações em que os estimadores têm diferentes perspectivas ou entendimentos da história do usuário. Contudo, busca-se o consenso com a participação de todos e diálogo. O consenso - em uma única estimativa - é normalmente alcançado em duas ou três rodadas. Se ainda houver grande divergência nas estimativas após três rodadas, um membro do time deve investigar a história - fora do evento - e trazê-la de volta com mais informações para serem analisadas na próxima sessão de refinamento do *backlog* do produto.

14.6.1. Foco da estimativa

Quanto mais perto da camada de valor, maior é o nível de abstração dos requisitos. Com isso é natural que haja maior dificuldade em estimar, pois os requisitos geralmente estão em um nível menor de detalhamento.

Figura 26: Detalhamento do Backlog do Produto



À medida que os requisitos são refinados em partes menores (histórias de usuário e tarefas) é natural que time tenha mais facilidade em estimar e consequentemente, a assertividade das estimativas é maior.

14.6.2. Passos para a contagem por *Story Points*

A seguir apresentam-se três passos, que devem ser realizados em conjunto pelo time, para auxiliar na contagem de histórias de usuário utilizando a métrica *story points*:

- **Primeiro passo:** definir a história mais simples, refletindo sobre como cada uma das histórias serão implementadas;
- **Segundo passo:** ordenar as histórias por complexidade, da mais simples para a mais complexa; e
- **Terceiro passo:** definir e registrar a pontuação das histórias;

Frequentemente utiliza-se a série de Fibonacci que apresenta uma sequência numérica, começando por 1 e 1, com os subsequentes correspondendo à soma dos dois anteriores [1, 1, 2, 3, 5, 8, 13, 21, 34, 55...]:

Sugere-se que o time utilize uma variedade pequena de *story points* para trabalhar e que se defina uma pontuação como teto (13 ou 21). Por exemplo: 1, 2, 3, 5, 8, 13.

Quanto maior a incerteza, maior a história e maior a distância entre os números da escala. O número superior da série representa uma história complexa que requer maior esforço. Nesses casos, o time deve analisar se a história deve ser dividida ainda mais.

14.6.3. Complexidade de histórias de usuário

Para ordenar as histórias de usuário por complexidade é necessário compreender e identificar o que as torna mais simples ou mais complexas.

Todo o time deve participar desse momento, trazendo detalhes importantes sobre a história que podem impactar a definição da sua complexidade.

É importante também que o time tenha histórias de referência, que já foram estimadas e desenvolvidas em outras *sprints*, para fazer uma comparação das histórias de referência com as histórias a serem estimadas: Qual a história mais complexa desenvolvida pelo time? E qual a menos complexa?

São muitos os fatores que podem tornar uma história mais ou menos complexa. A seguir foram listados alguns fatores que podem ser utilizados pelo time para avaliar a complexidade de cada história, facilitando assim sua organização e pontuação.

A lista abaixo pode ser usada como referência, podendo ser ajustada de acordo com as necessidades de cada organização, projeto ou time.

Tabela 43: Lista de fatores para avaliar a complexidade de uma história de usuário

FATORES PARA AVALIAR A COMPLEXIDADE DE UMA HISTÓRIA DE USUÁRIO

- | | |
|----|---|
| 1 | O número de subtarefas ou etapas necessárias para completar a história |
| 2 | História que afeta várias funcionalidades ou módulos |
| 3 | História que requer mudanças em várias partes do sistema (<i>backend, frontend</i> ou banco de dados) |
| 4 | Implementar funcionalidade que deve funcionar em múltiplos dispositivos (<i>desktop, mobile, tablet</i>) |
| 5 | História que envolve tecnologias ou ferramentas novas para a equipe |
| 6 | História que requer muito esforço de codificação, testes ou <i>design</i> |
| 7 | Trabalhar com código legado, que pode ser mal documentado ou estruturado |
| 8 | Volume grande de campos/atributos apresentados para o usuário |
| 9 | Volume de informações de controle e regras de negócio que influenciam a construção da funcionalidade |
| 10 | Tipo da função: entrada de dados, saída de dados ou mesmo consulta de dados dentro e fora da aplicação |
| 11 | Quantidade de validações de dados |
| 12 | Presença de cálculos matemáticos complexos (essa complexidade é muito variável, devendo ser definido pelo time o que é considerado cálculo matemático complexo) |
| 13 | Recuperação/consulta de dados de outras aplicações |
| 14 | Integração com uma ou mais aplicações |
-

15	Envio de dados para outras aplicações
16	Restrições de qualidade (por exemplo: confiabilidade, eficiência e portabilidade)
17	Restrições organizacionais como locais para operação, conformidade com padrões, hardware-alvo, etc
18	Restrições de interoperabilidade, segurança, privacidade
19	Restrições de implementação, como linguagem de desenvolvimento e prazo de entrega

14.6.4. Exemplo de aplicação da métrica

Como exemplo, considere uma *Sprint* que vai desenvolver as 4 histórias de usuário. Para utilizar a métrica, vamos seguir os passos indicados no item 14.6.2.

O **Primeiro passo** é definir a história mais simples. O time tem 4 histórias do usuário para desenvolver. Um membro da equipe pode descrever cada uma das histórias e esclarecer as dúvidas dos demais membros da equipe. Com isso, o time já começa a refletir sobre a complexidade de cada uma. Suponha que o time entende que a história 4 é a mais simples a ser implementada. Esse é o primeiro passo do processo de contagem.

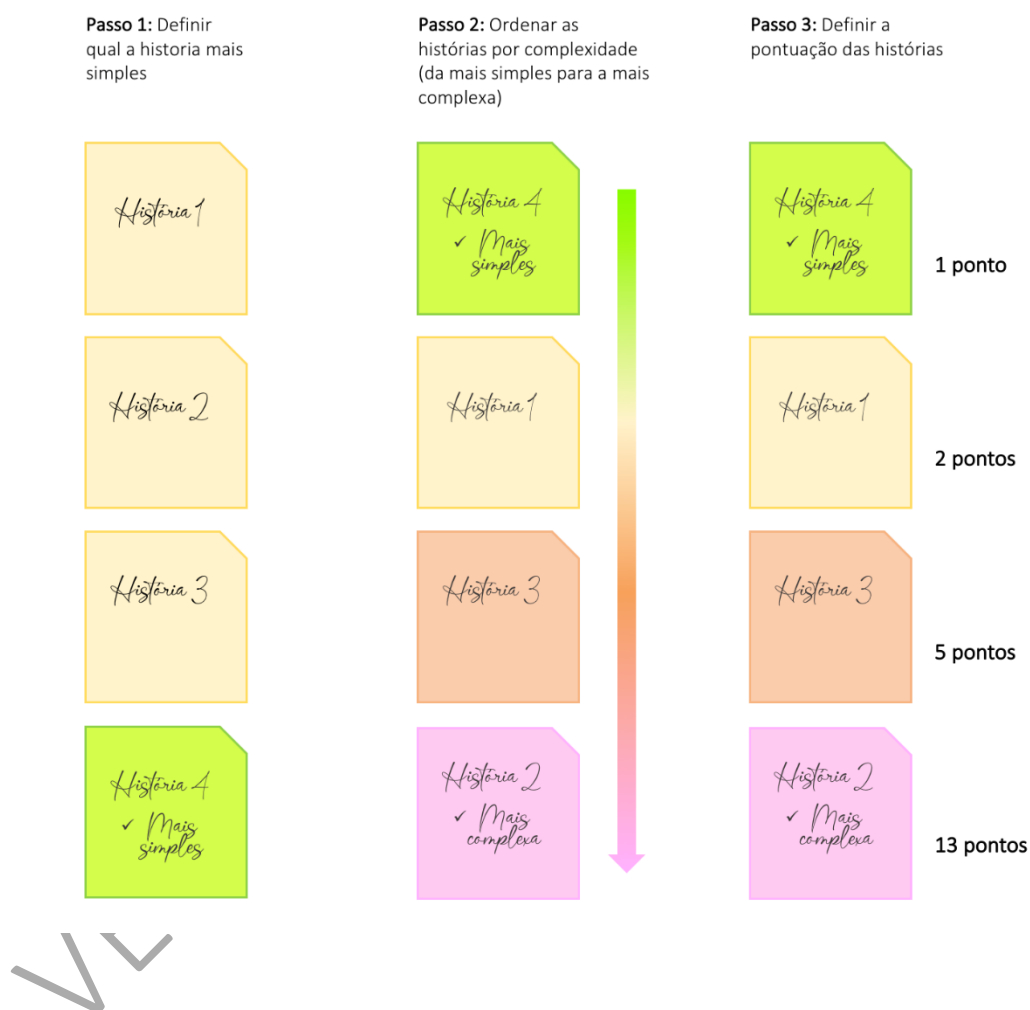
O **Segundo passo** é ordenar as histórias por complexidade, em ordem crescente de complexidade: da mais simples para a mais complexa. Para atender o segundo passo, o time discute em conjunto e analisa tudo o que está escrito em cada história e ordena as histórias por complexidade, dá mais simples para a mais complexa. Nesse momento, é importante que o time tenha o registro de histórias de referência, já estimadas e desenvolvidas pelo time. Assim, fica mais prático seguir para o 3º passo.

