

# ROTEIRO DE MÉTRICAS DE SOFTWARE DO **SISP**

VERSÃO 3.0

Técnicas de mensuração de software  
na Administração Pública Federal.



## MINISTÉRIO DA GESTÃO E DA INOVAÇÃO EM SERVIÇOS PÚBLICOS

**Esther Dweck**

Ministra da Gestão e da Inovação em Serviços Públicos

### SECRETARIA DE GOVERNO DIGITAL

**Rogério Souza Mascarenhas**

Secretário de Governo Digital

### DIRETORIA DE GESTÃO DE RECURSOS DE TECNOLOGIA DA INFORMAÇÃO

**Luiz Augusto Fonseca Monfardini**

Diretor de Gestão de Recursos de Tecnologia da Informação

### COORDENAÇÃO GERAL DE NORMAS DE AQUISIÇÕES DE TIC

**Cristiano Jorge Poubel de Castro**

Coordenador Geral de Normas de Aquisições de TIC

### COORDENAÇÃO GERAL DE ANÁLISE DE AQUISIÇÕES DE TIC

**Cícero Padilha de Almeida**

Coordenador Geral de Análise de Aquisições de TIC

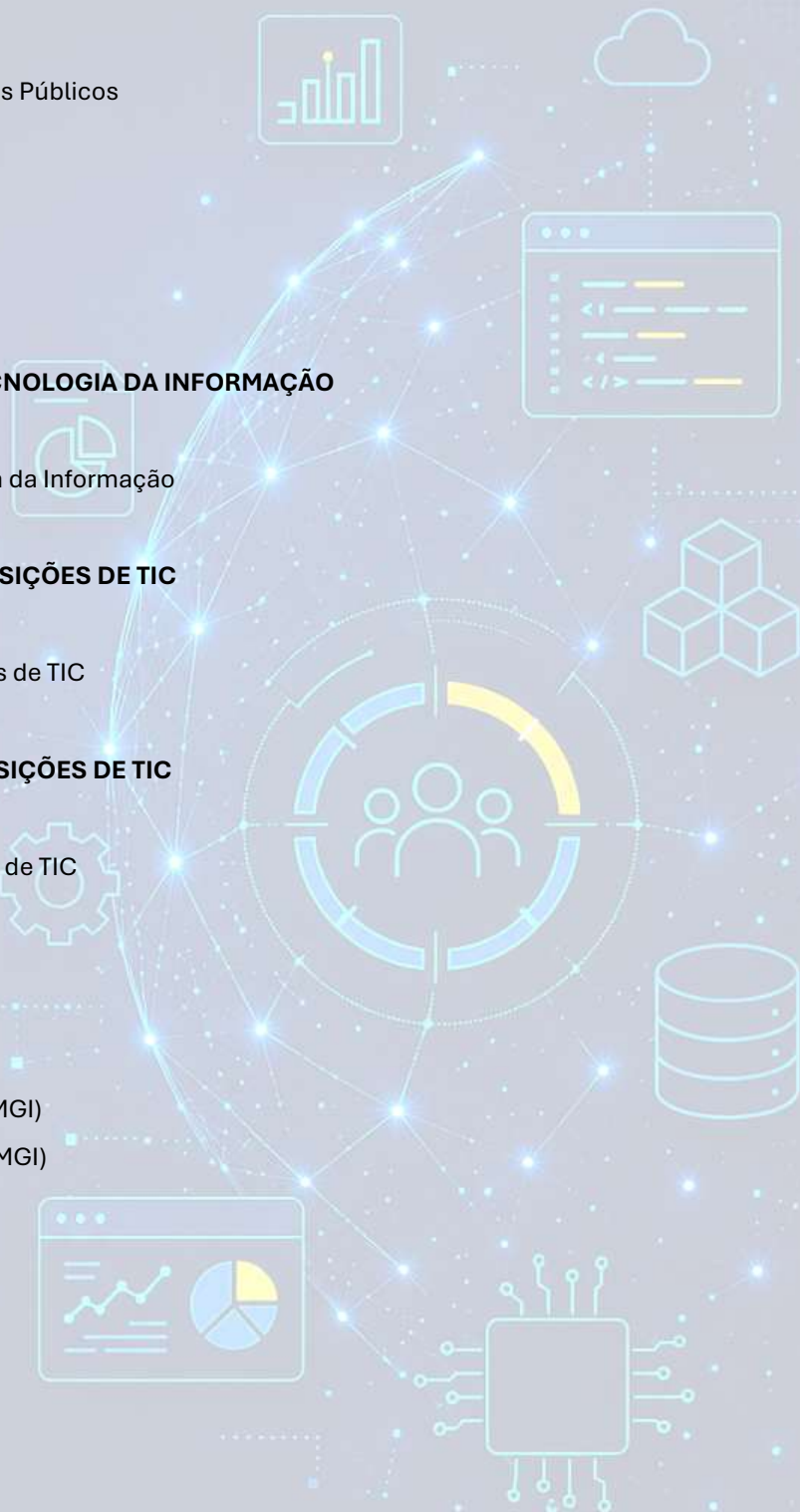
### EQUIPE DE ELABORAÇÃO DA VERSÃO 3.0

Cristiano Jorge Poubel de Castro (SGD/MGI)

Cícero Padilha de Almeida (SGD/MGI)

Loidianne Alves Marinho Silva Ramos (SGD/MGI)

Lucas Marcelo Ribeiro de Sousa Mota (SGD/MGI)



Ministério da Gestão e da Inovação em Serviços Públicos, Brasília, 2026

Qualquer parte desta publicação pode ser reproduzida para fins não comerciais, desde que citada a fonte e que as obras derivadas sejam compartilhadas sob a mesma licença, conforme os termos da Creative Commons Atribuição-Não Comercial-Compartilha Igual 4.0 Internacional (CC BY-NC-SA 4.0).

Disponível em: <https://www.gov.br/governodigital/pt-br/contratacoes-de-tic>

Normalização Bibliográfica: CGNAT/DEGTI/SGD/MGI

B823r

Brasil. Ministério da Gestão e da Inovação em Serviços Públicos.  
Secretaria de Governo Digital.

Roteiro de Métricas de Software do SISP: versão 3.0 / Ministério da  
Gestão e da Inovação em Serviços Públicos. Secretaria de Governo  
Digital - SGD. - Brasília: MGI, 2026.  
272 p.: il.

1. Tecnologia da Informação. 2. Roteiro de Métricas de Software. 3. Ponto de Função - Métricas. 4. Simple Function Point (SFP). 5. Story Points Padronizados (SPP). 6. Horas de Serviço Técnico (HST). 7. Sistema de Administração dos Recursos de Tecnologia da Informação - SISP. 8. Data Warehouse - Contagem de PF. 9. Inteligência Artificial - Mensuração. I. Título.

CDU 004.4  
CDD 658.403

# SUMÁRIO

1. INTRODUÇÃO .....	14
2. OBJETIVO DO ROTEIRO.....	17
3. ESCOPO DO ROTEIRO.....	18
4. TERMOS E DEFINIÇÕES.....	19
<b>PARTE I</b> 28	
<i>Análise por Pontos de Função (APF) e Simple Function Point (SFP)</i> .....	28
5. MÉTODO APF .....	29
5.1. PROCESSO DE CONTAGEM APF .....	30
6. MÉTODO SFP.....	42
6.1. PROCESSO DE CONTAGEM SFP .....	44
6.2. CORRESPONDÊNCIA E CONVERSIBILIDADE ENTRE OS MÉTODOS .....	49
6.3. EXEMPLO DE APLICAÇÃO DA CONVERSIBILIDADE.....	50
7. CÁLCULO DE PONTOS DE FUNÇÃO PARA O SISP .....	51
7.1. PROJETO DE DESENVOLVIMENTO .....	53
7.2. <i>BASELINE</i> APÓS DESENVOLVIMENTO INICIAL .....	53
7.3. PROJETO DE MELHORIA .....	54
7.4. <i>BASELINE</i> APÓS MELHORIA - MÉTODO SFP .....	56
7.5. <i>BASELINE</i> APÓS MELHORIA - MÉTODO APF .....	57
7.6. PROJETOS DE MIGRAÇÃO DE DADOS .....	57
7.7. MANUTENÇÃO CORRETIVA.....	58
7.8. MUDANÇA DE PLATAFORMA .....	60
7.8.1.      Mudança de Plataforma - Linguagem .....	60
7.8.2.      Mudança de Plataforma - Banco de Dados.....	61
7.9. ATUALIZAÇÃO DE VERSÃO .....	63
7.9.1.      Atualização de Versão - Linguagem .....	64
7.9.2.      Atualização de Versão - <i>Browser</i> .....	65
7.9.3.      Atualização de Versão - Banco de Dados.....	65
7.10. MANUTENÇÃO EM <i>INTERFACE</i> .....	66
7.11. ADAPTAÇÃO SEM ALTERAÇÃO DE REQUISITOS FUNCIONAIS.....	67
7.12. APURAÇÃO ESPECIAL .....	68
7.12.1.      Apuração especial - Base de dados.....	69

7.12.1.a.	Atualização de dados sem consulta prévia.....	70
7.12.1.b.	Consulta prévia sem atualização.....	70
7.12.1.c.	Atualização de dados com consulta prévia.....	71
7.12.2.	Apuração especial – Geração de Relatórios.....	71
7.12.3.	Apuração especial – Alteração .....	72
7.12.4.	Apuração especial – Reexecução.....	73
7.13.	ATUALIZAÇÃO DE DADOS.....	73
7.14.	DESENVOLVIMENTO, MANUTENÇÃO E PUBLICAÇÃO DE PÁGINAS ESTÁTICAS .....	74
7.15.	MANUTENÇÃO DE DOCUMENTAÇÃO DE SISTEMAS LEGADOS.....	75
7.16.	VERIFICAÇÃO DE ERROS .....	75
7.17.	PONTOS DE FUNÇÃO DE TESTE .....	76
7.18.	COMPONENTE REUSÁVEL.....	77
8.	ORIENTAÇÕES COMPLEMENTARES PARA CONTAGEM DE PF.....	79
8.1.	CONTAGEM COM MÚLTIPLAS MÍDIAS .....	79
8.1.1.	CENÁRIO 1: Mesmos dados apresentados em tela e impressos .....	81
8.1.2.	CENÁRIO 2: Dados de saída idênticos .....	81
8.1.3.	CENÁRIO 3: Dados de saída distintos .....	81
8.1.4.	CENÁRIO 4: Dados de entrada e saída batch e on-line idênticos.....	82
8.1.5.	CENÁRIO 5: Múltiplos canais de entrega da mesma funcionalidade.....	82
8.1.6.	CENÁRIO 6: Relatórios em múltiplos formatos.....	82
8.1.7.	CENÁRIO 7: Funcionalidades fornecidas via aplicação e <i>Web service</i> .....	83
8.1.8.	CENÁRIO 8: Desenvolvimento Multiplataforma .....	83
8.2.	LOG, TRILHA DE AUDITORIA E HISTÓRICO .....	83
8.2.1.	<i>Log</i> .....	83
8.2.2.	Trilha de auditoria .....	84
8.2.3.	Histórico .....	84
8.3.	ABAS E IDENTIFICAÇÃO DE PROCESSO ELEMENTAR.....	85
8.4.	PRINCIPAIS FALHAS DE CONTAGEM IDENTIFICADAS .....	86
8.4.1.	Dados de Código.....	87
9.	ORIENTAÇÕES PARA REALIZAR ESTIMATIVAS DE PROJETOS DE SW.....	90
9.1.	REUNIR A DOCUMENTAÇÃO DISPONÍVEL.....	91
9.2.	ESTIMATIVA DE TAMANHO COM SFP.....	92
9.2.1.	Exemplo de Estimativa de Tamanho com SFP.....	93
9.3.	ESTIMATIVA DE ESFORÇO .....	93
9.3.1.	Exemplo de Estimativa de Esforço .....	94
9.3.2.	Distribuição de Referência do Esforço por fase de projeto .....	95
9.4.	ESTIMATIVA DE PRAZO .....	97

9.4.1.	Passo 1: Definir a composição do time de referência .....	97
9.4.2.	Passo 2: Estimar a produtividade do time de referência.....	98
9.4.3.	Passo 3: Estimar o prazo do projeto .....	101
9.5.	ESTIMATIVA DE CUSTO .....	102
9.5.1.	Exemplo de Estimativa de Custo .....	103
9.6.	ESTIMATIVA DE RECURSOS DE TIC.....	106
10.	FATORES DE IMPACTO.....	107
10.1.	DEFINIÇÃO DOS FATORES DE IMPACTO (FI).....	109
10.2.	MEMÓRIA DE CÁLCULO DOS FATORES DE IMPACTO (FI).....	114
11.	CONTAGEM DE PROJETOS ÁGEIS COM SFP .....	121
11.1.	ORIENTAÇÕES GERAIS.....	123
11.2.	MUDANÇAS EM FUNCIONALIDADES.....	125
11.2.1.	Fatores que influenciam o número de mudanças .....	126
11.3.	REGISTRO E CONTROLE.....	127
11.3.1.	Exemplo de aplicação - Contagem Estimada da <i>Release N</i> .....	128
11.3.2.	Exemplo de aplicação - Contagem Final da <i>Release</i> .....	130
11.3.3.	Exemplo de aplicação - Contagem para a <i>Baseline</i> da Aplicação .....	132
11.4.	RECOMENDAÇÕES PARA O USO DO FATOR ÁGIL .....	133
11.4.1.	Exemplo de Remuneração da Contratada .....	134
11.5.	CANCELAMENTO DE PROJETOS ÁGEIS .....	134
12.	CONTAGEM DE PF EM CENÁRIOS ESPECÍFICOS.....	135
12.1.	CONTAGEM DE IA (Inteligência Artificial).....	135
12.1.1.	Determinar a Fronteira da Aplicação.....	136
12.1.2.	Escopo da contagem de IA .....	137
12.1.3.	Contagem para Processamento <i>Batch</i> de IA.....	138
12.1.4.	Contagem para a Execução Interativa de IA.....	145
12.1.5.	Manutenções de Soluções de IA.....	147
12.2.	CONTAGEM DE PF EM <i>CHATBOT</i> .....	147
12.2.1.	Visão <i>Chatbot</i> para contagem PF .....	148
12.2.2.	Escopo da contagem PF para <i>ChatBot</i> .....	150
12.2.3.	Contagem de funções de dados para <i>Chatbot</i> .....	150
12.2.4.	Contagem de funções transacionais para <i>Chatbot</i> .....	151
12.2.5.	Manutenção em soluções de <i>Chatbot</i> .....	152
12.2.6.	Itens sem contagem de PF em soluções de <i>Chatbot</i> .....	153
12.3.	CONTAGEM DE PF EM PAINÉIS ANALÍTICOS .....	153
12.3.1.	Criação e carga de tabelas .....	153
12.3.2.	Leitura de dados de outras aplicações .....	155

12.3.3.	Gráficos e tabelas dinâmicas .....	155
12.4.	CONTAGEM PF EM PROJETOS BASEADOS EM GEOTECNOLOGIAS.....	156
12.4.1.	Medição de camadas georreferenciadas .....	156
12.4.2.	Apresentação de dados em formatos diferentes.....	157
12.4.3.	Apresentação simultânea de vários elementos visuais.....	157
12.4.4.	Funcionalidades complexas de georreferenciamento .....	159
12.5.	CONTAGEM DE PF EM DESENVOLVIMENTO MULTIPLATAFORMA .....	161
12.5.1.	Plataforma de desenvolvimento híbrida .....	162
12.5.1.a.	Contagem de Projetos de Desenvolvimento.....	162
12.5.1.b.	Contagem de Projetos de Melhoria.....	163
12.5.2.	Plataforma de desenvolvimento nativa.....	163
12.5.2.a.	Contagem de Projetos de Desenvolvimento.....	163
12.5.2.b.	Contagem de Projetos de Melhoria.....	164
12.6.	CONTAGEM DE INGESTÃO DE DADOS PARA DATA LAKE.....	164
12.6.1.	Contagem de Ingestão <i>Full</i> .....	167
12.6.2.	Contagem de Ingestão Incremental.....	169
12.6.3.	Ingestão de dados com Utilitários .....	169
12.6.4.	Ingestão de dados com Testes não-funcionais .....	170
12.6.5.	Ingestão de dados com Criação de Arquivos no <i>Data Lake</i> .....	170
12.6.6.	Ingestão de dados com Controle de Qualidade .....	171
13.	CONTAGEM DE PF EM PROJETOS DE DW.....	172
13.1.	OBJETIVO.....	172
13.2.	MODELO DE ARQUITETURA DE REFERÊNCIA DE DATA WAREHOUSE/DATA MART .....	173
13.3.	ESTIMATIVA DE PROJETO DE DW.....	177
13.4.	VISÃO GERAL DA CONTAGEM DE PROJETO DW .....	179
13.5.	ESCOPO DA CONTAGEM E FRONTEIRA DA APLICAÇÃO .....	180
13.6.	CONTAGEM DE FUNÇÕES DE DADOS .....	181
13.6.1.	Contagem funções de dados internas do DW .....	181
13.6.2.	Contagem de funções de dados do processo ETL.....	184
13.6.3.	Considerações finais sobre funções de dados em projetos DW.....	185
13.7.	CONTAGEM DE FUNÇÕES TRANSACIONAIS.....	186
13.7.1.	Contagem de funções de carga de dados.....	186
13.7.2.	Contagem de funções de geração de relatórios ou gráficos.....	188
13.7.3.	Contagem de funções de extração de dados do Sistema Transacional de Origem.....	189
13.8.	PROJETO DE MELHORIA DE DW .....	195
13.9.	DEMANDAS TÍPICAS EM PROJETOS DE DW .....	195
13.9.1.	Alteração de Dados de Dimensões Estáticas .....	195

13.9.2.	Criação/Alteração de Campos em Tabelas Fato e Dimensão.....	196
13.9.3.	Alteração de Dados em Tabelas Fato e Dimensão.....	197
13.9.4.	Extração de Dados em Tabelas Fato e Dimensão .....	197
13.9.5.	Criação, Configuração e Disponibilização de um Filtro de Relatório.....	197
13.9.6.	Criação de Métricas .....	198
13.9.7.	Reorganização de Bancada .....	198
13.9.8.	Exclusão de Dados Antigos do DW .....	199
13.9.9.	Metadados Relacionados ao Controle do Processo de ETL .....	199
13.9.10.	Metadados Relacionados ao Significado dos Dados - Metadados do Negócio .....	200
13.9.11.	Migração de Banco de Dados.....	200
13.9.12.	Migração de ferramenta ETL .....	201
14.	ATIVIDADES SEM CONTAGEM DE PF .....	201
14.1.	ATIVIDADES SEM CONTAGEM DE PF EM PROJETOS DE DESENVOLVIMENTO DE SW .....	202
14.2.	ATIVIDADES SEM CONTAGEM DE PF EM PROJETOS DE DW .....	203
14.2.1.	Tabelas Sem Contagem de Ponto de Função.....	203
14.2.2.	Customização de Ferramenta OLAP .....	204
14.2.3.	Integração da Ferramenta OLAP ao AD/LDAP.....	204
14.2.4.	Customização da Ferramenta ETL.....	204
14.2.5.	Criação de usuários e perfis de acesso às ferramentas OLAP e ETL.....	204
14.2.6.	Automação do Processo de ETL.....	205
<b>PARTE II</b>	<b>206</b>	
	<b>Story Points Padronizados (SPP) .....</b>	<b>206</b>
15.	MÉTRICA STORY POINTS PADRONIZADOS (SPP) .....	207
15.1.	O QUE MUDA EM RELAÇÃO AO STORY POINTS TRADICIONAL .....	208
15.2.	ORIENTAÇÕES GERAIS.....	209
15.3.	BACKLOG DO PRODUTO ( <i>PRODUCT BACKLOG</i> ).....	210
15.3.1.	ITENS DO BACKLOG .....	211
15.3.2.	Refinamento contínuo do <i>Backlog</i> do Produto .....	212
15.4.	BACKLOG DA <i>SPRINT</i> .....	213
15.5.	HISTÓRIAS DE USUÁRIO ( <i>user stories</i> ).....	214
15.5.1.	Formato das Histórias do usuário.....	215
15.5.2.	Critérios de qualidade - INVEST.....	215
15.5.3.	Métodos adicionais para levantamento das Histórias de Usuário .....	217
15.5.4.	Critérios de Aceitação da História do Usuário .....	219
15.5.4.a.	Relação entre os critérios de aceitação da história e o SPP .....	219
15.5.4.b.	Qualidade dos critérios de aceitação.....	220
15.5.4.c.	Formatos dos critérios de aceitação.....	222

15.5.4.d.	Checklist para validação dos critérios de aceitação .....	224
15.5.4.e.	Riscos de medição decorrentes de critérios de aceitação inadequados .....	226
15.5.5.	Interface de usuário (ou UI, do inglês <i>User Interface</i> ) .....	228
15.5.6.	Tarefas ( <i>tasks</i> ) .....	229
15.5.7.	Definição de Preparado.....	231
15.6.	INCREMENTO DO PRODUTO.....	232
15.6.1.	Definição de Pronto .....	232
15.6.2.	Exemplos de Definição de Pronto.....	232
15.7.	CONTAGEM POR STORY POINTS PADRONIZADOS (SPP).....	234
15.7.1.	Entradas para o SPP .....	235
15.7.2.	Passos para a contagem por SPP .....	236
15.7.2.a.	Passo 1: Definir o Tipo Base da História .....	237
15.7.2.b.	Passo 2: Identificar os Fatores Técnicos .....	238
15.7.2.c.	Passo 3: Somar pontuação.....	241
15.7.2.d.	Passo 4: Aplicar faixa final automática .....	241
15.7.3.	Exemplos de aplicação da métrica SPP .....	242
15.7.3.a.	Exemplo 1: Aplicação do SPP para história de baixa complexidade.....	243
15.7.3.b.	Exemplo 2: Aplicação do SPP para história de média complexidade .....	244
15.7.3.c.	Exemplo 3: Aplicação do SPP para história de alta complexidade.....	245
15.7.4.	Tratamento de Melhorias.....	247
15.7.5.	Tratamento de Correção de Defeitos com SPP.....	248
15.7.6.	Tratamento de Atividades Técnicas na contagem com SPP .....	249
15.7.7.	O que contar e o que não contar com SPP .....	250
15.7.8.	Boas práticas na estimativa por SPP .....	251
15.7.9.	Boas práticas na definição do <i>Backlog</i> da <i>Sprint</i> .....	252
15.7.10.	Erros a serem evitados na contagem por SPP .....	252

### PARTE III

Horas De Serviço Técnico - HST.....	254
16. CONTAGEM DE HST .....	255
16.1.COMPOSIÇÃO DO CATÁLOGO DE SERVIÇOS TÉCNICOS .....	255
16.2.CUSTO DA HST.....	259
16.3.VALOR A SER REMUNERADO PELO SERVIÇO .....	259

### PARTE IV

Linhas de Código Funcionais (LoCF).....	265
17. MÉTRICA DE LINHAS DE CÓDIGO FUNCIONAIS (LoCF) .....	266
17.1.EXEMPLO DA APLICAÇÃO DA MÉTRICA LoCF .....	269

### PARTE V

<b>Seleção e Utilização das Métricas .....</b>	<b>271</b>
<b>18. COMO SELECIONAR A MÉTRICA A SER ADOTADA .....</b>	<b>272</b>
<b>19. PRÁTICAS DE ADOÇÃO E MONITORAMENTO DAS MÉTRICAS.....</b>	<b>275</b>
<b>19.1. PLANEJAMENTO DA ADOÇÃO .....</b>	<b>276</b>
<b>19.2. IMPLANTAÇÃO FORMAL .....</b>	<b>276</b>
<b>19.3. MONITORAMENTO E ACOMPANHAMENTO .....</b>	<b>276</b>
<b>19.4. GOVERNANÇA DA MEDIÇÃO .....</b>	<b>277</b>
<b>19.5. MELHORIA CONTÍNUA .....</b>	<b>278</b>
<b>20. USO DE FERRAMENTAS DE IA NA MENSURAÇÃO DE SOFTWARE.....</b>	<b>278</b>
<b>21. CONCLUSÃO.....</b>	<b>279</b>
<b>22. REFERÊNCIAS BIBLIOGRÁFICAS .....</b>	<b>281</b>

# TABELAS

TABELA 1 - REGRAS PARA CONTAGEM DE FUNÇÕES DE DADOS (FONTE: CPM 4.3.1) .....	34
TABELA 2- REGRAS PARA CONTAGEM DE FUNÇÕES TRANSACIONAIS (FONTE: CPM 4.3.1) .....	37
TABELA 3 - COMPLEXIDADE DAS FUNÇÕES DE DADOS (FONTE: CPM 4.3.1).....	39
TABELA 4 - COMPLEXIDADE FUNCIONAL DAS ENTRADAS EXTERNAS (FONTE: CPM 4.3.1).....	39
TABELA 5 - COMPLEXIDADE FUNCIONAL DAS CONSULTAS EXTERNAS E SAÍDAS EXTERNAS (FONTE: CPM 4.3.1) .....	39
TABELA 6 - CONTRIBUIÇÃO FUNCIONAL DOS TIPOS FUNCIONAIS (FONTE: CPM 4.3.1) .....	40
TABELA 7 - DIFERENÇA NA IDENTIFICAÇÃO DOS TIPOS FUNCIONAIS ENTRE OS MÉTODOS SFP E APF .....	46
TABELA 8 - PONTUAÇÃO DOS COMPONENTES FUNCIONAIS BÁSICOS DO MÉTODO IFPUG SFP .....	48
TABELA 9 - DIFERENÇAS ENTRE DADOS DE CÓDIGO E DADOS DE NEGÓCIO .....	87
TABELA 10 - TIPOS E EXEMPLOS DE DADOS DE CÓDIGO CONFORME CPM 4.3.1 .....	88
TABELA 11 - DISTRIBUIÇÃO DO ESFORÇO UTILIZANDO PROCESSOS TÉCNICOS DA NORMA ISO/IEC 12207 .....	96
TABELA 12 - EXEMPLO DE COMPOSIÇÃO MÍNIMA DE REFERÊNCIA PARA TIMES DE DESENVOLVIMENTO .....	98
TABELA 13 - CARGA HORÁRIA MENSAL DO TIME DE REFERÊNCIA .....	99
TABELA 14 - CÁLCULO DA ESTIMATIVA DE CUSTO MENSAL DO TIME DE REFERÊNCIA .....	104
TABELA 15 - ESTIMATIVAS DO PROJETO EXEMPLO .....	105
TABELA 16 - PROCESSOS TÉCNICOS DA NORMA ISO/IEC 12207 UTILIZADOS PARA DEFINIÇÃO DOS FIS .....	111
TABELA 17 - MEMÓRIA DE CÁLCULO DE PONTO DE FUNÇÃO PARA O SISP – PARTE 1.....	114
TABELA 18 - MEMÓRIA DE CÁLCULO DE PONTO DE FUNÇÃO PARA O SISP – PARTE 2.....	116
TABELA 19 - MEMÓRIA DE CÁLCULO DE PONTO DE FUNÇÃO PARA O SISP - PARTE 3 .....	118
TABELA 20 - PLANEJAMENTO DO <i>BACKLOG</i> DAS <i>SPRINTS</i> (N1, N2 E N3) DA <i>RELEASE N</i> .....	129
TABELA 21 - CONTAGEM SFP DA <i>RELEASE N</i> .....	131
TABELA 22 - CONTAGEM DE SFP DA <i>RELEASE N</i> PARA A BASELINE DA APLICAÇÃO .....	132
TABELA 23 - CONTAGEM PF DE DADOS DE ORIGEM .....	139
TABELA 24 - CONTAGEM PF PARA O MACROPROCESSO EXPLORAR DADOS .....	140
TABELA 25 - CONTAGEM PF PARA DADOS RESULTANTES.....	143
TABELA 26 - CONTAGEM PF PARA DISPONIBILIZAR DADOS.....	144
TABELA 27 - CONTAGEM PF PARA PREPARAR DADOS EM EXECUÇÃO INTERATIVA DE IA .....	146
TABELA 28 - CONTAGEM PF PARA CRIAR MODELOS EM EXECUÇÃO INTERATIVA DE IA.....	146
TABELA 29 - ARQUIVOS LÓGICOS DE UM <i>BOT</i> .....	151
TABELA 30 - FUNÇÕES PARA CRIAÇÃO DE CARGA DE TABELAS DE PAINÉIS ANALÍTICOS .....	154
TABELA 31 - FUNÇÃO PARA LEITURA DE DADOS DE OUTRAS APLICAÇÕES.....	155
TABELA 32 - FUNÇÃO PARA APRESENTAÇÃO DE GRÁFICOS OU TABELAS DINÂMICAS .....	156
TABELA 33 - CONTAGEM DE FUNÇÕES TRANSACIONAIS E DE DADOS DE UM SISTEMA DE GEO .....	160
TABELA 34 - CONTAGEM DE PF PARA INGESTÃO FULL NO <i>DATA LAKE</i> .....	167
TABELA 35 - PONTUAÇÃO DOS COMPONENTES FUNCIONAIS BÁSICOS DO MÉTODO IFPUG SFP .....	178

TABELA 36 - RESUMO DA CONTAGEM ESTIMADA DE PROJETO DE DW, UTILIZANDO O MÉTODO SFP .....	178
TABELA 37 - DIFERENÇA NA IDENTIFICAÇÃO DAS FUNÇÕES DE DADOS ENTRE OS MÉTODOS SFP E APF .....	179
TABELA 38 - ENTIDADES RECONHECIDAS EM UM MODELO DE DADOS MULTIDIMENSIONAL .....	181
TABELA 39 - RESUMO DA CONTAGEM DE TABELAS DE AGREGAÇÃO PARA OS MÉTODOS SFP E APF .....	184
TABELA 40 - RESUMO DA CONTAGEM DE FUNÇÕES DE CARGA .....	187
TABELA 41 - RESUMO DA CONTAGEM DE FUNCIONALIDADES DE GERAÇÃO DE RELATÓRIOS.....	189
TABELA 42 - RESUMO DA CONTAGEM DE FUNÇÕES DE EXTRAÇÃO DE DADOS DO SISTEMA DE ORIGEM.....	194
TABELA 43 - EXEMPLO DE CRIAÇÃO/ALTERAÇÃO DE CAMPOS EM TABELAS FATO E DIMENSÃO .....	196
TABELA 44 - EXEMPLO DE CRIAÇÃO/ALTERAÇÃO DE CAMPOS EM TABELA DIMENSÃO COMPARTILHADA .....	197
TABELA 45 - COMPARAÇÃO ENTRE SP TRADICIONAL E SPP .....	208
TABELA 46 - MÉTODOS DE LEVANTAMENTO DE REQUISITOS .....	218
TABELA 47 - QUALIDADE DOS CRITÉRIOS DE ACEITAÇÃO (REFERÊNCIA PARA SPP).....	221
TABELA 48 - <i>CHECKLIST</i> DE VALIDAÇÃO DOS CRITÉRIOS DE ACEITAÇÃO DA HISTÓRIA DO USUÁRIO .....	225
TABELA 49 - RISCOS DECORRENTES DE CRITÉRIOS DE ACEITAÇÃO INADEQUADOS.....	226
TABELA 50 - TIPOS BASE DE UMA HISTÓRIA DO USUÁRIO .....	237
TABELA 51 - FATORES TÉCNICOS DE REFERÊNCIA PARA A CONTAGEM POR SPP .....	239
TABELA 52 - FAIXA DE NORMALIZAÇÃO SPP .....	241
TABELA 53 - APLICAÇÃO DO SPP PARA HISTÓRIA DE BAIXA COMPLEXIDADE.....	243
TABELA 54 - APLICAÇÃO DO SPP PARA HISTÓRIA DE MÉDIA COMPLEXIDADE.....	245
TABELA 55 - APLICAÇÃO DO SPP PARA HISTÓRIA DE ALTA COMPLEXIDADE .....	246
TABELA 56 - RISCOS NO TRATAMENTO DE MELHORIAS DE FORMA DIFERENCIADA .....	247
TABELA 57 – TRATAMENTO DE CORREÇÃO DE DEFEITOS COM SPP.....	249
TABELA 58 – O QUE CONTAR E O QUE NÃO CONTAR COM SPP .....	250
TABELA 59 - ERROS A SEREM EVITADOS NA CONTAGEM POR SPP.....	253
TABELA 60 - DESCRIÇÃO DE CAMPOS DO CATÁLOGO DE SERVIÇOS TÉCNICOS .....	256
TABELA 61 - CAMPOS ADICIONAIS DO CATÁLOGO DE SERVIÇOS TÉCNICOS QUE DEVEM CONSTAR NO ETP.....	258
TABELA 62 - MAPA DE PESQUISA SALARIAL DE REFERÊNCIA (PORTARIA SGD/MGI Nº 750) .....	261
TABELA 63 - APOIO NA SELEÇÃO DE MÉTRICAS.....	272
TABELA 64 - BOAS PRÁTICAS PARA ADOÇÃO E MONITORAMENTO DE MÉTRICAS.....	<b>ERRO! INDICADOR NÃO DEFINIDO.</b>

# FIGURAS

FIGURA 1 - ESCOPO DAS MÉTRICAS CONTRATUAIS DE SOFTWARE .....	18
FIGURA 2 - PROCESSO DE CONTAGEM DE PONTOS DE FUNÇÃO CONFORME CPM 4.3.1 .....	30
FIGURA 3 - PROCESSO DE CONTAGEM DE PONTOS DE FUNÇÃO COM ATIVIDADES EM DESTAQUE.....	44
FIGURA 4 - PROCESSO DE DERIVAÇÃO DE ESTIMATIVAS DE PROJETOS DE SOFTWARE .....	90
FIGURA 5 - PROCESSOS DO CICLO DE VIDA DO SOFTWARE DA ISO/IEC 12207 (FONTE: ISO/IEC 12207) .....	108
FIGURA 6 - CICLO DE VIDA DO DESENVOLVIMENTO DE UM PROJETO DE IA.....	137
FIGURA 7 - VISÃO SIMPLIFICADA DO PROCESSAMENTO EM LOTE (BATCH) DE IA (ADAPTADA SERPRO, 2025) .....	138
FIGURA 8 - VISÃO SIMPLIFICADA DO PROCESSAMENTO INTERATIVO DE IA (ADAPTADA SERPRO, 2025) .....	145
FIGURA 9 - VISÃO SIMPLIFICADA DO FUNCIONAMENTO DE UM <i>CHATBOT</i> (ADAPTADA SERPRO, 2025) .....	149
FIGURA 10 - ELEMENTOS VISUAIS GEORREFERENCIADOS .....	158
FIGURA 11 - VISÃO SIMPLIFICADA DO PROCESSO DE INGESTÃO DE DADOS NO <i>DATA LAKE</i> .....	166
FIGURA 12 - MODELO DE ARQUITETURA DE REFERÊNCIA DE PROJETO DE DW / DM .....	174
FIGURA 13 - ENTIDADES DE UM ESQUEMA ESTRELA DE DATA WAREHOUSE.....	175
FIGURA 14 - VISÃO GERAL DA CONTAGEM DE PF COM OS MÉTODOS APF E SFP EM UM PROJETO DE DW/DM .....	180
FIGURA 15 - VISÃO DA FRONTEIRA DW .....	181
FIGURA 16 - EXEMPLO DE MODELO DE DADOS MULTIDIMENSIONAL – ESQUEMA ESTRELA .....	183
FIGURA 17 - VISÃO DA FRONTEIRA DW.....	185
FIGURA 18 - RESUMO DA CONTAGEM DE FUNÇÕES DE DADOS EM UM PROCESSO ETL .....	185
FIGURA 19 - CENÁRIO 1 – FLAT FILES SENDO UMA CÓPIA DE DADOS DE TABELA DO SISTEMA DE ORIGEM.....	190
FIGURA 20 - CENÁRIO 2 – GERAÇÃO DE FLAT FILES NA FRONTEIRA DO SISTEMA DE ORIGEM.....	191
FIGURA 21 - CENÁRIO 3 – INTERFACE DIRETA ENTRE A BASE DE DADOS DO SISTEMA DE ORIGEM E O DSA .....	192
FIGURA 22 - CARGA DE DADOS DO SISTEMA DE ORIGEM NA DSA .....	193
FIGURA 23 - CARGA DE DADOS DO SISTEMA DE ORIGEM NO ODS .....	194
FIGURA 24 - ALINHAMENTO DE TERMOS DO <i>BACKLOG</i> DO PRODUTO .....	212
FIGURA 25 - FLUXO DO PBI – DEFINIÇÃO DE PREPARADO E DEFINIÇÃO DE PRONTO .....	214
FIGURA 26 - UI DE BAIXA FIDELIDADE .....	229
FIGURA 27 - FLUXO DE CONTAGEM DA MÉTRICA SPP .....	236
FIGURA 28 - ADOÇÃO E MONITORAMENTO DAS MÉTRICAS .....	275

# 1. INTRODUÇÃO

As métricas de software são importantes instrumentos que auxiliam na tomada de decisões nas atividades de engenharia de software, especialmente no planejamento, na construção, no aprimoramento e na sustentação de sistemas de informação. Nesse contexto, as contratações de serviços de desenvolvimento, manutenção e sustentação de software, devem adotar métricas objetivas que permitam a mensuração dos produtos de software efetivamente entregues pela empresa contratada, em consonância com o processo de desenvolvimento de software adotado pelo órgão ou entidade, com vistas a assegurar uma gestão contratual eficiente, com foco na entrega de produtos de software seguros e de qualidade, assegurando a adequada remuneração pelos serviços prestados.

A Portaria SGD/MGI nº 750, de 20 de março de 2023, e suas atualizações estabelecem o modelo para a contratação de serviços de desenvolvimento, manutenção e sustentação de software no âmbito dos órgãos e entidades integrantes do SISP, orientando que a aferição e a entrega de produtos sejam realizadas por meio de métricas de software, independentemente da modalidade de remuneração adotada.

Esse modelo apresenta uma seção específica sobre mensuração de software (12. MENSURAÇÃO DE SOFTWARE), reforçando a importância e a obrigatoriedade do uso de métricas de software nas contratações em todos os tipos de contratos dessa natureza, conforme destacado 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 funcionais implementadas;*

*c) Pontos de história Padronizados (Story Point Padronizados - SPP);*

O Roteiro de Métricas do SISP, em sua versão 3.0, apresenta avanços relevantes com o objetivo de apoiar os órgãos na utilização de métricas para aferição de produtos de software. Sua finalidade é disponibilizar aos gestores de TIC uma ampla gama de opções de métricas e recursos para utilização em diferentes tipos de projetos e contratos de desenvolvimento, manutenção e sustentação de software. Nesse sentido, o roteiro apresenta e detalha as seguintes métricas:

- a) o método tradicional de Análise de Pontos de Função (APF), descrito no Manual de Práticas de Contagem de Pontos de Função (CPM 4.3.1);
- b) o método Simple Function Point (SFP), descrito no SPM 2.1 do IFPUG, que permite avaliar o tamanho funcional de forma mais simples, ágil e objetiva;
- c) o método Story Points Padronizados (SPP), adaptado de métricas ágeis, com foco na mensuração padronizada em ambientes fluentes em técnicas de desenvolvimento e em processos ágeis;
- d) o método Linhas de Código Funcionais (LoCF), uma evolução do método clássico de contagem de linhas de código, com adaptações estruturais para a mensuração efetiva de incrementos funcionais de software;
- e) o método de Horas de Serviço Técnico (HST), métrica auxiliar para a contabilização de produtos complementares e derivados de atividades de diferentes perfis profissionais que integram o ciclo de entrega de produtos de software.

O método APF é amplamente utilizado no âmbito do SISP e apresenta como principal benefício a independência em relação à tecnologia adotada. O método SFP, por sua vez, mantém os conceitos fundamentais de processos elementares e arquivos lógicos da APF, preservando a independência tecnológica, ao mesmo tempo em que proporciona maior simplicidade, rapidez e previsibilidade no processo de contagem, reduzindo a curva de aprendizado dos profissionais envolvidos.

Para ambos os métodos (APF e SFP), foram incorporadas orientações específicas para contagem de projetos de desenvolvimento e melhoria em contextos como Inteligência Artificial (IA), Chatbots, Geoprocessamento e Painéis Analíticos, além de diretrizes para contagem em Data Warehouse (DW) e Data Lake.

Além das métricas baseadas em pontos de função (APF e SFP), este roteiro apresenta a métrica SPP (Story Points Padronizados), estruturada para estimativa de esforço e mensuração de produtividade, baseada na atribuição de pontuação a histórias de usuário por meio de critérios objetivos e verificáveis.

De forma complementar às métricas vinculadas a produtos de software, o roteiro padroniza a utilização da métrica de Horas de Serviço Técnico (HST), baseada na quantidade de horas necessárias para a execução de atividades técnicas associadas ao processo de desenvolvimento de software, assegurando a vinculação à entrega de produtos aferíveis e relevantes à solução. Este documento apresenta orientações para cálculo, remuneração e aplicação da HST, com o objetivo de apoiar os órgãos do SISP na adequada mensuração de serviços técnicos eventuais, contribuindo para a mitigação de riscos apontados no Acórdão nº 1508/2020 – TCU – Plenário, tais como:

- antieconomicidade decorrente de superestimação de esforços, produtos ou perfis profissionais;

- criação de unidades de medida de forma unilateral sem a supervisão do órgão central do SISP e sem padronização;
- adoção de critérios subjetivos de mensuração;
- compartilhamento de metodologias sem consistência ou justificativas técnicas e econômicas;
- ausência de comparabilidade de preços, heterogeneidade e assimetria de informações entre a administração e o mercado.