O **Terceiro passo** é definir a pontuação das histórias. Finalmente, o time define a pontuação de cada uma das 4 histórias a serem desenvolvidas e testadas, utilizando a sequência de Fibonacci modificada (1,2,3,5,8,13,21).

- A história 4 é a mais simples, por isso ganha 1 ponto. Trata-se apenas de enviar um e-mail com o resultado de uma consulta.
- A história 1 tem o dobro de complexidade da história 4, na opinião do time.
- A história 3, é mais complexa que a história 1 pois envolve cálculos matemáticos complexos (na visão do time). Dessa forma, o time atribuiu 5 pontos para essa história.

- E por fim, o time considerou por unanimidade que a história 2 é a mais complexa. Para desenvolvê-la será necessária a integração com duas bases de dados externas. Inicialmente o time entendeu que ela seria o dobro de complexidade da anterior, mas na escala utilizada pelo time o dobro não existe. Restaram as opções de 13 ou 21 pontos. Como o time entende que a história 2 não é 20 vezes mais complexa que a história 4, e também não é a história mais complexa já desenvolvida pelo time, todos concordaram em atribuir 13 pontos para a história 2. A organização da pontuação das histórias ficou da seguinte maneira:

Figura 27: Passos para contagem de *story points*



14.6.5. Boas práticas na estimativa por *Story Points*

A seguir listam-se boas práticas - que devem ser observadas pelos times - ao realizar a contagem por *story points*:

- Certificar-se de que cada história de usuário esteja bem detalhada, com critérios de aceitação claros, que permitam à equipe entender facilmente os requisitos e objetivos da história;

- Envolver a equipe no refinamento do *backlog* para identificar potenciais riscos, dependências e garantir estimativas realistas;
- Definir faixas de pontuação para as histórias. Por exemplo, a pontuação das histórias deve estar na seguinte faixa de pontuação: 1, 2, 5, 8, 13 e 21 pontos; sendo que 21 é a pontuação máxima;
- Definir diretrizes em relação a pontuação das histórias e documentar. Por exemplo, o que torna uma história mais complexa? Qual foi a história mais complexa desenvolvida pelo time em *sprints* anteriores? E a menos complexa?
- Discutir com a equipe de desenvolvimento a viabilidade técnica das histórias, identificando possíveis riscos, dificuldades, interrupções, imprevistos etc.
- Criar um mapa visual das histórias de usuário que dependem de outras equipes ou de recursos externos;
- Caso exista uma história muito complexa a ser estimada, o time deve analisar se é possível dividir a história e refiná-la ainda mais;
- Caso o time não entre em consenso sobre a estimativa de uma história, o time deve verificar se a história foi bem compreendida por todos e se deve ser mais refinada;
- Verificar quais histórias foram subestimadas ou superestimadas em estimativas anteriores e assim, estimar as próximas histórias com maior precisão;
- Em projetos com muitas incertezas, para evitar a subestimativa, incluir *buffers* de tempo para lidar com atrasos, interrupções ou imprevistos.

14.6.6. Boas práticas na definição do Backlog da Sprint

Aqui destacam-se boas práticas que devem ser observadas ao definir as histórias que vão compor o *backlog* da sprint:

- Estabelecer metas claras para a *sprint*;
- Incluir apenas histórias com definição de preparado (item 14.4.6) atendidas;
- Entender a capacidade da equipe a fim de: estabelecer metas realistas, manter a qualidade das entregas e evitar a ociosidade ou a sobrecarga de trabalho;
- Discutir abertamente com o time sobre os possíveis riscos que podem bloquear o progresso da sprint, tais como falta de recursos, questões técnicas ou de negócios;

- Manter uma distribuição equilibrada das histórias dentro da *sprint*. Por exemplo: não incluir muitas histórias complexas dentro do *backlog da Sprint*;
- Manter uma razão entre as histórias que vão compor uma *Sprint* para que o time consiga atender a *Sprint* dentro do *time-box* definido. Por exemplo: três histórias de baixa ou média complexidade e, no máximo, uma história de alta complexidade.

14.6.7. Erros comuns na contagem por Story points

Finalmente, convém expor alguns erros comuns, que devem ser evitados ao utilizar a métrica *story points*:

- Utilizar *story points* como métrica para pagamento;
- Igualar *story points* a hora;
- Não esclarecer dúvidas a respeito da história e assim, não compreender qual o trabalho necessário a ser realizado;
- Não analisar corretamente a complexidade da história e se contentar com a média;
- Não definir uma faixa de pontuação a ser utilizada pelo time;
- Alguém que não vai fazer o trabalho de desenvolvimento (PO, Gerente de Projetos, *Scrum Master*, etc) definir a pontuação das histórias;
- Deixar de manter atualizados os pontos de referência entre os itens de *backlog*, histórias e épicos;
- Utilizar *story points* para fazer comparações entre times, projetos e organizações;
- Utilizar *story points* para medidas milimétricas;
- Confundir a medição do fluxo de trabalho (produto) com o fluxo de valor (resultado);
- Priorizar a maximização dos *story points* completados em vez da entrega de valor real aos usuários;
- Não manter um histórico e registro das contagens por *story points*.

15. CONTAGEM DE HORAS DE SERVIÇO TÉCNICO - HST

Horas de Serviço Técnico (HST) é uma métrica baseada na quantidade de horas necessárias para se alcançar um resultado ou entregar um produto, por meio de atividades executadas por um ou mais perfis profissionais e aferidas por meio de indicadores de níveis mínimos de serviço e critérios de aceitação previamente estabelecidos.

A Portaria SGD/MGI nº 750 de 20 de março de 2023 padroniza a modalidade de Remuneração por pontos de função complementados por horas de serviço técnico. Nessa modalidade, a remuneração do serviço deve ser feita por meio da métrica Ponto de Função, combinada, quando couber, ao pagamento por Horas de Serviço Técnico baseado em catálogos de atividades previamente definidas.

Conforme orienta a portaria mencionada, deve-se distinguir o escopo das macroatividades abrangidas pela métrica Ponto de Função e das atividades a serem remuneradas por meio de Horas de Serviço Técnico, relacionadas em catálogo específico.

Nesse sentido, nos subitens a seguir apresenta-se:

- A descrição dos campos que compõem o catálogo de serviços técnicos (subitem 15.1);
- O catálogo de serviços técnicos, contendo serviços técnicos complementares aos serviços de desenvolvimento e manutenção do software E NÃO MENSURÁVEIS EM PONTOS DE FUNÇÃO (subitem 15.2);
- Orientações para calcular o fator de ajuste por perfil profissional (subitem 15.4);
- Orientações e fórmula para calcular o valor a ser remunerado pelo serviço técnico executado (15.4).

15.1. COMPOSIÇÃO DO CATÁLOGO DE SERVIÇOS TÉCNICOS

Conforme subitem 5.2.1.6 da Portaria SGD/MGI nº 750 de 20 de março de 2023, o catálogo de atividades remuneradas pela métrica HST deve conter, no mínimo, para cada atividade:

a) a descrição da atividade;

b) o volume de unidades de HST a ser remunerado;

c) os perfis profissionais aptos a executarem a atividade;

d) os produtos e os resultados esperados;

e) o prazo máximo de execução;

f) os critérios de aceitação.