O Roteiro de Métricas 3.0 foi reestruturado para ampliar o potencial de utilização das métricas de software em contratos de desenvolvimento, manutenção e sustentação de software, considerando a evolução tecnológica observada nos últimos anos. Nesse contexto, o uso de recursos de Inteligência Artificial, associado a processos bem definidos, práticas de engenharia de software e equipes qualificadas, pode contribuir para ganhos de produtividade, qualidade e previsibilidade na entrega de produtos de software.

Por fim, este roteiro de métricas substitui as versões anteriores publicadas no âmbito das Portarias SLTI/MPOG nº 31, de 2011, e STI/MP nº 04, de 2017, ampliando o uso de diferentes métricas de software e incorporando práticas atuais de engenharia de software, desenvolvimento ágil, mensuração de tecnologias emergentes e diretrizes de codificação segura.

Espera-se que este roteiro instituído pela Portaria SGD/MGI nº 3656, de 2026, ao consolidar práticas padronizadas de mensuração de produtos de software nas contratações dos serviços previstos no escopo da Portaria SGD/MGI nº 750, de 2023, amplie a previsibilidade, a transparência e o embasamento baseado em dados para a tomada de decisões estratégicas. Espera-se, ainda, que sua adoção favoreça a entrega de soluções mais seguras, garantindo maior qualidade aos softwares que sustentam as diversas políticas públicas brasileiras.

## 2. OBJETIVO DO ROTEIRO

O objetivo deste documento é apresentar e padronizar a adoção de métricas objetivas que promovam maior eficiência na gestão contratual, na mensuração dos produtos de *software* implementados e na adequada remuneração dos serviços e produtos efetivamente entregues pela empresa contratada, independentemente da modalidade de remuneração adotada.

O roteiro orienta-se pelos seguintes princípios:

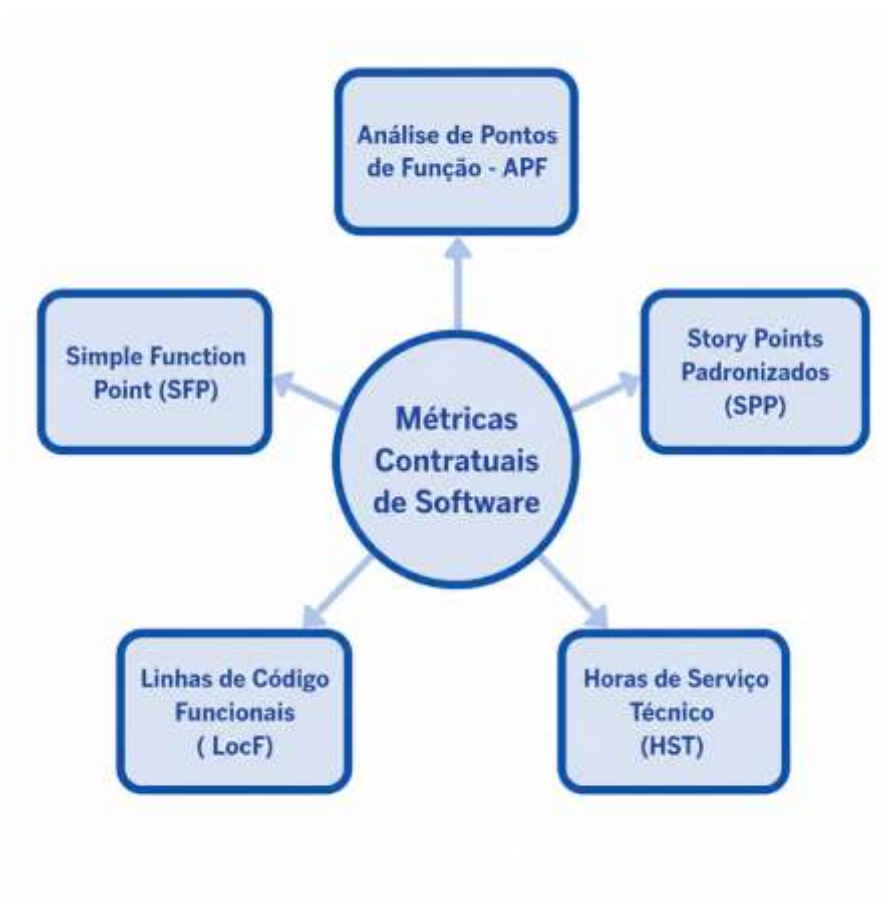
- **Objetividade:** apresentação de métricas objetivas, comparáveis, rastreáveis e transparentes para a mensuração de *software*;
- **Padronização:** adoção e compartilhamento de métricas padronizadas, confiáveis e baseadas em normas técnicas internacionais, bem como em metodologias consistentes e amplamente utilizadas no mercado;
- **Transparência dos custos:** consolidação da cultura de mensuração de produtos de *software* desenvolvidos, mantidos e sustentados, de modo a subsidiar a tomada de decisão quanto aos investimentos públicos em sistemas de informação no contexto da implementação de políticas públicas;
- **Conformidade:** observância das diretrizes estabelecidas na Portaria SGD/MGI nº 750/2023 e suas atualizações, bem como das recomendações e determinações constantes no Acórdão nº 2.037/2019 – TCU – Plenário, no Acórdão nº 1.508/2020 – TCU – Plenário e na jurisprudência consolidada sobre mensuração e desenvolvimento de *software* na administração pública federal;
- **Vinculação a resultados:** as métricas de softwares desempenham um papel fundamental em auxiliar a gestão e a tomada de decisão durante o processo de construção, sustentação e aprimoramento de sistemas de informação, sempre com o foco no alinhamento entre os recursos aplicados aos serviços e os resultados esperados em termos de produtos de software com qualidade.

### 3. ESCOPO DO ROTEIRO

O roteiro aplica-se aos órgãos e entidades integrantes do Sistema de Administração dos Recursos de Tecnologia da Informação (SISP), podendo ser utilizado como referência na definição, estimativa, mensuração e remuneração de serviços de software, conforme as diretrizes estabelecidas na Instrução Normativa SGD/ME nº 94, de 2023, Portaria SGD/MGI nº 750/2023 e na Portaria SGD/MGI nº 3656/2026.

Este documento abrange a utilização de diferentes métricas de software, incluindo Análise de Pontos de Função (APF), Simple Function Point (SFP), Story Points Padronizados (SPP), Horas de Serviço Técnico (HST) e Linhas de Código Funcionais (LoCF); bem como orientações complementares para a aplicação dessas métricas em contextos específicos, tais como Inteligência Artificial, *Chatbots*, Geoprocessamento, Painéis Analíticos, *Data Warehouse* e *Data Lake*.

**Figura 1 - Escopo das Métricas Contratuais de Software**



A aplicação das métricas deve observar o contexto de cada contratação, os processos de desenvolvimento adotados no órgão ou entidade e as definições estabelecidas nos instrumentos convocatórios.

## 4. TERMOS E DEFINIÇÕES

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

**Abordagem de desenvolvimento:** é 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:** é o conjunto de cultura organizacional, infraestrutura e metodologias que apoiam o desenvolvimento ágil (ISO/IEC/IEEE 26515:2018, item 3.2).

**Análise de decisão envolvendo critérios múltiplos:** é uma 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 por meio da análise da documentação existente e da identificação das informações relevantes aos requisitos.

**Análise de Pontos de Função:** é um método de medição de tamanho funcional de *software* definido pela ISO/IEC 14143-1:2007, ISO/IEC 20926:2009, bem como outros métodos reconhecidos, derivados desses padrões internacionais, como COSMIC (ISO/IEC 19761:2011), *Netherlands Software Metrics Association* (NESMA) ou *Simple Function Point* (SFP) do 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:** é um grupo lógico de dados, mantido ou referenciado pela aplicação, utilizado para atender aos requisitos de armazenamento de dados internos ou externos.

**Backlog do Produto:** representa o conjunto de itens necessários 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 atual de uma aplicação, representando o conjunto de funções disponíveis ao usuário em um determinado momento.

**Benchmarking:** envolve a comparação de produtos, processos e práticas reais ou planejadas adotadas por outras organizações e entidades para identificar as melhores práticas, gerar ideias para melhorias e fornecer uma base para medir o desempenho dos processos de desenvolvimento de software da organização.

**Brainstorming:** é uma 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. A técnica é composta 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:** corresponde a instruções de computador e definições de dados, expressas em uma forma adequada para entrada em um montador, compilador ou outro tradutor.

**Complexidade Funcional:** corresponde ao grau de complexidade associada a uma função utilizando as regras definidas no CPM 4.3.1.

**Componente Funcional Básico (CFB):** corresponde a unidade elementar de Requisitos Funcionais do Usuário definida e utilizada 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:** corresponde a 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):** é um conjunto de condições que precisam ser atendidas antes das entregas serem aceitas.

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

**Desenvolvimento ágil:** é uma 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:** é o método de desenvolvimento Test-Driven Development (TDD), no qual os desenvolvedores escrevem testes unitários antes de implementar a funcionalidade correspondente (NEN NPR 5326:2019).

**DevOps:** é o conjunto de princípios e práticas que promovem a integração e colaboração entre as partes envolvidas no desenvolvimento e operação de software, com o objetivo de especificar, desenvolver, implantar e operar produtos e serviços de forma contínua ao longo do ciclo de vida (IEEE 2675-2021, item 3.1).

**Diagrama de contexto:** é uma representação visual de alto nível que define o escopo do produto e os limites de um sistema, evidenciando sua interação com o ambiente externo. Apresenta o sistema de negócio (processo, equipamentos, sistema computacional etc.) como uma entidade, bem como os agentes externos (pessoas ou outros sistemas) que interagem com ele.

**Diagramas de afinidade:** é uma técnica de organização visual utilizada para estruturar grandes volumes de ideias, opiniões ou dados qualitativos, por meio do agrupamento em categorias com base em relações de similaridade, com o objetivo de apoiar a análise, a identificação de padrões e a tomada de decisão.

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

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

**Entrevistas:** é uma técnica de elicitación utilizada para obter informações sobre requisitos de alto nível, premissas ou restrições, critérios de aprovação ou outras informações relevantes. Consiste na realização de perguntas — estruturadas ou abertas — com o devido registro das respostas. Geralmente conduzida de forma individual, entre entrevistador e entrevistado, podendo também envolver múltiplos participantes.

**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:** corresponde 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:** é uma função transacional ou de dados destinada à transformação, migração ou adequação de dados, conforme requisitos de conversão definidos pelo usuário.

**Grupos de discussão:** é uma técnica de elicitación de requisitos que reúne partes interessadas pré-qualificadas e especialistas para explorar percepções, expectativas e atitudes em relação a um produto, serviço

ou resultado proposto, em um formato estruturado de interação coletiva, mais dinâmico e colaborativo do que entrevistas individuais.

**História de usuário:** é uma descrição, em linguagem natural, de uma necessidade ou funcionalidade do sistema, sob a perspectiva do usuário ou de outras partes interessadas, utilizada para apoiar o entendimento e a definição de requisitos.

**Horas de Serviço Técnico (HST):** corresponde à 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 (International Function Point Users Group):** organização internacional responsável pela padronização, evolução e disseminação da Análise de Pontos de Função (APF).

**Implantação:** é o processo de disponibilização do sistema ou de funcionalidades aos usuários em ambiente de produção, incluindo a migração ou carga de dados e a integração com outros sistemas no ambiente operacional.

**Implementação:** é o 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, por meio de técnicas e disciplinas de desenvolvimento de *software*. Esse processo resulta em um elemento de *software* que segue uma arquitetura e *design* estabelecidos.

**Incremento de produto:** é a versão do produto que agrega funcionalidades concluídas e potencialmente utilizáveis, passível de liberação ao final de um período definido (timebox).

**Independente:** corresponde à característica de um requisito funcional do usuário que pode ser implementado, testado e entregue sem dependência de processamento anterior ou subsequente.

**Informações:** são dados estruturados e processados para uma finalidade específica, de modo a torná-los significativos, relevantes e úteis em determinado contexto.

**Joint Application Design/Development (JAD):** é a técnica estruturada de elicitação de requisitos que consiste na realização de sessões facilitadas, reunindo especialistas de negócio e equipe de desenvolvimento, com o objetivo de coletar, validar e refinar requisitos e apoiar a melhoria do processo de desenvolvimento de software.

**Manutenção Adaptativa (ISO/IEC 14764:2006):** é a modificação de um produto de software, realizada após a entrega, com o objetivo de mantê-lo utilizável em ambientes operacionais alterados ou em evolução, incluindo adaptações a mudanças tecnológicas, regulatórias ou de integração.

**Manutenção Corretiva (ISO/IEC 14764:2006):** é a modificação de um produto de software, realizada após a entrega, para corrigir defeitos identificados, com o objetivo de restaurar o atendimento aos requisitos especificados.

**Manutenção Perfectiva (ISO/IEC 14764:2006):** é a modificação de um produto de software, realizada após a entrega, com o objetivo de melhorar seu desempenho, manutenibilidade ou outros atributos de qualidade, bem como aperfeiçoar funcionalidades existentes.

**Manutenção Preventiva (ISO/IEC 14764:2006):** é a modificação de um produto de software, realizada após a entrega, com o objetivo de detectar e corrigir falhas latentes, prevenindo a ocorrência de defeitos futuros.

**Mapa de Empatia:** é a ferramenta utilizada para sintetizar o perfil do usuário ou cliente, a partir da análise de seus objetivos, comportamentos, percepções e necessidades em relação a um problema ou contexto, apoiando a identificação de oportunidades e a tomada de decisão.

**Mapeamento de Histórias (Story Mapping):** é a técnica de representação bidimensional do backlog, que organiza histórias de usuário de forma lógica e visual, permitindo sua priorização, agrupamento e compreensão das relações e do fluxo funcional do produto.

**Mapeamento Mental:** é a técnica de organização visual de ideias, frequentemente utilizada no contexto de brainstorming, que consolida e estrutura contribuições individuais em um mapa único, evidenciando relações, convergências e lacunas, com o objetivo de apoiar a análise e a geração de novas ideias.

**Medição:** corresponde ao processo de determinação de um valor para um atributo, por meio da aplicação de uma escala de referência, bem como ao resultado dessa determinação.

**Método Delphi:** é a técnica estruturada de coleta e consolidação de opiniões de especialistas, realizada de forma iterativa e, geralmente, anônima, com o objetivo de alcançar consenso sobre determinado tema.

**Método de Dimensionamento de Camisetas (T-Shirt Sizing):** é o método de estimativa relativa que classifica itens, como histórias de usuário, em categorias de tamanho (por exemplo, extra pequeno, pequeno, médio, grande e extragrande), com o objetivo de apoiar comparações rápidas, evitando falsa precisão.

**Método de Medição:** corresponde à sequência lógica e estruturada de operações e regras utilizadas para realizar medições e produzir medidas consistentes.

**Metodologias ágeis:** correspondem ao conjunto de práticas e princípios voltados à entrega contínua e incremental de valor, com ênfase em colaboração, inspeção e adaptação frequentes.

**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:** correspondem às técnicas de elicitação que consistem na análise direta do comportamento dos usuários em seu ambiente de trabalho e na interação com eles, permitindo compreender como executam suas atividades e identificar requisitos, inclusive aqueles não explicitados. Podem incluir a observação passiva (“*job shadowing*”) ou a participação ativa do observador na execução das tarefas.

**Parte Interessada / Stakeholder:** corresponde ao indivíduo, grupo ou organização que pode afetar, ser afetado ou perceber-se afetado por decisões, atividades ou resultados de um projeto, programa ou portfólio.

**Personas:** corresponde à técnica que utiliza perfis fictícios, baseados em dados reais, para representar grupos de usuários, com o objetivo de apoiar o entendimento de suas necessidades, comportamentos e objetivos, orientando decisões de design e desenvolvimento do produto.

**Pesquisa Exploratória:** corresponde à técnica de investigação utilizada para compreender o contexto do problema, permitindo levantar informações iniciais sobre usuários, processos, ambientes e demais elementos relevantes ao desenvolvimento do produto ou serviço.

**PF\_INCLUÍDO ou SFP\_INCLUÍDO:** corresponde aos pontos de função associados às novas funcionalidades que farão parte da aplicação, decorrentes de projeto de desenvolvimento ou manutenção.

**PF\_ALTERADO ou SFP\_ALTERADO:** corresponde aos pontos de função associados às funcionalidades existentes na aplicação que sofrem alteração em projeto de manutenção.

**PF\_EXCLUÍDO ou SFP\_EXCLUÍDO:** corresponde aos pontos de função associados às funcionalidades existentes na aplicação que serão removidas/excluídas em projeto de manutenção.

**PF\_CONVERSÃO ou SFP\_CONVERSAO:** corresponde aos pontos de função associados às funcionalidades de conversão de dados em projetos de desenvolvimento ou manutenção. Incluem, por exemplo, processos de migração ou carga inicial de dados para popular as novas tabelas criadas e relatórios associados à migração de dados, quando requisitados pelo usuário. Os dados carregados em um processo de migração não devem ser contados como Arquivos Lógicos da aplicação.

**PF\_REFINADO ou SFP\_REFINADO:** corresponde aos pontos de função associados às alterações realizadas em funcionalidades já tratadas na mesma *release* (*release* corrente), decorrentes do detalhamento, refinamento ou complementação de requisitos ao longo do processo de desenvolvimento.

**Produto de Software ou Software:** corresponde ao conjunto de programas, procedimentos, rotinas, scripts e componentes, incluindo interfaces (*Application Programming Interface* – API), *web services*, dados e documentação associada.

**Produto com Mínima Viabilidade (MVP):** corresponde à versão inicial de um produto que possui funcionalidades suficientes para atender às necessidades básicas dos usuários e possibilitar a coleta de feedback para sua evolução.

**Produto pronto (Definição de Pronto):** corresponde ao estado do incremento quando atende aos critérios de qualidade, aceitação e níveis de serviço estabelecidos para o produto, devendo ser compreendida e observada por todo o time ágil.

**Projeto:** corresponde ao esforço temporário empreendido para criar um produto, serviço ou resultado único.

**Projeto ágil:** corresponde ao projeto de desenvolvimento de *software* que adota abordagem de desenvolvimento ágil, baseada em ciclos iterativos e incrementais, com foco na entrega contínua de valor.

**Proprietário/Dono do Produto (Product Owner):** corresponde ao responsável por representar os interesses do negócio e da contratante, definir e priorizar as funcionalidades do produto e estabelecer critérios de aceitação, assegurando o alinhamento com os objetivos do projeto.

**Protótipo:** corresponde ao modelo funcional ou representacional do produto, utilizado para explorar e validar requisitos de forma preliminar, antes de sua implementação definitiva.

**Prototipagem:** é a técnica utilizada para elaborar protótipos com o objetivo de obter respostas iniciais sobre requisitos, por meio da criação iterativa de modelos do produto. Pode envolver desde representações simples até simulações ou modelos digitais (2D/3D), permitindo a experimentação pelos usuários, a coleta de *feedback* e o refinamento progressivo dos requisitos.

**Qualidade de Software:** corresponde à capacidade de um produto, serviço, sistema, componente ou processo de atender às necessidades, expectativas e requisitos dos usuários e demais partes interessadas. No contexto de software e sistemas de TI, a qualidade pode ser avaliada por meio de características como confiabilidade, segurança, usabilidade, adequação funcional, manutenibilidade, portabilidade, eficiência de desempenho e compatibilidade, conforme a ISO/IEC 25010.

**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 se requer resposta rápida, 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 pelo desenvolvedor(es) de *software*.

**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:** corresponde à liberação de um ou mais incrementos de produto para uso por clientes ou usuários, conforme planejamento previamente definido, podendo agrupar uma ou mais *sprints*.

**Requisito:** corresponde à condição ou capacidade que deve estar presente em um produto, serviço ou resultado, de modo a atender a uma necessidade de negócio.

**Requisitos das Partes Interessadas:** são requisitos que descrevem as necessidades, expectativas e condições expressas por uma ou mais partes interessadas.

**Requisitos de Qualidade:** correspondem às condições e critérios necessários para verificar e validar a conformidade de uma entrega com os padrões estabelecidos, assegurando o atendimento aos requisitos do projeto. Incluem, por exemplo, critérios de teste, validação, certificação e níveis de serviço.

**Requisitos de Negócio:** correspondem às necessidades de alto nível da organização, incluindo problemas, oportunidades e objetivos estratégicos que justificam a realização de um projeto.

**Requisitos de Projeto:** correspondem às condições, restrições e obrigações que devem ser observadas na execução do projeto, tais como prazos, marcos, requisitos contratuais e limitações operacionais.

**Requisitos de Solução:** correspondem às funcionalidades, atributos e características do produto, serviço ou resultado que atendem aos requisitos de negócio e das partes interessadas, sendo classificados em requisitos funcionais e não funcionais.

**Requisitos do Usuário:** representam as necessidades expressas pelos usuários, descrevendo o que se espera que o sistema realize sob sua perspectiva.

**Requisitos Funcionais:** correspondem ao conjunto de requisitos que definem o comportamento do sistema, especificando as funções, serviços, processos, dados e interações que o *software* deve executar.

**Requisitos Não Funcionais:** são os requisitos que definem as características e restrições relacionadas à qualidade, desempenho, segurança, usabilidade, tecnologia e ambiente de operação do software, não estando diretamente associados às funções executadas pelo sistema.

**Reunião Diária (Daily):** é a reunião curta e periódica realizada pelo time ágil para inspecionar o progresso, alinhar as atividades e identificar impedimentos.

**Roadmap ou Visão do Produto:** representa o plano de alto nível que orienta a evolução do produto ao longo do tempo, apresentando marcos, entregas e direcionadores estratégicos.

**Simple Function Point (SFP):** é a métrica de dimensionamento funcional simplificada, baseada na identificação de processos elementares e arquivos lógicos, utilizada para estimar o tamanho funcional com maior agilidade e menor nível de detalhamento.

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

**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”.

**SPM (Simple Function Point Method – Counting Practices Manual):** é o manual de práticas de contagem do método de Pontos de Função Simples, que estabelece diretrizes para a aplicação do SFP na mensuração do tamanho funcional de software.

**Sprint:** é um ciclo de iteração, com duração limitada (tipicamente de até 4 semanas), em que um conjunto acordado de histórias de usuário ou funcionalidades é planejado, desenvolvido, testado e validado, resultando em incrementos aptos à entrega.

**Storyboarding:** corresponde à técnica de prototipagem que representa visualmente a sequência de navegação ou interação por meio de imagens ou ilustrações. No contexto de *software*, é utilizada para demonstrar fluxos de uso e caminhos de navegação em interfaces, como páginas *web* ou telas de sistemas.

**Story Points Padronizados (SPP):** é a métrica estruturada para estimativa de esforço e mensuração de produtividade, baseada na atribuição de pontos a histórias de usuário a partir de critérios objetivos, verificáveis e padronizados.

**Tamanho Funcional (ISO/IEC 14143-1:2007 (R2019)):** corresponde ao tamanho do *software* obtido pela quantificação dos requisitos funcionais do usuário.

**Técnica de Grupo Nominal:** é a técnica estruturada de geração e priorização de ideias que, no contexto de *brainstorming*, incorpora etapas de registro individual, compartilhamento e votação, com o objetivo de ordenar e selecionar as contribuições mais relevantes.

**Time/Equipe Ágil:** é o grupo multifuncional, geralmente composto por 3 a 10 pessoas, que colabora de forma integrada no desenvolvimento de um produto, compartilhando responsabilidades pela entrega de valor em ciclos iterativos.

**Timebox:** corresponde ao período de tempo fixo e previamente definido, durante o qual atividades são executadas com o objetivo de alcançar uma meta específica.

**Visão do Usuário:** representa a perspectiva do usuário sobre o produto, expressa por meio de suas necessidades, expectativas e requisitos funcionais.

**Votação:** é a técnica de tomada de decisão coletiva utilizada para avaliar alternativas, permitindo classificar, selecionar e priorizar opções, como requisitos ou soluções.

**Workshop de Cocriação:** é o encontro estruturado que reúne participantes para, por meio de atividades colaborativas, estimular a criatividade, alinhar entendimentos e desenvolver soluções de forma conjunta.



# PARTE I

Análise por Pontos de Função (APF) e  
Simple Function Point (SFP)

## 5. MÉTODO APF

A métrica de Pontos de Função (PF) mede o tamanho funcional de um software com base nas funcionalidades implementadas sob a perspectiva 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 padronizado para a medição de projetos de desenvolvimento e de manutenção de sistemas, com o objetivo de estabelecer uma medida de tamanho funcional em pontos de função, a partir da quantificação das funcionalidades solicitadas e entregues, sob o ponto de vista do usuário. Seu foco é medir o que o software faz, por meio de uma avaliação estruturada dos requisitos de negócio.

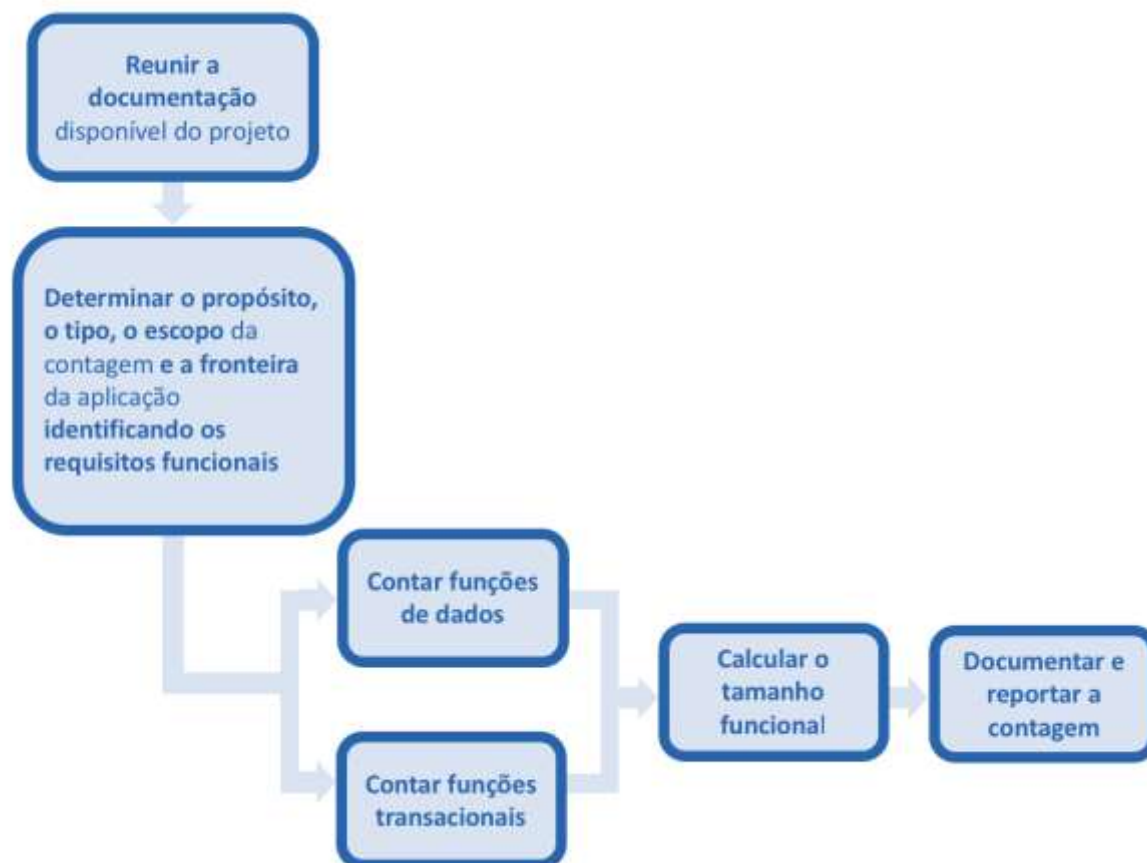
Além do método tradicional de Análise de Pontos de Função (APF), amplamente utilizado pelos órgãos do SISP, este roteiro apresenta também o método de Ponto de Função Simples (Método SFP, descrito no item 6 deste roteiro) como uma alternativa confiável, segura e capaz de acelerar consideravelmente o processo de contagem de PF.

Ressalta-se que a métrica de Pontos de Função foi concebida como uma medida de tamanho funcional para projetos de desenvolvimento e de melhoria (manutenção evolutiva) de software. O Manual de Práticas de Contagem (CPM 4.3.1) tem como objetivo mensurar o tamanho funcional de projetos de software, não sendo voltado, em sua essência, à operacionalização de contratos de prestação de serviços. Dessa forma, torna-se necessário o estabelecimento de roteiros complementares que contemplem aspectos práticos e recorrentes na realidade dos órgãos e entidades do SISP.

## 5.1. PROCESSO DE CONTAGEM APF

A Figura a seguir ilustra o processo de contagem de pontos de função utilizando o método APF, descrito nos subitens seguintes.

**Figura 2 - 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, com o objetivo de identificar os requisitos funcionais.
2. Determinar o propósito, o tipo e o escopo da contagem, bem como a fronteira da aplicação:
  - **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 a 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.
  - **Escopo da contagem:** delimita quais funcionalidades serão incluídas na contagem de pontos de função;

- **Tipo de contagem** é determinado com base no propósito da medição. Conforme CPM 4.3.1, existem três tipos de contagem:
  - **Contagem de projeto de desenvolvimento:** mede as funcionalidades entregues ao final da primeira versão da aplicação, incluindo eventuais funções de conversão de dados;
  - **Contagem de projeto de melhoria:** mede as funcionalidades alteradas, incluídas e excluídas, bem como eventuais funções de conversão de dados;
  - **Contagem de aplicação (ou baseline):** mede as funcionalidades de uma aplicação já instalada.
- **Fronteira da aplicação:** corresponde à 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. Seu estabelecimento pode envolver subjetividade. Em uma aplicação com vários módulos, por exemplo, a fronteira pode ser definida por módulo, por subsistema ou para a aplicação como um todo, a depender da visão do usuário. De fato, a definição da fronteira depende de processos de negócios, além disso, o posicionamento da fronteira influencia diretamente o resultado da contagem.

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.

Recomenda-se que as fronteiras das aplicações sejam previamente definidas em editais e mantidas atualizadas.

3. **Contar funções de dados:** após a definição da fronteira, devem ser identificadas as funções de dados, que atendem aos requisitos funcionais relacionados ao armazenamento e à manutenção 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 consideram-se 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.

A identificação das funções de dados deve seguir rigorosamente as regras do CPM 4.3.1, incluindo a contagem de DERs e RLRs, conforme descrito na tabela a seguir:

- **DERs (Dados Elementares Referenciados):** campos únicos, reconhecidos pelo usuário e não repetidos;
- **RLRs (Registros Lógicos Referenciados):** subgrupos de dados dentro de uma função de dados.

**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	<p>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.</p>	<p>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.</p>
	<p>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.</p>	<p>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.</p>
	<p>3. Contar um DER para cada parte de dado requisitada pelo usuário para estabelecer um relacionamento com outra função de dado.</p>	<p>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 porque é necessário para relacionar um funcionário a uma função existente na organização. Este tipo de dado elementar é conhecido como chave estrangeira.</p>
	<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,</p> <p>primeiros nomes (primeiro nome e inicial do meio) e sobrenome se, além do que está acima, o sobrenome for utilizado independentemente, ou</p> <p>primeiro nome, inicial do meio e sobrenome, se os três puderem ser utilizados independentemente.</p>

<p><b>RLR</b></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: Funcionário assalariado; incluindo informações gerais;</p> <p>Funcionário horista; incluindo informações gerais;</p> <p>Dependente do funcionário.</p>
--	--	---

4. **Contar funções transacionais:** uma função transacional 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 transacional 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:
  - A lógica de processamento de uma Saída Externa (SE) apresenta processamento adicional, exigindo lógica de processamento que envolva cálculos, dados derivados, manutenção em um ou mais ALIs, ou alterações no 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 transacionais (EEs, CEs e SEs)	Exemplos/Notas
<b>ALRs</b>	Um tipo de arquivo referenciado (ALR) é uma função de dados lida e/ou mantida pela função transacional	Para cada função transacional, um ALR deve ser contado para cada função de dados única que for acessada (lida e/ou gravada) pela função transacional.
<b>DERs</b>	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 transacional.	<p>DERs que atravessam a fronteira incluem:</p> <ul style="list-style-type: none"> <li>- Atributos que o usuário introduz por meio de uma tela, bem como aqueles apresentados em um relatório ou tela,</li> <li>— 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,</li> <li>— Atributos fornecidos pelo usuário da função transacional, ou apresentados a ele, e</li> <li>— Atributos em um arquivo eletrônico que entrem ou saiam pela fronteira.</li> </ul>
	3. Contar apenas um DER por função transacional 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 transacional 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 transacional 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 podem ser consultados no manual de práticas de contagem do método (CPM 4.3.1).

5. **Avaliar a complexidade 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 – 19	20 – 50	> 50
RLRs	1	Baixa	Baixa	Média
	2 - 5	Baixa	Média	Alta
	>5	Média	Alta	Alta

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

**Tabela 4 - Complexidade Funcional das Entradas Externas (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 Consultas Externas e Saídas Externas (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, método APF:

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

Tipos Funcionais	Complexidade		
	Baixa	Média	Alta
<b>Arquivo Lógico Interno (ALI)</b>	7 PF	10 PF	15 PF
<b>Arquivo de Interface Externa (AIE)</b>	5 PF	7 PF	10 PF
<b>Entrada Externa (EE)</b>	3 PF	4 PF	6 PF
<b>Consulta Externa (CE)</b>	3 PF	4 PF	6PF
<b>Saída Externa (SE)</b>	4 PF	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, ALRs, os exemplos de identificação das funções de dados e transacionais, bem como as regras de contagem do método APF podem ser consultados no CPM 4.3.1.

6. **Calcular o Tamanho Funcional:** é 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.

7. **Documentar e reportar a contagem:** é a última etapa do processo de contagem da figura 2.

- **Documentar:** a contagem de pontos de função deve ser devidamente registrada, de forma a garantir rastreabilidade, reprodutibilidade e manutenibilidade. A documentação deve contemplar todas as premissas adotadas, as decisões de medição, os padrões e diretrizes utilizados, bem como as referências e 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 20926:2009-sisp); PF\* na versão em português.

## 6. MÉTODO SFP

O Ponto de Função Simples (Simple Function Point - SFP) é um método de medição funcional compatível e consistente com o IFPUG APF, descrito no CPM 4.3.1, em termos de 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.

O método é aplicável ao longo de todo o ciclo de desenvolvimento de software, incluindo atividades de estimativa, desenvolvimento, manutenção, apoio a análises de produtividade e definição de métricas de equipe. Ele 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 2.1.

A decisão de usar o método IFPUG SFP pode ser influenciada por diversos fatores, tais 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:

- estimar tamanho, custo e os recursos necessários para o desenvolvimento, evolução e manutenção;
- estimar o tamanho funcional de software a ser desenvolvido;
- apoiar análises de qualidade e produtividade;
- complementar métricas de equipe;
- realizar contagens de pontos de função em projetos de desenvolvimento, melhoria e aplicações em produção.

O método tem correspondência de conceitos, tipos de medição e fórmulas de cálculo do método APF tradicional, o que leva à 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 dos órgãos do SISP demanda métodos de medição ágeis, com baixo impacto nos processos produtivos, que não exijam elevada especialização, sejam confiáveis, independentes de tecnologia e estejam correlacionados ao esforço, custo e duração dos projetos.

Nesse sentido, este roteiro apresenta um método de medição leve, ágil e consistente com a estrutura da família de padrões ISO 14143, além de totalmente compatível com o método IFPUG APF. O método SFP é um método confiável, menos sujeito a subjetividade de interpretações, mais fácil, mais simples de aplicar e manter e com menor necessidade de especialização em comparação ao método APF tradicional. Pode ser aplicado no início do processo de desenvolvimento do software. A sua utilização pode reduzir custos, tempo e discussões.

### O que torna o método SFP mais ágil, simples e leve?

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 aos componentes funcionais são: 7,0 SFP para Arquivo Lógico e 4,6 SFP para Processo elementar. Essa padronização elimina a necessidade de avaliação de complexidade e reduz significativamente o esforço de contagem.

Outro ponto importante é a previsibilidade. Como não há classificação de complexidade entre as funcionalidades, a contribuição funcional de cada processo elementar ou função de dados permanece constante ao longo do processo de desenvolvimento ou de melhoria do software.

Essas características tornam o método especialmente adequado para cenários que demandam maior agilidade na estimativa e menor esforço de medição, mantendo consistência com os conceitos fundamentais da Análise de Pontos de Função.

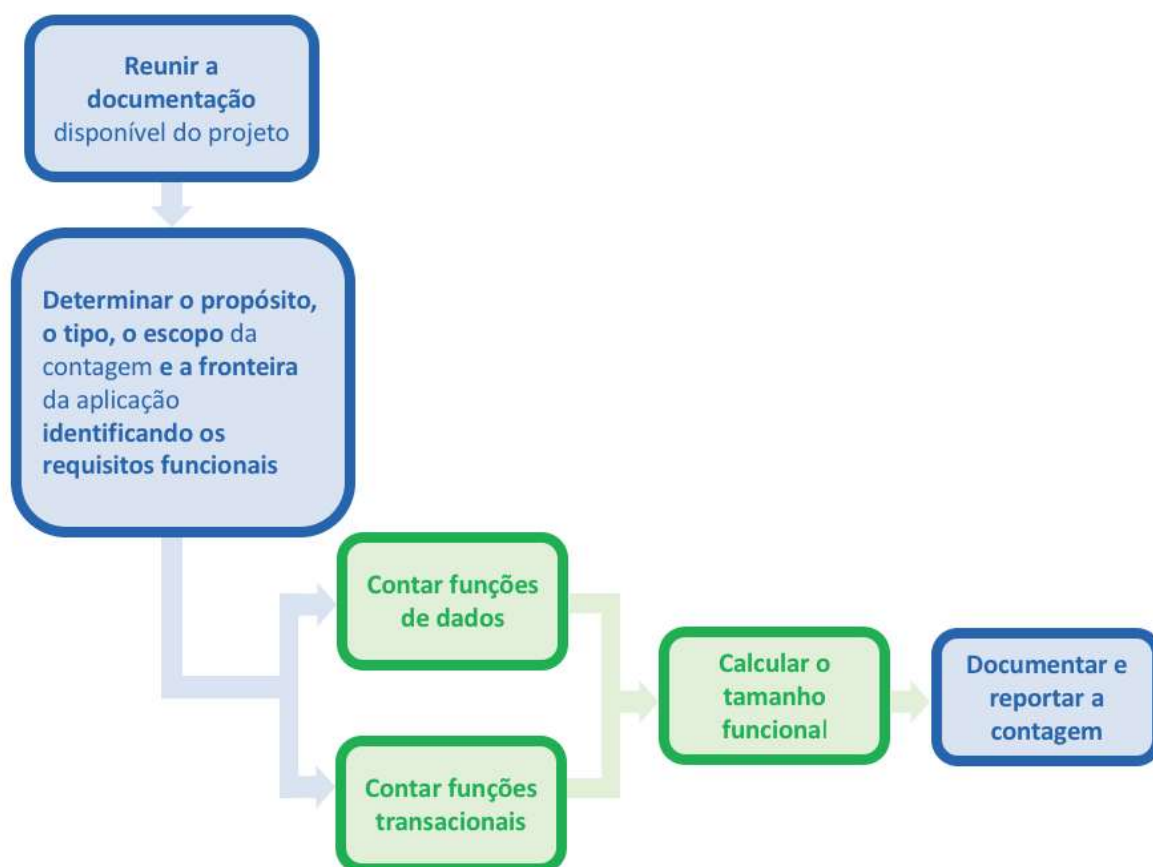
A seguir, descreve-se todo o processo de contagem do método SFP e a fórmula para conversão do método IFPUG APF em IFPUG SFP.

## 6.1. PROCESSO DE CONTAGEM SFP

O processo de mensuração do método SFP segue a mesma estrutura 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 5.1 deste roteiro.

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

**Figura 3 - 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 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 subconjunto 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, de seus usuários e de suas interações com 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 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.

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**

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



## TOME NOTA

Toda contagem ou estimativa de pontos de função é realizada dentro de uma fronteira de aplicação. Como a definição dessa fronteira é baseada em processos de negócio, ela pode envolver certo grau de subjetividade. Por exemplo: um sistema de capacitação de empregados deve ou não fazer parte da fronteira de um sistema de recursos humanos?

Em algumas organizações, a capacitação de empregados é tratada como parte do processo de gestão de recursos humanos. Nesse caso, considera-se uma única fronteira de aplicação, abrangendo tanto o sistema de recursos humanos quanto o módulo de capacitação.

Em outras organizações, a capacitação é tratada como um processo de negócio independente. Nessa situação, devem ser consideradas duas fronteiras distintas: uma para o sistema de recursos humanos e outra para o sistema de capacitação.

Diante disso, é fundamental estabelecer e documentar claramente as fronteiras das aplicações. Recomenda-se que as fronteiras dos sistemas a serem mantidos estejam formalmente registradas no edital de contratação e também no roteiro de métricas do órgão. Da mesma forma, é importante definir previamente as fronteiras das novas aplicações a serem contratadas, garantindo sua conformidade com o Plano Diretor de TI do órgão contratante.

Adicionalmente, nos casos em que são demandadas manutenções em componentes que implementam regras de negócio específicas e que são reutilizados por diferentes funcionalidades da aplicação, a abordagem de contagem deve ser cuidadosamente definida. Considerando o método de contagem de projetos de melhoria do CPM, todas as funcionalidades impactadas pela alteração poderiam ser contabilizadas.

No entanto, conforme o item 7.18 Componente Reusável) deste roteiro, o componente a ser modificado deve ser tratado como um processo elementar independente. Assim, sua alteração deve ser mensurada por meio da aplicação do fator de impacto (FI) sobre o PF\_ALTERADO ou SFP\_ALTERADO, conforme os conceitos apresentados na seção 7.3 (Projeto de Melhoria).

Outro aspecto relevante refere-se à adoção de múltiplas fronteiras (por exemplo, por segurança, domínio, contexto ou interface de usuário), que pode resultar em múltiplas instâncias da aplicação (como web, móvel e API). Essa abordagem pode impactar a quantidade de AIEs, EEs, SEs e CEs, aumentar a duplicidade de componentes e influenciar significativamente o custo final do produto.

Neste roteiro, os cenários envolvendo múltiplas instâncias são tratados no item 8.1 (Contagem com Múltiplas Mídias).

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” para fins de contagem. Ainda assim, convém realçar que o método utiliza os mesmos conceitos de processo elementar e arquivo lógico do método APF e 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 apresentadas na tabela a seguir; são valores constantes definidos a partir de estudos de correlação com o método APF:

**Tabela 8 - Pontuação dos Componentes funcionais básicos do método IFPUG SFP**

<b>Componentes funcionais básicos do IFPUG SFP</b>	<b>Pontuação</b>
<b>Arquivo Lógico (AL)</b>	7,0 SFP
<b>Processo Elementar (PE)</b>	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. A simplificação do método não elimina a necessidade de documentação das decisões de medição. 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 7 deste roteiro aplicam-se aos dois métodos de contagem de PF (APF e SFP) .

## 6.2. 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 um projeto de melhoria funcional, uma vez que não há mudança na complexidade. Não é necessário computar o PF\_ALTERADO, uma vez que na contagem SFP o SFP\_ALTERADO anterior e o SFP\_ALTERADO atual são idênticos.

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 ALIs 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:


$$SFP = (QtdEES + QtdCEs + QtdSEs) \times 4,6 + (QtdALIs + QtdAIEs) \times 7$$

Onde:

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

### 6.3. 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:



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

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

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

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);
- 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. Essa padronização elimina a necessidade de avaliação de complexidade.

## 7. 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 o seu dimensionamento baseadas nas regras de contagem do CPM 4.3.1 e SPM 2.1.



Conforme exposto em outras partes deste documento, o método SFP apresenta correspondência de conceitos, tipos de medição e fórmulas de cálculo em relação ao 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 das 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). Ressalta-se 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 à 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.
- **Fator de Impacto (FI) conforme processos técnicos da ISO/IEC 12207:** Os percentuais de FI utilizados em algumas fórmulas de contagem observam a Tabela 16 (item 10.1), que utiliza os processos técnicos e atividades de referência do ciclo de vida do *software*, padronizados pela ABNT NBR ISO/IEC

12207:2017. A utilização dos processos técnicos e das atividades previstas na norma ISO/IEC 12207 como referência conceitual proporciona maior flexibilidade aos órgãos contratantes para ajustar, se necessário, os fatores de impacto (FI), conforme as particularidades de seus processos e projetos de software. Essa abordagem favorece uma aferição transparente e aderente à realidade de cada projeto, especialmente naqueles em que se faz necessária a aplicação de fatores de impacto na contagem de pontos de função. Podem existir cenários em que:

- **O órgão contratante não tem a necessidade de contratar todas as fases do ciclo de vida do software:** nesses cenários, para a remuneração da contratada, o órgão deve considerar os macroprocessos a serem contratados e a composição do time de referência para definição do Fator de impacto (FI) a ser aplicado nas contagens de PF, observando as orientações contidas no item 10.1 deste roteiro.
- **O órgão contratante tem a necessidade de executar atividades de processos não contemplados no cálculo de FI do SISP:** caso um projeto de desenvolvimento ou manutenção exija a execução de um número diferente de processos e/ou atividades utilizados na memória de cálculo do SISP, o órgão contratante poderá recalcular o fator de impacto (FI) observando os processos e atividades da ISO/IEC 12207 e registrar/documentar a justificativa e o ajuste realizado no FI, indicando os entregáveis correspondentes como evidência dos processos e atividades aplicados.



**A utilização dos processos técnicos e das atividades previstas na ISO/IEC 12207** como referência conceitual para definição dos Fatores de Impacto (FI) utilizados nas contagens de PF dos projetos de desenvolvimento e manutenção de software, **proporciona aos órgãos contratantes maior flexibilidade para ajustar os fatores de impacto (FI) na contagem de PF, conforme as particularidades de seus processos e projetos de software.** Essa abordagem favorece uma aferição transparente e aderente à realidade de cada projeto

- **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).

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.

## 7.1. 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, por exemplo, a carga inicial de arquivos lógicos.

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



$$\begin{aligned} \text{PF\_DESENVOLVIMENTO} &= \text{PF\_INCLUÍDO} + \text{PF\_CONVERSÃO} \\ \text{SFP\_DESENVOLVIMENTO} &= \text{SFP\_INCLUÍDO} + \text{SFP\_CONVERSÃO} \end{aligned}$$

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.

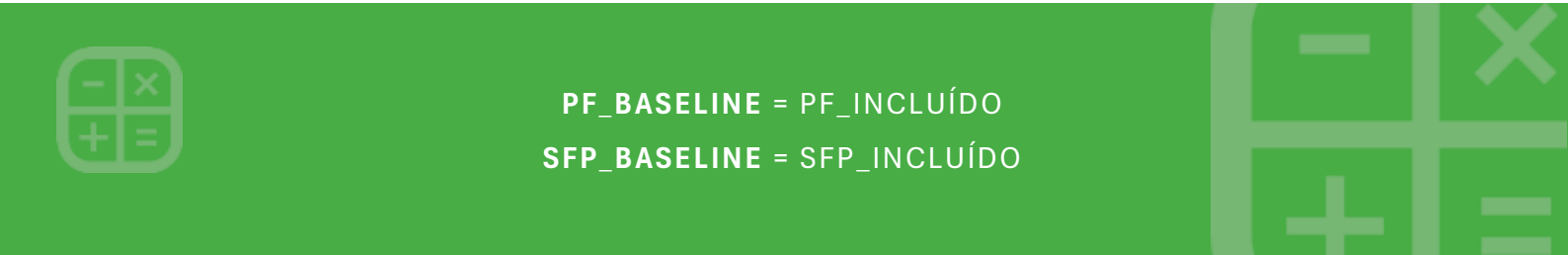
Este roteiro recomenda a supressão do PF\_CONVERSÃO / SFP\_CONVERSÃO das fórmulas de contagem de projetos de desenvolvimento quando for caracterizado um esforço relativamente maior dessa atividade. Nesses casos, recomenda-se tratá-los como projetos separados de migração de dados, conforme item 7.6.

## 7.2. BASELINE APÓS DESENVOLVIMENTO INICIAL

Ao final de um projeto de desenvolvimento inicial, a medição da aplicação implantada ou *baseline* (linha de base) do *software* considera apenas as funções transacionais e as funções de dados incluídas (PF\_INCLUÍDO ou SFP\_INCLUÍDO); não considerará 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 *baseline* após desenvolvimento inicial, que é a mesma para os métodos APF e SFP:



The banner features a green background with a grid of mathematical symbols: minus, multiply, plus, and equals. The formulas are centered in white text.

$$\text{PF\_BASELINE} = \text{PF\_INCLUÍDO}$$
$$\text{SFP\_BASELINE} = \text{SFP\_INCLUÍDO}$$

### 7.3. PROJETO DE MELHORIA

O Projeto de Melhoria (*enhancement*), também denominado projeto de melhoria funcional ou manutenção evolutiva, está associado a mudanças nos requisitos funcionais da aplicação, ou seja, à inclusão, alteração ou exclusão de funções em aplicações implantadas

Conforme o CPM 4.3.1, as mudanças nas funcionalidades podem decorrer de novos requisitos, revisão de requisitos do usuário, alterações legais ou regulamentares, ou ainda da inclusão de novos usuários. A ênfase deve estar nos requisitos de negócio, de forma que a contagem de PF do projeto de melhoria reflita a intenção dos requisitos do usuário:

- Uma função de dados (ALI ou AIE/ Arquivo Lógico) é considerada alterada quando houver mudanças nos tipos de dados, inclusão ou exclusão de dados, ou alteração de tamanho (número de posições), ou alteração de tipo de campo (por exemplo, mudança de numérico para alfanumérico), desde que decorrentes de alteração em regra de negócio.
- 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, conforme definido nas regras do CPM 4.3.1.



Este roteiro distingue o projeto de melhoria (associado a requisitos funcionais) do projeto de manutenção adaptativa (associado a requisitos não funcionais).

O dimensionamento de um projeto de melhoria consiste na soma de funções incluídas, alteradas, excluídas e funções de conversão de dados. A fórmula de cálculo utilizada no dimensionamento de um projeto de melhoria para os métodos APF e SFP é:



$$\begin{aligned} \text{PF\_MELHORIA} &= \text{PF\_INCLUÍDO} + (\text{FI}^* \times \text{PF\_ALTERADO}) + \\ &\quad (0,50 \times \text{PF\_EXCLUÍDO}) + \text{PF\_CONVERSÃO} \\ \text{SFP\_MELHORIA} &= \text{SFP\_INCLUÍDO} + (\text{FI}^* \times \text{SFP\_ALTERADO}) + \\ &\quad (0.50 \times \text{SFP EXCLUÍDO}) + \text{SFP CONVERSÃO} \end{aligned}$$



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

- **FI = 63%** para funcionalidades desenvolvidas ou mantidas pela própria empresa contratada.
- **FI = 84%** para funcionalidades não desenvolvidas ou mantidas previamente 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 não tenha participado do desenvolvimento ou de manutenções anteriores da funcionalidade objeto do projeto de melhoria.
  - Em demandas futuras de melhoria sobre a mesma funcionalidade, será considerado que a contratada já a desenvolveu.

Os percentuais de FI (fator de impacto) acima observam a Tabela 16 (item 10.1) que utiliza processos técnicos de referência do ciclo de vida do software, padronizados pela 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 significativamente elevado dessa

atividade. Nesses casos, recomenda-se tratá-los como projetos separados de migração de dados, conforme item 7.6 deste roteiro.



**As diretrizes para definição dos Fatores de Impacto (FI) utilizados nas fórmulas de cálculo de PF do SISP encontram-se no item 10 deste roteiro.**

A utilização dos processos técnicos e das atividades previstas na ISO/IEC 12207, proporciona aos órgãos contratantes maior flexibilidade para, se necessário, ajustar os FIs na contagem de PF conforme as particularidades de seus processos e projetos de software. Essa abordagem favorece uma aferição transparente e aderente à realidade de cada projeto

## 7.4. **BASELINE APÓS MELHORIA - MÉTODO SFP**

Conforme exposto no item 6.2 as fórmulas dos métodos de contagem de PF (APF e SFP) são semelhantes. A principal diferença está na forma de atualização da *baseline* após um projeto de melhoria, uma vez que, no método SFP, não há alteração na complexidade ou na pontuação dos componentes funcionais básicos (Arquivo Lógico – AL e Processo Elementar – PE).

Após um projeto de melhoria, a medição da nova *baseline* do aplicativo lançado (SFP\_BASELINE) corresponde à medição da *baseline* anterior ao projeto de melhoria, acrescida das novas funções (SFP\_INCLUÍDO) e reduzida das funções removidas (SFP\_EXCLUÍDO):



$$\text{SFP\_BASELINE} = \text{SFP\_BASELINE\_ANTERIOR} + \text{SFP\_INCLUÍDO} - \text{SFP\_EXCLUÍDO}$$

Para o método SFP, na contagem da *baseline* da aplicação não devem constar as funções alteradas em um projeto de melhoria, uma vez que estas funções já foram consideradas em medições anteriores e permanecem contabilizadas na *baseline*; ademais, o método SFP não determina a complexidade das funções, não gerando, portanto, alterações em sua pontuação.

## 7.5. **BASELINE APÓS MELHORIA – MÉTODO APF**

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

No método APF, quando uma função é modificada por um projeto de melhoria, sua pontuação de complexidade pode ser alterada. Neste caso o analista deve:

- Determinar a complexidade das funções antes da alteração (a partir da documentação da última medição ou medindo como existiam antes da mudança);
- Determinar a complexidade das funções depois da alteração;
- Atualizar a *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 à 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 funções de conversão nem as excluídas, visto que a *baseline* representa o tamanho funcional da aplicação medida.

## 7.6. **PROJETOS DE MIGRAÇÃO DE DADOS**

Este roteiro recomenda a supressão do PF\_CONVERSÃO ou SFP\_CONVERSÃO das fórmulas de contagem de pontos de função em projetos de desenvolvimento e de melhoria, nos casos em que for caracterizado um esforço

significativamente elevado 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 essa demanda como um projeto separado de migração de dados. Os projetos de migração de dados devem seguir a fórmula abaixo, idêntica para os métodos APF e SFP:


$$PF\_CONVERSÃO = PF\_INCLUÍDO \times 0,61$$

$$SFP\_CONVERSÃO = SFP\_INCLUÍDO \times 0,61$$


Um projeto de migração deve contemplar, no mínimo:

- Para o método APF: os ALLs mantidos pela migração, as Entradas Externas (considerando as cargas de dados nos ALLs) e, caso seja solicitado pelo usuário, os relatórios gerenciais das cargas, que devem ser contabilizados 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 16 (item 10.1), que utiliza os processos técnicos de referência do ciclo de vida do *software*, padronizados pela ISO/IEC 12207.

Todas as contagens devem ser realizadas com base nas funções 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 compor o tamanho funcional do software, pois não integram as funcionalidades disponibilizadas ao usuário após a implantação do sistema.

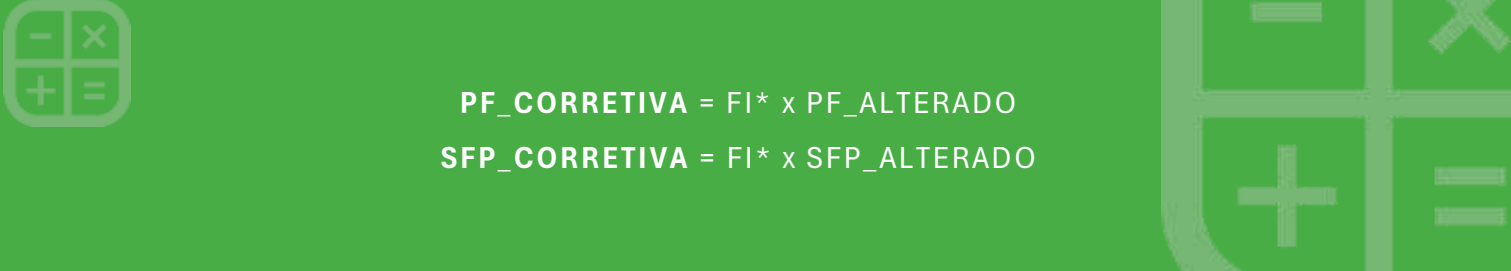
## 7.7. MANUTENÇÃO CORRETIVA

Mesmo com a execução de atividades de garantia da qualidade, defeitos podem ser identificados na aplicação entregue. A manutenção corretiva altera o software para correção de defeitos. Encontram-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 específica, deve ser considerada a garantia prevista no Código de Defesa do Consumidor.

Quando o sistema estiver fora da garantia ou não tiver sido desenvolvido pela empresa contratada, deverá ser estimado 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:


$$\text{PF\_CORRETIVA} = \text{FI}^* \times \text{PF\_ALTERADO}$$
$$\text{SFP\_CORRETIVA} = \text{FI}^* \times \text{SFP\_ALTERADO}$$

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 de correção da funcionalidade em questão, será considerado que a contratada já desenvolveu a funcionalidade.

Os fatores de impacto acima observam a Tabela 16 (item 10.1) que utiliza processos técnicos de referência do ciclo de vida do *software*, padronizados pela ISO/IEC 12207.



**As diretrizes para definição dos Fatores de Impacto (FI) utilizados nas fórmulas de cálculo de PF do SISP encontram-se no item 10 deste roteiro.**

A utilização dos processos técnicos e das atividades previstas na ISO/IEC 12207, proporciona maior flexibilidade aos órgãos contratantes para, se necessário, ajustar os FIs na contagem de PF conforme as particularidades de seus processos e projetos de software. Essa abordagem favorece uma aferição transparente e aderente à realidade de cada projeto

## 7.8. 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 *Laravel*; o banco de dados de um sistema legado que precisa ser migrado para *PostgreSQL*.

Recomenda-se a realização de 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 testes, apresentados na Tabela 16 (item 10.1).

As diretrizes para definição dos Fatores de Impacto (FI) utilizados nas fórmulas de cálculo de PF do SISP encontram-se no item 10 deste roteiro. A utilização dos processos técnicos e das atividades previstas na ISO/IEC 12207 proporciona aos órgãos contratantes maior flexibilidade para, se necessário, ajustar os FIs na contagem de PF conforme as particularidades de seus processos e projetos de software.

### 7.8.1. 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 61% para as funções de conversão, conforme fórmulas a seguir:



$$\text{PF\_REDESENV\_LINGUAGEM} = (\text{PF\_INCLUÍDO} \times \text{FI}^*) + (\text{PF\_CONVERSÃO} \times 0,61)$$
$$\text{SFP\_REDESENV\_LINGUAGEM} = (\text{SFP\_INCLUÍDO} \times \text{FI}^*) + (\text{SFP\_CONVERSÃO} \times 0,61)$$

Caso não existam mudanças nas funções de dados, estas 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 16 do item 10.1 deste roteiro.

Este roteiro recomenda a supressão do PF\_CONVERSÃO ou SFP\_CONVERSÃO da fórmula de contagem de projetos de redesevolvimento quando for caracterizado um esforço relativamente maior dessa atividade. Nesses casos, recomenda-se tratá-los como projetos separados de migração de dados, conforme item 7.6.

### 7.8.2. Mudança de Plataforma – Banco de Dados

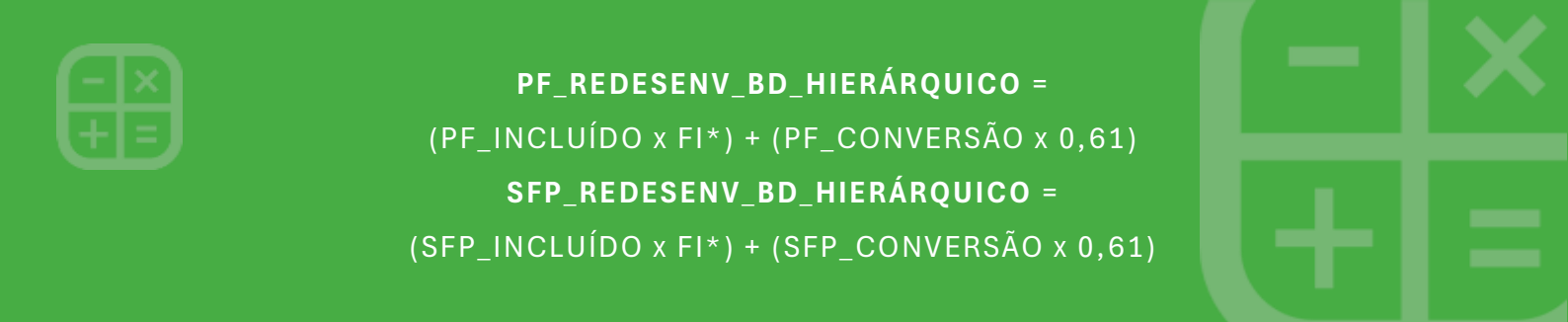
Nesta categoria encontram-se as demandas de redesevolvimento de sistemas para utilizar outro sistema gerenciador de banco de dados.

Caso não exista mudança nas funções de dados, ou seja, o banco de dados da aplicação seja mantido, então as funções de dados não devem ser consideradas para fins de pagamento. No entanto, recomenda-se a contagem das funções de dados para fins de documentação da *baseline* 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 não possua documentação;
- **FI = 73%**, caso o sistema já possua documentação de requisitos; e
- **FI = 61%** para as funções de conversão.

Assim, será utilizada a fórmula abaixo:


$$\begin{aligned} \text{PF\_REDESENV\_BD\_HIERÁRQUICO} &= \\ &(\text{PF\_INCLUÍDO} \times \text{FI}^*) + (\text{PF\_CONVERSÃO} \times 0,61) \\ \text{SFP\_REDESENV\_BD\_HIERÁRQUICO} &= \\ &(\text{SFP\_INCLUÍDO} \times \text{FI}^*) + (\text{SFP\_CONVERSÃO} \times 0,61) \end{aligned}$$

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 7.6.



Se o projeto já possui documentação de requisitos, o percentual referente aos processos de definição de requisitos (ver tabela 16 no item 10.1) não deve ser considerado para fins de pagamento.

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:



$$\begin{aligned} \text{PF\_REDESENV\_BD\_RELACIONAL} &= \\ &(\text{PF\_ALTERADO} \times 0,39) + (\text{PF\_CONVERSÃO} \times 0,61) \\ \text{SFP\_REDESENV\_BD\_RELACIONAL} &= \\ &(\text{SFP\_ALTERADO} \times 0,39) + (\text{SFP\_CONVERSÃO} \times 0,61) \end{aligned}$$



A definição dos percentuais dos fatores de impacto (FI) apresentados nas fórmulas acima, baseia-se nos processos técnicos de referência do ciclo de vida do *software*, padronizados pela ISO/IEC 12207 e descritos na tabela 16 (item 10.1).

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

- 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 referente aos processos de teste, conforme tabela 16, item 10.1;
- Em mudanças de banco relacional para relacional, geralmente a estrutura de dados não é alterada, desta forma não se contam as funções de dados;
- Nos projetos de redesevolvimento de banco de dados relacional para outro relacional, recomenda-se tratar o PF\_CONVERSÃO/SFP\_CONVERSÃO dentro do mesmo projeto.

## 7.9. ATUALIZAÇÃO DE VERSÃO

São consideradas nesta categoria demandas relacionadas à adaptação de uma aplicação existente, ou parte dela, para execução em diferentes versões de *browsers* (ex.: *Internet Explorer*, *Firefox*, *Chrome*) ou de linguagens de programação (ex.: versão mais atual do Java). Também se enquadram nesta categoria as atualizações 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 7.18).

Recomenda-se 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 16 do item 10.1 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 nas diretrizes estabelecidas no item 10 deste roteiro.

### 7.9.1. 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:



$$\text{PF\_ATUALIZAÇÃO\_VERSÃO\_LINGUAGEM} = \text{PF\_ALTERADO} \times 0,39$$
$$\text{SFP\_ATUALIZAÇÃO\_VERSÃO\_LINGUAGEM} = \text{SFP\_ALTERADO} \times 0,39$$

O PF\_ALTERADO/SFP\_ALTERADO deve considerar apenas as funcionalidades impactadas.

As funcionalidades que apresentarem apenas demandas de testes, devem ser contadas usando o percentual correspondente, conforme tabela 16 (item 10.1).

## 7.9.2. 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 múltiplos *browsers*. É importante destacar que este tipo de procedimento é usualmente realizado quando é necessário resolver algum problema de incompatibilidade. Nesse caso, 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 16 (item 10.1).

Essas atualizações podem implicar em manutenções em componentes comuns da plataforma utilizada. Nesses casos, a demanda deve ser contada como componente interno reusável, descrita no item 7.18 deste roteiro.

## 7.9.3. 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:



$$\text{PF\_ATUALIZAÇÃO\_VERSÃO\_BD} = \text{PF\_ALTERADO} \times 0,39$$
$$\text{SFP\_ATUALIZAÇÃO\_VERSÃO\_BD} = \text{SFP\_ALTERADO} \times 0,39$$



O PF\_ALTERADO/SFP\_ALTERADO deve considerar apenas as funcionalidades impactadas. As funcionalidades que apresentarem apenas demandas de testes, devem ser contadas usando o percentual relativo aos processos técnicos de testes, conforme diretrizes estabelecidas no item 10 deste roteiro.

## 7.10. 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:

- mudança 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;
- mudança do título de um relatório;
- mudança de *labels* de uma tela de consulta.

Para esses casos, a aferição do tamanho em pontos de função das funções transacionais impactadas será realizada da seguinte forma:

- Para o método APF: com a aplicação de um fator de impacto de 13% sobre a contagem de uma função transacional de mais baixa complexidade (3 PF), ou seja 0,39 PF, independentemente da complexidade da(s) funcionalidade(s) alterada(s);
- Para o método SFP: com a aplicação de um fator de impacto de 13% sobre a contagem de um processo elementar (4,6 SFP), ou seja 0,59 SFP.

Neste tipo de manutenção não são contadas funções de dados.



$PF\_INTERFACE = 0,39 \text{ PF} \times \text{QTD de Funções transacionais impactadas}$   
 $SFP\_INTERFACE = 0,59 \text{ SFP} \times \text{QTD de Funções transacionais impactadas}$

## 7.11. 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;
- adaptar uma funcionalidade para possibilitar a chamada por um *webservice* ou para outro tipo de integração com outros sistemas;
- replicar funcionalidade: chamar uma consulta existente em outra tela da aplicação;
- alterar a aplicação para adaptação às adaptações 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 alterações associadas a regras de negócio não funcionais. 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 no item 7.3), conforme fórmula a seguir, igual para os dois métodos:



$$PF\_ADAPTATIVA = FI^* \times PF\_ALTERADO$$
$$SFP\_ADAPTATIVA = FI^* \times SFP\_ALTERADO$$



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

- **FI = 55%** para funcionalidades de sistema desenvolvidas ou mantidas por meio de um projeto de melhoria pela empresa contratada.
- **FI = 63%** para funcionalidades de sistema não desenvolvidas ou mantidas por meio de um projeto de melhoria pela empresa contratada. Em demandas futuras de manutenção adaptativa sobre a mesma funcionalidade, será considerado que a contratada já desenvolveu a funcionalidade.

Destaca-se que, além da adequação das funcionalidades impactadas, a atualização da documentação do projeto de manutenção adaptativa deve ser realizada.

## 7.12. 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 7.12.1;
- Gerar um relatório específico ou arquivo para o usuário, por meio de recuperação de informações nas bases de dados da aplicação, detalhados na subseção 7.12.2.

Caso a apuração envolva a correção de dados decorrentes de erros em 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 parâmetros em uma apuração especial, tratados na subseção 7.12.3; e
- REEXECUÇÃO de uma apuração especial, tratados na subseção 7.12.4.



Recomenda-se que o órgão contratante **sempre solicite formalmente** à empresa contratada o **armazenamento do script** para permitir posterior alteração e/ou reexecução.

Cumpre 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*.

### 7.12.1. 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 essas funções são executadas apenas uma vez, não fazendo parte da aplicação, visando à correção de dados incorretos na base de dados da aplicação ou à atualização decorrente de modificações na estrutura de dados; como exemplo temos a inclusão de valor “sim” ou “não” no campo “indicador de matriz” referente ao CNPJ.

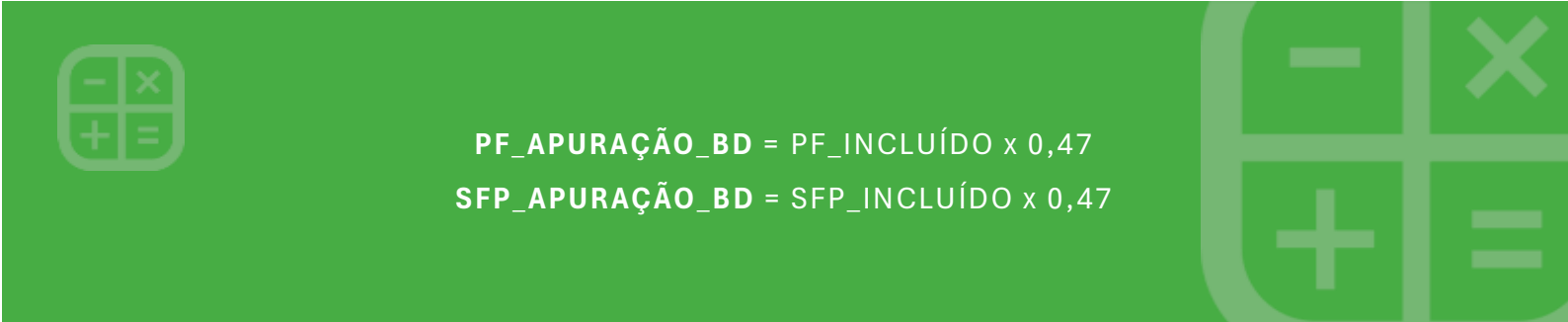
Normalmente, nesse tipo de atualização são afetados múltiplos registros. Nesses casos, considera-se apenas a contagem das funcionalidades desenvolvidas. Para o método APF, geralmente essas funcionalidades são classificadas como Entradas Externas. Para esse cenário, como artefato de homologação da demanda, 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 ISO/IEC 12207 e descritos na tabela 16 (item 10.1) deste roteiro.

### 7.12.1.a. Atualização de dados sem consulta prévia

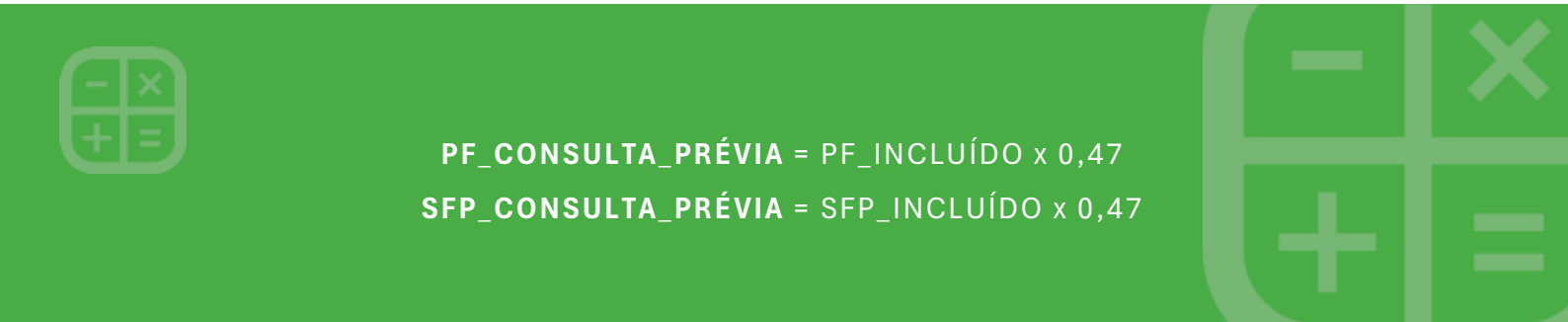
Existem casos de Apuração Especial – Base de Dados, em que o usuário não solicita uma consulta prévia das informações. Nesses casos, deve ser utilizada a fórmula a seguir, análoga para os dois métodos:


$$\begin{aligned} PF\_APURAÇÃO\_BD &= PF\_INCLUÍDO \times 0,47 \\ SFP\_APURAÇÃO\_BD &= SFP\_INCLUÍDO \times 0,47 \end{aligned}$$

### 7.12.1.b. Consulta prévia sem atualização

Em outros casos de Apuração Especial – Base de Dados, o usuário solicita uma consulta prévia das informações. Ressalta-se que essa consulta deve ser realizada antes da construção da funcionalidade, não se tratando da etapa de homologação. Trata-se de uma prática importante com o fim de evitar atualizações incorretas na base de produção.

A necessidade de consulta prévia deve ser definida pelo órgão contratante. Deve ser solicitada obrigatoriamente pelo órgão contratante para a avaliação da viabilidade de implementar a Apuração Especial - Base de Dados, conforme fórmula a seguir:


$$\begin{aligned} PF\_CONSULTA\_PRÉVIA &= PF\_INCLUÍDO \times 0,47 \\ SFP\_CONSULTA\_PRÉVIA &= SFP\_INCLUÍDO \times 0,47 \end{aligned}$$

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 (PE) cuja pontuação funcional é 4,6 SFP, visto que neste método não é necessário diferenciar os tipos de processo elementar, tampouco a sua complexidade.

### 7.12.1.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:



$$\begin{aligned} \text{PF\_APURAÇÃO\_BD\_PÓS\_CONSULTA\_PRÉVIA} &= \text{PF\_INCLUÍDO} \times 0,36 \\ \text{SFP\_APURAÇÃO\_BD\_PÓS\_CONSULTA\_PRÉVIA} &= \text{SFP\_INCLUÍDO} \times 0,36 \end{aligned}$$

### 7.12.2. 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 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:



$$\text{PF\_APURAÇÃO\_RELATÓRIOS} = \text{PF\_INCLUÍDO} \times 0,47$$
$$\text{SFP\_APURAÇÃO\_RELATÓRIOS} = \text{SFP\_INCLUÍDO} \times 0,47$$



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.

### 7.12.3. Apuração especial – Alteração

Recomenda-se sempre solicitar formalmente à contratada o armazenamento do *script* para permitir posterior alteração. Uma apuração especial pode ser considerada alterada quando houver a inclusão, exclusão ou alteração de parâmetros no *script* já executado:

- Solicitação de mudança no conteúdo de uma variável que já está codificada no *script* de apuração especial; ou
- Uma solicitação para alterar o *script*, incluindo, excluindo ou modificando parâmetros. Por exemplo, se há um parâmetro chamado "x" e foi solicitado incluir, alterar ou excluir o parâmetro "y" dentro do mesmo processo de negócio, que afeta os mesmos arquivos lógicos impactados na apuração especial já executada.

Nesse sentido, se for solicitada a alteração de uma apuração especial - conforme cenários descritos nos subitens 7.12.1 ou 7.12.2 - esta deve ser dimensionada conforme fórmula a seguir:

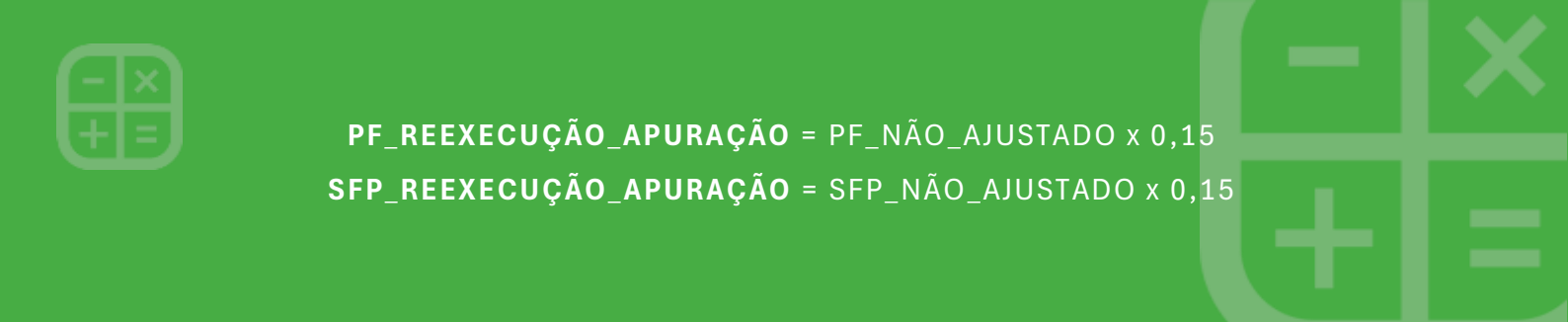


$$\text{PF\_ALTERAÇÃO\_APURAÇÃO} = \text{PF\_ALTERADO} \times 0,36$$
$$\text{SFP\_ALTERAÇÃO\_APURAÇÃO} = \text{SFP\_ALTERADO} \times 0,36$$



#### 7.12.4. Apuração especial – Reexecução

Como já foi mencionado no início do item 7.12, recomenda-se sempre solicitar formalmente à 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:


$$\begin{aligned} \text{PF\_REEXECUÇÃO\_APURAÇÃO} &= \text{PF\_NÃO\_AJUSTADO} \times 0,15 \\ \text{SFP\_REEXECUÇÃO\_APURAÇÃO} &= \text{SFP\_NÃO\_AJUSTADO} \times 0,15 \end{aligned}$$

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.

### 7.13. ATUALIZAÇÃO DE DADOS

Em alguns casos, as demandas de correção de problemas em base de dados estão associadas a atualizações manuais (de forma interativa), diretamente no banco de dados em um único registro e que não envolvem cálculos ou procedimentos complexos. São exemplos desse tipo de demanda, atualização de valor cadastrado incorretamente em campo de tabela ou a exclusão de um registro de uma tabela.

Nestes casos:

- Para o método APF, a aferição deve considerar 15% do PF de uma Entrada Externa (EE) e os Tipos de Dados da Entrada Externa são todos os TD considerados na funcionalidade – campos atualizados e campos utilizados para a seleção do registro.
- Para o método SFP, a aferição deve considerar 15% de um processo elementar (PE) cuja pontuação funcional é 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:



$$\text{PF\_ATUALIZAÇÃO\_BD} = \text{PF\_ALTERADO} \times 0,15$$
$$\text{SFP\_ATUALIZAÇÃO\_BD} = \text{SFP\_ALTERADO} \times 0,15$$



Ressalta-se que normalmente neste tipo de demanda não há gestão de configuração (armazenamento de *script*, versionamento, etc.) das atualizações. Caso a contratante identifique a necessidade de realização de gestão de configuração das atualizações no banco de dados, então a demanda será classificada como Apuração Especial - Base de Dados (subseção 7.12.1).

## 7.14. DESENVOLVIMENTO, MANUTENÇÃO E PUBLICAÇÃO DE PÁGINAS ESTÁTICAS

Nesta seção são tratadas demandas de desenvolvimento e manutenção 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 incluído 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:



$$\text{PF\_PUBLICAÇÃO} = \text{PF\_ALTERADO} \times 0,26$$
$$\text{SFP\_PUBLICAÇÃO} = \text{SFP\_ALTERADO} \times 0,26$$



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 16 deste roteiro.

## 7.15. MANUTENÇÃO DE DOCUMENTAÇÃO DE SISTEMAS LEGADOS

Nesta seção são tratadas demandas de documentação ou atualização de documentação de sistemas legados. Nessas demandas, o analista geralmente realiza 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 ISO/IEC 12207 e descritos na tabela 16 (item 10.1):


$$PF\_DOCUMENTAÇÃO = PF\_NÃO\_AJUSTADO \times 0,21$$

$$SFP\_DOCUMENTAÇÃO = SFP\_NÃO\_AJUSTADO \times 0,21$$

Caso a demanda envolva a 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.

## 7.16. VERIFICAÇÃO DE ERROS

As verificações de erro, bem como a análise e a solução de problemas, correspondem às demandas apontadas pelo órgão, relacionadas aos comportamentos anormais ou indevidos 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 7.7).

Caso não seja constatado o problema reportado ou este decorra 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, nestes casos, a verificação de erros.

## 7.17. 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, ou seja, além das funcionalidades diretamente afetadas 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:



**PFT = 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:



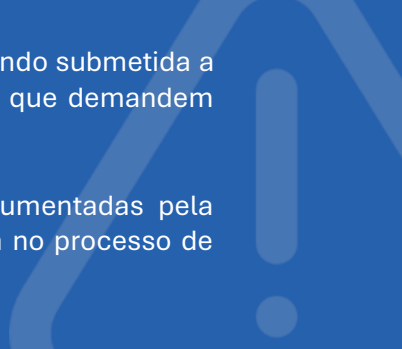
$$PF\_TESTES = PFT \times 0,15$$

$$SFP\_TESTES = PFT \times 0,15$$



**ATENÇÃO!** Uma função deve ser contabilizada apenas uma vez, mesmo quando submetida a múltiplos testes com diferentes cenários. As funcionalidades da aplicação que demandem testes devem ser formalmente requisitadas pela contratante.

As funções testadas, consideradas no PFT, devem ser devidamente documentadas pela contratada, em conformidade com a documentação de testes estabelecida no processo de desenvolvimento da contratante.



## 7.18. COMPONENTE REUSÁVEL

Em alguns casos, são demandadas inclusões ou manutenções em componentes internos de uma aplicação que implementam regras de negócio e 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 desenvolvido ou mantido, seja considerado como um processo elementar independente, sendo sua inclusão ou alteração contabilizada da seguinte forma:

- Inclusão de componente: contabilizado como PF incluído;
- Alteração de componente: aplicando-se um fator de impacto (FI) sobre o PF\_ALTERADO ou SFP\_ALTERADO, seguindo os conceitos apresentados na seção 7.3 - Projeto de Melhoria.