O catálogo de referência de serviços técnicos que será apresentado nesse roteiro está alinhado com a portaria mencionada. Ele apresenta os campos mínimos indicados na portaria 750 e outros campos, alinhados com Acórdão nº 1508/2020 – TCU – Plenário, com o fim de mitigar riscos de

- Antieconomicidade oriunda de superestimações de esforços, produtos ou perfis profissionais;
- Criação de unidades de medida de forma unilateral e sem padronização;
- Fixação e de disseminação de critérios subjetivos;
- Compartilhamento de metodologias e práticas sem a devida consistência e sem justificativas técnica e econômica;
- Cenários de incomparabilidade de preços, de heterogeneidade e de assimetria de informações entre a administração e o mercado;

A tabela a seguir apresenta a definição de cada um dos campos que devem compor um catálogo de serviços técnicos a serem remunerados pela métrica HST:

Tabela 44: Descrição de campos do catálogo de serviços técnicos

Campos	Descrição dos campos
ID	Identificador do serviço técnico
Serviço Técnico	Título/Identificação do serviço técnico
Descrição do Serviço	Descrição do serviço técnico.
Unidade de Medida	Item mensurável utilizado para aferição do quantitativo de HST associadas ao serviço.

Exemplo: Serviço Técnico realizado por artefato, por funcionalidade, por caso de teste, etc.

Descrição dos Entregáveis	<p>Produto resultante da atividade ou serviço realizado.</p> <p>Os entregáveis são resultados tangíveis; são evidências e/ou artefatos que caracterizam os resultados dos serviços técnicos realizados. Fornecem não apenas as evidências da realização das atividades, mas também insumos para atividades subsequentes.</p>
Descrição das atividades mínimas	<p>Descrição das atividades mínimas a serem realizadas para execução do serviço técnico.</p>
Complexidade	<p>Definição do grau de complexidade do serviço.</p> <p>*Os serviços apresentados neste catálogo não têm variação da complexidade e foram classificados como “Padrão”.</p>
Critérios de identificação do grau de complexidade	<p>Descrição dos critérios utilizados na definição de complexidade do serviço.</p>
Qualificação profissional do executor	<p>Perfil profissional de referência, apto a realizar o serviço técnico. No catálogo e ordem de serviço de HST, o perfil profissional deve estar associado a natureza e ao nível de complexidade do serviço técnico a ser realizado.</p> <p>A qualificação e experiência dos profissionais que executarão o serviço técnico devem estar descritas em seção específica (ex: Requisitos de Experiência Profissional) do instrumento convocatório.</p> <p>*Caso a contratada disponha apenas de profissional com perfil superior ao necessário para execução do serviço, o valor do serviço técnico, bem como as suas características (atividades mínimas, complexidade, esforço, prazo máximo etc.) não devem ser alteradas para acolher perfil superior.</p>

Esforço (h) necessário à execução do serviço (A)	Corresponde a estimativa de horas necessárias para a realização de um serviço.
Memória de cálculo do esforço	Corresponde ao detalhamento do esforço (em horas) o para realização das atividades envolvidas na execução do serviço técnico.
Critérios de aceitação do serviço	Corresponde ao detalhamento dos critérios para aceitação do serviço prestado.

Além dos campos listados acima, o catálogo de serviços técnicos que vai compor o estudo técnico preliminar e/ou termo de referência da contratação deve apresentar também os campos a seguir, com o fim de apresentar a memória de cálculo da estimativa de custo e estimativa de execução das horas de serviços técnicos:

Tabela 45: Campos adicionais do catálogo de serviços técnicos que devem constar nos estudos técnicos preliminares

Campos	Descrição dos campos
Prazo máximo de execução do serviço	Corresponde ao nível mínimo de serviço esperado, com base em histórico de produtividade da organização, relacionados à execução da atividade técnica.
Quantidade estimada de execuções do serviço (B)	Quantidade estimada de execuções de cada serviço técnico, a ser definido pela organização, conforme base histórica.
Quantitativo estimado total de HST no contrato (C = A x B)	Corresponde ao produto do esforço em horas (A) necessário à execução do serviço e a sua quantidade estimada de execuções (B).

Memória de cálculo da estimativa de execuções do serviço	Corresponde ao detalhamento do cálculo utilizado para definir a quantidade estimada de execuções do serviço técnico.
Observações	Notas complementares, dicas, comentários e/ou orientações gerais relacionadas ao serviço técnico.

15.2. CATÁLOGO DE SERVIÇOS TÉCNICOS

Os serviços técnicos, mensurados por Hora de Serviço Técnico (HST) e descritos do catálogo a seguir, estão divididos em 5 categorias, a saber:

- Categoria 1- Requisitos de Negócio: compreende serviços técnicos para levantamento da visão do negocio, planejamento da release e elaboração de backlog do produto;
- Categoria 2- Testes: compreende serviços técnicos para elaboração de casos de teste e planos de testes não funcionais, bem como a execução, monitoramento e análise destes testes;
- Categoria 3 – Homologação: compreende serviços para disponibilização de ambiente para realização de homologação assistida e geração de massa de dados especifica para esse fim.
- Categoria 4 - Manual do Usuário: compreende serviços técnicos para elaboração de atualização de help /manual de usuário.

Convém destacar que este catálogo apresenta alguns dos serviços técnicos complementares aos serviços de desenvolvimento e manutenção de SW. O órgão pode incluir outros serviços técnicos, não mensuráveis com a métrica de PF, SEGUINDO O PADRÃO APRESENTADO NESTE ROTEIRO.

Tabela 46: Catálogo de serviços técnicos mensurados por HST

1. CATEGORIA: REQUISITOS DE NEGÓCIO

ID	Serviço Técnico	Descrição do serviço	Unidade de medida	Descrição dos Entregáveis	Descrição das atividades mínimas	Complexidade	Critérios de	Qualificação profissional do	Esforço (Horas)	Memória de cálculo do esforço
RN1	Construir a Visão do Negócio	Desenvolver atividades para entender as necessidades, expectativas, objetivos específicos do negócio e definir do escopo da demanda. Elaborar documento que contempla aspectos relacionados a problemas, objetivos de negócio, necessidades, expectativas; registra uma proposta de solução, a qual envolve elementos tecnológicos, descreve as características-chaves do produto, os processos de negócio, os consumidores da solução, os requisitos de ambientes, requisitos de documentação e requisitos do produto; menciona tecnologias importantes, entre outros.	por artefato	Documento de Visão (DV), Visão da Solução (VS) ou Visão da Demanda (VD) ou outro artefato previsto na MDS da Contratante	Realizar reuniões e utilizar técnicas para entendimento da visão do produto e solução; Elaborar documento que contempla aspectos relacionados a problemas, objetivos de negócio, necessidades, expectativas; descreve a proposta de solução, as características-chaves do produto, os consumidores da solução, os requisitos de ambientes, requisitos de documentação e requisitos do produto; menciona tecnologias importantes, entre outros.	Padrão	N/A	ANR-02 Analista de Negócios/ Requisitos - Pleno	44h	Realizar reuniões para identificar e analisar criticamente o problema (8h); Elaborar documento (32h); Obter consenso sobre o documento junto aos envolvidos (2h); Disponibilizar a versão final revisada (2h).
RN2	Atualizar a visão do Negócio	Desenvolver atividades para o entendimento das necessidades, expectativas, objetivos específicos do	por artefato	Documento de Visão (DV), Visão da Solução (VS) ou Visão da Demanda	Realizar reuniões e utilizar técnicas para atualizar a visão do produto e solução;	Padrão	N/A	ANR-02 Analista de Negócios/	10h	Realizar reuniões e utilizar técnicas para atualizar a visão do produto e solução (2h);

		negócio e proposta de solução do projeto e atualizar a documentação associada.		(VD) ou outro artefato previsto na MDS da Contratante atualizado	Atualizar documento que contempla aspectos relacionados a problemas, objetivos de negócio, necessidades, expectativas; descreve a proposta de solução, as características-chaves do produto, os consumidores da solução, os requisitos de ambientes, requisitos de documentação e requisitos do produto; menciona tecnologias importantes, entre outros.			Requisitos - Pleno		<p>Atualizar o documento de visão ou outro artefato previsto na MDS da Contratante (4h);</p> <p>Obter consenso sobre o documento junto aos envolvidos (2h);</p> <p>Disponibilizar a versão final revisada (2h).</p>
RN3	Planejar o Roadmap do produto	Desenvolver atividades para construir o planejamento de entregas ou plano cronológico de liberação dos releases (versões do produto), dividindo os objetivos de negócio e as características-chaves ou macrofunções do produto em partes entregáveis, por ordem de prioridade.	por artefato	Roadmap do Produto	<p>Organizar reuniões em que os envolvidos participem ativamente da construção do roadmap;</p> <p>Agrupar os itens (objetivos e características-chaves) por ordem de prioridade, em quantidade compatível com a capacidade de produção do projeto (instituição e contratada) e no tempo disponível para o desenvolvimento;</p> <p>Estabelecer uma cronologia de entregas ou a periodicidade;</p> <p>Elaborar e divulgar o Roadmap para todos os envolvidos.</p>	Padrão	N/A	Analista de Negócios/ Requisitos - Sênior	24h	<p>Organizar reuniões com os envolvidos para construção do Roadmap (8h);</p> <p>Elaborar Roadmap (12h);</p> <p>Obter consenso sobre o roadmap junto aos envolvidos (2h);</p> <p>Disponibilizar a versão final revisada (2h).</p>

RN4	Elaborar o backlog do produto	Desenvolver atividades para construir e disponibilizar o backlog do produto, que é a lista priorizada dos itens necessários para o desenvolvimento e entrega do produto de software. O Backlog do Produto representa tudo que é necessário para desenvolver e lançar um produto de valor agregado ao negócio; é uma lista de todos os requisitos (funcionais e não funcionais), funções, tecnologias, melhorias e correções de defeitos que constituem as mudanças que serão efetuadas no produto para versões futuras.	por artefato	Backlog do Produto	<p>Realizar reuniões e utilizar técnicas para levantamento de tudo o que é necessário para desenvolver o produto;</p> <p>Decompor os objetivos de negócio e características-chaves - estabelecidas no Documento de Visão - em requisitos de software (funcionais e não funcionais) e em tarefas técnicas de produção de software, chamados itens de backlog;</p> <p>Construir o Backlog do Produto, agrupando os itens necessários para o desenvolvimento do produto, por ordem de prioridade.</p>	Padrão	N/A	Analista de Negócios/ Requisitos - Sênior	24h	<p>Realizar reuniões para levantamento do que é necessário para desenvolver o produto (8h);</p> <p>Construir o Backlog do Produto (12h);</p> <p>Obter consenso sobre o backlog do produto junto aos envolvidos (2h);</p> <p>Disponibilizar a versão final revisada (2h).</p>
RN5	Planejar o release	Desenvolver atividades para definir o plano do release com a meta a ser alcançada em função dos objetivos de negócio, características-chaves e prazos do produto.	por artefato	Plano do release	<p>Realizar a estimativa de tamanho e esforço para implementar os itens do backlog do produto selecionados para o release (Backlog do Release);</p> <p>Definir a duração e a quantidade de iterações do release a partir da estimativa de tamanho e esforço definida;</p> <p>Inserir no plano do release: as premissas, os impedimentos e os riscos envolvidos no release;</p>	Padrão	N/A	Gerente de projetos	16h	<p>Realizar a estimativa de tamanho e esforço para implementar os itens do backlog do produto selecionados para o release (Backlog do Release) e definir a duração e a quantidade de iterações do release (6h);</p> <p>Inserir no plano do release: as premissas,</p>

					Prever atividades prévias ao início das iterações para que a equipe execute a criação/disponibilização dos ambientes de desenvolvimento e de testes necessários.					<p>impedimentos e riscos envolvidos no release.</p> <p>Prever atividades prévias ao início das iterações para que a equipe execute a criação/disponibilização dos ambientes de desenvolvimento e de testes necessários (6h);</p> <p>Obter consenso sobre o plano do release junto aos envolvidos (2h);</p> <p>Disponibilizar a versão final revisada (2h).</p>
--	--	--	--	--	--	--	--	--	--	--

2. CATEGORIA: TESTES

ID	Serviço Técnico	Descrição do serviço	Unidade de medida	Descrição dos Entregáveis	Descrição das atividades mínimas	Complexidade	Critérios de	Qualificação profissional do	Esforço (Horas)	Memória de cálculo do esforço
TS1	Elaborar caso de testes não funcional de	Elaborar casos de teste não funcional de uma aplicação. Definir as premissas de um teste de stress ou carga.	por caso de teste	Caso de Teste não funcional	<p>Reunir documentação relacionada e entender o projeto</p> <p>Elaborar caso de testes informando o objetivo, as pré-condições, as entradas,</p>	Padrão	N/A	Analista de Testes/Q	10h	<p>Reunir documentação relacionada e entender o projeto (2h);</p> <p>Elaborar caso de testes</p>

	stress ou carga				os passos a serem seguidos e os resultados esperados para atender aos objetivos de teste.			ualidade - Pleno		(4h); Obter consenso sobre o documento junto aos envolvidos (2h); Disponibilizar a versão final revisada (2h).
TS2	Executar, monitorar, controlar e analisar os testes não funcionais de stress ou carga	Desenvolver atividades para executar, monitorar, controlar e analisar testes não funcionais, produzindo resultados reais: Testes de stress ou carga – para verificar a quantidade de usuários simultâneos suportados pela aplicação;	por caso de teste	Relatório de testes não funcionais	Preparar ambiente e executar testes não-funcionais; Monitorar e controlar os testes; analisar os dados coletados e registrar a execução dos procedimentos, os resultados e os incidentes identificados durante a execução dos testes.	Padrão	N/A	Analista de Testes/Qualidade - Pleno	20h	Preparar ambiente para execução dos testes (8h); Executar, monitorar e controlar os testes (8h); Analisar os dados coletados; documentar a execução, os procedimentos, os resultados e os incidentes identificados durante a execução dos testes (2h); Disponibilizar relatório da execução dos testes para os envolvidos (2h).

TS3	Elaborar Plano de Testes	Desenvolver atividades para criação do Plano Testes Mestre (ou MTP - Master Test Plan) de um projeto. Trata-se da elaboração de documentação usada para planejamento e gerenciamento das atividades de teste; descreve as decisões tomadas durante o planejamento inicial do projeto. Ele apresenta uma descrição detalhada dos objetivos a serem alcançados, bem como os meios e cronograma para alcançá-los; direciona, orienta e restringe o esforço de teste priorizando os produtos liberados úteis e necessários. O plano de testes do projeto inclui a estratégia de teste específica do projeto, a equipe e o agendamento do teste.	por plano de testes do projeto	Plano de testes do projeto	<p>Reunir documentação relacionada e realizar reuniões para entendimento do projeto;</p> <p>Elaborar plano estabelecendo os testes a serem realizados para a entrega da solução, incluindo:</p> <p>a definição das metas e os objetivos dos testes no escopo do projeto; a identificação e análise de risco, bem como as abordagens para mitigação dos riscos;</p> <p>as estratégias e abordagens de teste adotadas; os recursos necessários e os produtos que serão liberados.</p>	Padrão	N/A	Analista de Testes/Qualidade - Pleno	16h	<p>Reunir documentação relacionada e realizar reuniões para compreender o projeto (2h);</p> <p>Elaborar o plano de testes do projeto (10h);</p> <p>Obter consenso sobre o documento junto aos envolvidos (2h);</p> <p>Disponibilizar a versão final revisada (2h).</p>
TS4	Atualizar Plano de Testes	Desenvolver atividades para ATUALIZAÇÃO do Plano Testes Mestre (ou MTP - Master Test Plan) de um projeto; documentação usada para planejamento e gerenciamento das atividades de teste. Descreve as decisões tomadas durante o planejamento inicial do projeto. Ele apresenta uma descrição detalhada dos objetivos a serem alcançados, bem como os meios e cronograma para alcançá-los; direciona, orienta e restringe o esforço de teste	por plano de testes do projeto	Plano de testes do projeto atualizado	<p>Reunir documentação relacionada e/ou realizar reuniões para compreender o projeto e as atualizações a serem feitas no plano de teste</p> <p>Atualizar o plano de testes do projeto e disponibilizar a versão final revisada</p>	Padrão	N/A	Analista de Testes/Qualidade - Pleno	8h	<p>Reunir documentação relacionada e/ou realizar reuniões para compreender o projeto e as atualizações a serem feitas no plano de teste do projeto (2h);</p> <p>Atualizar o plano de testes do projeto (2h);</p>