Vale ressaltar que na contagem da baseline da aplicação não deve constar a inclusão ou alteração de componentes, visto que a *baseline* representa o tamanho funcional da aplicação que está sendo medida.



$$\text{PF\_COMPONENTE} = \text{PF\_INCLUÍDO} \text{ ou } \text{FI} * \text{x} \text{ PF\_ALTERADO}$$
$$\text{SFP\_COMPONENTE} = \text{SFP\_INCLUÍDO} \text{ ou } \text{FI} * \text{x} \text{ SFP\_ALTERADO}$$



Além disso, as funcionalidades da aplicação que necessitem de teste devem ser requisitadas pela contratante e dimensionadas conforme proposto na seção 7.17.

Um exemplo de manutenção de componentes é a mudança em tópico de 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.

Para esses casos, a aferição do tamanho em pontos de função dos arquivos alterados deve ser feita da seguinte forma:

- Para o método APF: com a aplicação de um fator de impacto de 13% sobre a contagem de uma EE de baixa complexidade (3 PF), ou seja 0,39 PF;

- Para o método SFP: com a aplicação de um fator de impacto de 13% sobre a contagem de um processo elementar (4,6 SFP), ou seja 0,59 SFP.



$$\text{PF\_COMPONENTE\_ARQUIVO} = 0,39 \text{ PF} \times \text{QTD\_ARQUIVOS\_ALTERADOS}$$
$$\text{SFP\_COMPONENTE\_ARQUIVO} = 0,59 \text{ SFP} \times \text{QTD\_ARQUIVOS\_ALTERADOS}$$

## 8. ORIENTAÇÕES COMPLEMENTARES PARA CONTAGEM DE PF

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, em caso de falhas, podem impactar significativamente o resultado das contagens, a saber:

- Contagem com múltiplas mídias (item 8.1 e subitens relacionados);
- Log, trilha de auditoria e histórico (item 8.2 e subitens relacionados);
- Abas e identificação de processo elementar (item 8.3); e
- Principais falhas de contagem identificadas (item 8.4).

### 8.1. 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 à adoção de duas abordagens alternativas, a saber: *single instance* e *multiple instance*.

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 formatos como PDF, DOC e XLS, bem como as 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 ou para fora da 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 em oposição à 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. Nesse caso, a notícia completa somente 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.

### 8.1.1. 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 seja requisitada pelo usuário, na tela em questão.

Nesses casos, recomenda-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.

### 8.1.2. 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.

### 8.1.3. 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 seguir o padrão da ferramenta de desenvolvimento e tiver que ser customizada para o cliente, então será utilizada a abordagem *multiple instance*.

#### 8.1.4. CENÁRIO 4: Dados de entrada e saída batch e on-line idênticos

Uma informação pode ser carregada na aplicação por meio de dois métodos: arquivo *batch* e entrada *on-line*. Um mesmo conjunto de informações pode ser emitido *batch* e *on-line*. O processamento do arquivo *batch* executa validações durante o processamento. O processamento *on-line* também executa validações das informações.

Para esses casos, deve-se considerar a utilização da abordagem *multiple instance*, que conta duas funcionalidades: a entrada ou saída de dados *batch* e a entrada ou saída 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*, por isso são contadas duas funcionalidades.

#### 8.1.5. CENÁRIO 5: Múltiplos canais de entrega da mesma funcionalidade

Caso uma funcionalidade deva ser disponibilizada em múltiplos canais, por exemplo consulta de dados em página *Web* e consulta de dados no telefone celular.

Se a funcionalidade for desenvolvida duas vezes para os dois canais (consulta de dados na *Web* e consulta de dados via celular), deve ser utilizada a abordagem *multiple instance*.

#### 8.1.6. CENÁRIO 6: Relatórios em múltiplos formatos

Um relatório deve ser entregue em diferentes formatos, por exemplo, em um arquivo *html* e um formato de valores separados por vírgula.

Nestes casos, conforme sugerido na abordagem *multiple instance*, considera-se a ferramenta utilizada na geração dos relatórios:

- Se a equipe de desenvolvimento precisar desenvolver o relatório nos dois formatos na ferramenta em questão, serão contadas duas funcionalidades. Porque, a lógica de processamento de análise de condições para verificar quais são aplicáveis é identificada.

- Se a ferramenta de desenvolvimento suportar um gerador de relatórios que permite ao usuário imprimir o relatório, salvar em html ou salvar no formato de valores separados por vírgula, então a funcionalidade será contada apenas uma vez, observando que a funcionalidade será da ferramenta e não da aplicação.

### 8.1.7. CENÁRIO 7: Funcionalidades fornecidas via aplicação e *Web service*

Algumas funcionalidades são fornecidas pela aplicação via interface com o usuário final e também via *Web service*. Quando há esforço para o desenvolvimento de duas funcionalidades, estas devem ser contadas duas vezes, considerando a abordagem *multiple instance*.

### 8.1.8. CENÁRIO 8: Desenvolvimento Multiplataforma

Para a construção de um determinado sistema foi solicitado que fossem entregues, em um único projeto, versões para plataformas distintas (ex: *Android* e *iOS*). Existem frameworks para desenvolvimento que permitem criar produtos em multiplataformas, contudo muitas vezes são necessários testes, homologação e implantação diferenciados para cada plataforma.

Para esses casos, deve-se utilizar a abordagem *multiple instance*, conforme descrito no item 0 (CONTAGEM DE DESENVOLVIMENTO MULTIPLATAFORMA) deste roteiro.

## 8.2. LOG, TRILHA DE AUDITORIA E HISTÓRICO

O objetivo dessa seção é descrever orientações sucintas a respeito de contagem de *log*, trilha de auditoria e histórico.

### 8.2.1. *Log*

Conceituamos o termo “*Log*” como o registro de procedimentos ou ações realizadas pela aplicação, em determinado período de tempo, com o objetivo de apoiar a auditoria do ambiente tecnológico e a identificação das causas raiz de falhas em sistemas. Diante desse conceito, definimos que o Log não deve ser mensurado em pontos de função, já que ele não armazena informações negociais reconhecidas pelo usuário da aplicação.

## 8.2.2. Trilha de auditoria

Conceituamos “Trilha de Auditoria” como a funcionalidade que tem o objetivo de armazenar informações referentes às ações realizadas pelos usuários da aplicação no passado, de modo que seja possível apurar quais foram as ações executadas durante a utilização do sistema.

Para isso, devem existir no mínimo as informações para identificar quem realizou a ação, quando e o que foi realizado, além de outras informações que o usuário da aplicação defina como necessárias.

A trilha de auditoria deve ser solicitada pelo usuário da aplicação e, para a contagem, deve existir funcionalidade específica de consulta a esses dados.

Caso a trilha de auditoria faça parte da política corporativa de segurança da informação adotada pelo contratante para todos os sistemas do órgão, ela deve ser considerada como um requisito não funcional e, portanto, não será mensurável em ponto de função.

Diante do exposto, a principal diferença entre o *Log* e a Trilha de Auditoria é:

- **Log:** apoia a coleta de informações no âmbito tecnológico, ou seja, em problemas decorrentes da arquitetura tecnológica que precisam ser investigados, por meio da análise do conjunto de procedimentos executados pela aplicação, como exemplo a baixa performance no sistema, travamentos e outros comportamentos inesperados.
- **Trilha de Auditoria:** apoia a auditoria para os dados de negócio, armazenando informações das ações realizadas pelo usuário na aplicação.

## 8.2.3. Histórico

Conceituamos “Histórico” como um registro de estados com informações anteriores de um registro em determinado momento. O usuário poderá consultar a evolução dessas informações em uma linha do tempo e sua existência é justificada por necessidades de negócio. Assim, para fazer parte do tamanho funcional, deve ser solicitado pelo gestor e deverá existir funcionalidade de consulta a tais dados.

A função de consulta aos dados de um histórico deverá ser contada de acordo com as regras de contagem das funções transacionais do CPM.

Não devem ser contabilizadas funções transacionais separadas para incluir, alterar e/ou excluir as informações históricas, pois o armazenamento dessas informações é parte integrante das mesmas funcionalidades que processam os dados de negócio. Apenas quando o histórico for mantido de forma independente do registro principal, por exemplo no caso do arquivo lógico interno (APF: ALI; SFP: AL) principal ter sido excluído, o histórico se torna um arquivo lógico ALI/AL independente e não um registro lógico do ALI/AL relacionado.

### 8.3. ABAS E IDENTIFICAÇÃO DE PROCESSO ELEMENTAR

Um Processo Elementar é a menor unidade de atividade que é significativa para o usuário. 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 deixa de reunir os critérios de um processo elementar.

Como exemplo, podemos considerar o processo elementar “Incluir Empregado”. Para sua *interface*, foi utilizado o recurso de abas: uma destinada ao cadastro dos dados gerais do empregado, outra para os dados de endereço e uma terceira para o registro dos dependentes.

Na contagem de PF, esse cenário corresponde a apenas um processo elementar – “Incluir Empregado” (EE no método APF; PE no método SFP) –, já que cada aba, isoladamente, não constitui uma transação completa.

Neste exemplo, o cadastro do empregado somente é finalizado quando todas as informações distribuídas entre as três abas são preenchidas, conforme os requisitos do usuário. Assim, neste caso, a menor unidade de atividade significativa para o usuário é justamente o processo elementar “Incluir Empregado”.

Podem existir cenários em que cada aba representa um processo elementar autocontido, reconhecido pelo usuário, independente e com requisitos funcionais e lógica de processamento distintos. Por isso é importante ressaltar que, independentemente da tecnologia empregada, cabe ao analista consultar as orientações do CPM 4.3.1 e do SPM 2.1 para identificar o processo elementar e aplicar as regras de contagem de PF corretamente.

## 8.4. 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 divergências significativas no quantitativo de pontos de função mensurados. A listagem abaixo apresenta situações que constituem falhas de contagem:

- a) Identificar 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) Contar DER para variáveis de paginação, número de páginas e informação de posicionamento;
- e) Contar DER para ajudas de navegação como habilidade de navegar com uma lista utilizando “anterior”, “próximo”, “primeiro”, “último” e seus equivalentes gráficos;
- f) Contar mais de um DER por função transacional para a habilidade de iniciar ações quando há múltiplos meios para isso;
- g) Contar mais de um DER quando a função transacional tem capacidade de enviar várias mensagens.



Convém destacar que algumas 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 é necessária a diferenciação das funções transacionais e das funções de dados, tampouco as suas complexidades. Também não é necessária a identificação de DERs, ALRs ou RLRs, tampouco a identificação da “intenção primária”.

## 8.4.1. Dados de Código

Dentre as principais falhas de contagem identificadas, destacamos que a contagem de Dados de Código não deve ser considerada 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].

Ao contrário dos dados de código, os dados de negócio refletem a informação necessária a ser armazenada e recuperada pela área funcional, geralmente um usuário do negócio.

A seguir apresenta-se um quadro comparativo contendo algumas características lógicas que distinguem dados de código dos dados de negócio, descritas no CPM 4.3.1:

**Tabela 9 - Diferenças entre Dados de Código e Dados de Negócio**

	<b>Características lógicas dos Dados de Código</b>	<b>Características lógicas dos Dados de Negócio</b>
<b>1</b>	O dado é obrigatório para a área funcional, mas é opcionalmente armazenado como um arquivo de dados	Os dados são obrigatórios para a operação da área funcional do usuário
<b>2</b>	Não é geralmente identificado como parte dos requisitos funcionais do usuário; é geralmente identificado como parte da solução para atender requisitos não-funcionais do usuário	É identificável pelo usuário (geralmente por um usuário do negócio)
<b>3</b>	É algumas vezes mantido pelo usuário (geralmente por uma pessoa de suporte ao usuário)	Capaz de ser mantido pelo usuário (geralmente um usuário do negócio)
<b>4</b>	Armazena dados para padronizar e facilitar atividades e transações de negócio. As transações de negócio referenciam os Dados de Código para melhorar a facilidade da entrada de dados, melhorar a consistência dos dados, garantir a integridade dos dados, etc	Armazena os Dados Principais do usuário para apoiar transações de negócio
<b>5</b>	É essencialmente estático – apenas muda em resposta a mudanças na forma como o negócio funciona	Muito dinâmicos – as operações normais de negócio fazem com que sejam regularmente referenciados e rotineiramente incluídos, alterados ou excluídos

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 mantêm devem ser contadas. Como consequência, o CPM 4.3.1 afirma que:

- Dados de Código não podem ser considerados como Arquivos Lógicos (ALI ou AIE);
- Dados de Código não podem ser considerados com RLR ou DER em um ALI ou AIE;
- Não se deve contar Funções Transacionais para manutenção de Dados de Código;
- Dados de Código não podem ser considerados Arquivos Lógicos Referenciados (ALR) ao avaliar a complexidade de uma função transacional.

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.

Para auxiliar a identificação de Dados de Código, o referido manual traz uma lista, não exaustiva, de três áreas gerais em que os vários tipos diferentes de dados de código podem ser enquadrados:

**Tabela 10 - Tipos e Exemplos de Dados de Código conforme CPM 4.3.1**

TIPOS	Descrição	Exemplos
<b>Dados de substituição</b>	Código + Descrição  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.  Este tipo de Dado de Código contém um código e um nome explicativo ou descrição.	Estados: Código do Estado, Nome do Estado;  Cores: Código da Cor, Descrição da Cor.
<b>Dados Estáticos e Constantes</b>	Uma ocorrência  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.	Uma entidade com dados sobre uma organização em particular; por exemplo, nome e endereço.
	Dados estáticos  Contém dados que são estáticos. A quantidade de instâncias e o conteúdo de uma instância raramente muda.	Uma entidade elementos químicos: símbolo, número atômico, descrição.

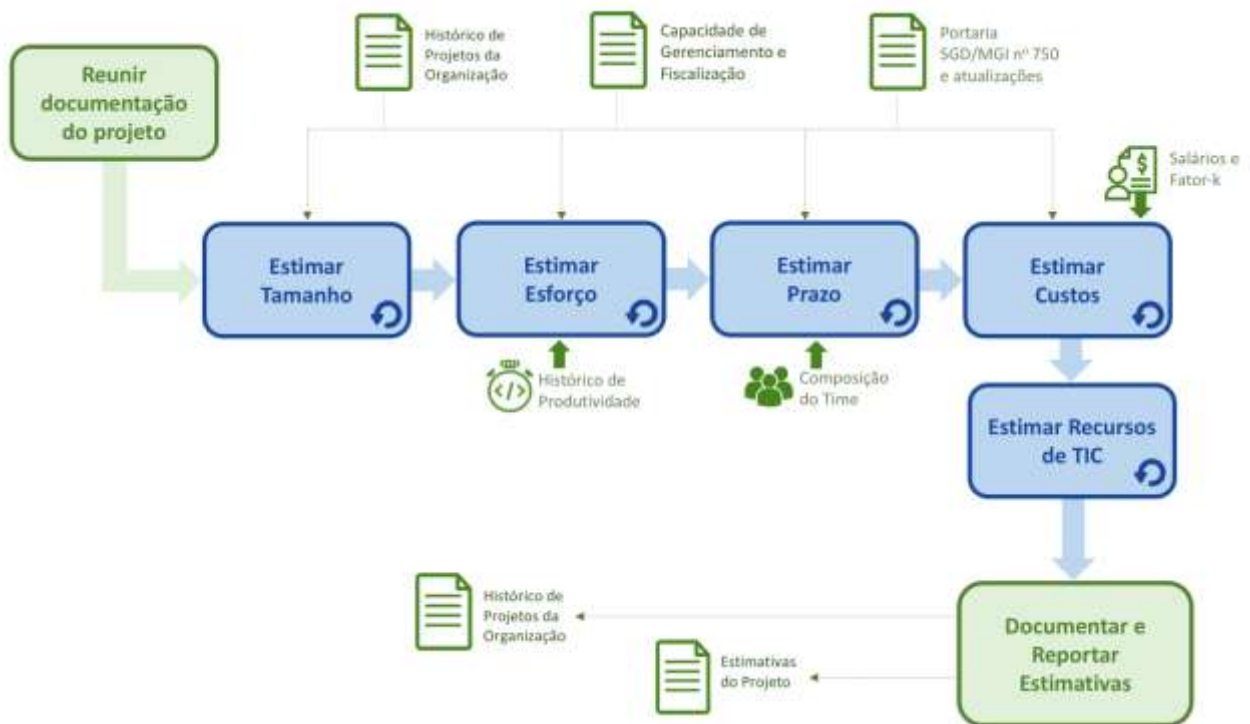
	Valores <i>Default</i>	Este tipo de Dado de Código contém valores <i>default</i> para (alguns atributos em) novas instâncias de um objeto de negócio.	
<b>Dados com Valores Válidos</b>	Valores Válidos	Fornece uma lista de valores válidos para um atributo de um ou mais tipos de objetos de negócio. É normalmente 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 válidos	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.

## 9. ORIENTAÇÕES PARA REALIZAR ESTIMATIVAS DE PROJETOS DE SW

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

**Figura 4 - Processo de Derivação de Estimativas de Projetos de Software**



Com o objetivo de facilitar o entendimento, nos tópicos seguintes são descritas as 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 de decisões, adaptações, inovação e entrega de 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.

## 9.1. REUNIR A DOCUMENTAÇÃO DISPONÍVEL

As estimativas normalmente são realizadas no início do processo de desenvolvimento ou melhoria do *software*. A primeira atividade 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: documentos 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.

Ressalta-se que as estimativas apresentadas neste capítulo têm caráter referencial para planejamento e não substituem os critérios de medição e pagamento definidos contratualmente.

## 9.2. 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. Como já foi mencionado, o Ponto de Função Simples (Simple Function Point - SFP) é um método de medição funcional compatível e consistente com o IFPUG APF, descrito no CPM 4.3.1, em termos de terminologia e definições.

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 6.1 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\_INCLUÍDO) e as funcionalidades de conversão de dados (SFP\_CONVERSÃO) que apoiam o início do uso do software, conforme fórmula a seguir:



$$\text{ESTIMATIVA DE TAMANHO (SFP)} = \text{SFP\_INCLUÍDO} + \text{SFP\_CONVERSÃO}$$



A definição dos termos SFP\_INCLUÍDO e SFP\_CONVERSÃO pode ser encontrada no item 4 deste roteiro.

### 9.2.1. Exemplo de Estimativa de Tamanho com SFP

Considere o Projeto Exemplo em que foram realizadas todas as etapas do procedimento de contagem exposto no item 6.1 deste roteiro para estimar o tamanho do software a ser desenvolvido com o método SFP.

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

Para essas funções de dados e transacionais, foi atribuída a pontuação funcional determinada pelo método SFP (PE=4,6 SFP / AL=7 SFP). O tamanho estimado do Projeto Exemplo foi obtido usando a fórmula apresentada no item 9.2, conforme a seguir:

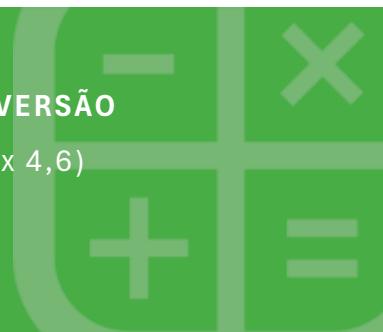


$$\text{ESTIMATIVA DE TAMANHO (SFP)} = \text{SFP\_INCLUIDO} + \text{SFP\_CONVERSÃO}$$

$$\text{ESTIMATIVA DE TAMANHO (SFP)} = [(95 \times 4,6) + (20 \times 7)] + (5 \times 4,6)$$

$$\text{ESTIMATIVA DE TAMANHO (SFP)} = 437 + 140 + 23$$

$$\text{ESTIMATIVA DE TAMANHO (SFP)} = 600 \text{ SFP}$$



### 9.3. ESTIMATIVA DE ESFORÇO

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:



$$\text{ESTIMATIVA DE ESFORÇO (HORAS)} = \text{Tamanho (SFP)} \times \text{Índice de Produtividade (H/SFP)}$$



A produtividade para projetos ágeis de TI deve considerar dados históricos dos projetos já realizados, tecnologias, ferramentas e processos de desenvolvimento de software aplicados pelo órgão, devendo ser formalizada no instrumento convocatório ou em padrão definido pelo órgão.

Destaca-se que a utilização de recursos de Inteligência Artificial pode impactar a produtividade dos times. Assim, recomenda-se que o órgão avalie os efeitos do uso desses recursos na execução dos serviços e considere seus reflexos na definição do índice de produtividade.

Para fins deste roteiro, adota-se como referência o índice de produtividade de 10 horas por Ponto de Função (10 H/PF).

### 9.3.1. Exemplo de Estimativa de Esforço

Para o cálculo do esforço, vamos considerar o **Projeto Exemplo** descrito no subitem 9.2.1, cuja estimativa de tamanho é de 600 SFP. Aplicando a fórmula para estimativa de esforço e utilizando índice de produtividade de 10 H/SFP temos:



$$\text{ESTIMATIVA DE ESFORÇO (HORAS)} = \text{Tamanho (SFP)} \times \text{Índice de Produtividade (H/SFP)}$$

$$\text{ESTIMATIVA DE ESFORÇO (HORAS)} = 600 \times 10$$

$$\text{ESTIMATIVA DE ESFORÇO (HORAS)} = 6000 \text{ horas}$$



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.

### 9.3.2. Distribuição de Referência do 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 *software*; para tanto, serão utilizados os processos técnicos da ISO/IEC 12207 conforme exposto no item 10.1 deste roteiro.

A seguir, apresenta-se a tabela com os processos técnicos da norma ISO/IEC 12207, detalhada no item 10.1 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, sempre vinculando às entregas efetivas, inerentes a cada processo ou atividade previstas na norma ISO/IEC 12207.

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

**Tabela 11 - Distribuição do Esforço utilizando processos técnicos da norma ISO/IEC 12207**

Processos técnicos da ISO/IEC 12207		Quantidade de Atividades do processo técnico (A)	Percentual por processo (C= A/B)
P1	Processo de definição das necessidades e requisitos das partes interessadas;	6	16 %
P2	Processo de definição de requisitos de Sistema/ <i>Software</i> ;	4	11 %
P3	Processo de definição de Arquitetura;	6	16 %
P4	Processo de definição do Projeto/ <i>design</i> ;	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 %
<b>Total de Atividades (B)</b>		<b>38</b>	<b>100%</b>



**ATENÇÃO!** O órgão pode definir outros macroprocessos e subdividi-los para melhor aderência à sua metodologia e aos marcos de entrega, sempre vinculando às entregas efetivas, inerentes a cada processo ou atividade previstas na norma ISO/IEC 12207.

## 9.4. ESTIMATIVA DE PRAZO

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.

### 9.4.1. 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 Exemplo**:

**Tabela 12 - 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

### 9.4.2. Passo 2: Estimar a produtividade do time de referência

Estimar a produtividade do time de referência também contribui para 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** correspondem à quantidade de profissionais por perfil, que compõem o time de referência;
- **Aloc** é o percentual de alocação de cada profissional;
- **160** é a carga horária mensal estimada (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 Exemplo**, descrito no subitem 9.2.1.

A tabela a seguir apresenta o cálculo da carga horária mensal de cada perfil profissional do time de referência (ver tabela do subitem 9.4.1), 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 13 - Carga horária mensal do time de referência**

<b>Perfis Profissionais</b>	<b>Qtd estimada de profissionais por time (A)</b>	<b>% Alocação (B)</b>	<b>Carga Horária mensal do perfil = A*B*160</b>
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
<b>CARGA HORÁRIA MENSAL DO TIME:</b>			<b>586 H</b>

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/SFP. Contudo, o índice de produtividade deve considerar dados históricos dos projetos já realizados, tecnologias, ferramentas e processos de desenvolvimento de software aplicados pelo órgão, devendo ser formalizada no instrumento convocatório ou em padrão definido pelo órgão.

Destaca-se que a utilização de recursos de Inteligência Artificial pode impactar a produtividade dos times. Assim, recomenda-se que o órgão avalie os efeitos do uso desses recursos na execução dos serviços e considere seus reflexos na definição do índice de produtividade.

### 9.4.3. 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 duas 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:


$$\text{(Fórmula 1) ESTIMATIVA DE PRAZO (MESES) = ESFORÇO (H) / CARGA HORÁRIA MENSAL DO TIME (H)}$$

$$\text{(Fórmula 2) ESTIMATIVA DE PRAZO (MESES) = TAMANHO (SFP) / PRODUTIVIDADE MÍNIMA PARA O MÊS (SFP/MÊS)}$$


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


$$\text{ESTIMATIVA DE PRAZO (SPRINTS) = TAMANHO (SFP) / PRODUTIVIDADE MÍNIMA POR SPRINT}$$


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 Exemplo:

- Estimativa de tamanho: 600 SFP;

- Estimativa de esforço: 6000 Horas;
- Carga horária mensal do Time de referência: 586 horas;
- Produtividade mínima esperada para o time de referência: 58,6 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)**

ESTIMATIVA DE PRAZO (MESES) = 6000/ 586 ≈ 10 MESES

**ESTIMATIVA DE PRAZO (MESES) = TAMANHO (SFP) / PRODUT. MÍN. P/ O MÊS (SFP/MÊS)**

ESTIMATIVA DE PRAZO (MESES) = 600/58,6 ≈ 10 MESES

**ESTIMATIVA DE PRAZO (SPRINTS) = TAMANHO (SFP) / PRODUT. MÍN. POR SPRINT**



## 9.5. ESTIMATIVA DE CUSTO

Esse roteiro recomenda 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 9.4.1 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 do ponto de função é derivado do custo mensal do time e da produtividade esperada, conforme fórmulas a seguir:



$$\text{ESTIMATIVA DE CUSTO DO PF (EcPF)} = E_{cm} / P_e$$
$$\text{ESTIMATIVA DE CUSTO DO PROJETO (EcP)} = \text{Tamanho SFP} \times E_{cPF}$$



Onde:

- **E<sub>cm</sub>** corresponde a Estimativa de Custo Mensal do Time Ágil de Referência;
- **P<sub>e</sub>** corresponde a Produtividade Esperada para o mês.
- **E<sub>cPF</sub>** corresponde a estimativa de custo do PF.

No subitem 9.2 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-se o termo ESTIMATIVA DE CUSTO DO PF (E<sub>cPF</sub>). Vale lembrar que SFP 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.

### 9.5.1. 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 Exemplo**:



**ATENÇÃO!** Os valores constantes na tabela a seguir são meramente ilustrativos, devendo-se utilizar os valores vigentes na Portaria SGD/MGI nº 750 e suas atualizações, ou os valores de novos perfis incluídos pelo órgão observando as diretrizes desta portaria.

**Tabela 14 - Cálculo da estimativa de custo mensal do time de referência**

Identificação do Perfil	Salário (S)	Fator K	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\$11.966,67	1,94	R\$23.215,34	33,00%	52,8	1	R\$145,10	R\$7.661,06
Desenvolvedor Pleno	R\$10.713,73	1,98	R\$21.213,18	50%	80	1	R\$132,58	R\$10.606,59
Desenvolvedor Sênior	R\$15.750,00	1,95	R\$30.712,50	100%	160	1	R\$191,95	R\$30.712,50
Arquiteto Sênior	R\$18.542,27	1,94	R\$35.971,99	33%	52,8	1	R\$224,82	R\$11.870,76
Analista de Negócios/ Requisitos Pleno	R\$8.744,98	2,01	R\$17.577,41	50%	80	1	R\$109,86	R\$8.788,70
Analista de Negócios/ Requisitos Sênior	R\$11.227,93	1,98	R\$22.231,30	50%	80	1	R\$138,95	R\$11.115,65
Administrador de Dados Sênior	R\$11.345,67	1,97	R\$22.350,97	25%	40	1	R\$139,69	R\$5.587,74
Analista de Testes/Qualidade – Sênior	R\$11.081,16	1,98	R\$21.940,70	25%	40	1	R\$137,13	R\$5.485,17
<b>CUSTO MENSAL DO TIME</b>								R\$91.828,18

A tabela acima é meramente ilustrativa e 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 presentes na Portaria SGD/MGI nº 750, de 20 de março de 2023 (alterada pela Portaria SGD/MGI nº 6.679, de 17 de setembro de 2024).

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 Exemplo**:

**Tabela 15 - Estimativas do Projeto Exemplo**

<b>Estimativa de Tamanho do Projeto em SFP</b>	<b>600 SFP</b>
<b>Índice de Produtividade Mínima</b>	10 H/SFP
<b>Estimativa de Esforço do Projeto</b>	6000 Horas
<b>Total de Horas do Time por mês</b>	586 Horas
<b>Produtividade esperada para o mês</b>	58,6 SFP/mês
<b>Estimativa de Prazo em meses</b>	10 meses
<b>Estimativa de Custo mensal do Time (Ecm*)</b>	R\$91.828,18
<i>*Ecm corresponde a soma do custo mensal dos profissionais do time, conforme a sua taxa de alocação</i>	

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



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

$$\text{ESTIMATIVA DE CUSTO DO PF (EcPF)} = 91.828,18 / 58,6$$

$$\text{ESTIMATIVA DE CUSTO DO PF (EcPF)} = \text{R}\$1.567,03$$



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):



$$\text{ESTIMATIVA DE CUSTO DO PROJETO} = \text{Tamanho PF} \times \text{EcPF}$$

$$\text{ESTIMATIVA DE CUSTO DO PROJETO} = 600 \times \text{R}\$1.567,03$$

$$\text{ESTIMATIVA DE CUSTO DO PROJETO} = \text{R}\$ 940.218,00$$



## 9.6. 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.

## 10. FATORES DE IMPACTO

Considerando a diversidade de realidades existentes nos órgãos da Administração Pública Federal, adotou-se, neste roteiro, uma abordagem baseada em processos de referência que se adaptam aos diferentes ciclos de vida de software das organizações do SISP.

Para a definição dos percentuais de esforço e dos Fatores de Impacto (FI) utilizados nas fórmulas de contagem de pontos de função, foram considerados processos técnicos derivados da norma ABNT NBR ISO/IEC 12207:2017, utilizados como referência conceitual para a estruturação da memória de cálculo apresentada neste documento.

Ressalta-se que a referida norma estabelece uma estrutura de referência para os processos do ciclo de vida de software, sem prescrever percentuais de esforço ou pesos relativos entre atividades. Dessa forma, os valores de FI definidos neste roteiro representam uma modelagem própria, construída a partir da seleção e agregação de processos técnicos considerados relevantes para o contexto de desenvolvimento e manutenção de software no âmbito do SISP.

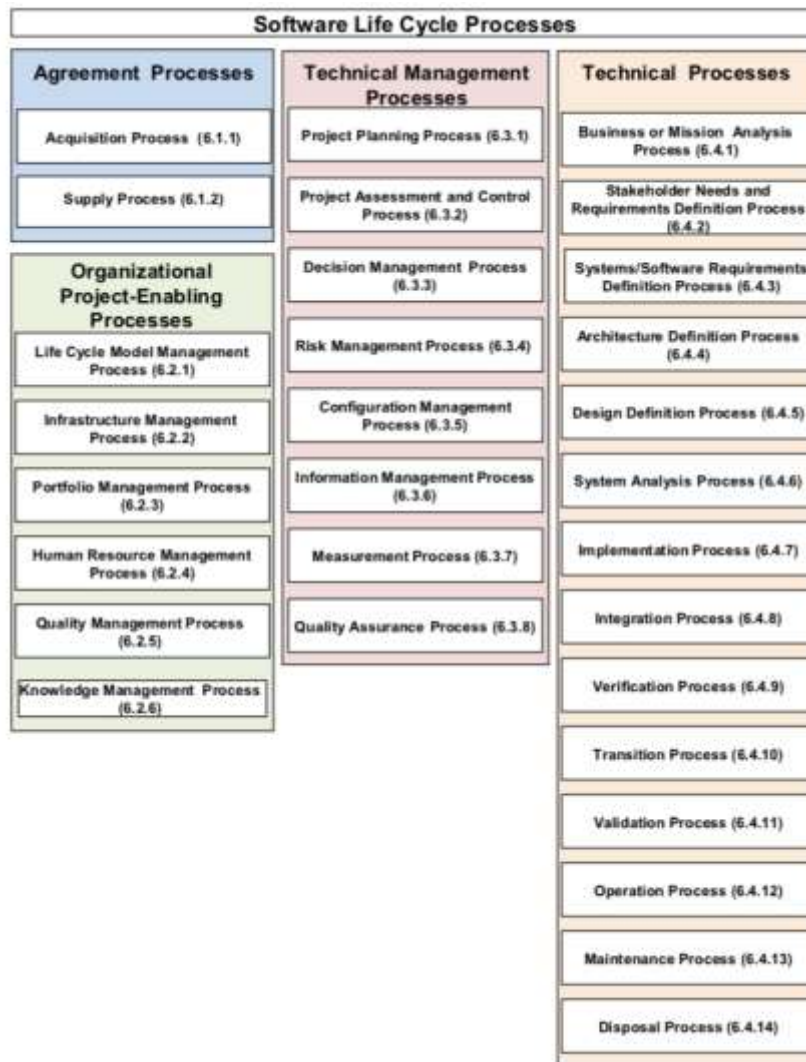
Nesse sentido, foram definidos:

- Os percentuais de esforço para cada fase ou macroprocesso do projeto (tabela 16);
- os Fatores de Impacto (FI) aplicados nas fórmulas de contagem de PF para projetos de desenvolvimento e manutenção de software (tabelas 17, 18 e 19).

A ISO/IEC 12207 baseia-se 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, conforme figura a seguir:

- Processos contratuais;
- Processos de habilitação de projetos organizacionais;
- Processos técnicos de gestão; e
- Processos técnicos.

Figura 5 - Processos do ciclo de Vida do Software da ISO/IEC 12207 (Fonte: ISO/IEC 12207)



O escopo deste roteiro abrange exclusivamente 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 este 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 referência conceitual para estruturação da memória de cálculo dos fatores de impacto (FI) relacionados à contagem de pontos de função.

A seguir, listam-se 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 fins deste roteiro, foram utilizados exclusivamente processos técnicos diretamente relacionados ao desenvolvimento de software, não sendo considerados, na memória de cálculo dos Fatores de Impacto, processos de natureza contratual, organizacional, operacional ou de descarte.

Adicionalmente, adotou-se uma abordagem objetiva para fins de mensuração, na qual as atividades associadas aos processos técnicos selecionados foram consideradas com peso uniforme, permitindo a padronização do cálculo dos Fatores de Impacto.

No que diz respeito a conformidade com a norma 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. Nesse sentido, foi adotada uma aplicação parcial e adaptada dos processos técnicos, conforme necessidade do modelo de mensuração.

## **10.1. DEFINIÇÃO DOS FATORES DE IMPACTO (FI)**

Para a definição dos Fatores de Impacto (FI) utilizados nas fórmulas de contagem apresentadas neste roteiro, foram considerados dez processos técnicos aplicados ao desenvolvimento de *software* da ISO/IEC 12207, bem como suas respectivas atividades de referência.

Este roteiro não tem por finalidade prescrever um modelo específico de ciclo de vida para os órgãos do SISP, nem determinar uma sequência obrigatória de processos dentro desse ciclo. Assim, recomenda-se que cada

órgão adote os processos que compõem o seu ciclo de vida de software em conformidade com sua própria metodologia de desenvolvimento.

A utilização dos processos técnicos e das atividades definidas na ISO/IEC 12207 confere maior flexibilidade aos órgãos contratantes para ajustar os fatores de impacto (FI), conforme as particularidades de seus projetos e dos seus processos de desenvolvimento de *software*.

Podem existir cenários em que:

- **O órgão contratante não tem a necessidade de contratar todas as fases do ciclo de vida de desenvolvimento de *software*:** nesses cenários, a contratada será remunerada pela contagem de pontos de função considerando apenas os percentuais dos macroprocessos contratados, conforme os níveis percentuais da tabela 16.
- **O órgão contratante identifique a necessidade de executar processos e/ou atividades não contemplados no cálculo de FI do SISP:** por exemplo, quando um projeto de desenvolvimento ou manutenção demandar um conjunto distinto de processos e/ou atividades presentes na memória de cálculo de FI do SISP (tabelas 17, 18 e 19), será possível recalculá-lo. Nessa situação, o órgão deverá considerar os processos e atividades previstos na ISO/IEC 12207 e registrar a justificativa para o ajuste, documentando a memória de cálculo do novo FI adotado. Também deverá indicar os entregáveis correspondentes como evidências dos processos e atividades efetivamente aplicados.



**A utilização dos processos técnicos e das atividades previstas na ISO/IEC 12207 para definição dos Fatores de Impacto (FI) proporciona aos órgãos contratantes maior flexibilidade para ajustar os FI na contagem de PF, conforme as particularidades de seus processos e projetos de software.** Essa abordagem favorece uma aferição transparente e aderente à realidade de cada projeto.

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

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

Ressalta-se que os nomes dos processos técnicos, bem como da finalidade e atividades relacionadas, apresentados a seguir correspondem a adaptações terminológicas dos processos definidos na norma, mantendo sua equivalência conceitual.

**Tabela 16 - Processos técnicos da norma ISO/IEC 12207 utilizados para definição dos FIs**

Processos Técnicos ISO/IEC 12207		Finalidade	Atividades dos Processos Técnicos da ISO/IEC 12207		Qtd de atividades por processo técnico (A)	Percentual por processo técnico (C=A/B)
<b>P1</b>	Processo de definição das necessidades e requisitos das partes interessadas	O objetivo do processo de definição das necessidades e requisitos das partes interessadas é definir os requisitos das partes interessadas para um sistema que possa fornecer os recursos necessários aos usuários e outras partes interessadas em um ambiente definido.	P1.1	Preparar-se para a definição das necessidades e requisitos das partes interessadas	6	16%
			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		
<b>P2</b>	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	4	11%
			P2.2	Definir requisitos de sistema/software		
			P2.3	Analisar requisitos de sistema/software		
			P2.4	Gerenciar requisitos de sistema/software		
<b>P3</b>	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	P3.1	Preparar-se para a definição da arquitetura.	6	16%
			P3.2	Desenvolver pontos de vista de arquitetura		
			P3.3	Desenvolver modelos e visões de arquiteturas candidatas.		

		expressar isso em um conjunto de visões consistentes.	P3.4	Relacionar a arquitetura com o design		
			P3.5	Avaliar arquiteturas candidatas		
			P3.6	Gerenciar a arquitetura selecionada		
<b>P4</b>	Processo de definição do projeto	O objetivo é fornecer dados e informações detalhadas suficientes sobre o sistema e seus elementos para permitir a implementação consistente com as entidades arquitetônicas definidas nos modelos e visualizações da arquitetura do sistema.	P4.1	Preparar-se para a definição do projeto do sistema de software	4	11%
			P4.2	Estabelecer projetos relacionados a cada elemento do sistema de software		
			P4.3	Avaliar alternativas para obtenção de elementos de sistemas de software		
			P4.4	Gerenciar o projeto		
<b>P5</b>	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	3	8%
			P5.2	Realizar a análise do sistema		
			P5.3	Gerenciar a análise do sistema		
<b>P6</b>	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	3	8%
			P6.2	Realizar a implementação		
			P6.3	Gerenciar os resultados da implementação		
<b>P7</b>	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	3	8%
			P7.2	Realizar a integração		
			P7.3	Gerenciar resultados de integração		
<b>P8</b>	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	3	8%
			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 partes interessadas no ambiente operacional.	P9.1	Preparar-se para a transição do sistema de software	3	8%
			P9.2	Executar a transição		
			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	3	8%
			P10.2	Realizar a validação		
			P10.3	Gerenciar resultados de validação		
<b>Total de Atividades (B):</b>					<b>38</b>	<b>100%</b>



**ATENÇÃO!** O órgão pode definir outros macroprocessos e subdividi-los para melhor aderência à sua metodologia e aos marcos de entrega, sempre vinculando às entregas efetivas, inerentes a cada processo ou atividade previstas na norma ISO/IEC 12207.

Para definir os percentuais associados a cada processo, todas as atividades foram tratadas com o mesmo peso (peso igual a 1), de forma a facilitar a aplicação prática do método.

As tarefas relacionadas a cada uma dessas atividades podem ser consultadas na documentação da norma ISO/IEC 12207.

## 10.2. MEMÓRIA DE CÁLCULO DOS FATORES DE IMPACTO (FI)

A seguir, apresenta-se a memória de cálculo para a definição dos fatores de impacto (FI) utilizados no item 7 e em outras partes deste roteiro:

- **Na tabela 17** apresenta-se a memória de cálculo dos fatores de impacto (FI) dos projetos descritos nos subitens 7.1 a 7.8.2 (item 7 – CÁLCULO DE PF PARA O SISP) deste roteiro;
- **Na tabela 18** apresenta-se a memória de cálculo dos fatores de impacto (FI) dos projetos descritos nos subitens 7.9.1 a 7.12.4 (item 7 – CÁLCULO DE PF PARA O SISP), deste roteiro; e
- **Na tabela 19** apresenta-se a memória de cálculo dos fatores de impacto (FI) dos projetos descritos nos:
  - Subitens 7.13 a 7.18, presentes no item 7 – CÁLCULO DE PF PARA O SISP deste roteiro;
  - Subitens 12.5.1 e 12.5.2, referente a contagem de PF para desenvolvimento Multiplataforma;
  - Subitens 13.9.1 e 0, referente a Contagem de PF para projetos DW (item 13).

**Tabela 17 - Memória de Cálculo de Ponto de Função para o SISP – Parte 1**

Cálculo de PF para o SISP		Projeto de Desenvolvimento	Projeto de Melhoria / Funcionalidade desenvolvida	Projeto de Melhoria / Funcionalidade não desenvolvida COM	PF Excluído	Projeto de Migração de dados	Manutenção Corretiva	Manutenção Corretiva (em PE desenvolvido por outra empresa)	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/
Processos Técnicos e Atividades (ISO/IEC 12207)		7.1	7.3	7.3		7.6	7.7	7.7	7.8.1	7.8.1	7.8.2	7.8.2	7.8.2
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		1		
	2.2	1	1	1			1	1	1		1		

	2.3	1	1	1	1		1	1	1		1		
	2.4	1	1	1	1		1	1	1		1		
<b>P3</b>	3.1	1		1					1	1	1	1	
	3.2	1		1					1	1	1	1	
	3.3	1		1					1	1	1	1	
	3.4	1		1					1	1	1	1	
	3.5	1		1					1	1	1	1	
	3.6	1		1		1			1	1	1	1	
<b>P4</b>	4.1	1		1		1			1	1	1	1	
	4.2	1	1	1	1	1			1	1	1	1	
	4.3	1	1	1	1	1			1	1	1	1	
	4.4	1	1	1	1	1			1	1	1	1	
<b>P5</b>	5.1	1	1	1		1		1	1	1	1	1	
	5.2	1	1	1	1	1		1	1	1	1	1	
	5.3	1	1	1	1	1		1	1	1	1	1	
<b>P6</b>	6.1	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
	6.3	1	1	1	1	1	1	1	1	1	1	1	1
<b>P7</b>	7.1	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
	7.3	1	1	1	1	1	1	1	1	1	1	1	1
<b>P8</b>	8.1	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
	8.3	1	1	1	1	1	1	1	1	1	1	1	1
<b>P9</b>	9.1	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
	9.3	1	1	1	1	1	1	1	1	1	1	1	1
<b>P10</b>	10.1	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

	10.3	1	1	1	1	1	1	1	1	1	1	1	1
<b>Qtd Atividades</b>		38	24	32	19	23	18	21	32	28	32	28	15
<b>FI</b>		1	0,63	0,84	0,50	0,61	0,47	0,55	0,84	0,73	0,84	0,73	0,39

**Tabela 18 - Memória de Cálculo de Ponto de Função para o SISP – Parte 2**

<b>Cálculo de PF para o SISP</b>		Atualização de Versão - Linguagem	Atualização de Versão - Browser	Atualização de Versão - Banco de Dados	Manutenção em Interface	Adaptação sem alteração de requisitos funcionais	Adaptação sem alteração de requisitos funcionais (PE não)	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
<b>Processos Técnicos e Atividades (ISO/IEC 12207)</b>		7.9.1	7.9.2	7.9.3	7.10	7.11	7.11	7.12.1.a	7.12.1.b	7.12.1.c	7.12.2	7.12.3	7.12.4
<b>P1</b>	1.1												
	1.2												
	1.3												
	1.4												
	1.5												
	1.6												
<b>P2</b>	2.1												
	2.2					1	1	1	1		1		
	2.3					1	1	1	1		1	1	
	2.4					1	1	1	1		1	1	
<b>P3</b>	3.1												
	3.2												
	3.3												

	3.4												
	3.5												
	3.6												
<b>P4</b>	4.1												
	4.2						1						
	4.3						1						
	4.4						1						
<b>P5</b>	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	
<b>P6</b>	6.1	1	1	1		1	1	1	1		1		
	6.2	1	1	1	1	1	1	1	1	1	1	1	
	6.3	1	1	1		1	1	1	1	1	1	1	
<b>P7</b>	7.1	1	1	1		1	1						
	7.2	1	1	1	1	1	1						
	7.3	1	1	1		1	1						
<b>P8</b>	8.1	1	1	1		1	1	1	1	1	1	1	
	8.2	1	1	1	1	1	1	1	1	1	1	1	1
	8.3	1	1	1		1	1	1	1	1	1	1	1
<b>P9</b>	9.1	1	1	1		1	1	1	1	1	1		
	9.2	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
<b>P10</b>	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
<b>Qtd Atividades</b>	15	15	15	5	21	24	18	18	14	18	14	6	
<b>FI</b>	0,39	0,39	0,39	0,13	0,55	0,63	0,47	0,47	0,36	0,47	0,36	0,15	

**Tabela 19 - Memória de Cálculo de Ponto de Função para o SISP - Parte 3**

Cálculo de PF para o SISP		Atualização de dados	Desenvolvimento, Manutenção e Publicação de Páginas Estáticas	Manutenção de Documentação de Sistemas Legados	Verificação de Erros	Pontos de Função de Teste	Componente - Arquivos alterados	Desenvolvimento Multiplataforma Híbrida	Desenvolvimento Multiplataforma Nativa	Alteração de Dados de Dimensões Estáticas	Metadados relacionados ao significado dos dados
<b>Processos Técnicos e Atividades (ISO/IEC 12207)</b>		7.13	7.14	7.15	7.16	7.17	7.18	12.5.1	12.5.2	13.9.1	0
<b>P1</b>	1.1										
	1.2										
	1.3										
	1.4										
	1.5										
	1.6										
<b>P2</b>	2.1			1							
	2.2			1							
	2.3			1							
	2.4		1	1							
<b>P3</b>	3.1										
	3.2										
	3.3										
	3.4										
	3.5										
	3.6										
<b>P4</b>	4.1										
	4.2										
	4.3										
	4.4		1	1							

P5	5.1			1	1						
	5.2	1		1	1						
	5.3			1	1						
P6	6.1								1		
	6.2	1	1				1		1	1	1
	6.3								1		
P7	7.1							1	1		
	7.2	1	1				1	1	1	1	1
	7.3							1	1		
P8	8.1				1	1		1	1		
	8.2	1	1		1	1	1	1	1	1	1
	8.3				1	1		1	1		
P9	9.1		1					1	1		
	9.2	1	1				1	1	1	1	1
	9.3		1					1	1		
P10	10.1				1	1		1	1		
	10.2	1	1		1	1	1	1	1	1	1
	10.3		1		1	1		1	1		
<b>Qtd Atividades</b>		6	10	8	9	6	5	12	15	5	5
<b>FI</b>		0,15	0,26	0,21	0,23	0,15	0,13	0,31	0,39	0,13	0,13

Nas duas últimas linhas das tabelas acima temos:

- **Qtd Atividades:** corresponde a estimativa da quantidade (soma) de atividades a serem executadas em projetos dos itens 7, 12 e 13 deste roteiro.
- **FI:** corresponde ao cálculo do Fator de impacto (FI) para cada tipo de projeto.

Para o cálculo do fator de impacto (FI) de cada projeto apresentado acima utilizou-se a seguinte fórmula:



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



Onde:

- **Qtd. Atividades:** corresponde à quantidade (soma) de atividades estimadas para projetos dos itens 7, 12 e 13 deste roteiro (penúltimas linhas das tabelas 17, 18 e 19);
- **38:** representa o total de atividades dos processos técnicos considerados, com base na ISO/IEC 12207, assumindo peso uniforme entre elas para fins de cálculo do FI.

## 11. CONTAGEM DE PROJETOS ÁGEIS COM SFP

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 objetivo de subsidiar a gestão, o planejamento, a medição e a contratação desses serviços na Administração Pública Federal (APF).

Uma das principais dificuldades e desafios na adoção de métodos ágeis em contratações 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.

Em função das características do processo ágil, é natural que refinamentos e mudanças em funcionalidades ocorram com maior frequência ao longo do desenvolvimento. Ainda assim, essa dinâmica não deve comprometer os princípios de economicidade e efetividade dos resultados previstos e entregues, sendo fundamental garantir a exequibilidade do projeto.

A seguir, destacam-se duas premissas, citadas na Portaria SGD/MGI nº 750, de 20 de março de 2023 e atualizações, 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 mencionada, possuem um manifesto público que estabelece 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.

A Portaria SGD/MGI nº 750 de 20 de março de 2023 e atualizações informa 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. *Independentemente 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);

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 de 20 de março de 2023 e atualizações orienta que **independente da modalidade de contratação** (remuneração por ponto de função complementado por HST, remuneração por alocação de profissionais de TI ou remuneração 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.

## 11.1. 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, devendo ser referenciado 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 e atualizações, apresenta orientações quanto à 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 mencionada, 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 a 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 dimensionar 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.

## 11.2. 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 7.3 (Projeto de Melhoria) deste roteiro.

Conforme definição apresentada no item 4 deste roteiro, refinamentos correspondem a 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 acréscimo de ponto de função ou 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 do 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 o histórico dos refinamentos e do esforço empregado em seus atendimentos.

### 11.2.1. 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, listam-se alguns fatores que 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.

### 11.3. 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.

### 11.3.1. 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 do 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 padronizada de 4,6 SFP;
- A função de dados (AL) incluída apresenta contribuição funcional padronizada 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 7.3 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 20 - Planejamento do Backlog das Sprints (N1, N2 e N3) da Release N**

<b>Release N</b>	<b>Nome da Função</b>	<b>Categoria</b> (Inc, Alt, Exc, Refin, etc)	<b>Tipo</b> (PE, AL)	<b>SFP</b>	<b>Observação</b>
<b>Sprint N1</b>	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 7.3 (7 SFP *0,47=3,29 SFP)
<b>Sprint N2</b>	Alterar Beneficiário	Inc	PE	4,6	
	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 7.3 (4,6 SFP *0,47 = 2,16 SFP)
<b>Sprint N3</b>	Emitir Relatório de Beneficiários	Inc	PE	4,6	
	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 7.3 (4,6 SFP *0,47 = 2,16 SFP)
<b>Total SFP da Release</b>				<b>28,41 SFP</b>	

### 11.3.2. 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 dois componentes funcionais básicos, com contribuição funcional padronizada: Processo Elementar (4,6 SFP) e Arquivo Lógico (7 SFP);
- Não há necessidade de contagem detalhada dos DER, ALR;
- Não há alteração da complexidade das 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 classificação de complexidade entre as funcionalidades, a contribuição funcional de cada processo elementar ou arquivo lógico não vai ser alterada 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, para a gestão de mudanças e para a gestão do projeto;
- As alterações em funcionalidades já desenvolvidas em *releases* anteriores foram caracterizadas como projeto de melhoria, conforme item 7.3 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 21 - Contagem SFP da Release N**

<b>Release N</b>	<b>Nome da Função</b>	<b>Categoria</b> (Inc, Alt, Exc, Refin)	<b>Tipo</b> (PE, AL)	<b>SFP</b>	<b>Observação</b>
<b>Sprint N1</b>	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 Release N-1). Aplicado FI de 47%, conforme item 7.3 (7 SFP *0,47=3,29 SFP)
<b>Sprint N2</b>	Alterar Beneficiário	Inc	PE	4,6	-
	Beneficiário	Refin	AL	-	Mudança caracterizada como refinamento, pois o arquivo lógico (AL) foi incluído na Sprint N1, da mesma release.
	Incluir Benefício	Alt	PE	2,16	Alteração caracterizada como Projeto de Melhoria (PE desenvolvido e pronto na Release N-1). Aplicado FI de 47%, conforme item 7.3 (4,6 SFP *0,47 = 2,16 SFP)
<b>Sprint N3</b>	Emitir Relatório de Beneficiários	Inc	PE	4,6	-
	Incluir Benefício	Refin	PE	-	Mudança caracterizada como refinamento. Apesar de ter sido desenvolvida na Release N-1, a funcionalidade foi alterada na Sprint N2.
	Incluir Beneficiário	Refin	PE	-	Mudança caracterizada como refinamento. A funcionalidade foi desenvolvida em sprint da mesma release (Sprint N1).
	Alterar Beneficiário	Refin	PE	-	Mudança caracterizada como refinamento. A funcionalidade foi desenvolvida em sprint da mesma release (Sprint N2).

Alterar Benefício	Alt	PE	2,16	Alteração caracterizada como Projeto de Melhoria (PE desenvolvido e pronto na Release N-1). Aplicado FI de 47%, conforme item 7.3 (4,6 SFP *0,47 = 2,16 SFP)
<b>Total SFP da Release N:</b>			28,41 SFP	-

### 11.3.3. 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 22 - Contagem de SFP da Release N para a Baseline da Aplicação**

	Nome da Função	Tipo	SFP	Observação
<b>Contagem SFP da Release N Para a Baseline</b>	Beneficiário	AL	7	Na contagem da <i>Release</i> para a Baseline da aplicação com o método SFP, não devem constar as funcionalidades alteradas, excluídas e refinamentos (ver item 7.4)
	Incluir Beneficiário	PE	4,6	
	Alterar Beneficiário	PE	4,6	
	Emitir Relatório de Beneficiários	PE	4,6	
	<b>Total SFP da Release N</b>			

## 11.4. 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 contabilizados 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:

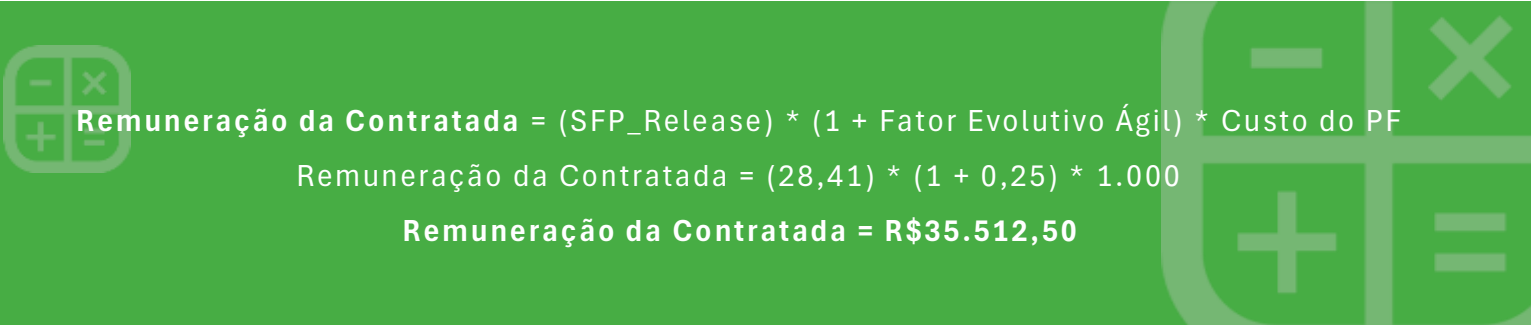
- O Fator Ágil deve ser aplicado em *releases* com quantidade de *sprints* estabelecidas. Por exemplo: para *Releases* com até 4 *sprints*, com *sprints* de até 4 semanas;
- O Fator Ágil não é aplicável para projetos com *releases* excessivamente grandes ou para projetos executados em apenas 1 (uma) *sprint*. Sugere-se que o tamanho mínimo de uma *release* seja de 2 (duas) *sprints* e o máximo de 4 (quatro) *sprints*;

- A aplicação do fator ágil é adequada para processos em que há necessidade de grande volume de reconstruções entre sprints de uma mesma *release*.

### 11.4.1. 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:


$$\begin{aligned} \text{Remuneração da Contratada} &= (\text{SFP\_Release}) * (1 + \text{Fator Evolutivo Ágil}) * \text{Custo do PF} \\ \text{Remuneração da Contratada} &= (28,41) * (1 + 0,25) * 1.000 \\ \text{Remuneração da Contratada} &= \text{R\$35.512,50} \end{aligned}$$

No exemplo acima foi utilizado o Total de SFPs apurados na *Release N* (exemplo descrito no item 11.3.2), 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 APF tradicional.

## 11.5. 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.

## 12. CONTAGEM DE PF EM CENÁRIOS ESPECÍFICOS

Considerando que a métrica de Ponto de Função é independente de tecnologia, os tópicos a seguir apresentam diretrizes para a contagem de PF em diferentes contextos e ambientes tecnológicos, utilizando os métodos APF e SFP.

- Contagem de Projetos de IA (item 12.1)
- Contagem de Soluções de ChatBot (item 12.2)
- Contagem de Painéis Analíticos (item 12.3)
- Contagem de Projetos baseados em Geotecnologia (item 12.4)
- Contagem de Desenvolvimento Multiplataforma (item 0)
- Contagem de Ingestão de dados em Data Lake (item 12.6)

### 12.1. CONTAGEM DE IA (Inteligência Artificial)

Este tópico apresenta diretrizes para contagem de PF em projetos de IA. As orientações expostas a seguir foram elaboradas com base no *Guia de Referência de Contagem de Pontos de Função do SERPRO (2025)*.

Esse item contempla também instruções para a realização de contagem de PF por meio do método SFP (*Simple Function Point* / Ponto de Função Simples), com o propósito de conferir maior agilidade, consistência e previsibilidade ao processo de contagem.

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 do uso de lógica, árvores de decisão e algoritmos de aprendizado de máquina, incluindo aprendizado profundo.

Considerando o seu potencial para transformar, inovar e melhorar significativamente a vida das pessoas, essa tecnologia vem sendo utilizada nas organizações públicas para apoiar a tomada de decisão, além de aumentar a eficácia e a eficiência do poder público.

Em caráter propositivo, este instrumento reconhece que a inteligência artificial possui diversas vertentes de aplicação. Assim, este roteiro foi concebido com caráter evolutivo, podendo ser refinado continuamente à medida que novas técnicas e aplicações de IA se desenvolvem.

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.

### 12.1.1. 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 atento para não contar a mesma funcionalidade mais de uma vez, por exemplo, uma funcionalidade que atua como 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 CEs), potencialização da duplicidade de componentes e impacto no custo final do produto.

Para reduzir riscos de incorreções ou imprecisões na contagem de pontos de função (PF) em projetos que envolvem mais de uma fronteira da aplicação, este roteiro apresenta orientações específicas sobre: (i) o tratamento de componentes utilizados por múltiplas funcionalidades (item 7.18. Componente Reusável) e (ii) a contagem de PF para funcionalidades entregues em diferentes mídias (item 8.1. 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 7.18 Componente Reusável)
- Funcionalidades entregues em mais de uma mídia (item 8.1- Contagem com Múltiplas Mídias).

## 12.1.2. 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 6 - Ciclo de Vida do Desenvolvimento de um projeto de IA (Adaptada SERPRO, 2025)**



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 16 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.

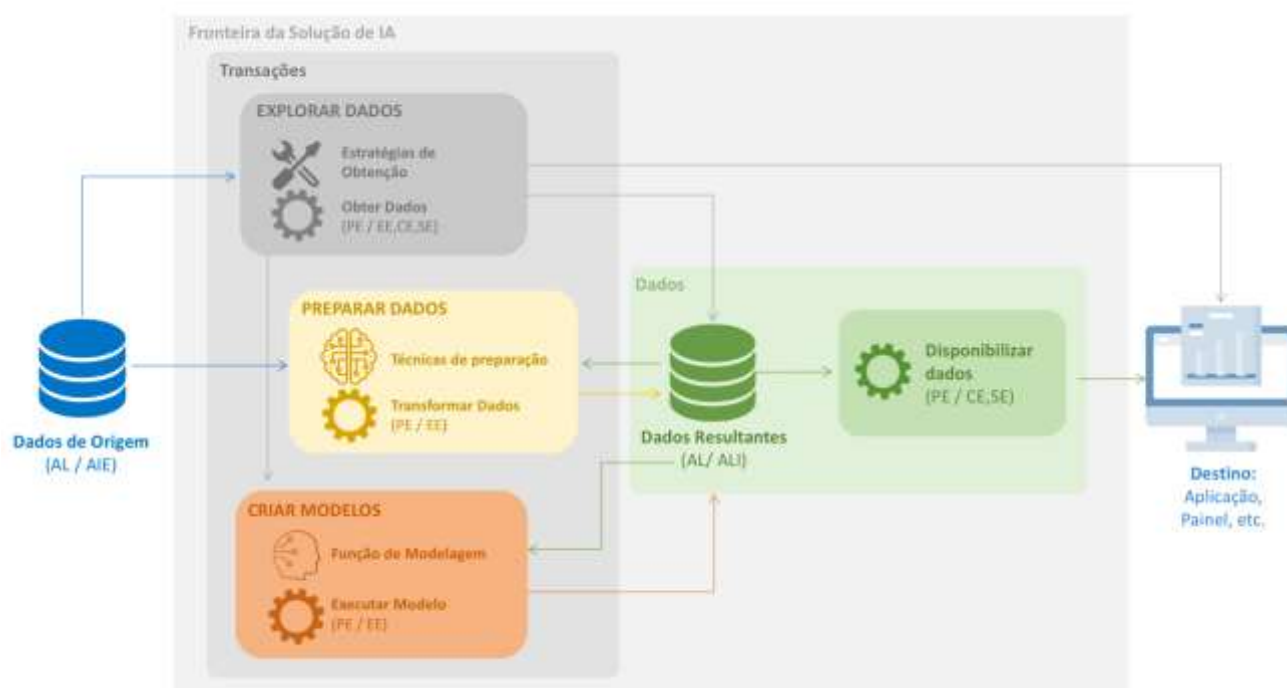
Sistemas ou modelos de IA podem ser operacionalizados seguindo arquiteturas de processamento em lote (batch) ou interativo. Dessa forma, este roteiro foi orientado por essas variações arquiteturais, que estão relacionadas com o fluxo de informação dentro da aplicação, conforme explicado nas próximas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

### 12.1.3. Contagem para Processamento *Batch* de IA

No contexto da IA, o Processamento em Lote (Batch) refere-se 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 7 – Visão simplificada do Processamento em Lote (Batch) de IA (Adaptada SERPRO, 2025)**



A imagem ilustra o fluxo de dados de uma solução de processamento que utiliza IA. O fluxo é 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 são 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.

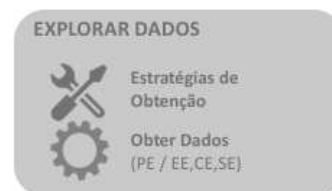
Para efeitos de contagem nos métodos APF tradicional e SFP, temos:

**Tabela 23 - Contagem PF de Dados de Origem**

	<b>Método APF identifica:</b>	<b>Método SFP identifica:</b>	<b>OBS</b>
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 13 deste roteiro.

Para efeito de contagem será considerada apenas a tarefa “**Obter Dados**”, conforme exposto na figura 7. 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.

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 24 - Contagem PF para o macroprocesso Explorar Dados**

	<b>Método APF identifica:</b>	<b>Método SFP identifica:</b>	<b>OBSERVAÇÕES</b>
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, mas 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 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 considerada uma EE, no método APF tradicional;
- b) Será considerado um Processo elementar (PE), no método SFP, com pontuação igual a 4,6 SFP.

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.



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**, mesmo aquelas que transformarem o dado de maneira que seu formato seja necessário para ser aceito em uma função de modelagem analítica.

#### 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 armazene 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 25 - Contagem PF para Dados Resultantes**

	<b>Método APF identifica:</b>	<b>Método SFP identifica:</b>	<b>OBSERVAÇÕES</b>
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 7 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 26 - Contagem PF para Disponibilizar Dados**

	<b>Método APF identifica:</b>	<b>Método SFP identifica:</b>
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

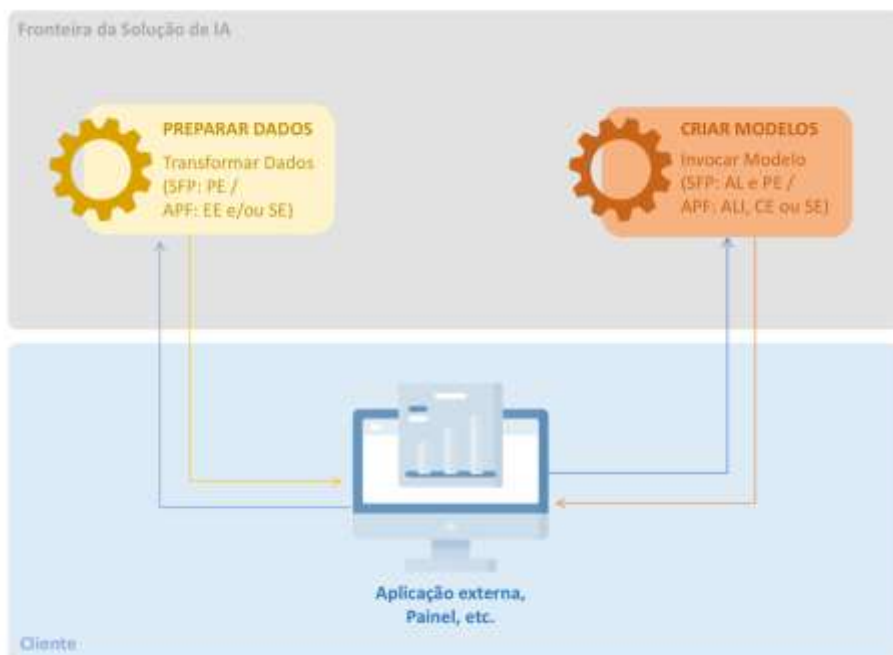
g) DESTINO

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

## 12.1.4. Contagem para a Execução Interativa de IA

O processamento interativo, no contexto da IA, refere-se 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 8 – Visão simplificada do Processamento Interativo de IA (Adaptada SERPRO, 2025)



### 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**”.

Para os métodos APF e SFP, temos:

**Tabela 27 - Contagem PF para Preparar Dados em Execução Interativa de IA**

	<b>Método APF identifica:</b>	<b>Método SFP identifica:</b>
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 citadas na alínea “c)” da seção 12.1.3.

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 28 - Contagem PF para Criar Modelos em Execução Interativa de IA**

	<b>Método APF identifica:</b>	<b>Método SFP identifica:</b>
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

Trata-se de aplicações externas à fronteira do sistema, que podem acionar a solução de IA e, portanto, devem ter contagem própria. Ressalta-se que esse roteiro não leva em conta a tecnologia e o formato das aplicações externas.

### 12.1.5. 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 7.3 deste roteiro.

## 12.2. CONTAGEM DE PF EM CHATBOT

Este item apresenta diretrizes para contagem de PF em soluções de *ChatBot*. As orientações a seguir foram elaboradas com base no *Guia de Referência de Contagem de Pontos de Função do SERPRO (2025)*.

Essa seção contempla também instruções para a realização de contagem de PF por meio do método SFP (*Simple Function Point / Ponto de Função Simples*), com o propósito de conferir maior agilidade, consistência e previsibilidade ao processo de contagem.

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 autoexplica 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, Twitter, 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.

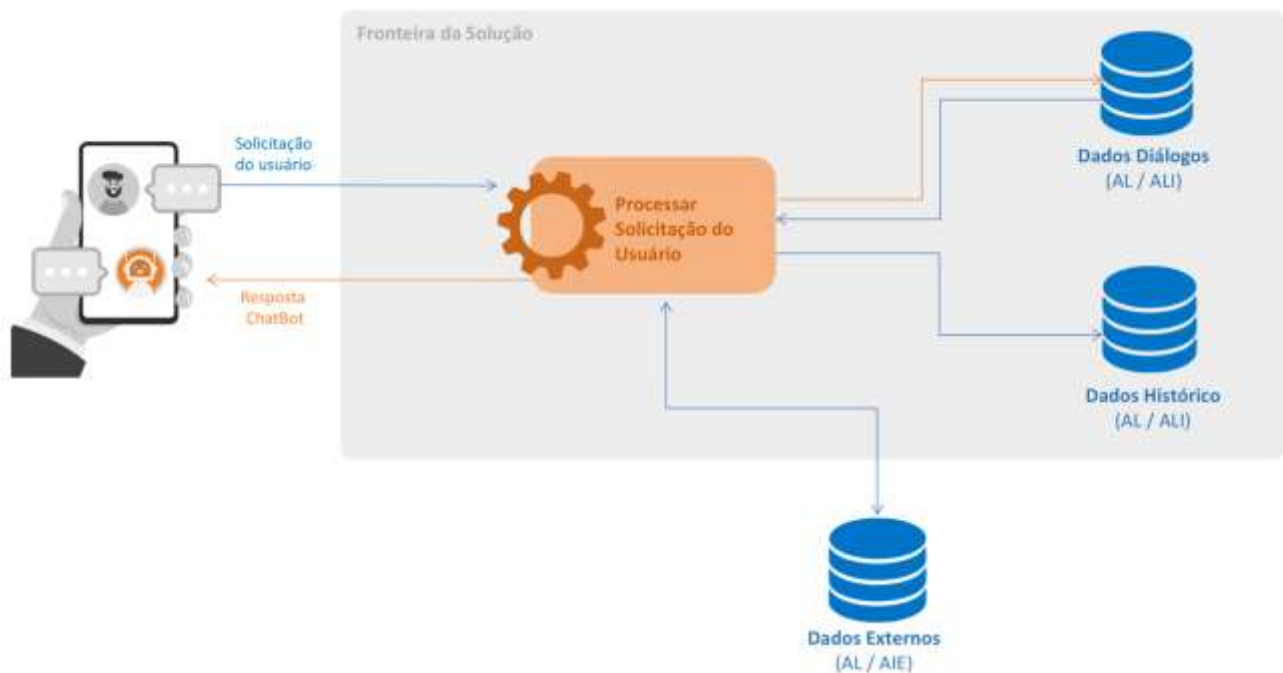
Convém destacar que a métrica PF considera a visão do usuário e é independente da metodologia e tecnologia utilizadas.

### 12.2.1. 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 9 - Visão simplificada do funcionamento de um ChatBot (adaptada SERPRO, 2025)**



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*.

### 12.2.2. Escopo da contagem PF para *ChatBot*

O escopo da contagem abrange geralmente o Projeto de Desenvolvimento e/ou de Melhoria de aplicações que se integrarão 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 7.18- Componente Reusável) e sobre a contagem de PF de funcionalidades entregues em mais de uma mídia (item 8.1- Contagem com Múltiplas Mídias).

### 12.2.3. Contagem de funções de dados para *Chatbot*

Considerando a figura 9, 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 à 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 29 - Arquivos lógicos de um Bot**

<b>Funções de Dados</b>	<b>Método APF identifica:</b>	<b>Método SFP identifica:</b>	<b>Tipos de dados básicos</b>
Dados de Diálogo: dados referentes às configurações dos nós de diálogo do <i>chatbot</i> ;	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 <i>chatbot</i> ;	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 7 deste roteiro.

#### 12.2.4. Contagem de funções transacionais para *Chatbot*

Considerando ainda a figura 9, 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, representada por funções transacionais:

- 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 Improriedades;
- Exibir mensagem Transbordo;
- Exibir mensagem Sugestões.

Esse rol de funções transacionais é uma referência não exaustiva e pode ser ajustado conforme as particularidades de cada projeto. Para cada novo tipo de nó de diálogo identificado para o *bot* do negócio, deve ser computado um processo elementar adicional, desde que represente comportamento funcional distinto.

### 12.2.5. Manutenção em soluções de *Chatbot*

Convém destacar novamente que a figura 9 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 7 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.

### 12.2.6. 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 16 deste roteiro.

## 12.3. CONTAGEM DE PF EM 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, este roteiro de métricas contém orientações sobre como tratar componentes que são utilizados por mais de uma funcionalidade (item 7.18- Componente Reusável) e sobre a contagem de PF de funcionalidades entregues em mais de uma mídia (item 8.1- 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.

### 12.3.1. Criação e carga de tabelas

Os principais elementos de um painel analítico compreendem gráficos, tabelas, indicadores-chave de desempenho (KPIs) e filtros interativos.

No que se refere às tabelas, é importante destacar que nem toda tabela deve ser considerada um arquivo lógico. Em outras palavras, na contagem de Pontos de Função (PF), não se deve contabilizar um arquivo lógico para cada tabela existente.

Um grupo lógico tem caráter orgânico, funcional, podendo ser implementado por uma ou mais tabelas e mantido por vários processos elementares diferentes. As 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.

No que diz respeito às cargas de dados, cada processo de carga deve ser considerado um processo elementar. Assim, cada carga de dados é contada 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 e SFP:

**Tabela 30 - Funções para criação de carga de tabelas de painéis analíticos**

	<b>APF</b>	<b>SFP</b>	<b>OBSERVAÇÕES</b>
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 ( <i>Delta</i> )	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.

### 12.3.2. 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 lidos de outros sistemas, devendo ser consideradas na contagem de pontos de função.

**Tabela 31 - 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.

### 12.3.3. 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 apresentados nos painéis contêm dados gerenciais resultantes de cálculos específicos e requisitos de negócio independentes.

Nesse sentido, um painel pode apresentar vários gráficos, indicadores e tabelas independentes; e a geração de cada gráfico e tabela dinâmica deve ser contada como um processo elementar distinto.

**Tabela 32 - 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 apresentados 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.</p> <p>Os Tipos de Dados (TDs) são os dados apresentados nos gráficos ou tabelas e os filtros utilizados para geração do gráfico ou tabelas.</p>

## 12.4. CONTAGEM PF EM 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 de funções de dados e transacionais, não tratados diretamente pelo CPM 4.3.1 e SPM 2.1.

### 12.4.1. Medição de camadas georreferenciadas

Esta seção se refere a camadas de georreferenciamento, tais como “camadas *overlayer*”, “*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.

### 12.4.2. 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 8.1 deste roteiro de métricas.

Considere a necessidade de uma consulta ‘C’ realizando a leitura em um grupo de dados georreferenciados. 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 dois ou mais processos elementares distintos, em aderência à 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.

### 12.4.3. Apresentação simultânea de vários elementos visuais

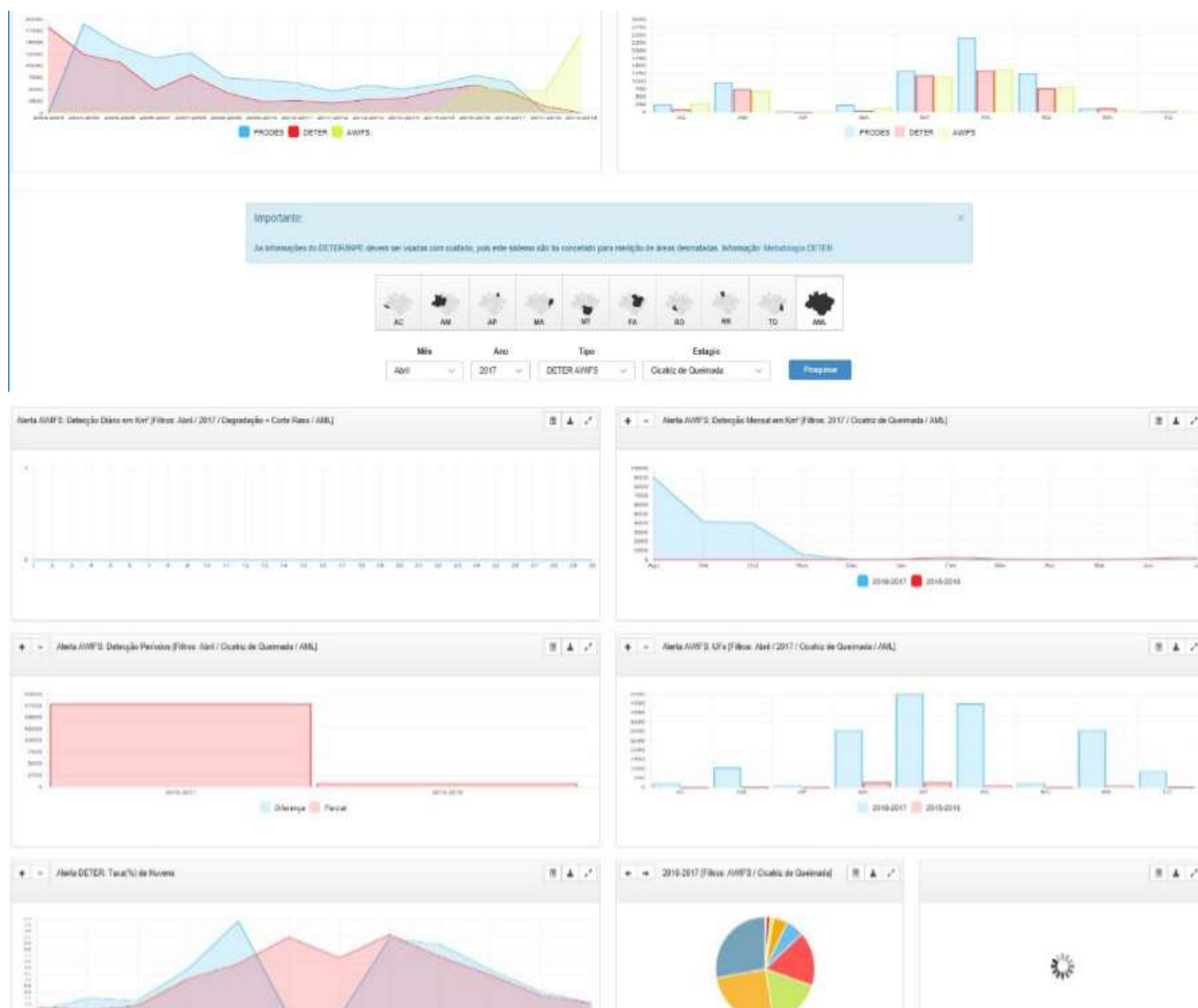
Esta seção trata cenários em que vários “elementos visuais georreferenciados” são exibidos simultaneamente, em uma mesma tela.

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 10 - 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).

#### 12.4.4. 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 de 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 33 - Contagem de funções transacionais e de dados de um sistema de Geo**

	<b>APF</b>	<b>SFP</b>	<b>OBSERVAÇÕES</b>
Cada camada georreferenciada	ALI ou AIE	AL	Ex: “camadas <i>overlay</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 <i>Multiple Instance</i> , descrita no item 8.1 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 APF, estas consultas gerenciais normalmente são contadas como SE.
Cada elemento visual georreferenciado	SE	PE	Exemplo no item 12.4.3.
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

## 12.5. CONTAGEM DE PF EM DESENVOLVIMENTO MULTIPLATAFORMA

O desenvolvimento multiplataforma tem se expandido significativamente, impulsionado por frameworks que viabilizam a construção de aplicações para diferentes plataformas de produção. Embora possa haver compartilhamento de código, as atividades de testes, homologação e implantação tendem a ocorrer de forma específica para cada plataforma, podendo demandar perfis profissionais distintos.

Considerando a abordagem de *multiple instance* e reconhecendo a plataforma de entrega como uma característica de diferenciação na identificação da unicidade da função transacional, este roteiro sugere a utilização de fatores de impacto (FI) diferenciados para o desenvolvimento multiplataforma. A distinção fundamenta-se na forma como os aplicativos são desenvolvidos para as plataformas de produção:

- **O desenvolvimento nativo** consiste na criação de aplicativos específicos para cada plataforma de produção (por exemplo: *iOS* e *Android*). Cada aplicativo é desenvolvido utilizando as linguagens e ferramentas oficiais de cada plataforma (por exemplo: *Swift* ou *Objective-C* para *iOS* e *Kotlin* ou *Java* para *Android*).
- **O desenvolvimento híbrido** consiste na criação de um único aplicativo destinado a múltiplas plataformas (como *iOS* e *Android*). Normalmente utiliza tecnologias web (*HTML*, *CSS* e *JavaScript*) e *frameworks* específicos. O aplicativo é executado dentro de um contêiner nativo e posteriormente disponibilizado para cada ambiente, resultando em aplicativos derivados de uma mesma implementação lógica.

Além do tipo de plataforma de desenvolvimento, esse roteiro considera também fatores de impacto (FI) diferenciados pelo tipo da plataforma (Base e Adicional) de instalação do aplicativo:

- **A plataforma base** é a plataforma de referência escolhida, podendo ser uma na qual o aplicativo será disponibilizado; para esta, aplicam-se os fatores de impacto já definidos neste roteiro de métricas;
- **A plataforma adicional** é qualquer outra plataforma, distinta da Base, à qual também o aplicativo será disponibilizado, aplicando-se fatores de impacto específicos que contemplem, na contagem, o esforço

empreendido nas atividades dos processos de Integração, Verificação, Transição e Validação, conforme disposto no item 10.1 deste roteiro.

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

## 12.5.1. Plataforma de desenvolvimento híbrida

### 12.5.1.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 31%, que corresponde ao esforço empreendido em atividades dos processos de Integração, Verificação, Transição e Validação, conforme memória de cálculo presente na tabela 19 (item 10.1) 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).

### 12.5.1.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.

## 12.5.2. Plataforma de desenvolvimento nativa

### 12.5.2.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.

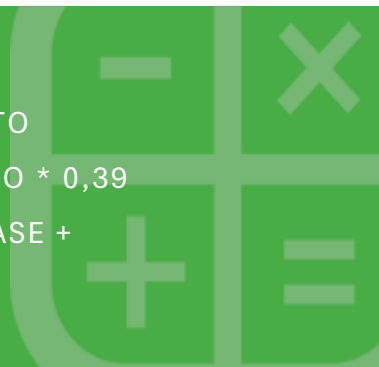
Para a fórmula de contagem da plataforma de desenvolvimento nativa adicional, utilizou-se um fator de impacto de 39%, que corresponde ao esforço empreendido nos processos de Implementação, Integração, Verificação, Transição e Validação, conforme memória de cálculo presente na tabela 19 (item 10.1) deste roteiro.