		priorizando os produtos liberados úteis e necessários. O plano de testes do projeto inclui a estratégia de teste específica do projeto, a equipe e o agendamento do teste.								Obter consenso sobre o documento junto aos envolvidos (2h); Disponibilizar a versão final revisada (2h).
3. CATEGORIA: HOMOLOGAÇÃO										
ID	Serviço Técnico	Descrição do serviço	Unidade de medida	Descrição dos Entregáveis	Descrição das atividades mínimas	Complexidade	Critérios de	Qualificação profissional do	Esforço (Horas)	Memória de cálculo do esforço
HM1	Disponibilizar Ambiente para Realizar Homologação Assistida	Desenvolver atividades para criação e preparação do ambiente específico para homologação assistida. Implementar os produtos ou componentes de apoio à homologação assistida. Apoiar os procedimentos, incluindo configuração de parâmetros, saneamento de dúvidas, e depuração de problemas.	por demanda	Ambiente de homologação assistida criado Assistência do profissional durante a homologação	Criar e preparar ambiente de homologação assistida; configurar parâmetros, sanear dúvidas e depurar problemas.	Padrão	N/A	Desenvolvedor de Software - Pleno	20h	Criar e preparar ambiente específico para homologação assistida (8h); Implementar os produtos ou componentes de apoio à homologação assistida (8h); Apoiar os procedimentos, incluindo configuração de parâmetros, saneamento de dúvidas e depuração de problemas (4h).

HM2	Elaborar/gerar massa de dados específica no ambiente de homologação	Elaborar/gerar massa de dados específica no ambiente de homologação, com a finalidade de atingir os resultados esperados na aplicação, de acordo com os requisitos aprovados pelo requisitante.	por plano de testes do projeto	Massa de dados elaborada/gerada em ambiente de homologação, conforme definido pelo requisitante	Reunir documentação relacionada e/ou realizar reuniões para compreender a necessidade do requisitante. Elaborar/gerar massa de dados específica, conforme solicitação do requisitante.	Padrão	N/A	Desenvolvedor de Software - Pleno	12h	Reunir documentação relacionada e/ou realizar reuniões para compreender a necessidade do requisitante (2h); Elaborar/gerar massa de dados específica, conforme solicitação do requisitante (6h); Obter consenso sobre a massa de dados, junto ao requisitante (2h); Disponibilizar a massa de dados em ambiente de homologação (2h).
-----	---	---	--------------------------------	---	---	--------	-----	-----------------------------------	-----	---

4. CATEGORIA: MANUAL DO USUÁRIO

ID	Serviço Técnico	Descrição do serviço	Unidade de medida	Descrição dos Entregáveis	Descrição das atividades mínimas	Complexidade	Critérios de	Qualificação profissional do	Esforço (Horas)	Memória de cálculo do esforço
----	-----------------	----------------------	-------------------	---------------------------	----------------------------------	--------------	--------------	------------------------------	-----------------	-------------------------------

MU1	Elaborar Help / Manual de Usuário de Sistema em Desenvolvimento	Desenvolver atividades para elaboração de Help da funcionalidade ou manual do usuário. durante o projeto de desenvolvimento ou manutenção.	por funcionalidade	Help da Funcionalidade ou Manual de Usuário	Reunir documentação relacionada e/ou realizar reuniões para entendimento da funcionalidade; Elaborar manual de usuário ou help da funcionalidade.	Padrão	N/A	ANR-02 Analista de Negócios/ Requisitos - Pleno	4h	Reunir documentação relacionada e elaborar manual de usuário ou help da funcionalidade (4h);
MU2	Atualizar Help / Manual de Usuário de Sistema em Desenvolvimento	Desenvolver atividades para atualização de Help da funcionalidade ou manual do usuário, durante o projeto de desenvolvimento ou manutenção.	por funcionalidade	Help da Funcionalidade ou Manual de Usuário atualizado	Reunir documentação relacionada e/ou realizar reuniões para entendimento da funcionalidade e alterações a serem feitas no documento; Atualizar manual de usuário ou help da funcionalidade.	Padrão	N/A	ANR-02 Analista de Negócios/ Requisitos - Pleno	2h	Reunir documentação relacionada e realizar alterações no documento (2h).
MU3	Elaborar Manual do Usuário de sistema já implantado, desenvolvido pela contratada	Desenvolver atividades para elaboração de Manual de Usuário para sistema já implantado, desenvolvido pela contratada.	por funcionalidade	Manual de Usuário	Reunir documentação relacionada e/ou realizar reuniões para entendimento da funcionalidade; Elaborar manual de usuário.	Padrão	N/A	ANR-02 Analista de Negócios/ Requisitos - Pleno	6h	Reunir documentação relacionada (2h); Elaborar manual de usuário ou help da funcionalidade (4h).

MU4	Elaborar Manual de usuário de sistema NÃO desenvolvido pela contratada	Desenvolver atividades para a elaboração de Manual de Usuário para sistema que não foi originalmente desenvolvido pela contratada.	por funcionalidade	Manual de Usuário	Reunir documentação relacionada e/ou realizar reuniões para entendimento da funcionalidade; Elaborar manual de usuário.	Padrão	N/A	ANR-02 Analista de Negócios/ Requisitos - Pleno	8h	Reunir documentação relacionada e/ou realizar reuniões para entendimento da funcionalidade (2h); Elaborar manual de usuário para funcionalidade (6h);
------------	---	--	--------------------	-------------------	--	--------	-----	---	----	--

15.3. CUSTO DA HST

O custo da HST adjudicado na licitação será usado para remuneração dos serviços técnicos, conforme descrito no item

15.4. Normalmente, para obter o custo da HST utiliza-se a fórmula a seguir:


$$\text{Custo da HST} = \text{Ct} / 160$$


Onde:

- o **Ct** corresponde ao custo total do profissional de referência dividido por 160 horas mensais.

O custo total do profissional de referência (Ct) é composto pelo custo mensal do perfil profissional mais os custos adicionais. Desta forma, uma hora de serviço técnico do profissional de referência corresponde a 1 HST.

Para facilitar o entendimento, vale observar a Tabela 13 (Cálculo da estimativa de custo mensal do time de referência) do item 8.5.1 deste roteiro, que apresenta parte de uma planilha de composição de custos e formação de preços. Lá é possível observar a fórmula e cálculo para obter o custo da HST na penúltima coluna.

15.4. VALOR A SER REMUNERADO PELO SERVIÇO

Como já foi dito, Horas de Serviço Técnico (HST) é uma métrica baseada na quantidade de horas necessárias para se alcançar um resultado ou entregar um produto, por meio de atividades executadas por um ou mais perfis profissionais e aferidas por meio de indicadores de níveis mínimos de serviço e critérios de aceitação previamente estabelecidos.

Nesse sentido, o valor a ser remunerado para cada serviço técnico deve se basear em 3 elementos principais, a saber:

- **Natureza** do Serviço técnico;
- **Esforço**: Tempo estimado para realização da atividade técnica;
- **Perfil profissional**: perfil do profissional que irá executar o serviço técnico.