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).



$$\begin{aligned} \text{PF\_DESENV\_NATIVO\_PLAT\_BASE} &= \text{PF\_DESENVOLVIMENTO} \\ \text{PF\_DESENV\_NATIVO\_PLAT\_ADICIONAL} &= \text{PF\_DESENVOLVIMENTO} * 0,39 \\ \text{PF\_DESENV\_NATIVO\_TOTAL} &= \text{PF\_DESENV\_PLAT\_NATIVO\_BASE} + \\ &\quad \text{PF\_DESENV\_NATIVO\_PLAT\_ADICIONAL} \end{aligned}$$



### 12.5.2.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.

## 12.6. CONTAGEM DE INGESTÃO DE DADOS PARA *DATA LAKE*

Este tópico tem por finalidade apresentar diretrizes para contagem de pontos de função (PF) nos processos de ingestão de dados em Data Lakes. As orientações aqui definidas foram elaboradas com base no estudo e análise do Guia de Referência de Contagem de Pontos de Função do SERPRO (2025).

Essa seção contempla também instruções para a realização de contagem por meio do método SFP (*Simple Function Point / Ponto de Função Simples*), com o propósito de conferir maior agilidade, consistência e previsibilidade ao processo de contagem de ingestão de dados para *Data Lake*.

Conforme já exposto neste documento, o método SFP constitui uma abordagem de medição funcional compatível e alinhada ao IFPUG APF - especificado no *Counting Practices Manual (CPM) v.4.3.1* -, mantendo convergência de terminologia e definições.

O *Data Lake* é um repositório destinado ao armazenamento centralizado de grandes volumes de dados, em diferentes formatos e provenientes de múltiplas fontes. Esses dados são armazenados em seu estado bruto, sem prévio processamento ou análise, permitindo que a organização mantenha informações potencialmente relevantes para uso estratégico e suporte à tomada de decisões.

Os *Data Lakes* são estruturados para viabilizar o consumo de dados — processo que abrange a coleta, a importação e o processamento das informações para posterior armazenamento e utilização. A etapa responsável pela obtenção e transferência dos dados dos sistemas ou serviços de origem para o *Data Lake* é denominada ingestão de dados.

A ingestão de dados consiste em um serviço que envolve a extração de informações na origem e sua carga na plataforma, fazendo uso de ferramentas de *Big Data* e de processos internos definidos pela organização.

Os principais tipos de ingestão de dados são:

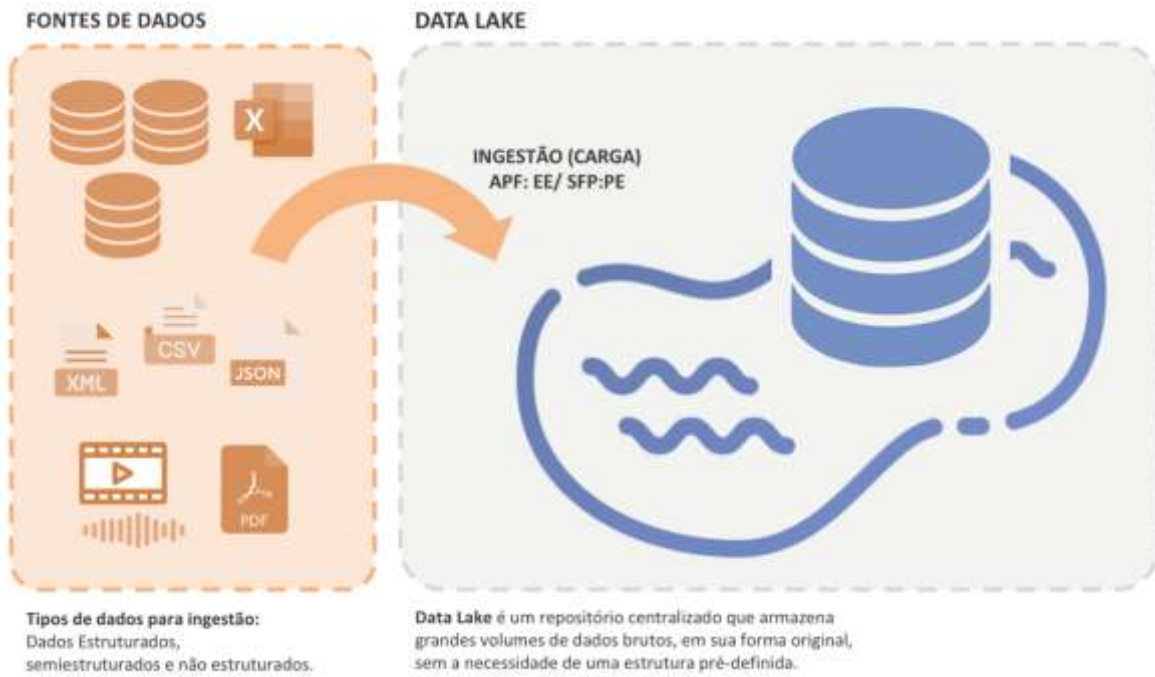
- **Ingestão Full:** realiza a carga completa dos dados, sem aplicação de transformações, disponibilizando no *Data Lake* as informações exatamente como se encontram na fonte.
- **Ingestão incremental:** Carrega apenas os dados novos ou modificados desde a última carga, otimizando volume e desempenho.

Vale observar que as cargas de dados que envolvem tratamento, aplicação de filtros ou transformações devem ser consideradas no processo de ETL do Data Warehouse, e não no processo de ingestão para o *Data Lake*.

Os tipos de dados considerados na ingestão são:

- **Dados estruturados:** Organizados em modelos formais (ex.: tabelas, bancos relacionais).
- **Dados semiestruturados:** Possuem alguma estrutura, mas não rígida (ex.: *JSON*, *XML*, *logs*).
- **Dados não estruturados:** Não seguem modelo pré-definido (ex.: textos livres, imagens, vídeos, PDFs).

**Figura 11 - Visão simplificada do processo de ingestão de dados no *Data Lake***



## 12.6.1. Contagem de Ingestão Full

A tabela a seguir apresenta cenários de ingestão Full de dados estruturados, semiestruturados e não estruturados no *Data Lake*.

**Tabela 34 - Contagem de PF para Ingestão Full no *Data Lake***

		O Método APF identifica	O Método SFP identifica	OBSERVAÇÕES
<b>Ingestão de dados de File Adabas – campo fixo</b>	Este processo consiste na exportação de dados do File Adabas e a importação de dados para o <i>Data Lake</i> .	1 CE para Extração de Dados File XXX 1 EE para a Carga de Dados File XXX	1 PE para Extração de Dados File XXX 1 PE para a Carga de Dados File XXX	ATENÇÃO! Observe que o <i>Data Lake</i> armazena dados brutos, assim o Arquivo Físico será tratado apenas como <u>Arquivo Lógico (AL) referenciado</u> nos processos elementares contados.  Nesse sentido, <u>não será contado PF para funções de dados no <i>Data Lake</i></u> , considerando que nas demandas de ingestão de dados não há definição de entidade lógica de dados para o <i>Data Lake</i> .
<b>Ingestão de Dados de File Adabas – campo multivalorado</b>	Este processo consiste na exportação de dados do File Adabas com algoritmo de transformação de dados para tratar o campo multivalorado e a importação de dados para o <i>Data Lake</i> .	1 SE para Extração de Dados com Transformação File XXX 1 EE para Carga de Dados File XXX	1 PE para Extração de Dados com Transformação File XXX 1 PE para Carga de Dados File XXX	ATENÇÃO! O <i>Data Lake</i> armazena dados brutos. Assim, o Arquivo Físico será tratado apenas <u>como Arquivo Lógico (AL) referenciado</u> nos processos elementares contados.  Nesse sentido, <u>não será contado PF para funções de dados no <i>Data Lake</i></u> , considerando que nas demandas de ingestão de dados não há definição de entidade lógica de dados para o <i>Data Lake</i> .
<b>Ingestão de Dados de Tabelas de SGBD Relacional</b>	Este processo consiste na importação de dados de tabelas de Banco de Dados Relacionais para o <i>Data Lake</i> .	1 EE para Carga de Dados Tabela XXX	1 PE para Carga de Dados Tabela XXX	ATENÇÃO! Observe que o <i>Data Lake</i> armazena dados brutos, assim o Arquivo Físico será tratado apenas como <u>Arquivo Lógico (AL) referenciado</u> nos processos elementares contados.  Nesse sentido, <u>não será contado PF para funções de dados no <i>Data Lake</i></u> , considerando que nas demandas de ingestão de dados não há definição de entidade lógica de dados para o <i>Data Lake</i> .
<b>Ingestão de Dados de Estrelas de DW</b>	Este processo consiste na importação de dados de estrelas de DW para o <i>Data Lake</i> .	1 EE para a carga de Dados Estrela XXX	1 PE para a carga de Dados Estrela XXX	ATENÇÃO! O <i>Data Lake</i> armazena dados brutos. Assim, o Arquivo Físico será tratado apenas <u>como Arquivo Lógico (AL) referenciado</u> nos processos elementares contados.

	A estrela é composta por uma Tabela Fato e as Dimensões relacionadas a Tabela Fato em questão.			Nesse sentido, <u>não será contado PE para funções de dados no Data Lake</u> , considerando que nas demandas de ingestão de dados não há definição de entidade lógica de dados para o Data Lake.
<b>Ingestão de Dados de Arquivo de Dados SEMI ESTRUTURADOS</b>	Consiste na exportação de dados disponibilizados por serviços e importação de dados de arquivos semiestruturados (planilhas, xml, txt) para o <i>Data Lake</i> .	1 SE para a Extração de Dados com Transformação Arquivo XXX  1 EE para a Carga de Dados File XXX	1 PE para a Extração de Dados com Transformação Arquivo XXX  1 PE para a Carga de Dados File XXX	Cabe ressaltar que, normalmente, a exportação de dados exige a aplicação de algoritmos de derivação e transformação, a fim de adequar as informações ao formato específico necessário para a carga no Data Lake  E novamente, cabe lembrar que o Data Lake armazena dados brutos, assim o Arquivo Físico será tratado como Arquivo Lógico referenciado nos processos elementares contados. Dessa forma, <u>não será contado PE para funções de dados no Data Lake</u> , considerando que nas demandas de ingestão de dados não há definição de entidade lógica de dados para o Data Lake.
<b>Ingestão de Dados de Arquivo Json</b>	Consiste na ingestão de dados semi-estruturados de arquivos Json para o <i>Data Lake</i>	1 ALI para Arquivo de Dados XXX (Tabela Parquet)  1 SE para Realizar Tratamento de Dados do Arquivo de Dados XXX  1 EE para Realizar Carga de Dados do Arquivo Dados XXX	1 AL para Arquivo de Dados XXX (Tabela Parquet)  1 PE para Realizar Tratamento de Dados do Arquivo de Dados XXX  1 PE para Realizar Carga de Dados do Arquivo Dados XXX	Em demandas de Ingestão de Dados de Arquivos Json pode ser necessária a definição de estrutura e a criação de uma Tabela Parquet no Data Lake para ser carregada.  Esta tabela será mantida com os dados do Arquivo Json.
<b>Ingestão de Dados de Arquivo de Dados NÃO ESTRUTURADOS</b>	Consiste na ingestão de dados não estruturados, tais como imagens, vídeos, áudios, Pdf para o <i>Data Lake</i>	1 ALI para o Arquivo de Dados XXX  1 SE para Realizar Tratamento de Dados Arquivo XXX  1 EE para Realizar Carga de Dados Arquivo XXX	1 AL para o Arquivo de Dados XXX  1 PE para Realizar Tratamento de Dados Arquivo XXX  1 PE para Realizar Carga de Dados Arquivo XXX	Esta categoria de demandas requer o desenvolvimento de aplicativos.

## 12.6.2. Contagem de Ingestão Incremental

A ingestão de dados incremental consiste na implementação de um processo destinado à atualização periódica dos dados previamente carregados no *Data Lake*, considerando apenas os registros novos ou alterados desde a última carga. Os processos de ingestão incremental são independentes da carga *full* e devem ser contabilizados como processos elementares distintos.

De modo geral, as ingestões incrementais são solicitadas com periodicidade diária, semanal ou mensal; mas em alguns cenários pode ser requerida uma ingestão incremental em tempo real.

Quando solicitada, a ingestão incremental em tempo real normalmente demanda a realização de testes não-funcionais de desempenho. Para esses casos, recomenda-se a mensuração dos serviços de teste não-funcionais por meio da métrica HST, descrita na seção 16 deste roteiro.

Cumprir observar que na ausência de carga incremental — quando a atualização do *Data Lake* ocorre exclusivamente por meio de cargas *full* executadas de forma recorrente, na periodicidade definida —, desenvolve-se e contabiliza-se apenas o processo correspondente à carga *full*. Nessa situação específica, não há processo adicional de ingestão incremental, tampouco contagem de pontos de função aplicável a esse tipo de carga.

## 12.6.3. Ingestão de dados com Utilitários

A ingestão de dados com utilitário no *Data Lake* refere-se ao processo de utilizar ferramentas específicas (os "utilitários") para coletar, transferir e carregar grandes volumes de dados de diversas fontes para o *Data Lake*.

Esse roteiro sugere que a contagem da ingestão de dados com utilitário seja contada como a configuração de componente para ingestão de dados no *Data Lake*. Para isso, vamos utilizar a fórmula a seguir, considerando apenas as tabelas configuradas no aplicativo para ingestão:

- Para o método APF: com a aplicação de um fator de impacto de 13% sobre a contagem de uma EE de baixa complexidade (3 PF), ou seja 0,39 PF;
- Para o método SFP: com a aplicação de um fator de impacto de 13% sobre a contagem de um processo elementar (4,6 SFP), ou seja 0,59 SFP.



$PF\_INGESTÃO\_UTILITÁRIOS = 0,39 PF \times QTD\_TABELAS\_CONFIGURADAS$   
 $SFP\_INGESTÃO\_UTILITÁRIOS = 0,59 SFP \times QTD\_TABELAS\_CONFIGURADAS$

#### 12.6.4. Ingestão de dados com Testes não-funcionais

Na contagem de Pontos de Função de demandas de ingestão de dados, deve-se observar se existe a necessidade da realização de testes não funcionais para as extrações ou carga de dados. Em geral, os testes não funcionais de performance são necessários nas situações em que há grande volume de dados a serem extraídos/carregados.

Este roteiro recomenda a utilização da métrica HST (item 16) para aferição dos serviços de testes não-funcionais.

#### 12.6.5. Ingestão de dados com Criação de Arquivos no *Data Lake*

Em alguns casos, torna-se necessária a criação de um agrupamento de dados específico *no Data Lake*. Este agrupamento de dados será carregado nos processos de ingestão de dados. Nestes casos, o agrupamento de dados deve ser contado como arquivo lógico (APF: ALI/ SFP: AL) do *Data Lake*.

Além disso, é comum a necessidade da criação de dados de controle, visando garantir a qualidade da ingestão de dados. Nestes casos, o agrupamento de dados de controle deve ser contado como arquivo lógico (APF: ALI/ SFP: AL) do *Data Lake*.

Caso os Arquivos Criados tenham subgrupos de dados, na contagem com o método APF estes subgrupos devem ser contados como Registros Lógicos de dados do ALI.

Para os dois métodos, cada processo de carga de dados independente deve ser contado como um processo elementar (APF:EE/ SFP:PE).

### 12.6.6. Ingestão de dados com Controle de Qualidade

Em determinadas demandas de ingestão de dados, o cliente poderá requerer a implementação de mecanismos específicos de controle de qualidade. Nesses casos, os dados carregados no *Data Lake* devem ser submetidos a processos de validação, mediante comparação com os dados existentes nas bases de dados do sistema de origem.

Para esses cenários, aplicam-se as seguintes diretrizes:

- **Contagem de funções transacionais:** deve ser contabilizada uma função transacional adicional (APF: 1 EE / SFP: 1 PE) para cada processo de verificação da qualidade e correção de inconsistências executado.
- **Contagem das funções de dados:** os dados extraídos do sistema de origem e utilizados nos procedimentos de validação e correção devem ser contabilizados como arquivos lógicos (APF: AIE / SFP: AL).

Cumprido destacar que os processos de verificação e correção de inconsistências possuem arquivos referenciados distintos, uma vez que fazem uso simultâneo do arquivo lógico do sistema de origem e dos arquivos lógicos de dados e de controle mantidos no *Data Lake*.

## 13. CONTAGEM DE PF EM PROJETOS DE DW

Esta seção apresenta diretrizes para a contagem de projetos de desenvolvimento e manutenção de *Data Warehouse* (DW), contemplando os componentes e requisitos de sua arquitetura e das ferramentas associadas. Abrange não apenas o desenvolvimento, mas também a estimativa de tamanho, projetos de melhoria e atividades de manutenção em DW que, no contexto de contratações, devem ser dimensionadas por meio de uma métrica de *software*.

O conteúdo foi desenvolvido a partir da primeira versão do Guia de Contagem de Pontos de Função do SISP para Projetos *Data Warehouse*, que se fundamenta:

- 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 objetivo de trazer agilidade e previsibilidade para o processo de contagem de pontos de função para Projetos de DW. Como já apresentado neste documento, o método SFP é um método de medição funcional compatível e consistente com o IFPUG APF, descrito no CPM 4.3.1, em terminologia e definições.

### 13.1. OBJETIVO

Este roteiro apresenta diretrizes e regras de contagem de projetos de DW utilizando os métodos APF e SFP, com o objetivo 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 projetos 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, a geração de consultas e relatórios no OLAP (*On-line Analytical Processing*), além da especificidade de profissionais de mercado que atendem essas demandas.

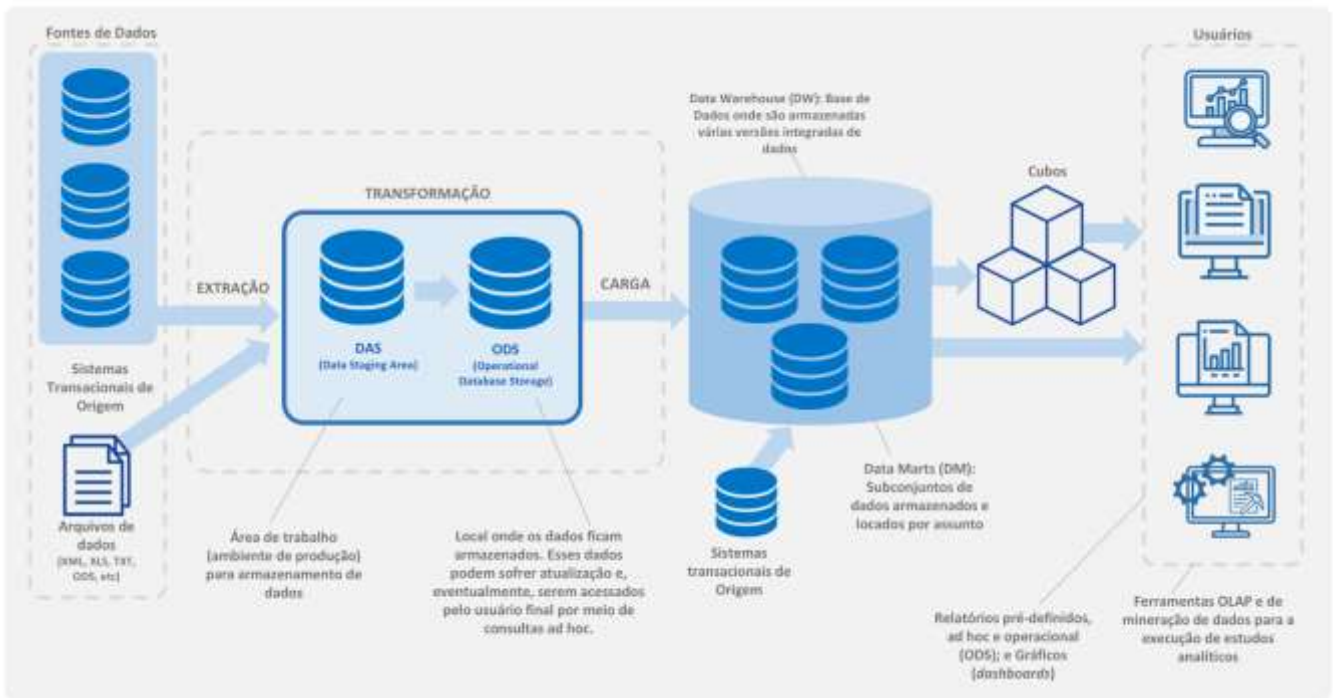
## 13.2. MODELO DE ARQUITETURA DE REFERÊNCIA DE DATA

### 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 maior volume de discussões 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 12 - Modelo de arquitetura de referência de projeto de DW / DM**



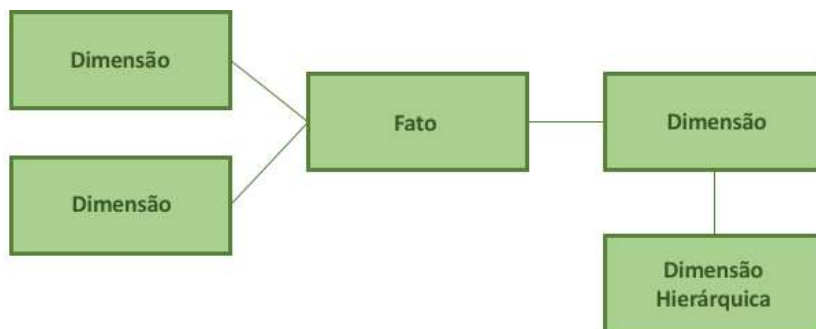
As camadas e componentes da arquitetura de referência de projetos DW/DM são:

- **(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 (DSA)*, *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 13.7.3 (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 13 - 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 localizados 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.

### 13.3. 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, 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 6.1 (Processo de Contagem do Método SFP) deste roteiro; e
- As orientações para realizar a contagem estimada com o método SFP estão descritas no item 9.2 (Estimativa de Tamanho com SFP) deste roteiro.

Convém lembrar que o método SFP tem correspondência de conceitos, tipos de medição, fórmulas de cálculo do método APF tradicional, o que leva a identificação dos mesmos componentes funcionais (processos elementares e arquivos lógicos).

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 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 35 - 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 36 - Resumo da contagem estimada de projeto de DW, utilizando o método SFP**

Componentes funcionais básicos do IFPUG SFP	Contribuição Funcional	Contagem com SFP no DW
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 dashboard solicitado
		1 PE para cada relatório solicitado (não ad hoc)
		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.

## 13.4. 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 13.5;
- a contagem de funções de dados: descrita no subitem 13.6;
- a contagem de funções transacionais: descrita no subitem 13.7.

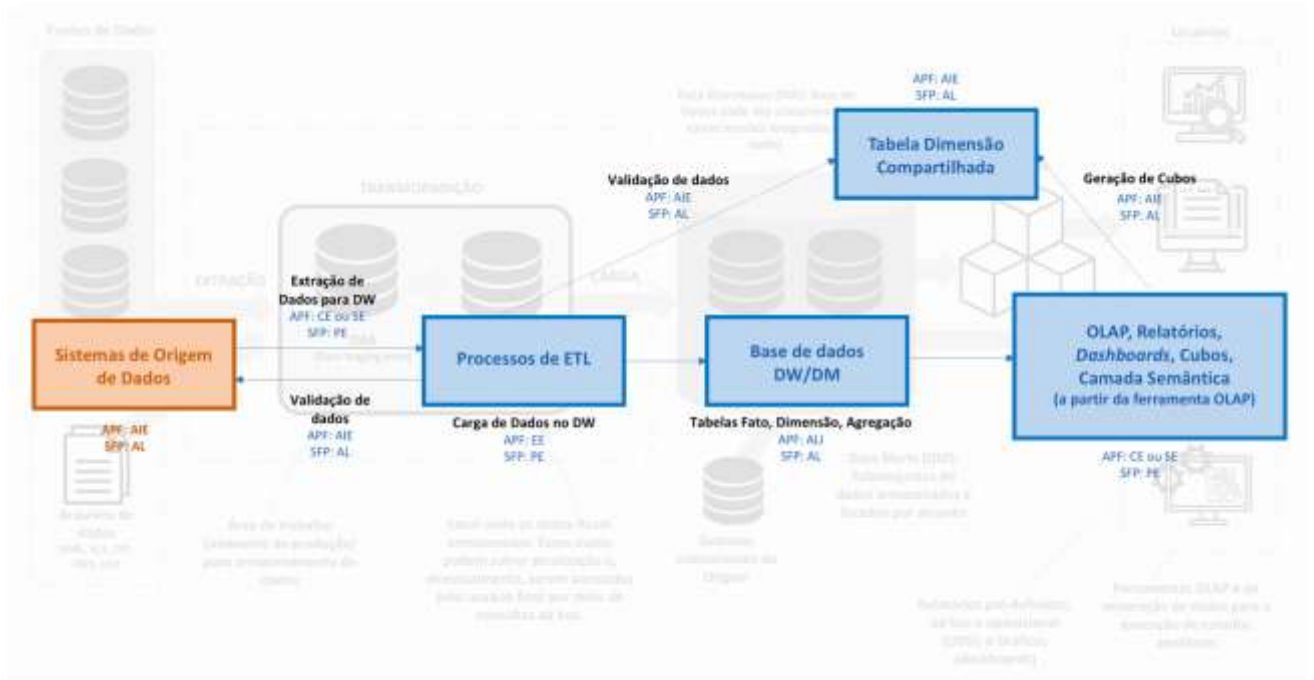
Destaca-se novamente que os dois métodos (APF e SFP) levam a identificação das mesmas funções de dados e das mesmas funções transacionais, entretanto no método SFP não é feita a diferenciação entre os arquivos lógicos internos (ALI, consultados e que armazenam dados) e os arquivos de interface externa (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 6.1 deste roteiro:

**Tabela 37 - Diferença na identificação das funções de dados entre os métodos SFP e APF**

	<b>O método SFP identifica:</b>	<b>O método APF identifica:</b>
<b>Funções de Dados</b>	Arquivo Lógico (AL)	Arquivo Lógico Interno (ALI)
		Arquivo de Interface Externa (AIE)
<b>Funções transacionais</b>	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 13.2:

**Figura 14 - Visão geral da contagem de PF com os métodos APF e SFP em um projeto de DW/DM**



### 13.5. ESCOPO DA CONTAGEM E FRONTEIRA DA APLICAÇÃO

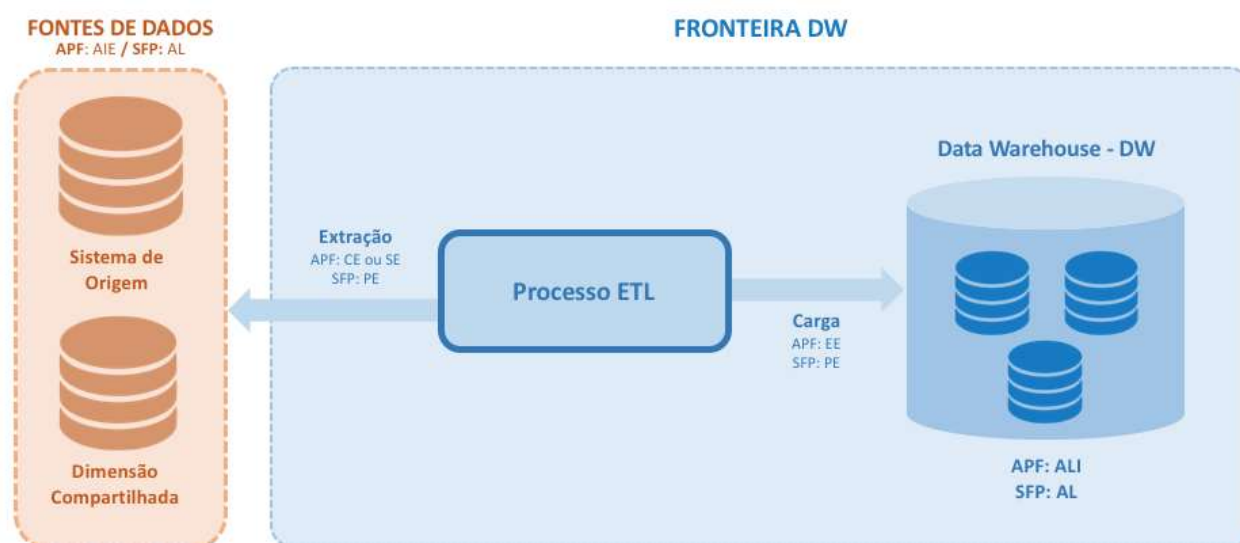
O escopo da contagem define quais são as funcionalidades objeto de determinada contagem. O escopo da contagem é determinado pelo propósito da contagem e identifica os sistemas, as aplicações ou seus componentes que serão dimensionados. Um escopo de contagem pode conter mais de uma aplicação.

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 dessas fronteiras.

Figura 15 - Visão da Fronteira DW



## 13.6. CONTAGEM DE FUNÇÕES DE DADOS

Os subitens a seguir apoiam a identificação das funções de dados em um projeto de DW, utilizando os métodos APF e SFP.

### 13.6.1. 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 38 - Entidades reconhecidas em um modelo de dados multidimensional

		O método SFP identifica:	O método APF identifica:
<b>Tabelas Fato</b>	Mantida por 1 ou mais processos ETL	1 Arquivo Lógico (AL)	1 Arquivo Lógico Interno (ALI)
<b>Tabelas Dimensão</b>	Mantida por 1 ou mais processos ETL	1 Arquivo Lógico (AL)	1 Arquivo Lógico Interno (ALI)
<b>Tabelas Dimensão Estática</b>	Se assemelha ao conceito de tabelas <i>Code Data</i> do	não é contabilizada	não é contabilizada

CPM, armazenando dados estáticos

**Tabelas Dimensão Hierárquica**

Ocorrem quando algumas tabelas Dimensão não estão ligadas diretamente na tabela Fato. São considerados níveis hierárquicos de uma outra tabela Dimensão.

\*Na Figura 16, o “Histórico Estado Civil” é um exemplo de dimensão hierárquica.

Não é contabilizada, visto que para o método SFP, na contagem de arquivos lógicos, não é necessária a identificação de DERs e RLRs.

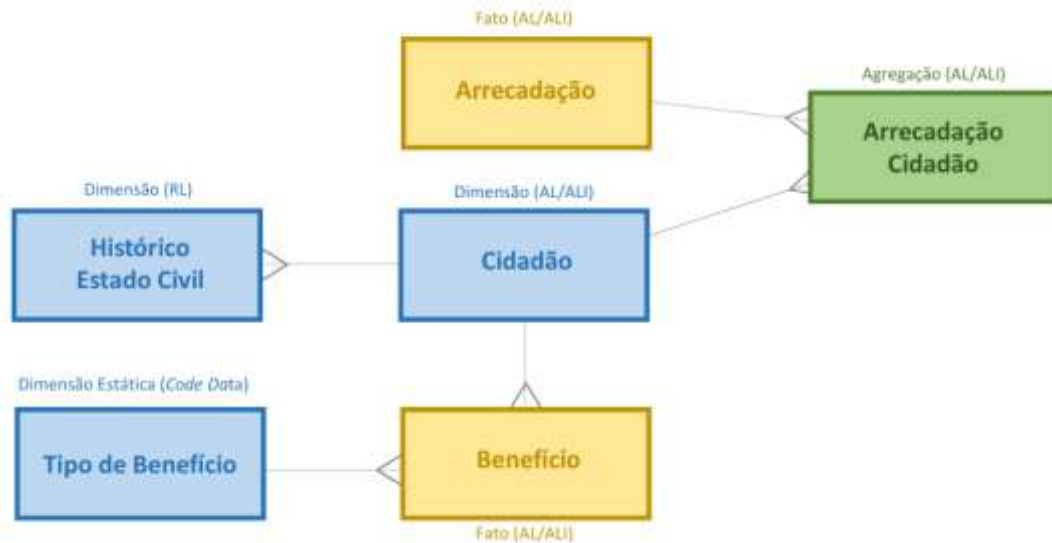
Contar 1 Registro Lógico para cada Dimensão Hierárquica ligada.

Caso não existam níveis hierárquicos ou subgrupos de dados dentro da Dimensão, deve-se considerar apenas 1 Registro Lógico.

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 16 - Exemplo de modelo de dados multidimensional – Esquema Estrela**



As tabelas de agregação podem ser contadas nos seguintes casos:

- (a) Quando existir o cruzamento de dados de dois ou mais arquivos lógicos do DW para gerar um novo Grupo de Informação, reconhecido pelo usuário. A figura acima mostra a agregação da tabela Fato “Arrecadação” com a tabela Dimensão “Cidadão”, resultando na tabela Agregação “Arrecadação Cidadão” conforme requisito do usuário e, nesse caso, deve ser contada como um arquivo lógico. Uma estrutura de Agregação pode acontecer, também, entre tabelas Dimensão;
- (b) Quando existir a necessidade de se guardar os dados no DM cuja data de armazenamento no DW já está expirada. Contudo, uma tabela de Agregação ou um novo arquivo gravado para armazenar resultado de sumarizações, médias, máximas e percentagens de tabelas do DW, para atender requisitos não funcionais de desempenho, por exemplo, não é considerado um arquivo lógico. Esses casos são tratados no item 13.9 (Demandas Típicas em Projetos de Data Warehouse), mais especificamente no item 13.9.6, de Criação de Métricas;
- (c) Caso seja requisitado pelo usuário que os dados dos sistemas de origem do DW sejam disponibilizados na área de armazenamento temporária do DW chamada ODS ( *Operational Data Store*) e esses dados não representam uma cópia dos dados do sistema de origem, mas são dados transformados e disponibilizados para consulta pelo usuário, então esses dados serão contados como arquivo lógico e as cargas desses dados serão contadas como um processo elementar.

**Tabela 39 - Resumo da contagem de tabelas de agregação para os métodos SFP e APF**

		<b>O método SFP identifica:</b>	<b>O método APF identifica:</b>
<b>Tabelas de agregação</b>	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

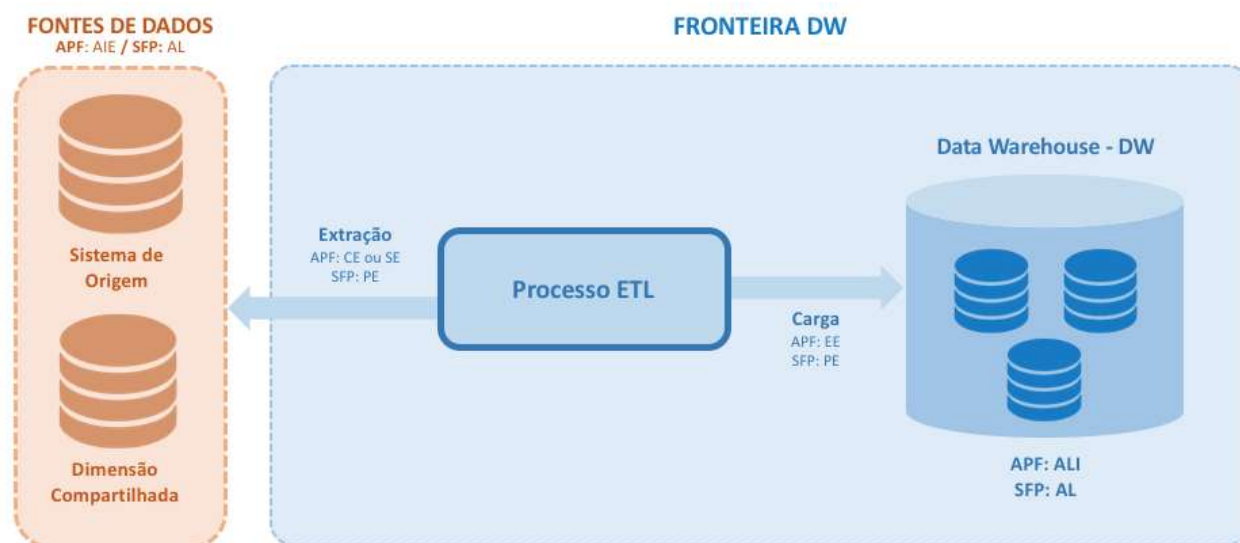
### 13.6.2. Contagem de funções de dados do processo ETL

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 17 - 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:

Figura 18 - Resumo da contagem de funções de dados em um processo ETL

		O método SFP identifica:	O método APF identifica:
<b>Sistema de Origem</b>	Para cada arquivo lógico do sistema de origem, usado na validação de dados do processo de ETL	1 AL	1 AIE
<b>Tabela dimensão compartilhada</b>	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

### 13.6.3. 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.

## 13.7. 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 apresentado neste 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”.

### 13.7.1. 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 roteiro, 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 13.9.8 - Exclusão de Dados Antigos do DW.

As cargas de dados em tabelas de Agregação do DW, conforme descrito no item 13.6.1 (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 40 - Resumo da contagem de funções de carga**

	<b>O método SFP identifica:</b>	<b>O método APF identifica:</b>
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 13.6	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

### 13.7.2. 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.

Geralmente, os relatórios solicitados para o os times de DW possuem dados calculados, sendo contabilizados como SE, para o método APF e como PE para o método SFP. Nesse sentido:

- 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 13.6.

**Tabela 41 - Resumo da contagem de funcionalidades de geração de relatórios**

	<b>O método SFP identifica:</b>	<b>O método APF identifica:</b>
Cada relatório ou gráfico requisitado pelo usuário e implementado pela equipe de desenvolvimento	1 PE	1 CE ou 1 SE
Para cada geração do Cubo por cada Estrela modelada	1 PE	1 SE

### 13.7.3. Contagem de funções de extração de dados do Sistema Transacional de Origem

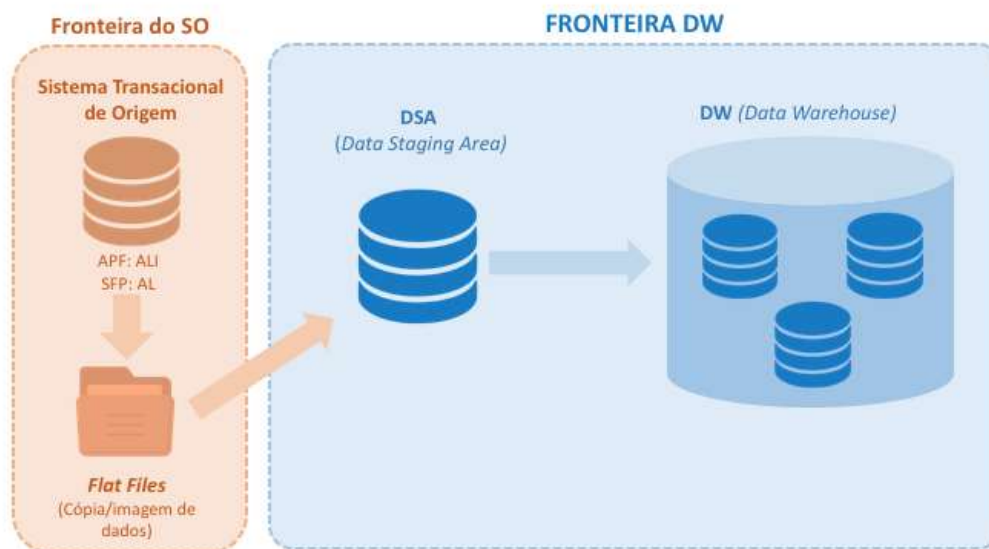
Segundo NESMA (2014), a extração de dados de um sistema transacional de origem para a carga nas tabelas do DW através do processo de ETL, acontece por dois cenários:

- **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:

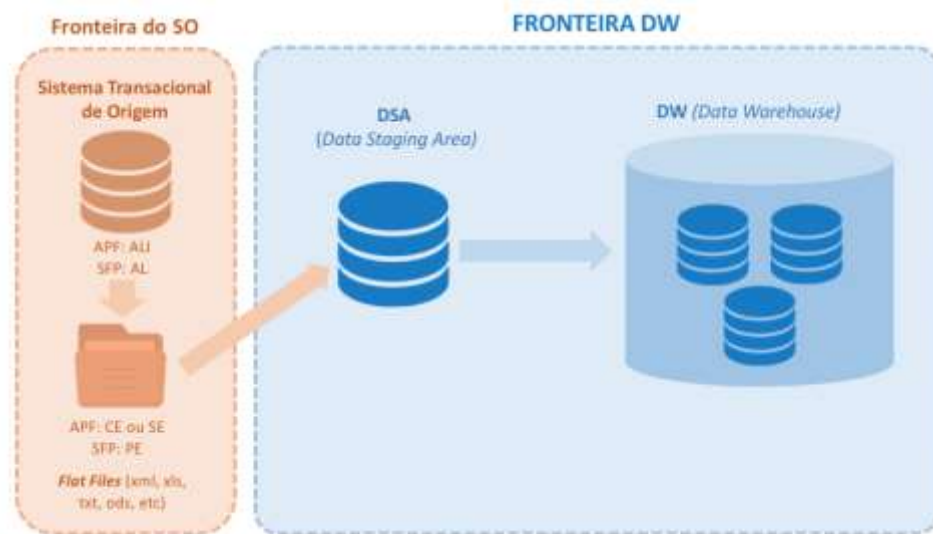
- **Cenário 1: flat files sendo uma cópia de dados de tabela do sistema de origem.** Neste caso, o *flat file* é uma cópia ou imagem de um arquivo lógico (APF: ALI/ SFP: AL) do sistema de origem, sem nenhum processamento adicional para a sua geração.
  - **Para o método APF,** considera-se apenas, no 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 ALRs e a contribuição funcional do processo elementar é sempre 4,6 SFP.