Considerando que os profissionais que atuarão nas ordens de serviço remuneradas por HST têm perfil igual ou compatível com os profissionais considerados para as ordens de serviço remuneradas por Pontos de Função, para demonstrar o cálculo do valor unitário da HST adotou-se os perfis profissionais (colunas 1 e 2) e valores salariais de referência (coluna 3), trazidos pela Portaria SGD/MGI nº 750, de 20 de março de 2023, conforme tabela abaixo:

Tabela 47: Mapa de Pesquisa Salarial de Referência (Portaria SGD/MGI nº 750)

Perfil	Descrição do Perfil	Salário de Referência	Fator de Ajuste
ARQSOF-01	Arquiteto de Software – Pleno	R\$12.073,70	2,23
ARQSOF-02	Arquiteto de Software – Sênior	R\$18.084,53	3,34
DESENV-01	Desenvolvedor de Software – Junior	R\$7.519,48	1,39
DESENV-02	Desenvolvedor de Software – Pleno	R\$10.677,45	1,97
DESENV-03	Desenvolvedor de Software – Sênior	R\$14.016,77	2,59
LDESENV	Líder Técnico de Desenvolvimento	R\$15.901,68	2,94
ANR-01	Analista de Negócios/Requisitos Júnior	R\$6.567,23	1,21
ANR-02	Analista de Negócios/Requisitos Pleno	R\$8.744,98	1,62
ANR-03	Analista de Negócios/Requisitos Sênior	R\$11.227,93	2,07
ADADOS-02	Administrador de Dados Pleno	R\$7.714,04	1,43
ADADOS-03	Administrador de Dados Sênior	R\$12.115,48	2,24

SCRUM	Scrum Master	R\$11.732,20	2,17
GEPRO	Gerente de projetos de TI	R\$13.949,62	2,58
ATQ-01	Analista de Testes/Qualidade – Junior	R\$5.412,32	1
ATQ-02	Analista de Testes/Qualidade – Pleno	R\$7.795,75	1,44
ATQ-03	Analista de Testes/Qualidade – Sênior	R\$11.081,16	2,05

Inicialmente, antes de expor a fórmula para o cálculo do valor a ser remunerado pelo serviço, vamos calcular o fator de ajuste apresentado na tabela acima (coluna 4). Para isso, utilizou-se como referência o salário do perfil profissional ATQ-1 (linha destacada em amarelo).

Para os demais perfis, com salários superiores ao perfil profissional ATQ-01, é apresentado fator de ajuste superior a um (1). Logo, para obter o fator de ajuste dos demais perfis profissionais, utilizou-se a fórmula a seguir

$$\text{Fator de Ajuste} = \frac{\text{Salário do Perfil Profissional Y}}{\text{Salário do Perfil Profissional de Referência}}$$

Após obter o fator de ajuste para cada perfil profissional, deve-se calcular o valor da HST para cada perfil profissional, utilizando o custo da HST (fórmula descrita no item 15.3), conforme fórmula a seguir:

$$\text{Valor HST do perfil profissional} = \text{Custo da HST} * \text{Fator de Ajuste do perfil profissional}$$

Em vista disso, para calcular o valor a ser remunerado para o serviço técnico executado, deve-se utilizar a seguinte fórmula:



$$\text{Valor a ser remunerado para o serviço} = \text{Valor HST do perfil profissional} * \text{Esforço (h)} * \text{Qtd Atv Exec}$$



Onde:

- **Valor HST do perfil profissional** corresponde ao custo da HST multiplicado pelo fator de ajuste do perfil profissional;
- **Esforço (h)** corresponde ao esforço em horas necessário para execução do serviço, previsto no catálogo de serviços, apresentado no item 15.2 deste roteiro; e
- **Qtd Atv Exec** corresponde a quantidade de vezes que o serviço técnico a ser remunerado foi executado.

Convém destacar que os serviços não devem ser executados por profissional com qualificação inferior ao estabelecido para o profissional de referência do serviço, previsto no catálogo de serviços.

Caso o serviço seja realizado por profissional com qualificação superior ao mínimo exigido, o valor da HST do perfil profissional não deve ser alterado, sendo considerado o valor relacionado ao profissional de referência para o serviço,

No que se refere à atomicidade no uso das HST, a menor divisão possível é um (1) HST.



OS SERVIÇOS NÃO DEVEM SER EXECUTADOS POR PROFISSIONAL COM QUALIFICAÇÃO INFERIOR ao estabelecido para o profissional de referência do serviço, previsto no catálogo de serviços.

Caso o serviço seja realizado por profissional com qualificação superior ao mínimo exigido, o valor da HST do perfil profissional não deve ser alterado.

VERSÃO CONSULTA PÚBLICA

16. DIRETRIZES DE CODIFICAÇÃO SEGURA

Este roteiro dispõe sobre as melhores práticas de codificação com vistas a assegurar uma codificação enxuta, limpa, clara e eficiente, observando as diretrizes de codificação segura publicadas pela Secretaria de Governo Digital.

16.1. DIRETRIZES GERAIS

Além das diretrizes constantes deste roteiro, deve-se adotar no que couber as orientações constantes dos:

- Guia de Requisitos e de Obrigações quanto a Segurança da Informação e Privacidade, disponível em: https://www.gov.br/governodigital/pt-br/seguranca-e-protecao-de-dados/guias/guia_requisitos_obrigacoes.pdf.
- Guia de Segurança em Aplicações Web, disponível em: https://www.gov.br/governodigital/pt-br/seguranca-e-protecao-de-dados/guias/guia_seguranca_aplicacoesweb.pdf.
- Guia de Requisitos Mínimos de Segurança e Privacidade para APIs, disponível em: https://www.gov.br/governodigital/pt-br/seguranca-e-protecao-de-dados/guias/guia_seguranca_apis.pdf.

16.2. DIRETRIZES DE ESCRITA DE CÓDIGO

16.2.1. Nomes Significativos

- Os nomes utilizados para variáveis, funções, classes e métodos devem ser autoexplicativos, ou seja, deve responder as seguintes perguntas: "Por que ele existe?", "o que ele faz?" e "como é usado?".
- Não se deve incluir no código informações inverídicas ou falas para mascarar o código ou dificultar sua manutenção.
- Os nomes devem ser pronunciáveis e ter significado adequado ao contexto em língua portuguesa.
- Evite codificações ou encriptações em nomes de variáveis, classes, funções e métodos.
- Os nomes devem ser de fácil entendimento associado ao contexto do domínio do negócio.

16.2.2. Funções e Classes

- I. As funções e classes devem ter o mínimo de linhas possível, mantendo-se a clareza e implementando o código necessário para traduzir o objetivo esperado da função ou da classe.
- II. As funções e classes devem ter objetivos específicos possuindo um nível de abstração (deve ter apenas uma responsabilidade ou objetivo, ou seja, deve fazer uma coisa).
- III. A escrita do código deve seguir uma narrativa *top-down*.
- IV. As funções e classes não devem possuir efeitos escondidos ou colaterais, ou seja, não devem produzir efeitos distintos daqueles que se propõe a realizar.
- V. Deve-se evitar níveis de herança exagerados, superior a 4 (quatro).

16.2.3. Comentários

- I. O código útil deve ser autoexplicativo, limpo e eficiente, devendo possuir apenas os mínimos de comentário possível.
- II. Evite comentários redundantes ou óbvios, que descrevem o que o código faz em acrescentar nenhuma informação útil.

16.2.4. Estruturas

- I. Não se deve criar estruturas de código redundantes ou sem efetiva geração de valor à finalidade pretendida.
- II. Não se deve replicar linhas de códigos ou estruturas tornando-as redundantes.
- III. Deve-se evitar duplicar ou replicar funções ou classes que possuam a mesma responsabilidade ou objetivo.

16.2.5. Indentação

- I. Deve-se adotar a indentação adequada para melhor entendimento e clareza do código, evitando-se indentações exageradas ou nenhuma indentação.

16.2.6. Complexidade ciclomática

- II. Deve-se adotar a menor complexidade ciclomática possível por algoritmo.
- III. Deve-se evitar o aninhamento excessivo de instruções de condições (If's dentro de If's).
- IV. Deve-se evitar o exagero de condições ("Case") dentro de instruções de escolha ("switch").
- V. Deve-se remover construções desnecessárias, ou seja, aquelas que podem ser removidas sem alterar a semântica e a finalidade do código.

16.3. DIRETRIZES DE SEGURANÇA DA INFORMAÇÃO

16.3.1. Armazenamento de Dados

- I. Nos casos de armazenamento de dados abertos (Dados públicos), não se deve utilizar meio de armazenamento que não possua acesso para escrita restrito por senha.
- II. Nos casos de armazenamento de dados fechados (Dados cujo acesso é restrito a um grupo específico de pessoas), não se deve utilizar meio de armazenamento que não possua acesso para leitura e escrita restrito por senha.
- III. Não se deve disponibilizar às aplicações acesso à algum banco de dados utilizando *login* de usuário com permissões de *root*.
- IV. Não se deve utilizar o armazenamento de senhas em código-fonte.
- V. Não se deve utilizar as mesmas senhas para ambientes de desenvolvimento ou homologação e produção
- VI. Deve-se armazenar de forma segura os dados de usuários e os sistemas que utilizam cada senha fornecida.

16.3.2. Gerenciamento de Acessos e Permissões de Usuários

- I. Não se deve armazenar senhas em texto plano sem utilizar um algoritmo de *hash* seguro e *salt*.
- II. Deve-se utilizar controle de usuário e senha nominais para determinar a identidade do usuário.
- III. Deve-se utilizar HTTPS para controle de autenticação ao menos nas telas de *login* do sistema web.

16.3.3. Comunicação

- I. Deve-se empregar canal de comunicação com controle de duplicação e perda de informações/mensagens.
- II. Deve-se empregar canal de comunicação com controle de autenticação (eg, HTTPS, certificados digitais gerados por autoridades confiáveis, VPNs).
- III. Deve-se empregar canal de comunicação que provenha confidencialidade dos dados transmitidos (eg, HTTPS, VPNs).

16.3.4. Medidas contra-ataques

- I. Deve-se prevenir ataques de injeção de SQL (SQL Injection), a exemplo das seguintes medidas mínimas:
 - II. Não se deve criar SQLs concatenando parâmetros textuais de origem não-segura, como parâmetros preenchidos pelo usuário ou mesmo armazenados no banco de dados.
 - III. Deve-se restringir permissões de acesso ao banco de dados para o usuário da aplicação.
 - IV. Deve-se, sempre que possível, passar parâmetros em comandos SQL (DML ou DDL) utilizando *prepared statements*.
 - I. Consultas que não podem ser parametrizadas deverão receber tratamento especial, como escapes ou codificação em hexadecimal.
 - II. Deve-se prevenir ataques de injeção de *HTML* e *Javascript*.
 - III. Deve-se prevenir ataques do tipo *cross-site scripting* (XSS).
 - IV. Deve-se prevenir ataques de quebra de autenticação e gerenciamento de sessão (*Broken Authentication and Session Management*).

16.3.5. Auditoria, rastreamento e Logs

Permitir que eventos sejam registrados a exemplo de:

- a) operações de *login* e *logout*;
- b) acessos a determinadas telas ou seções do sistema;
- c) acesso a informações com alguma restrição (eg, documentos sigilosos, processos em segredo de justiça, dados pessoais ou bancários);
- d) operações de inclusão, alteração ou exclusão de registros no banco de dados;
- e) alteração de perfil de acesso (para sistemas que possuem acesso com diferentes perfis); e
- f) execução de *jobs* e tarefas automatizadas.

Permitir que informações sejam auditáveis para cada tipo de evento selecionado para monitoramento:

- a) data e hora;
- b) usuário que efetuou a operação;
- c) endereço IP;
- d) identificador da sessão do usuário (quando aplicável, eg, *cookie*);
- e) tela (página) do sistema de onde a operação foi realizada;
- f) identificador da instância (para sistemas clusterizados);
- g) para operações de inserção, alteração ou exclusão, o tipo da operação, nome da tabela que foi manipulada, ID do registro e, se for o caso, valores anterior e atual de cada campo;
- h) parâmetros informados pelo usuário (eg, parâmetros GET ou POST), tomando cuidado de não armazenar dados sensíveis, como senhas;
- i) tempo de resposta do sistema;
- j) para execução de *jobs* e tarefas automatizadas, armazenar o resultado da operação; falha, sucesso, cancelada, etc.

Permitir formas de captura dos dados para auditoria, a exemplo:

- a) alterações aplicadas no banco de dados podem ser auditadas via *triggers*;
- b) auditar as alterações a partir da própria aplicação, sendo que algumas informações poderão não ser registradas (eg, operações SQL realizadas por fora da aplicação).

Em sistemas *web* desenvolvidos em Java, um Filtro pode interceptar as requisições feitas à aplicação.

Deve-se definir no documento de especificação de requisitos do sistema quais informações deverão ser registradas e o local de armazenamento dos dados da auditoria.

16.3.6. Cópias de Segurança (Backups)

- I. Deve-se incluir no plano de projeto a especificação da necessidade e a atribuição da responsabilidade de realização de *backups* do banco de dados e dos códigos-fonte do sistema, bem como as políticas de acesso a este *backup*.
- II. Deve-se definir um procedimento estruturado para a restauração de *backups*.
- III. Deve-se definir e capacitar responsáveis pela recuperação dos *backups*.

16.3.7. Testes

- I. Deve-se realizar testes manuais de segurança antes de cada versão do software que modifique sua estrutura (telas de *login*, serviços não autenticados, novos formulários com interação com o usuário, etc.).
- II. Deve-se garantir, através de testes automatizados, que os serviços e dados sigilosos estão protegidos e disponíveis apenas para os usuários detentores das informações.
- III. Deve-se elaborar uma política de testes, automatizados ou não, visando a garantia de não vulnerabilidade aos principais ataques conhecidos em sistemas.
- IV. Deve-se definir cenários de testes voltados à garantia dos requisitos não funcionais do *software*, preferencialmente realizado por uma equipe de testes diferente da equipe de desenvolvimento do *software*, com intuito de se evitar vícios.
- V. Deve-se definir cenários de testes, principalmente nos aspectos de segurança, para os casos de atualizações na arquitetura do sistema (servidores de aplicação, banco de dados, versões de *browser*, versões de sistema operacional, etc.).

16.3.8. Medidas contra falhas de segurança

- I. Deve-se manter procedimento planejado para imediata indisponibilização do sistema e realização de manutenção corretiva.
- II. Deve-se definir uma política de acompanhamento pós-correção de ocorrências de falha de segurança.

16.3.9. Diretrizes para a instalação, configuração e gerenciamento de ambientes de desenvolvimento de sistemas

- I. Deve-se utilizar um controle de versão distribuído, que mantém um repositório completo em cada máquina de desenvolvimento, *Eg, Git, Mercurial*.
- II. Deve-se utilizar bancos de dados distintos para cada ambiente (Desenvolvimento, produção).
- III. Deve-se utilizar servidores de aplicação/web distintos para cada ambiente (Desenvolvimento, produção).
- IV. Deve-se prover acesso ao ambiente de desenvolvimento/testes/homologação apenas aos integrantes da equipe de desenvolvimento e aos interessados no projeto (*stakeholders*).
- V. Deve-se realizar testes periódicos para assegurar a segurança do ambiente de desenvolvimento/testes/homologação.

16.3.10. Diretrizes para a configuração de proteção a dados sensíveis

- I. Deve-se utilizar um método criptográfico que siga o princípio de Kerckhoffs; o método de encriptação e seus parâmetros devem ser públicos e estar documentados, somente a chave criptográfica deve ser mantida em sigilo.
- II. Não se deve utilizar um cifrador que admita um método conhecido para quebra da chave criptográfica melhor do que a força bruta, baseada em tentativa e erro.
- III. Não se deve utilizar o modo de cifrador de bloco *electronic codebook* (ECB) ou modos menos seguros.

- IV. Não se deve utilizar um tamanho da chave menor que 128 bits (cifrador simétrico) ou 1024 bits (cifrador assimétrico).
- V. Não se deve utilizar função de *hash* sem algum tipo de *salt*.
- VI. Não se deve utilizar algoritmos considerados obsoletos para criptografia e *hash* criptográfico. Exemplos: MD5, SHA1, DES/3DES, RC2, RC4, MD4.
- VII. Não se deve utilizar um tamanho da chave menor que 192 bits (cifrador simétrico) ou 2048 bits (cifrador assimétrico).
- VIII. Não se deve distribuir chaves criptográficas sem a utilização de uma infraestrutura de chave pública e, portanto, sem a utilização de um cifrador assimétrico.

16.3.11. Geração de Senhas

- I. Não se deve utilizar senhas com menos de 8 caracteres.
- II. Não se deve utilizar somente um tipo de caractere (letras, dígitos e símbolos).
- III. Não se deve utilizar palavras comumente utilizadas para senhas (ou variantes destas), como, por exemplo: nome do animal de estimação, membro da família ou pessoa significativa; datas de aniversário; nome do feriado favorito; algo relacionado ao time esportivo favorito; e as palavras “senha” e “password”.
- IV. Não se deve utilizar periodicidade de troca superior a 1 ano.
- V. Não se deve armazenar senhas em claro.
- VI. Deve-se armazenar ao menos o *hash* criptográfico com *salt*.
- VII. Não se deve usar um canal em claro para a transmissão da senha ou elemento correspondente.
- VIII. Não se deve utilizar método de conferência menos seguro que desafios baseados em *hash* ou o uso de *hashes* armazenados.
- IX. Não se deve permitir uma taxa de tentativas de validação de senha superior a 5 tentativas por minuto.
- X. Não se deve mostrar diretamente a senha quando esta necessita ser digitada pelo usuário; deve haver opção de habilitar e desabilitar a visualização da senha digitada até então.