Figura 19 - 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 20 - Cenário 2 – geração de flat files na fronteira do sistema de origem

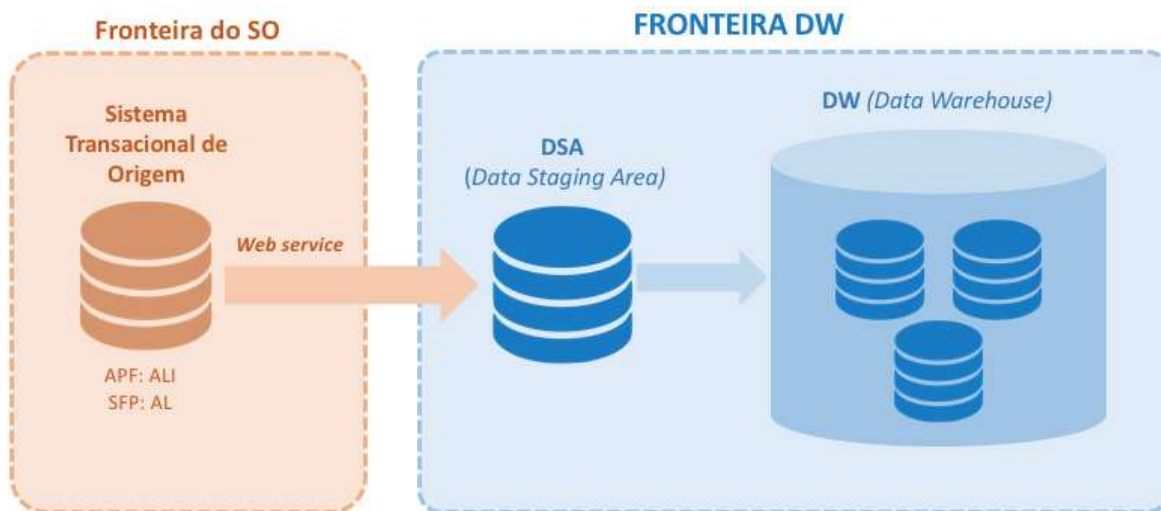


- **Cenário 3: interface direta entre a base de dados do sistema de origem e o DSA.** Neste caso, não ocorre a transferência de dados da base do sistema de origem para o DW, mas esses dados ficam disponíveis para leitura pelo DW. A consulta por *WebService* para a extração de dados dos sistemas de origem 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 21 - Cenário 3 – interface direta entre a base de dados do sistema de origem e o DSA**

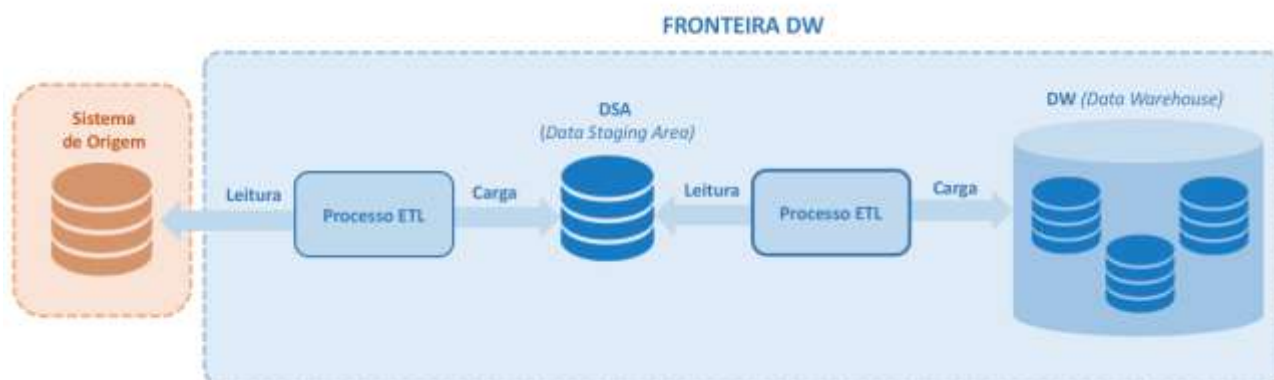


A funcionalidade de extração de dados de uma aplicação de origem, quando houver requisito de desenvolvimento, deve ser contada como uma função transacional (APF: CE ou SE/ SFP: PE) dentro da fronteira da aplicação de origem.

Conseqüentemente, 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.

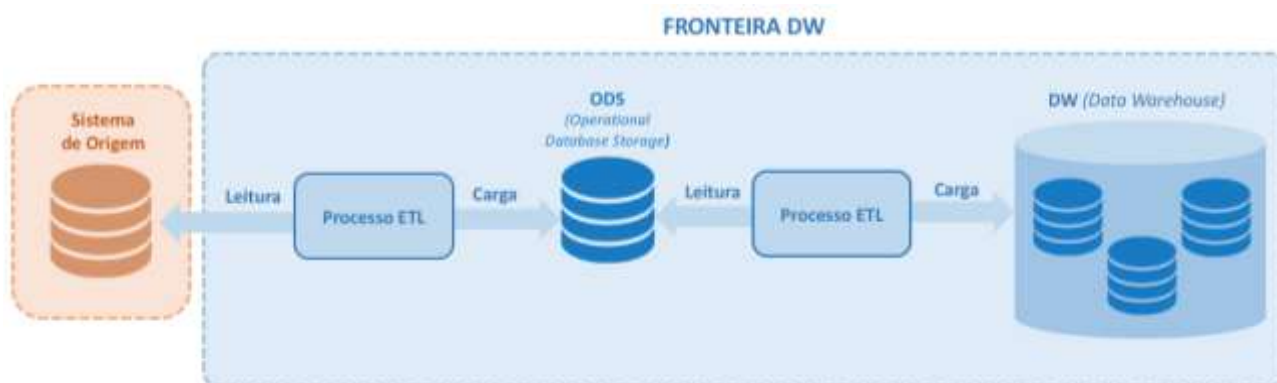
Figura 22 - Carga de dados do sistema de origem na DSA



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. 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 23 - 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 42 - 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

## 13.8. 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 7.3 (PROJETO DE MELHORIA) deste roteiro, que apresenta a contagem de PF utilizando os métodos SFP e APF.

## 13.9. 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 7 (CÁLCULO DE PONTOS DE FUNÇÃO PARA O SISF) 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.

### 13.9.1. 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, a aferição do tamanho em pontos de função das dimensões alteradas deve ser realizada da seguinte forma:

- Para o método APF: com a aplicação de um fator de impacto de 13% sobre a contagem de uma EE de baixa complexidade (3 PF), ou seja 0,39 PF;
- Para o método SFP: com a aplicação de um fator de impacto de 13% sobre a contagem de um processo elementar (4,6 SFP), ou seja 0,59 SFP.



$PF\_DIMENSÃO\ ESTÁTICA\_BD = 0,39\ PF\ X\ QTD\ de\ dimensões\ alteradas$

$SFP\_DIMENSÃO\ ESTÁTICA\_BD = 0,59\ PF\ X\ QTD\ de\ dimensões\ alteradas$

### 13.9.2. 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 SFP\_ALTERADO/PF\_ALTERADO e aplicar o Fator de Impacto (FI) conforme definido no Projeto de Melhoria de DW (item 13.8).

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 43 - Exemplo de Criação/Alteração de Campos em tabelas fato e dimensão**

	<b>O método SFP identifica:</b>	<b>O método APF identifica:</b>
Alteração de tabela Fato ou Dimensão (SFP_ALTERADO/PF_ALTERADO)	1 AL	1 ALI
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 44 - Exemplo de Criação/Alteração de Campos em tabela dimensão Compartilhada**

	<b>O método SFP identifica:</b>	<b>O método APF identifica:</b>
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

### 13.9.3. 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 7.12.1 (Apuração Especial – Base de dados) deste roteiro de métricas.

### 13.9.4. Extração de Dados em Tabelas Fato e Dimensão

A extração ou recuperação de dados de tabelas Fato e Dimensão não deve ser contabilizada em projetos de DW quando a ferramenta OLAP já fornecer 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 7.12.2 (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.

### 13.9.5. 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 7.18 (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.

### 13.9.6. 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 13.9.2 (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 7.10 (Manutenção em Interface), 7.11 (Adaptação sem Alteração de Requisitos Funcionais) ou 7.18 (Componente Reusável) deste roteiro de métricas.

### 13.9.7. 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 7.10 (Manutenção em Interface) deste roteiro de métricas e considerar o respectivo FI para cada item reposicionado/reorganizado da bancada no projeto de DW.

### **13.9.8. 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.

### **13.9.9. 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. Contudo, caso essas funcionalidades sejam requeridas e desenvolvidas para apresentar ou editar esses dados de controle de forma diferente do disponível na ferramenta de ETL, 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.

## 13.9.10. Metadados Relacionados ao Significado dos Dados -

### Metadados do Negócio

Os metadados do negócio descrevem o significado dos dados de um DW. Normalmente, tem-se o “rótulo” e a “descrição” do campo. Conforme STN (2013), representam as demandas para descrever atributos, métricas e pastas relacionadas a uma tabela Fato ou atributos e pastas associados a uma tabela Dimensão.

No projeto de desenvolvimento, esses metadados são parte do projeto de DW e, portanto, não conta PF. No projeto de manutenção, se houver demanda para alterar a “descrição” associada ao nome do campo criado, sugere-se a adoção das seguintes fórmulas, iguais para os dois métodos:



**PF\_METADADOS\_NEGÓCIO** = PF\_INCLUÍDO x 0,13 (para tabelas fato ou dimensão)  
**PF\_METADADOS\_NEGÓCIO** = 0,39 PF X QTD de dimensões alteradas

**SFP\_METADADOS\_NEGÓCIO** = SFP\_INCLUÍDO x 0,13 (para tabelas fato ou dimensão)  
**SFP\_METADADOS\_NEGÓCIO** = 0,59 SFP X QTD de dimensões alteradas

Se uma tabela Dimensão tiver mais de um registro lógico, então conta-se apenas uma vez esse AL/ALI em (SFP\_METADADOS\_NEGÓCIO/ PF\_METADADOS\_NEGÓCIO).

### 13.9.11. 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 7.8.2 (Mudança de Plataforma – Banco de Dados) ou 7.9.3 (Atualização de Versão – Banco de Dados) deste roteiro.

### 13.9.12. 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 7.8.1 (Mudança de Plataforma - Linguagem de Programação) ou 7.9.1 (Atualização de Versão - Linguagem de Programação) deste roteiro.

## 14. ATIVIDADES SEM CONTAGEM DE PF

Ressalta-se 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 nem serem inerentes ao processo de desenvolvimento do software.

Considerando um projeto de desenvolvimento de software que utiliza métodos ágeis, ao adotar a modalidade de remuneração por pontos de função, complementados por HST, conforme padronização da Portaria SGD/MGI nº 750 ou posterior:

- as atividades relacionadas à 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, que não são mensuráveis pela métrica de PF, devem ser remuneradas pela métrica de Horas de Serviço Técnico – HST, descrita no item 16 deste roteiro.

O catálogo de serviços técnicos aferidos em HST (disponível em <https://www.gov.br/governodigital/pt-br/contratacoes-de-tic>) inclui atividades como elaboração de manuais de usuário, execução de testes avançados, serviços de UX e outros serviços técnicos essenciais ao ciclo de vida do software.

Nos subitens a seguir apresentam-se:

- Atividades diversas, não remuneradas pela métrica de PF; e
- Atividades sem contagem de PF em projetos de DW.

## 14.1. ATIVIDADES SEM CONTAGEM DE PF EM PROJETOS DE DESENVOLVIMENTO DE SW

A seguir apresentam-se atividades não remuneradas pela métrica de PF:

- **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 Coaching*, 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 de 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.

## 14.2. ATIVIDADES SEM CONTAGEM DE PF EM PROJETOS DE DW

Destacam-se a seguir 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.

### 14.2.1. 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 se tratam de soluções técnicas;
- Tabelas Dimensão Estática;
- Tabelas de Agregação criadas para atender requisitos não funcionais, como desempenho.

### **14.2.2. 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 às customizações realizadas em desenvolvimento de *software*.

### **14.2.3. 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.

### **14.2.4. Customização da Ferramenta ETL**

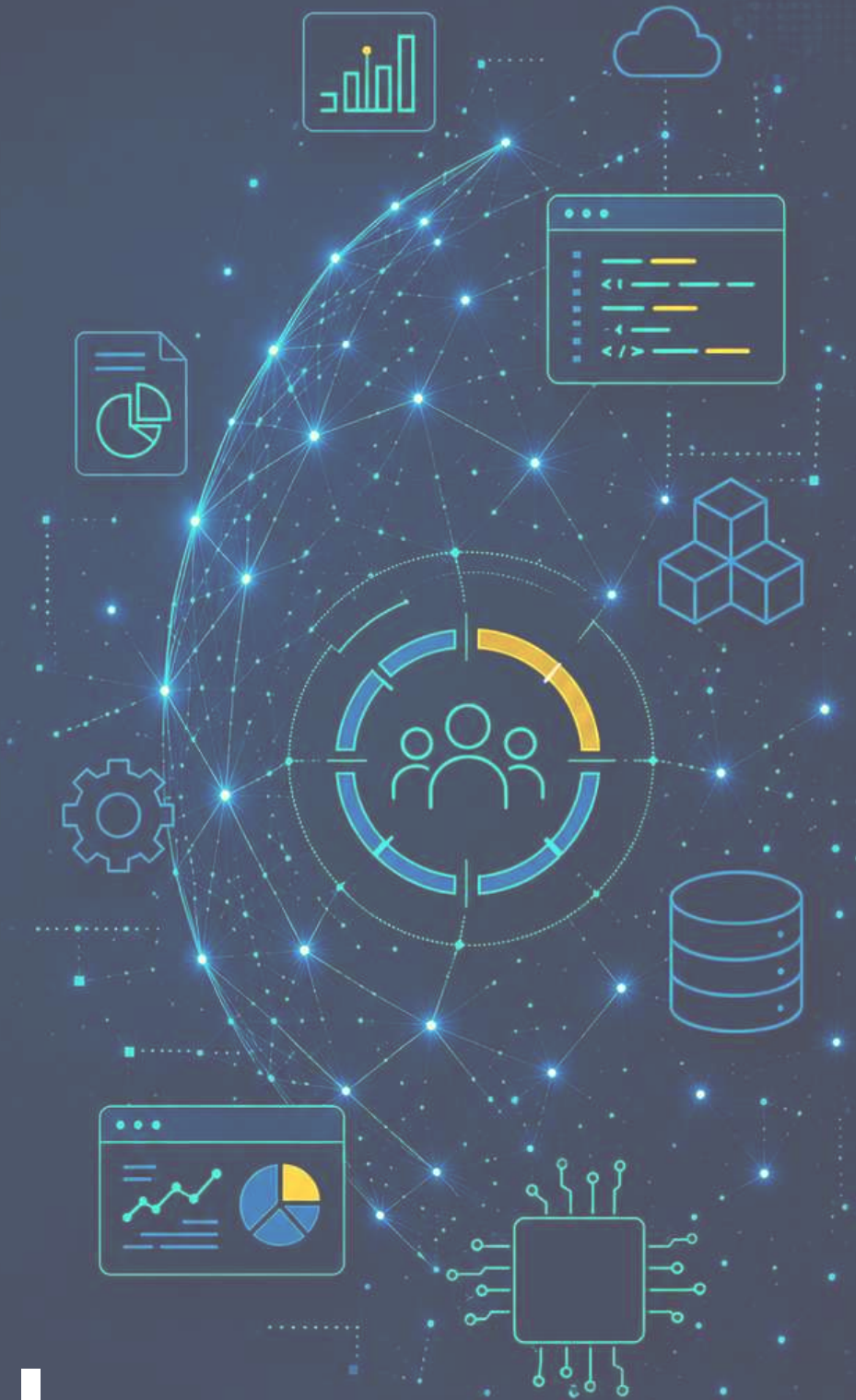
De forma similar ao item 14.2.2 (Customização da Ferramenta OLAP), a customização da ferramenta de ETL corresponde a customizações/personalizações feitas em ferramentas ETL de forma a atender as necessidades do cliente. Por exemplo, alterações de uma página *web* da ferramenta ETL para a substituição do logo do fornecedor pelo logo do cliente; disponibilização de mensagens, dentre outras necessidades.

### **14.2.5. 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.

### 14.2.6. 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.



# PARTE II

## Story Points Padronizados (SPP)

## 15. MÉTRICA *STORY POINTS* PADRONIZADOS (SPP)

Story Points Padronizados (SPP) é uma métrica estruturada para estimativa de esforço e mensuração de produtividade, baseada na atribuição de pontuação a histórias de usuário por meio de critérios objetivos e verificáveis.

A pontuação é definida a partir de duas dimensões observáveis:

- o tipo base da história; e
- os fatores técnicos.

Esses elementos devem ser identificados a partir do conteúdo da história de usuário e, principalmente, de seus critérios de aceitação. Esses critérios constituem a principal fonte de evidência para a aplicação da métrica. A variação da pontuação em SPP não decorre de percepção subjetiva do time, mas da presença explícita de atributos funcionais e fatores técnicos previamente definidos. Dessa forma, a métrica reduz significativamente a subjetividade, permitindo maior consistência, rastreabilidade e auditabilidade das estimativas.

Sua utilização deve estar associada à mensuração de entregas funcionais, de modo a assegurar que a métrica reflita o valor efetivamente entregue, e não apenas o esforço técnico interno. Essa abordagem contribui para maior transparência, comparabilidade e confiabilidade na medição de produtividade.

A métrica SPP não deve ser utilizada exclusivamente para fins de pagamento. Esse roteiro recomenda a sua utilização como instrumento de apoio ao planejamento e à estimativa das Sprints, bem como indicador para o acompanhamento da produtividade mínima esperada dos times de desenvolvimento. Sua aplicação possibilita a padronização da estimativa de histórias de usuário, promovendo comparabilidade não apenas dentro de um mesmo time, mas também entre diferentes equipes da organização.

A lógica central da métrica consiste na substituição da estimativa baseada exclusivamente na opinião do time por um modelo de contagem objetiva, fundamentado na identificação estruturada do tipo funcional predominante da história e dos fatores técnicos presentes. Antes de detalhar o método de contagem por SPP, serão apresentados conceitos e orientações essenciais para sua correta aplicação, especialmente no que se refere à construção do backlog do produto e à elaboração de histórias de usuário com critérios de aceitação adequados.



### A métrica SPP não deve ser utilizada exclusivamente para fins de pagamento.

Esse roteiro recomenda a sua utilização como instrumento de apoio ao planejamento e à estimativa das Sprints, bem como indicador para o acompanhamento da produtividade mínima esperada dos times de desenvolvimento.

## 15.1. O QUE MUDA EM RELAÇÃO AO *STORY POINTS* TRADICIONAL

O *story point* (SP) tradicional é uma estimativa relativa construída a partir da percepção do time, sendo, portanto, inerentemente subjetiva e dependente do contexto de cada equipe.

Story Points Padronizados (SPP) introduz um ajuste conceitual relevante: a pontuação não é baseada na opinião do time, mas sim em critérios objetivos, verificáveis e previamente definidos, extraídos da história de usuário e, principalmente, de seus critérios de aceitação.

Dessa forma, a pontuação no SPP é calculada a partir de atributos observáveis do produto funcional, permitindo registro estruturado, rastreabilidade e auditabilidade do processo de estimativa.

O quadro a seguir apresenta as principais diferenças entre o modelo tradicional de *story points* (SP) e o SPP:

**Tabela 45 - Comparação entre SP Tradicional e SPP**

Aspecto	SP Tradicional	SPP ( <i>Story Points</i> Padronizados)
<b>Base da estimativa</b>	Percepção relativa do time	Critérios objetivos e verificáveis
<b>Unidade de comparação</b>	Comparação entre histórias	Avaliação direta da própria história
<b>Subjetividade</b>	Alta	Controlada por regras claras
<b>Dependência da equipe</b>	Alta (cada time pontua de forma distinta)	Baixa (regras padronizadas)

<b>Consistência entre projetos</b>	Baixa	Alta
<b>Auditabilidade</b>	Limitada	Possível e estruturada
<b>Forma de estimar</b>	Técnicas colaborativas ( <i>Planning Poker, T-shirt size, etc.</i> )	Checklist estruturado com critérios definidos
<b>Fonte de evidência</b>	Conhecimento tácito do time	Critérios de aceitação e artefatos da história
<b>Rastreabilidade</b>	Baixa	Alta - pontuação justificável por evidências
<b>Replicabilidade</b>	Baixa (resultados variam entre equipes)	Alta (mesma entrada tende a gerar o mesmo resultado)
<b>Foco da métrica</b>	Esforço relativo percebido	Complexidade funcional padronizada da história
<b>Aplicação organizacional</b>	Limitada ao time	Aplicável em nível organizacional

## 15.2. ORIENTAÇÕES GERAIS

O desenvolvimento ágil tem sido amplamente adotado pelos órgãos da Administração Pública Federal (APF) no contexto de desenvolvimento de software. Independentemente do porte ou complexidade dos projetos, diferentes métodos ágeis podem ser aplicados em distintas fases do ciclo de vida, conforme as necessidades do contexto.

De modo geral, as organizações utilizam uma combinação de práticas, métodos e terminologias ágeis, ajustadas à sua cultura organizacional e às características dos projetos. Dentre os principais métodos adotados, destacam-se o *Scrum*, *Extreme Programming (XP)*, *Kanban*, *Lean* e *Feature Driven Development (FDD)*, entre outros.

No *Scrum*, o desenvolvimento ocorre de forma iterativa e incremental, por meio de ciclos denominados *sprints*, com duração típica entre 2 e 4 semanas. Conforme o *Scrum Guide (2020)*, o *framework* é estruturado em três artefatos principais:

- 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 (2020)*, 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 viabilizam a inspeção contínua, promovem transparência, foco e alinhamento entre os membros do time. Contudo, o Scrum não estabelece, de forma prescritiva, como esses artefatos devem ser detalhados ou estruturados.

A partir desses elementos, são construídas as histórias de usuário, que constituem a principal unidade de trabalho utilizada na aplicação da métrica *Story Points* Padronizados (SPP). Originadas no *Extreme Programming (XP)*, as histórias de usuário têm como objetivo descrever, de forma simples e orientada a valor, as necessidades dos usuários e as funcionalidades esperadas, permitindo priorização e implementação incremental.

Neste contexto, as seções seguintes apresentam orientações sobre a construção dos principais artefatos e elementos necessários para viabilizar a aplicação do SPP, com foco nos requisitos mínimos que asseguram a consistência, a objetividade e a rastreabilidade da contagem.

Ressalta-se que este roteiro não tem como objetivo detalhar exhaustivamente os métodos ágeis ou o *framework Scrum*, mas sim estabelecer as condições necessárias para a correta aplicação da métrica *Story Points* Padronizados (SPP).

### 15.3. **BACKLOG DO PRODUTO (PRODUCT BACKLOG)**

O *Backlog* do Produto (*Product Backlog*) é o principal artefato utilizado para organizar, priorizar e gerenciar o trabalho a ser realizado no desenvolvimento de software. Sua construção envolve o levantamento estruturado de requisitos, restrições, objetivos e demais itens necessários à evolução do produto.

Trata-se de um documento vivo e em contínua evolução. Entre os itens que devem ser considerados na construção do *backlog* do produto destacam-se: requisitos funcionais, requisitos não funcionais, requisitos de

*interface*, requisitos de interação, requisitos de domínio, itens de trabalho, problemas, restrições e preocupações das partes interessadas, tarefas técnicas, processos e resultados relacionados, cenários e contexto, restrições, e assim por diante.

A construção do *backlog* do produto requer conhecimento em práticas ágeis e entendimento do domínio do negócio. Recomenda-se que sua elaboração ocorra de forma colaborativa, envolvendo os membros do time e as partes interessadas relevantes, especialmente em sua versão inicial.

O *backlog* do produto é utilizado como referência central para acompanhamento da evolução do projeto, apoiando o planejamento, a priorização de entregas, a alocação de esforços e o alinhamento entre a organização e o time ágil.

A organização deve definir e disponibilizar ferramentas adequadas para gestão do *backlog*, de forma a garantir acesso, transparência e rastreabilidade dos itens (PBIs – *Product Backlog Items*).

Como resultado de um levantamento adequado do *backlog* do produto, espera-se a compreensão dos principais aspectos do domínio do problema, incluindo partes interessadas, prioridades de negócio, riscos envolvidos e possíveis conflitos de interesse.

O tipo mais comum de item do *backlog* do produto é a História de Usuário, embora também possam existir outros tipos de itens, como requisitos técnicos, itens de infraestrutura e demandas de teste.

A priorização dos itens do *backlog* é, em geral, responsabilidade do *Product Owner* (PO), que deve considerar o valor para o negócio, os riscos associados e os objetivos estratégicos do produto.

### 15.3.1. ITENS DO BACKLOG

Antes de iniciar a construção do *backlog* do produto, é fundamental que haja alinhamento quanto à terminologia adotada. Neste documento, são utilizados os seguintes conceitos:

- **Backlog do Produto:** conjunto estruturado de todos os itens de trabalho do produto;
- **Épico:** item de alto nível, que representa uma funcionalidade ampla ainda não detalhada;
- **História de Usuário:** unidade funcional que descreve uma necessidade do usuário, com valor de negócio associado;
- **Tarefa:** desdobramento técnico necessário para implementação de uma história de usuário.

A figura a seguir apresenta uma visão simplificada da relação entre esses elementos.

**Figura 24 - Alinhamento de termos do *Backlog* do Produto**



A **História de Usuário** é uma descrição concisa de uma funcionalidade, orientada à perspectiva do usuário, que explicita o objetivo, o benefício e o valor esperado.

Os **Épicos** correspondem a histórias de maior dimensão, ainda em estágio inicial de refinamento, que podem ser progressivamente decompostas em histórias menores. A associação entre histórias e épicos é opcional.

As **Tarefas** representam atividades técnicas necessárias à implementação de uma história de usuário. Elas detalham como o trabalho será executado e, em geral, não possuem valor de negócio isoladamente. Destaca-se que as tarefas não devem ser utilizadas como unidade de mensuração no SPP, uma vez que a métrica se baseia exclusivamente em entregas funcionais percebidas pelo usuário.

### 15.3.2. Refinamento contínuo do *Backlog* do Produto

O *backlog* do produto é um artefato dinâmico e, portanto, seu refinamento contínuo é essencial para o bom desempenho dos times ágeis.

O objetivo do refinamento contínuo é revisar, esclarecer, detalhar, reorganizar e priorizar os itens do *backlog* do produto, de forma a mantê-los adequados ao escopo de trabalho mais imediato.

Por meio dessa prática, funcionalidades podem ser decompostas em itens menores, e novas informações podem ser incorporadas, resultando em um *backlog* com itens (PBIs) mais claros, consistentes e precisos.

O time deve considerar continuamente os *insights* e *feedbacks* das partes interessadas a cada incremento do produto. Com isso, novos itens podem ser incluídos, enquanto itens existentes podem ser ajustados, reordenados ou detalhados, mantendo o backlog atualizado e alinhado às necessidades do negócio.

Nem todos os itens do *backlog* apresentam o mesmo nível de detalhe. Em geral, os itens mais próximos de implementação — normalmente aqueles planejados para as próximas *sprints* — devem possuir maior nível de refinamento.

O *Product Owner* (PO) deve atuar continuamente na preparação dos itens candidatos às próximas *sprints*. Recomenda-se que o refinamento seja concentrado nos itens de curto prazo, tipicamente aqueles previstos para uma a três *sprints*, evitando detalhamento excessivo de itens que ainda poderão sofrer alterações.

Itens previstos para um horizonte mais distante não devem ser refinados antecipadamente, a fim de evitar retrabalho e perda de foco.

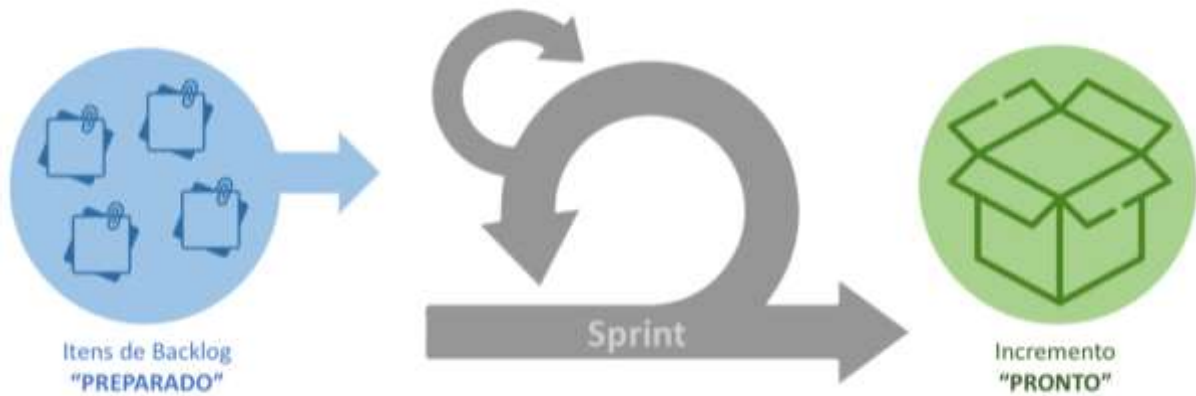
## 15.4. **BACKLOG DA SPRINT**

Após a definição, refinamento, organização e priorização dos itens do *backlog* do produto (PBIs), o time passa à definição do *Backlog da Sprint*, normalmente durante a reunião de Planejamento da *Sprint*.

O *Backlog da Sprint* corresponde a um subconjunto do *backlog* do produto, selecionado para desenvolvimento na iteração corrente. A qualidade dos itens que compõem esse subconjunto deve ser avaliada em dois momentos distintos:

- **Na entrada (antes da *sprint*):** por meio da **Definição de Preparado (*Definition of Ready*)**, estabelecida em comum acordo entre o *Product Owner* (PO) e o time de desenvolvimento. Essa definição determina as condições mínimas que um item do *backlog* do produto deve atender para ser considerado apto a ingressar no *Backlog da Sprint*.
- **Na saída (ao final da *sprint*):** por meio da **Definição de Pronto (*Definition of Done*)**, que estabelece os critérios necessários para que o item seja considerado concluído. Itens que atendem a esses critérios são apresentados na **Revisão da *sprint*** e passam a compor o incremento do produto.

**Figura 25 - Fluxo do PBI – Definição de preparado e definição de pronto**



A Definição de Preparado e a Definição de Pronto são, em geral, estabelecidas antes do início do desenvolvimento, podendo ser definidas ainda nas fases iniciais do projeto. No entanto, ambas devem ser tratadas como artefatos evolutivos, passíveis de revisão e ajuste ao longo do tempo, conforme novas necessidades sejam identificadas.

Nas seções seguintes, serão detalhados os critérios relacionados à construção das histórias de usuário e às definições de preparado e de pronto, os quais devem ser observados para assegurar a qualidade dos itens do *backlog* que serão implementados na *sprint*.

## 15.5. HISTÓRIAS DE USUÁRIO (*user stories*)

A história de usuário é a principal unidade de representação de requisitos em abordagens ágeis, permitindo a identificação e a elicitación de necessidades sob a perspectiva do usuário final. Trata-se de uma descrição concisa de uma funcionalidade, que explicita o objetivo, o benefício esperado e o valor associado para o negócio e para as partes interessadas.

As histórias de usuário deslocam o foco de especificações técnicas extensas para a compreensão das necessidades do usuário, permitindo que as equipes desenvolvam soluções mais aderentes ao contexto real de uso e orientadas à geração de valor.

Na elaboração de uma história de usuário, devem ser considerados, de forma integrada as necessidades e expectativas do usuário, o valor para o negócio, os requisitos funcionais e não funcionais, os requisitos de qualidade do software, os riscos, restrições e condições de uso, bem como o contexto operacional e os cenários de aplicação.

Uma história de usuário bem definida deve apresentar requisitos claros, compreensíveis e alinhados entre as partes envolvidas. A interação contínua com o *Product Owner* (PO) e demais *Stakeholders* contribui para o refinamento das histórias e para a incorporação de feedbacks ao longo do desenvolvimento.

Nesse sentido, recomenda-se que as histórias de usuário:

- sejam descritas em linguagem simples, clara e objetiva;
- representem uma necessidade real do usuário e evidenciem o benefício associado;
- possuam critérios de aceitação definidos e vinculados ao comportamento esperado;
- estejam orientadas à geração de valor para o negócio;
- sejam progressivamente refinadas e, quando necessário, decompostas em unidades menores.

### 15.5.1. Formato das Histórias do usuário

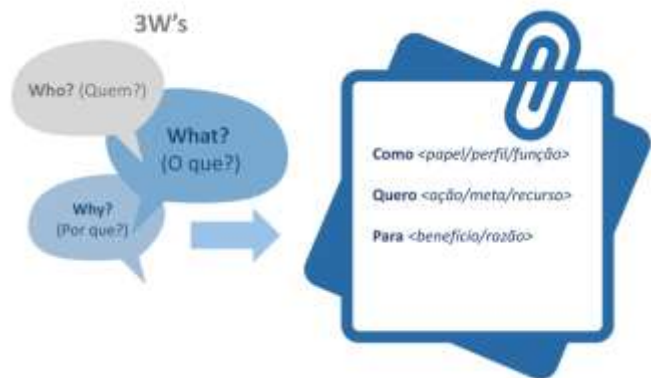
A História de usuário é um formato textual para a descrição de um requisito que busca responder três perguntas do acrônimo conhecido como 3Ws: *Who?* (Quem?), *What?* (O quê?) e *Why?* (Por quê?)

O formato recomendado de uma história do usuário é:

- 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.



Recomenda-se que a construção das histórias de usuário ocorra de forma colaborativa, envolvendo o time de desenvolvimento, o *Product Owner* (PO) e, sempre que possível, as partes interessadas, de modo a assegurar entendimento compartilhado.

### 15.5.2. Critérios de qualidade - INVEST

O acrônimo "INVEST" pode ser usado como referência avaliar a qualidade das histórias de usuário:

- **Independent (independente):** cada história de usuário, sempre que possível, 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 aos *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 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.



O INVEST funciona como um guia de boas práticas, não sendo necessário que todas as características estejam presentes de forma absoluta em todas as histórias. No entanto, quanto maior a aderência a esses critérios, maior tende a ser a qualidade da história e a previsibilidade da sua implementação

### 15.5.3. Métodos adicionais para levantamento das Histórias de Usuário

A elaboração de histórias de usuário em linguagem simples, clara, estimável, testável e orientada a valor não é uma tarefa trivial, exigindo conhecimento técnico e compreensão adequada do contexto do produto.

Embora, em tese, qualquer membro da equipe possa contribuir para a escrita de histórias de usuário, o *Product Owner* (PO) é o responsável pela gestão do *backlog* do produto, incluindo a criação, organização e comunicação dos itens que o compõem.

O uso do modelo 3W's (Quem? O quê? Por quê?) favorece a colaboração entre as partes interessadas e contribui para a estruturação inicial das histórias. No entanto, esse modelo não substitui a necessidade de aprofundamento do entendimento do problema e do contexto de uso.

As histórias de usuário podem ser identificadas e elaboradas em diferentes momentos do processo de desenvolvimento, tais como:

- durante reuniões de revisão de *sprint*;
- ao surgimento de novas demandas ou ideias;
- no refinamento do *backlog* do produto;
- em workshops de mapeamento de histórias;
- em interações diretas com usuários e stakeholders.

Não existe uma técnica única ou padrão para o levantamento ou mapeamento de histórias de usuário. A escolha da abordagem mais adequada depende do contexto do projeto, do nível de maturidade do time e do grau de complexidade do problema.

Nesse sentido, recomenda-se que os profissionais envolvidos possuam conhecimento de diferentes técnicas de levantamento de requisitos, de forma a selecionar, de maneira consciente, aquelas mais adequadas a cada situação.

A seguir, apresentam-se exemplos de técnicas que podem ser utilizadas no levantamento e mapeamento de histórias de usuário:

**Tabela 46 - Métodos de levantamento de requisitos**

<b>Técnicas/abordagens</b>	<b>Descrição</b>	<b>Exemplos</b>
<b>Técnicas tradicionais de coleta de dados</b>	Métodos de obtenção direta de informações, amplamente utilizados para coleta de dados relevantes sobre o domínio do problema.	Questionários, entrevistas, observação, pesquisas, análise de documentos, benchmarking, etc.
<b>Técnicas de elicitación em grupo, ideação</b>	Abordagens colaborativas voltadas à compreensão das necessidades dos usuários e geração de soluções.	<i>Brainstorming, Design Thinking, Joint Application Development (JAD), workshops</i> de cocriação, grupos de discussão.
<b>Prototipação</b>	Utilização de representações visuais ou interativas para apoiar a elicitación e validação de requisitos, especialmente em cenários de incerteza.	Protótipos em papel, <i>wireframes</i> , protótipos digitais, <i>storyboard</i>
<b>Técnicas contextuais</b>	Métodos voltados à compreensão do ambiente e do contexto de uso do sistema.	Etnografia, análise social, pesquisas exploratórias etc.
<b>Técnicas de análise, síntese e representação de dados</b>	Técnicas utilizadas para organização, interpretação e estruturação das informações coletadas.	Diagrama de afinidade, mapas conceituais, mapeamento mental, matriz de posicionamento
<b>Técnicas para identificação dos usuários</b>	Métodos voltados à compreensão do público-alvo e suas necessidades.	Personas, mapa de empatia, modelagem de papéis do usuário

#### 15.5.4. Critérios de Aceitação da História do Usuário

Cada história de usuário deve possuir critérios de aceitação (*Acceptance Criteria – ACs*), os quais estabelecem condições objetivas, explícitas e verificáveis que devem ser integralmente atendidas para que a entrega seja considerada concluída.

Os critérios de aceitação constituem o referencial técnico para validação da funcionalidade implementada, permitindo a verificação clara do atendimento ao escopo definido.

No contexto das contratações públicas que utilizam métricas baseadas em *story points*, conforme diretrizes da Portaria SGD nº 750/2023, a definição adequada dos critérios de aceitação é fundamental para assegurar:

- objetividade na medição;
- transparência na execução contratual;
- rastreabilidade entre demanda e entrega;
- redução de subjetividade no aceite;
- mitigação de riscos de questionamentos.

Nesse contexto, os critérios de aceitação configuram-se como condição necessária para a homologação da entrega e para a consequente contabilização dos *Story Points* Padronizados (SPP).

##### 15.5.4.a. Relação entre os critérios de aceitação da história e o SPP

Os critérios de aceitação são a principal fonte de evidência para aplicação da métrica SPP, uma vez que é a partir deles que se identificam os atributos funcionais e os fatores técnicos da história. Sua definição pode ocorrer por meio de diferentes práticas de elicitação, tais como interações com partes interessadas, aplicação de técnicas de levantamento de requisitos, *workshops* e sessões de refinamento, testes de usabilidade e validações exploratórias.

Cada história de usuário deve possuir critérios próprios, capazes de alinhar as expectativas entre a área demandante e a equipe de desenvolvimento. Os critérios devem ser redigidos de forma:

- **clara**, sem ambiguidades;
- **objetiva**, sem interpretações subjetivas;
- **verificável**, permitindo validação binária (atende ou não atende).

Não é admitida aprovação parcial de critérios de aceitação. Critérios imprecisos, incompletos ou ambíguos indicam a necessidade de refinamento adicional da história antes de sua estimativa.

Para fins de mensuração por SPP, é indispensável que os critérios de aceitação contemplem explicitamente, quando aplicável: as integrações com sistemas externos, os requisitos de auditoria e rastreabilidade e os requisitos não funcionais (desempenho, segurança, disponibilidade, entre outros). A ausência desses elementos pode resultar em subdimensionamento da complexidade e distorções na mensuração.

A estimativa por SPP deve considerar o esforço necessário para o atendimento integral dos critérios previamente definidos e acordados. Por essa razão, a definição dos critérios de aceitação constitui etapa obrigatória anterior à estimativa. A clareza e completude dos critérios de aceitação são essenciais para assegurar:

- objetividade e consistência na medição por SPP;
- rastreabilidade entre escopo estimado e produto entregue;
- sustentação técnica da pontuação atribuída;
- redução de divergências no aceite;
- segurança jurídica na execução contratual.