16.3.12. Diretrizes para reforço da segurança de software nas fases de especificação, projeto e design

- I. Deve-se empregar modelo de projeto de software que contemple, no mínimo:
 - a) a etapa de modelagem de ameaças;
 - b) definição clara dos riscos de segurança; e
 - c) nível de severidade que o comprometimento de dados sensíveis traria ao sistema e à instituição.
- II. Não se deve omitir, durante o projeto de desenvolvimento de sistema e sua execução, a definição de responsabilidades pela segurança de dados do sistema e como essa responsabilidade será verificada.
- III. Deve-se utilizar cronograma de projeto que contemple pontos de verificação de segurança do sistema desenvolvido ao longo de sua construção.

16.3.13. Atualização de softwares

- I. Não se deve habilitar as atualizações automáticas de software ou componentes utilizados na construção de um sistema, sob pena de introdução indevida de falhas de segurança.
- II. Somente utilize código de terceiros que sejam confiáveis.

17. REQUISITOS MÍNIMOS DE QUALIDADE E PADRONIZAÇÃO DOS ASPECTOS TÉCNICOS DA CODIFICAÇÃO

Os requisitos mínimos de qualidade e padronização de código devem ser integralmente observados pela contratada, salvo quando autorizado pela contratante.

Cada indicador de qualidade do código fonte possui meta a ser alcançada que estão relacionadas no quadro exemplificativo a seguir:

Tabela 48: Indicadores e metas de qualidade de código

Grupo	Indicador	Unidade	Meta
Projeto	<i>Complexity / file</i> ou equivalente	média total	<= 10
	<i>Complexity / class</i> ou equivalente	média total	<= 10
	<i>Complexity /function</i> ou equivalente	média total	<= 3
	<i>Duplications</i> ou equivalente	%	<= 4%
	<i>Security Issue Tags</i> ou equivalente	unidades	= 0
	<i>Technical Debt ratio</i> ou equivalente	%	<= 2,5%
	<i>SQALE RATING</i> ou equivalente	Nota	=A
Violações de código (possíveis bugs, estilo de codificação, más práticas de codificação)	<i>Critical Issues</i> ou equivalente	unidades	= 0
	<i>Blocker Issues</i> ou equivalente	unidades	= 0
Indicadores relacionados a testes	<i>Unit Tests Coverage</i> - camada negócio / Impl ou equivalente	%	>= 70%
	<i>Unit Test Success</i> ou equivalente	%	>= 100%
	<i>Skipped Tests</i> ou equivalente	unidades	= 0

As metas e indicadores constantes do quadro acima poderão ser ajustados pela organização e adequados a cada Ordem de Serviço. Caso a organização não informe um novo quadro na Ordem de Serviço ou no aditivo da Ordem de Serviço, deverá ser considerado o quadro acima.

18. ORIENTAÇÕES PARA O PROCESSO DE AMOSTRAGEM

O processo de amostragem para fins de verificação dos níveis mínimos de serviço deve observar ao menos as seguintes etapas:

- Definição do Tamanho amostral, que deve ser superior a 30 amostras.
- Seleção das amostras em diferentes funcionalidades ou módulos, orientando-se por aquelas de maior risco ou valor esperado ao usuário.
- Realização dos testes e análise sobre as amostras coletadas.
- Cálculo do valor percentual de atendimento do nível de serviço sobre o conjunto de amostras.

Se a diferença entre o indicador aferido no processo de amostragem for igual ou inferior a 5% ao limite do nível mínimo de serviço, deve-se ampliar a quantidade de amostras em no mínimo 30 unidades com vistas a certificar-se que a aferição não está situada na margem de erro.

19. CONCLUSÃO

VERSÃO CONSULTA PÚBLICA

20. COLABORADORES DA VERSÃO 3.0

As seguintes pessoas foram colaboradoras de texto, análise de dados, imagens e/ou conceitos e forneceram revisões e recomendações sobre as versões preliminares do Roteiro de Métricas do SISP - versão 3.0:

VERSÃO CONSULTA PÚBLICA

21. REFERÊNCIAS BIBLIOGRÁFICAS

- [Dekkers, 2003] DEKKERS, C. Measuring the “logical” or “functional” Size of Software Projects and Software Application. Spotlight Software, ISO Bulletin May 2003 pp10-13.
- [IFPUG, 2010] IFPUG. Counting Practices Manual. Version 4.3.1, January, 2010.
- [IFPUG, 2021] IFPUG. SFP Counting Practices Manual. Release 2.1, October, 2021, This is the first release of the IFPUG SFP published by IFPUG.
- Portaria SGD/MGI nº 750, de 20 de março de 2023, disponível em <https://www.gov.br/governodigital/pt-br/contratacoes/portaria-sgd-mgi-no-750-de-20-de-marco-de-2023>;
- SISP - Roteiro de Métricas de Software do SISP: versão 2.3 / Ministério do Planejamento, Desenvolvimento e Gestão. Secretaria de Tecnologia da Informação e Comunicação - Setic. — Brasília: MP, 2018. 96 p.: il. Disponível em <http://www.sisp.gov.br/metricas/wiki/download/file/Roteiro-M%99tricas2.3>
- Portaria SGD/ME Nº 6.432, DE 15 DE JUNHO DE 2021, disponível em https://www.in.gov.br/web/dou/-/portaria-sgd/me-n-6.432-de-15-de-junho-de-2021-*326528202).
- Orientações para Contratação e Renovação de Contratos Baseados em UST — Português (Brasil) (<https://www.gov.br/governodigital/pt-br/contratacoes/orientacoes-para-novas-contratacoes-e-renovacoes-de-contratados-baseados-em-ust>)
- Acórdão Nº 1508/2020 — TCU — Plenário;
- Manifesto ágil, disponível em <https://agilemanifesto.org/iso/ptbr/manifesto.html>
- Princípios por trás do Manifesto ágil, disponível em <https://agilemanifesto.org/iso/ptbr/principles.html>
- O Guia Scrum 2020, Guias Scrum.
- Caroli e Aguiar, Product Backlog Building: Um Guia Prático para Criação e Refinamento de Backlog para Produtos de Sucesso.
- Schwaber, K. e Sutherland, J. The Scrum Guide™ — O Guia Definitivo do Scrum: As Regras do Jogo.
- Engenharia de Software e Sistemas — Teste de Software — Parte 6: Diretrizes para o uso da ISO/IEC/IEEE 29119 em Projetos Ágeis.
- Catálogo UST do SERPRO, versão 14 de 04/02/2022

- ISO/IEC/IEEE 12207:2017, Engenharia de sistemas e software — Processos do ciclo de vida de software
- ISO/IEC/TR 24587:2021, Engenharia de software e sistemas — Desenvolvimento ágil — Considerações de adoção ágil
- ISO/IEC/IEEE 24765:2017, Engenharia de sistemas e software — Vocabulário
- ISO/IEC/IEEE 26515:2018, Engenharia de sistemas e software — Desenvolvendo informações para usuários em um ambiente ágil
- Rascunho final: ISO/IEC/DTR 7052, Engenharia de software – Controlando riscos que ocorrem frequentemente durante o desenvolvimento e manutenção de software personalizado
- Rascunho final: ISO_IEC_FDIS_33202, Engenharia de software e sistemas — Principais práticas ágeis
- User Story Essentials, Gartner
- Guia do Conhecimento em Gerenciamento de Projetos (Guia PMBOK). 7 ed. Estados Unidos: PMI, 2021. PROJECT MANAGEMENT INSTITUTE (PMI).
- Guia do Conhecimento em Gerenciamento de Projetos (Guia PMBOK). 6 ed. Estados Unidos: PMI, 2017. PROJECT MANAGEMENT INSTITUTE (PMI).
- M. Cohn, “Agile Estimating and Planning,” Prentice Hall, 2005.
- R. Jeffries, A. Anderson, C. Hendrickson, “Extreme Programming Installed”, Addison-Wesley, 2000.