Alterações nos critérios de aceitação após a estimativa devem ser formalmente tratadas como modificação de escopo, com registro explícito da versão dos critérios vigente no momento da estimativa.



Os critérios de aceitação constituem o referencial técnico mínimo para validação da entrega, sendo condição necessária para sua homologação e conseqüente contabilização por SPP.

**A estimativa por SPP considera o esforço necessário para o atendimento integral dos critérios previamente definidos e acordados.**

#### 15.5.4.b. Qualidade dos critérios de aceitação

O quadro a seguir apresenta exemplos ilustrativos de critérios de aceitação, evidenciando diferenças entre formulações inadequadas e adequadas para fins de mensuração por SPP.

**Tabela 47 - Qualidade dos Critérios de Aceitação (Referência para SPP)**

Situação	Critério de Aceitação	Avaliação	Justificativa
<b>✘ Inadequado (genérico)</b>	“O sistema deve funcionar corretamente”	Não aceitável	Não é verificável nem mensurável
<b>✘ Subjetivo</b>	“A tela deve ser intuitiva”	Não aceitável	Depende de interpretação individual
<b>✘ Incompleto</b>	“Validar CPF do beneficiário”	Não aceitável	Não especifica formato, regra ou comportamento esperado
<b>✘ Não testável</b>	“O sistema deve ser rápido”	Não aceitável	Não define parâmetro de desempenho
<b>✘ Ambíguo</b>	“Exibir dados corretamente”	Não aceitável	Não define quais dados ou critérios de correção
<b>✘ Incompleto</b>	“Validar CPF conforme regra padrão”	aceitável com ressalvas	Requer referência explícita da regra utilizada
<b>☑ Adequado (objetivo)</b>	“O sistema deve validar o CPF com 11 dígitos numéricos, rejeitando valores inválidos conforme algoritmo de validação”	Aceitável	Critério claro, objetivo e verificável
<b>☑ Adequado (testável)</b>	“O sistema deve impedir o salvamento quando campos obrigatórios não forem preenchidos”	Aceitável	Permite validação binária
<b>☑ Adequado (completo)</b>	“O sistema deve consultar o serviço de CEP e, em caso de indisponibilidade, permitir preenchimento manual dos campos”	Aceitável	Define comportamento principal e exceção
<b>☑ Adequado (requisito não funcional explícito)</b>	“O tempo de resposta da operação deve ser inferior a 2 segundos”	Aceitável	Define critério mensurável
<b>☑ Adequado (auditável)</b>	“O sistema deve registrar log contendo usuário, data, hora e alteração realizada”	Aceitável	Permite rastreabilidade

### 15.5.4.c. Formatos dos critérios de aceitação

Os critérios podem ser escritos em diferentes formatos, desde que assegurem objetividade e verificabilidade.

Os dois mais utilizados são:

- Formato orientado a regras; e
- Formato orientado a cenários.

Ambos são compatíveis com a mensuração por SPP e ambos podem coexistir na mesma história.

- a) **O Formato Orientado a Regras** normalmente apresenta uma lista de regras/condições, declarações verdadeiras que definem o comportamento do sistema. O formato orientado a regras é mais sintético e comum em contextos contratuais.

Para exemplificar, seguem critérios de aceitação no formato de regras de negócio, usando como exemplo a história do usuário apresentada no item 15.5.1:

- História do usuário: Como Técnico do Programa, quero incluir o endereço de um beneficiário na tela “Detalhes do Beneficiário”, para enviar correspondência para o beneficiário.
- Critérios de Aceitação:
  - O sistema deve permitir o cadastro de apenas um endereço principal por beneficiário.
  - O sistema deve permitir o cadastro de um endereço temporário/recado (opcional), desde que haja um principal cadastrado.
  - Um endereço válido no sistema deve ser composto, no mínimo, por: CEP, Logradouro, Número, Bairro, Cidade, UF e País (padrão Brasil).
  - O CEP informado no cadastro do beneficiário deve ser validado quanto ao formato (8 dígitos numéricos) e, preferencialmente, integrado a uma base de endereços (*Web Service* dos Correios ou similar) para preenchimento automático dos campos (Logradouro, Bairro, Cidade, UF), reduzindo erros de digitação.
  - Caso o CEP informado não seja encontrado na base externa, o sistema deve permitir a digitação manual dos campos.
  - O sistema deve impedir o salvamento caso o CEP esteja em formato inválido.
  - Após salvar o endereço, a informação deve permanecer registrada e visível na tela “Detalhes do Beneficiário”.
  - O sistema deve exibir mensagem de erro quando campos obrigatórios não forem preenchidos.

- Alterações no endereço do beneficiário devem ser versionadas (auditadas). O sistema deve armazenar quem (técnico) alterou, quando e qual era o endereço anterior.
- O salvamento do endereço deve ocorrer em até 2 segundos.
- As informações devem ser armazenadas conforme as regras de segurança e proteção de dados vigentes

b) **O Formato Orientado a Cenários** comumente utiliza a sintaxe *Gherkin*, que segue o formato “Dado/Quando/Então (*Given-When-Then*)” para estruturar textos simples, em linguagem natural, que descrevem o cenário, a ação do usuário e o resultado esperado da ação.

Trata-se de exemplos concretos do uso (cenários) e que melhoram a clareza e testabilidade:

- **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?

Esse formato é mais robusto e favorece a clareza comportamental, a testabilidade, automação de testes e a redução de ambiguidades interpretativas.

Utilizando como exemplo a história de usuário apresentada no item anterior, existem vários possíveis cenários a exemplo de:

- Cenário 1: Inclusão de endereço com sucesso
- Cenário 2: CEP não encontrado (cadastro manual)
- Cenário 3: Validação de campos obrigatórios
- Cenário 4: Duplicidade de endereço principal
- Cenário 5: CEP com formato inválido
- Cenário 6: UF inválida (prevenção via lista de UFs)

A seguir, vamos detalhar apenas os cenários 1 e 2 utilizando a sintaxe *Gherkin*:

- Cenário 1: Inclusão de endereço com sucesso
  - **Dado que** o Técnico do Programa acessou a tela "Detalhes do Beneficiário" para o beneficiário "João da Silva" E que o beneficiário não possui endereço cadastrado
  - **Quando** o técnico clica no botão "Incluir Endereço"

- E preenche o campo CEP com um CEP Válido
- E o sistema preenche automaticamente Logradouro, Bairro, Cidade e UF
- E o técnico informa o Número e Complemento
- E o técnico clica em "Salvar"
- **Então** o sistema deve exibir a mensagem "Endereço cadastrado com sucesso"
- E o endereço completo deve ser exibido na seção de endereços da tela de detalhes
- E o sistema deve registrar no log de auditoria: "Endereço incluído por técnico X em [data/hora]".
- Cenário 2: CEP não encontrado (cadastro manual)
  - **Dado** que o técnico está incluindo um novo endereço
  - **Quando** ele informa um CEP "99999-999" (inexistente)
  - E o sistema não encontra o endereço automaticamente
  - **Então** o sistema deve permitir a digitação manual de todos os campos (Logradouro, Bairro, Cidade, UF)
  - E deve exibir um aviso: "CEP não encontrado. Preencha os campos manualmente e revise antes de salvar."

#### 15.5.4.d. **Checklist para validação dos critérios de aceitação**

A seguir apresenta-se uma sugestão de *Checklist* a ser observada antes da estimativa por *Story Points* Padronizados (SPP). Cumpre observar que a ausência ou inadequação dos critérios de aceitação compromete diretamente:

- a identificação dos fatores técnicos;
- a consistência da estimativa;
- a rastreabilidade da pontuação atribuída;
- a confiabilidade da mensuração por SPP.

Nesse sentido, a estimativa por SPP não deve ser realizada caso qualquer item obrigatório não seja atendido:

- Se **qualquer item obrigatório** (☑) não for atendido: a história deve **retornar para refinamento**
- Se **itens condicionais** (⚠) forem aplicáveis e não estiverem descritos: A história deve **retornar para refinamento**
- Somente após a validação completa: A história está apta para estimativa por SPP

Dessa forma, sugere-se que cada história seja submetida à verificação dos critérios de aceitação por meio do *Checklist* de referência apresentado na tabela abaixo:

**Tabela 48 - Checklist de validação dos critérios de aceitação da história do usuário**

<b>Item</b>	<b>Verificação</b>	<b>Obrigatório</b>
1	Os critérios de aceitação estão descritos de forma clara e sem ambiguidades	<input checked="" type="checkbox"/>
2	Os critérios são objetivos e não dependem de interpretação subjetiva	<input checked="" type="checkbox"/>
3	Todos os critérios permitem validação binária (atende / não atende)	<input checked="" type="checkbox"/>
4	Não existem critérios genéricos (ex: “funcionar corretamente”, “ser intuitivo”)	<input checked="" type="checkbox"/>
5	Os critérios cobrem o comportamento principal da funcionalidade	<input checked="" type="checkbox"/>
6	Foram considerados cenários alternativos e exceções relevantes	<input checked="" type="checkbox"/>
7	Estão explicitadas as validações de dados e regras de negócio	<input checked="" type="checkbox"/>
8	Estão identificadas integrações com sistemas externos, quando aplicável	<input type="checkbox"/>
9	Estão definidos requisitos não funcionais relevantes (ex: desempenho, segurança)	<input type="checkbox"/>
10	Estão definidos requisitos de segurança, auditoria/rastreabilidade, quando aplicável	<input type="checkbox"/>
11	Os critérios estão alinhados com o valor de negócio da história	<input checked="" type="checkbox"/>
12	Não há dependência de informações implícitas ou não documentadas	<input checked="" type="checkbox"/>

### 15.5.4.e. Riscos de medição decorrentes de critérios de aceitação inadequados

Como já foi mencionado, a ausência de critérios de aceitação claros, objetivos e verificáveis compromete a segurança técnica da mensuração por SPP e pode gerar distorções na execução contratual.

Os riscos apresentados a seguir devem ser tratados preventivamente durante o refinamento *do backlog* e mitigados antes da estimativa por SPP.

**Tabela 49 - Riscos decorrentes de critérios de aceitação inadequados**

Riscos	Descrição Técnica	Impacto na mensuração por SPP	Consequências	Possíveis medidas de tratamento dos riscos
<b>Critérios genéricos ou vagos</b>	Uso de expressões como “sistema deve funcionar corretamente” ou “deve ser intuitivo”, sem parâmetros objetivos.	Impossibilidade de validação objetiva do aceite.	Contestações sobre entrega concluída e divergência na medição.	Revisar os critérios de aceitação utilizando linguagem objetiva e mensurável; substituir termos subjetivos por condições verificáveis; aplicar <i>Checklist</i> de validação antes da estimativa (item 15.5.4.d).
<b>Critérios incompletos</b>	Regras de negócio e/ou validações essenciais não documentadas.	Subestimação do esforço real na estimativa inicial.	Solicitação posterior de ajuste de escopo ou reestimativa; necessidade de fracionamento posterior.	Realizar refinamento adicional da história; validar critérios de aceitação com PO e stakeholders; garantir cobertura de regras de negócio, validações e cenários relevantes antes da estimativa.
<b>Critérios excessivamente amplos</b>	Inclusão de múltiplas funcionalidades dentro de um único critério.	Superestimação ou distorção da complexidade.	Solicitação posterior de ajuste de escopo ou reestimativa; necessidade de fracionamento posterior.	Decompor a história em unidades menores; revisar granularidade e separar funcionalidades distintas.
<b>Critérios alterados após estimativa</b>	Inclusão de novas validações ou integrações durante o desenvolvimento.	Aumento não previsto do esforço necessário.	Mudança de escopo e necessidade de revisão formal.	Formalizar a alteração como mudança de escopo; registrar versão dos critérios; realizar reestimativa quando aplicável; garantir rastreabilidade entre versões.
<b>Ausência de critérios não funcionais relevantes</b>	Falta de levantamento de requisitos não funcionais: de desempenho, segurança, auditoria, integrações, etc.	Complexidade não considerada na estimativa.	Entrega tecnicamente incompleta ou disputas sobre qualidade.	Incluir explicitamente requisitos não funcionais (desempenho, segurança, auditoria, etc.); utilizar <i>Checklist</i> de validação dos critérios de aceitação para garantir cobertura desses aspectos (item 15.5.4.d).

<b>Critérios não testáveis</b>	Critérios sem possibilidade de validação binária (atende/não atende).	Subjetividade no aceite.	Fragilidade na homologação e risco de questionamentos contratuais.	Reescrever critérios de aceitação para permitir validação binária (atende/não atende); eliminar termos subjetivos; definir parâmetros claros de verificação.
<b>Dependência externa não explicitada</b>	Integrações ou serviços externos não previstos nos critérios de aceitação.	Subavaliação da incerteza e do risco técnico.	Atrasos, pedidos de revisão da pontuação.	Identificar e explicitar integrações e dependências externas nos critérios de aceitação; definir comportamento em caso de falha; considerar riscos técnicos na estimativa.
<b>Múltiplos objetivos em uma única história</b>	História com escopo difuso e critérios heterogêneos.	Dificuldade de estimativa e mensuração.	Necessidade de desmembramento tardio e recontagem de pontos.	Reestruturar a história para garantir foco em um único objetivo de negócio; revisar critérios de aceitação para evitar sobreposição de escopo.

### 15.5.5. Interface de usuário (ou UI, do inglês *User Interface*)

Uma interface de usuário pode ser representada de diferentes formas, variando conforme o nível de detalhamento desejado:

- **Baixa fidelidade:** esboços simples (*sketches*);
- **Média fidelidade:** *wireframes*;
- **Alta fidelidade:** *mockups* ou protótipos interativos.

A forma de representação deve ser definida pelo time, considerando o contexto do projeto, o nível de incerteza e o esforço disponível. O mais importante é que haja alinhamento entre os envolvidos quanto ao entendimento da solução a ser construída.

O time pode estabelecer critérios para considerar uma história de usuário “preparada” também sob a perspectiva da interface, especialmente quando a experiência do usuário é relevante para a funcionalidade.

Ressalta-se que nem todo item de backlog ou história de usuário requer representação de interface. No entanto, quando houver interação com o usuário, recomenda-se que a interface seja representada e vinculada à história correspondente.

A representação da interface deve ser tratada como artefato de apoio, podendo contribuir para:

- refinamento da história de usuário;
- validação funcional com *stakeholders*;
- melhor compreensão do escopo;
- identificação de regras de negócio e exceções;
- apoio à estimativa de complexidade;
- suporte à mensuração por SPP.

É importante destacar que a interface não substitui os critérios de aceitação, sendo complementar a eles. A figura a seguir apresenta um exemplo de interface de baixa fidelidade (esboço simples) associada à história de usuário “*Incluir Endereço do Beneficiário*”.

**Figura 26 - UI de baixa fidelidade**

The image shows a wireframe for a form titled "Incluir Endereço do Beneficiário". It includes a note that the asterisked fields are mandatory. The form contains the following elements: a CEP field with a "Buscar" button; a checked checkbox for "Endereço carregado automaticamente"; a "Logradouro\*" field; "Complemento" and "Bairro\*" fields; "Cidade\*" (with "São Paulo" selected) and "UF\*" (with "SP" selected) dropdown menus; two checkboxes for "Este é o endereço principal" (checked) and "Este é um endereço temporário/recado" (unchecked); a "Salvar" button; and a footer note "Todas as alterações serão salvas.".

### 15.5.6. Tarefas (tasks)

As tarefas representam unidades menores de trabalho associadas a uma história de usuário, correspondendo às atividades necessárias para sua implementação.

Cada tarefa descreve como a solução será construída, detalhando aspectos técnicos da execução. Por esse motivo, são utilizadas principalmente pelo time de desenvolvimento para organização e acompanhamento do trabalho.

As tarefas possuem as seguintes características:

- são descritas em linguagem técnica;
- detalham a forma de implementação da solução;
- não seguem, necessariamente, um formato textual padronizado;
- podem apresentar dependências entre si;
- não representam, isoladamente, valor direto para o usuário.

Ao definir as tarefas, o time de desenvolvimento aprofunda o entendimento técnico da história de usuário, decompondo-a em atividades executáveis. A seguir, alguns exemplos de tarefas:

- Criar interface (UI) para cadastro de endereço do beneficiário;
- Criar dados de teste para validação do endereço;

- Automatizar *scripts* de geração de dados;
- Implementar validação de CEP no *backend*;
- Configurar integração com serviço externo.

Para fins de mensuração por *Story Points* Padronizados (SPP), apenas entregas funcionais são consideradas na contagem. A entrega funcional é percebida pelo usuário, pode ser validada por meio de critérios de aceitação e gera valor direto para o negócio. Nesse contexto:

- **Histórias de usuário** são unidades válidas para contagem em SPP;
- **Tarefas técnicas** são consideradas atividades-meio e não devem ser pontuadas.

A seguir, exemplos de atividades técnicas que não entram na contagem por SPP:

- refatoração sem alteração funcional;
- ajustes de infraestrutura ou *pipeline*;
- melhorias internas de código ou arquitetura;
- implementação de observabilidade interna;
- preparação de ambiente técnico.

A decomposição em tarefas não altera a pontuação da história em SPP, sendo utilizada normalmente para organização interna do trabalho. Nesse sentido, a inclusão de tarefas técnicas na contagem não é permitida pois compromete a objetividade da métrica SPP e gera distorções na mensuração.



**Para mensuração por *Story Points* Padronizados (SPP), apenas entregas funcionais são consideradas na contagem.** Nesse contexto:

- **Histórias de usuário** são unidades válidas para contagem em SPP;
- **Tarefas técnicas** são consideradas atividades-meio e **não devem ser pontuadas.**

### 15.5.7. Definição de Preparado

Antes de uma história de usuário ser selecionada para uma iteração, ela deve ser avaliada em relação a um conjunto de critérios que indiquem se está apta a ser desenvolvida, testada e entregue dentro de uma *sprint*. A inclusão de itens de backlog (PBIs) não preparados no backlog da *sprint* compromete a previsibilidade do trabalho, podendo gerar retrabalho, atrasos e perda de qualidade nas entregas.

A **Definição de Preparado** (*Definition of Ready – DoR*), assim como a Definição de Pronto (*Definition of Done – DoD*), é estabelecida antes do início do desenvolvimento do produto, geralmente nas fases iniciais do projeto. No entanto, deve ser tratada como um artefato evolutivo, passível de revisão e aprimoramento contínuo ao longo do tempo.

A Definição de Preparado é, em geral, estruturada como uma lista de critérios que um item do backlog deve atender para ser considerado apto à implementação, conforme exemplo a seguir:

- Deve existir compreensão compartilhada do item entre o *Product Owner* (PO) e o time de desenvolvimento;
- Os critérios de aceitação devem estar definidos, discutidos e acordados;
- O item deve estar claro, sem ambiguidades e com escopo compreensível;
- O item deve ter sido estimado;
- O item deve possuir tamanho adequado, compatível com a capacidade de entrega dentro de uma *sprint*;
- Dependências relevantes (técnicas ou externas) devem estar identificadas e compreendidas.

Outros critérios podem ser adicionados conforme o contexto do projeto. A Definição de Preparado deve ser continuamente revisada e ajustada para refletir a maturidade do time e as necessidades do produto. Assim como a Definição de Pronto, a Definição de Preparado deve ser:

- compreendida por todos os membros do time;
- acordada de forma colaborativa;
- mantida visível e acessível;
- aplicada de forma consistente ao longo do projeto.

## 15.6. INCREMENTO DO PRODUTO

O incremento do produto é um pedaço do produto que será entregue ao final da *sprint*. 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.

### 15.6.1. Definição de Pronto

A **Definição de Pronto** (*Definition of Done – DoD*) é a descrição formal do estado que um incremento deve alcançar para ser considerado concluído, atendendo aos padrões de qualidade estabelecidos para o produto. Quando um item do *backlog* do produto satisfaz integralmente a Definição de Pronto, ele passa a compor um incremento potencialmente utilizável.

A Definição de Pronto promove transparência e assegura entendimento compartilhado sobre o que significa uma entrega concluída. Todos os membros do time ágil devem conhecer, compreender e aplicar consistentemente essa definição.

Itens do backlog que não atendem à Definição de Pronto:

- não devem ser considerados concluídos;
- não devem ser apresentados como entrega final na Revisão da *Sprint*;
- devem retornar ao *backlog* do produto para ajustes e replanejamento.

### 15.6.2. Exemplos de Definição de Pronto

Os critérios abaixo representam exemplos que podem ser adaptados conforme o contexto 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 funcional 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, conseqüentemente, 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) Realização de testes funcionais nos principais fluxos da funcionalidade entregue;
- d) Validação do comportamento do sistema conforme os critérios de aceitação definidos para a história.

## 15.7. CONTAGEM POR STORY POINTS PADRONIZADOS (SPP)

Como já mencionado, o *Story Points* Padronizados (SPP) é uma métrica estruturada que permite estimar o esforço a partir da complexidade funcional observável, diferenciando-se das estimativas baseadas em tempo.

A métrica SPP não deve ser utilizada exclusivamente para fins de pagamento. Esse roteiro recomenda a sua utilização como instrumento de apoio ao planejamento e à estimativa das *Sprints*, bem como como indicador para o acompanhamento da produtividade mínima esperada dos times de desenvolvimento.

Sua utilização deve estar associada à mensuração de entregas funcionais, de modo a assegurar que a métrica reflita o valor efetivamente entregue, e não apenas o esforço técnico interno. Essa abordagem contribui para maior transparência, comparabilidade e confiabilidade na medição de produtividade.

O SPP possibilita a mensuração padronizada do esforço e dos fatores técnicos das histórias de usuário, tanto no contexto de um time quanto em nível organizacional, eliminando a dependência de percepções individuais. Nesse contexto, a sequência de Fibonacci é utilizada apenas como faixa de normalização, não sendo empregada como base para a estimativa.

### Por que usar a métrica Story Points Padronizados (SPP)?

O SPP é uma métrica estruturada baseada na atribuição de pontuação a histórias de usuário a partir de critérios objetivos e verificáveis.

Pode ser utilizada pela organização como instrumento de apoio ao planejamento, à estimativa das *sprints*, bem como indicador para acompanhamento e análise da produtividade esperada dos times de desenvolvimento.

A adoção da métrica SPP oferece as seguintes vantagens:

- É simples e de fácil aplicação;
- Permite estimativas em poucos minutos;
- Dispensa debates abstratos ou excessivamente subjetivos;
- Não depende de comparação relativa entre histórias;
- Mantém lógica semelhante à métrica de story points tradicional, amplamente utilizada por times ágeis;
- Reduz a variação de estimativas entre equipes;
- Possibilita rastreabilidade e auditabilidade das estimativas.

Nas seções seguintes, são apresentados os passos para a contagem por SPP, evidenciando os principais benefícios de sua aplicação no contexto organizacional.

### 15.7.1. Entradas para o SPP

À medida que os requisitos são refinados em partes menores o time passa a compreender melhor o que precisa ser desenvolvido. Esse refinamento facilita o processo de estimativa e, como consequência, aumenta a precisão das estimativas realizadas.

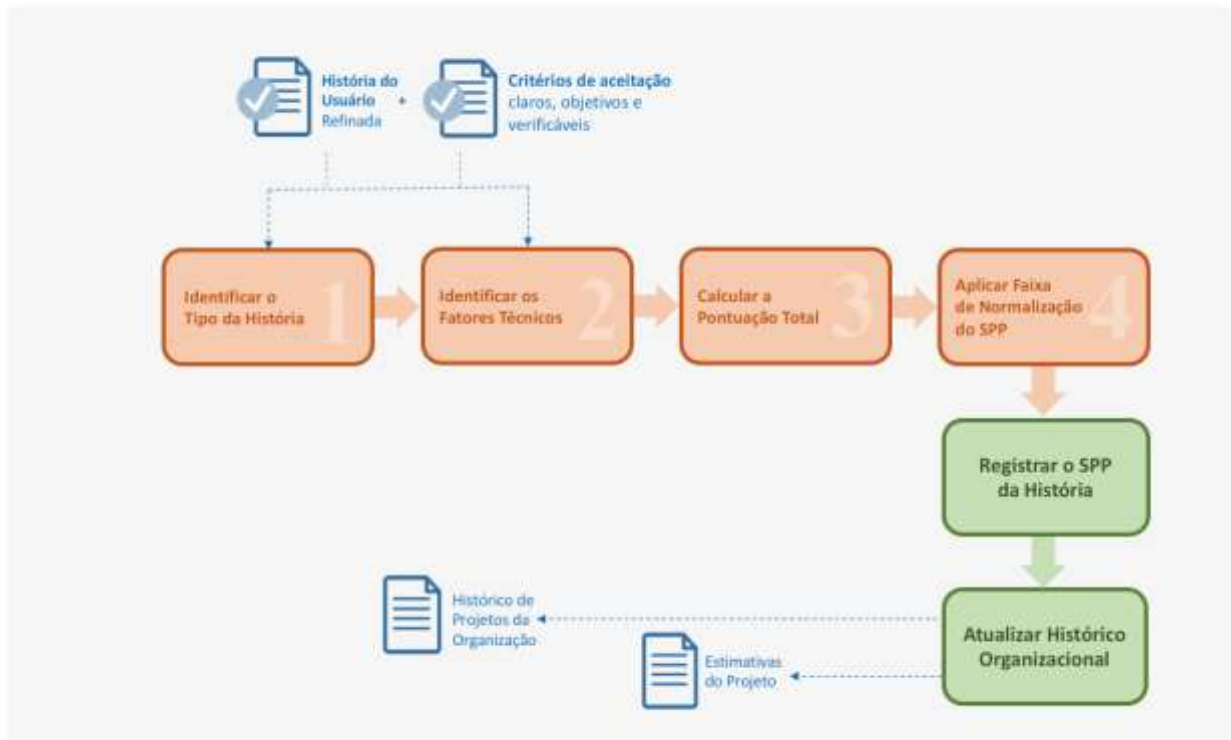
Nesse contexto, para aplicar o SPP, o ponto de partida é sempre uma história de usuário bem escrita e seus respectivos critérios de aceitação. Os critérios de aceitação são essenciais porque detalham o comportamento esperado da funcionalidade. Por meio deles é possível identificar não apenas os requisitos funcionais, mas também regras de negócio, validações, exceções, integrações, restrições técnicas e requisitos não funcionais.

Esses elementos ajudam o time a compreender melhor o esforço e a complexidade envolvidos no desenvolvimento. Muitas vezes, o texto da história de usuário, mesmo sendo claro e bem escrito, apresenta apenas uma visão resumida da necessidade do usuário. Já os critérios de aceitação revelam detalhes que impactam diretamente no trabalho de desenvolvimento.

Por esse motivo, na métrica SPP, a análise da história de usuário e de seus critérios de aceitação constitui a entrada do processo de estimativa, pois é a partir dessas informações que serão identificados o Tipo da História e os Fatores técnicos. Esses dois elementos formam a base para o cálculo padronizado do SPP.

O diagrama a seguir apresenta o fluxo de aplicação da métrica Story Points Padronizados (SPP) e nos itens seguintes, são detalhados os passos para a contagem com SPP.

Figura 27 - Fluxo de contagem da métrica SPP



Observe que a história de usuário e seus critérios de aceitação (ACs) constituem a principal fonte de informação para identificação do tipo da história e dos fatores técnicos.

Após identificação do tipo da história do usuário (passo 1) e identificação dos seus fatores técnicos (passo 2), realiza-se a soma das pontuações obtidas (passo 3). Depois, aplica-se uma faixa de normalização (passo 4), resultando no valor final em *Story Points* Padronizados (SPP).

O valor calculado deve ser registrado no *backlog* e incorporado ao histórico organizacional, permitindo análise de produtividade e comparabilidade entre *sprints* e entre diferentes times da organização.

### 15.7.2. Passos para a contagem por SPP

A contagem por SPP é realizada por meio de um processo estruturado, baseado exclusivamente em critérios objetivos extraídos da história de usuário e de seus critérios de aceitação.

A estimativa NÃO deve ser baseada em percepção, experiência individual ou comparação com outras histórias. O processo é composto por quatro passos:

- Passo 1: Definir o Tipo Base da História
- Passo 2: Identificar os Fatores Técnicos
- Passo 3: Calcular a Pontuação Total
- Passo 4: Aplicar a Faixa de Normalização

Nesse método, a equipe não "discute Fibonacci"; marcam-se os fatores presentes, soma a pontuação e contabiliza em *Story Points* Padronizados (SPP).

O órgão deve adotar uma escala reduzida e padronizada de pontuação em SPP, definindo previamente um valor máximo (teto) a ser utilizado. Recomenda-se a utilização da sequência: 1, 2, 3, 5, 8, 13 e 21, sendo 21 o valor máximo permitido. Para assegurar a padronização da métrica, não se deve utilizar valores intermediários ou criar novas faixas de pontuação fora da escala definida.

O número superior da série (13 ou 21) representa uma história complexa que requer maior esforço. Nesses casos, o time deve analisar se a história pode ser dividida ainda mais.

### 15.7.2.a. Passo 1: Definir o Tipo Base da História

A seguir, listam-se as regras para definir o tipo base da história

- Deve ser escolhido **apenas um tipo**, mesmo que a história envolva múltiplos aspectos
- Deve ser considerado o **aspecto predominante** da funcionalidade
- Integração externa **sempre prevalece sobre os demais tipos**

**Tabela 50 - Tipos base de uma história do usuário**

<b>Tipo Base da História</b>	<b>Descrição</b>	<b>Pontuação Base</b>
<b>1</b> Consulta simples	Apenas leitura de dados, sem persistência, sem regra de negócio além de exibição	1
<b>2</b> Entrada ou alteração de dados	Criação, edição ou exclusão de dados com persistência	2
<b>3</b> Processamento com regras	Aplicação de regras de negócio que alteram comportamento, cálculos ou validações condicionais	3

---

4	Integração / comunicação externa	Comunicação com sistema externo (envio, recebimento ou consulta de dados)	4
---	----------------------------------	---	---

---

### 15.7.2.b. Passo 2: Identificar os Fatores Técnicos

Uma regra fundamental da contagem por SPP é que os fatores técnicos devem ser considerados com base em evidências explícitas nos critérios de aceitação.

Convém destacar que no contexto do SPP, a participação do time na estimativa é recomendada como boa prática para promover alinhamento e validação coletiva, entretanto, não constitui requisito obrigatório, uma vez que a pontuação decorre da identificação de fatores técnicos objetivos, explicitamente definidos nos critérios de aceitação, permitindo a aplicação consistente da métrica independentemente da percepção individual dos membros do time.

Os fatores técnicos apresentados neste roteiro foram organizados em 5 blocos (A,B,C,D e E) devem ser adotados como referência para a aplicação da métrica SPP.

Caso o órgão identifique a necessidade de adequação, eventuais novos fatores deverão estar previamente definidos no Termo de Referência ou padrão técnico equivalente, com o respectivo registro da memória de cálculo. As adequações devem preservar os princípios de objetividade, rastreabilidade e comparabilidade da métrica.

O passo 2 da contagem por SPP consiste na identificação dos fatores técnicos aplicáveis à história em análise. A identificação dos fatores técnicos no SPP deve seguir regras claras e objetivas, garantindo consistência, rastreabilidade e redução de subjetividade na mensuração.

Para a correta aplicação da métrica, devem ser observadas as seguintes diretrizes na identificação dos fatores técnicos da história analisada:

- Os fatores técnicos devem ser considerados com base em evidências explícitas nos critérios de aceitação ou na documentação da história;
- Cada fator técnico corresponde a +1 ponto;
- Um fator somente pode ser marcado quando estiver claramente descrito;
- Não é permitida a inferência, suposição ou interpretação implícita de complexidade;

- Uma história de usuário pode apresentar zero, um ou múltiplos fatores técnicos em cada bloco;
- Cada fator deve ser contabilizado apenas uma vez por história de usuário, independentemente da quantidade de ocorrências.

A aplicação dessas regras assegura maior objetividade na contagem, além de permitir que diferentes avaliadores obtenham resultados consistentes a partir das mesmas informações.

**Tabela 51 - Fatores Técnicos de referência para a contagem por SPP**

<b>Fatores técnicos</b>	
<b>BLOCO A - Escopo Estrutural</b> (+ 1 ponto para cada fator)	A1. Implementação que envolva mais de uma funcionalidade distinta ou mais de um módulo do sistema, conforme descrito nos critérios de aceitação ou escopo da história
	A2. Implementação que envolva, no mínimo, duas camadas tecnológicas distintas (ex: <i>frontend + backend</i> , <i>backend + banco de dados</i> ), com atuação explícita em ambas as camadas
	A3. Necessidade de alteração, integração ou adaptação de código legado identificada na história ou nos critérios de aceitação
<b>BLOCO B – Dados e Regras</b> (+ 1 ponto para cada fator)	B1. Existência de dez (10) ou mais campos ou atributos distintos de entrada e/ou saída definidos nos critérios de aceitação
	B2. Existência de cinco (5) ou mais validações descritas nos critérios de aceitação
	B3. Existência de três (3) ou mais regras de negócio independentes, descritas nos critérios de aceitação
	B4. Existência de cálculos com múltiplas variáveis ou dependência entre etapas de cálculo (ex: fórmulas encadeadas), devidamente especificado
<b>BLOCO C – Tecnologia</b> (+ 1 ponto para cada fator)	C1. Uso obrigatório de tecnologia, ferramenta ou recurso técnico emergente (ex: IA, <i>blockchain</i> , etc.) não previamente utilizado pelo time ou projeto e definido nos critérios de aceitação
	C2. Implementação obrigatória para mais de um tipo de interface ou dispositivo (ex: <i>web e mobile</i> ), conforme descrito na história do usuário e/ou critérios de aceitação
	C3. Necessidade de desenvolvimento, testes ou adaptação para múltiplas plataformas tecnológicas distintas (ex: sistemas operacionais, navegadores ou ambientes), formalmente definida

	D1. Consulta a sistema externo, prevista nos critérios de aceitação
<b>BLOCO D – Integrações externas</b> <b>(+ 1 pontos para cada fator)</b>	D2. Envio de dados para sistema externo, previsto na história do usuário e/ou critérios de aceitação
	D3. Recebimento e/ou processamento de dados provenientes de sistema externo, conforme especificado
	D4. Dependência de disponibilidade de sistema externo para execução da funcionalidade, devidamente indicada
<b>BLOCO E – Restrições não funcionais</b> <b>(+ 1 ponto para cada fator)</b>	E1. Requisitos de segurança definidos nos critérios de aceitação, que exijam implementação de mecanismos como autenticação, autorização, criptografia ou registro de auditoria
	E2. Atendimento obrigatório a norma, lei ou regulamento expressamente citado (ex: LGPD, Bacen, eSocial), que imponha regras adicionais de implementação
	E3. Restrições organizacionais formalmente definidas, como exigência de ambiente, padrões institucionais obrigatórios, infraestrutura específica ou hardware-alvo
	E4. Requisitos de desempenho com critérios mensuráveis objetivamente definidos (ex: tempo máximo de resposta, volume de processamento, <i>throughput</i> mínimo)
	E5. Restrição técnica não prevista na pipeline padrão de implementação e claramente definida nos critérios de aceitação.



Os fatores apresentados na Tabela 51 - Fatores Técnicos de referência para a contagem por SPP devem ser adotados como referência para a aplicação da métrica SPP.

**Caso o órgão identifique a necessidade de adequação, eventuais novos fatores técnicos deverão estar previamente definidos no Termo de Referência ou padrão técnico equivalente, com o respectivo registro da memória de cálculo.** As adequações devem preservar os princípios de objetividade, rastreabilidade e comparabilidade da métrica.

### 15.7.2.c. Passo 3: Somar pontuação

O terceiro passo consiste em somar as pontuações obtidas nos passos 1 e 2 (Pontuação de Base + Soma dos Fatores técnicos marcados):



$$\text{Total} = \text{Tipo Base} + \sum(\text{Fatores Técnicos})$$



Onde:

- **Tipo Base:** corresponde ao tipo base da história do usuário que está sendo medida (ver item 15.7.2.a).
- $\sum$  (**Fatores Técnicos**): corresponde a soma dos fatores técnicos (estruturados em 5 blocos: A, B, C, D, E) da história do usuário (ver item 15.7.2.b).

### 15.7.2.d. Passo 4: Aplicar faixa final automática

Para contabilizar o SPP, manter padronização e evitar pontuações infladas, a pontuação total deve ser convertida conforme tabela abaixo:

Tabela 52 - Faixa de normalização SPP

Pontuação calculada	SPP Final	Complexidade
1-2	1	Baixa
3-4	3	Baixa
5-6	5	Média
7-9	8	Média
10-12	13	Alta
>12	21	Alta

Para esse passo, aplicam-se as seguintes regras:

- A conversão é automática e obrigatória.
- Não é permitido ajuste manual.
- Não é permitido arredondamento fora da tabela.
- Não é permitida a criação de multiplicadores.
- Regra de corte: se a pontuação da história for superior a 12, a história deve ser avaliada para possível divisão e refinamento.



**O órgão deve adotar uma escala reduzida e padronizada de pontuação em SPP, definindo previamente um valor máximo (teto) a ser utilizado.** Recomenda-se a utilização da sequência: 1, 2, 3, 5, 8, 13 e 21, sendo 21 o valor máximo permitido.

Não é permitida a utilização de valores intermediários, multiplicadores ou a criação de novas faixas de pontuação fora da escala definida.

### 15.7.3. Exemplos de aplicação da métrica SPP

Os exemplos a seguir têm finalidade pedagógica e demonstram a aplicação objetiva da métrica SPP em histórias com diferentes níveis de complexidade.

Cumpra observar que em todos os três exemplos, a pontuação foi atribuída apenas com base em evidências explícitas nos critérios de aceitação das histórias do usuário.

Como já foi mencionado, um fator só pode ser marcado se estiver claramente descrito. Não é permitido inferir ou assumir complexidade não documentada. Observe nos exemplos a seguir que a variação da pontuação não decorre de percepção subjetiva do time, mas da presença objetiva de atributos funcionais e fatores técnicos previamente definidos na métrica.

### 15.7.3.a. Exemplo 1: Aplicação do SPP para história de baixa complexidade

A seguir apresenta-se a aplicação do método SPP para uma história simples de consulta, sem evidência de integração, sem volume elevado de campos e sem regras complexas.

- História do usuário: **Como** atendente, **quero** consultar os dados cadastrais do beneficiário **para** visualizar suas informações básicas.
- Critérios de aceitação:
  - O sistema deve permitir a busca do beneficiário por CPF.
  - O sistema deve exibir nome, CPF, data de nascimento e situação cadastral do beneficiário.
  - O sistema deve apresentar mensagem quando o beneficiário não for encontrado.
  - A consulta não deve permitir alteração de dados.

**Tabela 53 - Aplicação do SPP para história de baixa complexidade**

Etapa	Elemento avaliado	Evidência objetiva	Pontuação
<b>Passo 1</b>	Tipo da História	Consulta simples, apenas leitura de dados	1
<b>Passo 2 – Bloco B</b>	B2 – Validações	Há somente uma (1) validação de existência do beneficiário.	+0
<b>Passo 2 – Demais blocos</b>	Não aplicável	Não há integração externa, múltiplas camadas explícitas, requisitos não funcionais relevantes ou regras extensas	+0
<b>Passo 3</b>	Soma total	1 + 0	1
<b>Passo 4</b>	Normalização	Faixa 1–2	1 SPP

### 15.7.3.b. Exemplo 2: Aplicação do SPP para história de média complexidade

Considerando os critérios de aceitação listados para a história 'Incluir endereço do beneficiário' (item 15.5.4.c):

- História do usuário: **Como** Técnico do Programa, **quero** incluir o endereço de um beneficiário na tela “Detalhes do Beneficiário”, **para** enviar correspondência para o beneficiário.
- Critérios de Aceitação:
  - O sistema deve permitir o cadastro de apenas um endereço principal por beneficiário.
  - O sistema deve permitir o cadastro de um endereço temporário/recado (opcional), desde que haja um principal cadastrado.
  - Um endereço válido no sistema deve ser composto, no mínimo, por: CEP, Logradouro, Número, Bairro, Cidade, UF e País (padrão Brasil).
  - O CEP informado no cadastro do beneficiário deve ser validado quanto ao formato (8 dígitos numéricos) e, preferencialmente, integrado a uma base de endereços (*Web Service* dos Correios ou similar) para preenchimento automático dos campos (Logradouro, Bairro, Cidade, UF), reduzindo erros de digitação.
  - Caso o CEP informado não seja encontrado na base externa, o sistema deve permitir a digitação manual dos campos.
  - O sistema deve impedir o salvamento caso o CEP esteja em formato inválido.
  - Após salvar o endereço, a informação deve permanecer registrada e visível na tela “Detalhes do Beneficiário”.
  - O sistema deve exibir mensagem de erro quando campos obrigatórios não forem preenchidos.
  - Alterações no endereço do beneficiário devem ser versionadas (auditadas). O sistema deve armazenar quem (técnico) alterou, quando e qual era o endereço anterior.
  - O salvamento do endereço deve ocorrer em até 2 segundos.
  - As informações devem ser armazenadas conforme as regras de segurança e proteção de dados vigentes

Aplica-se a seguir a contagem com o método SPP:

**Tabela 54 - Aplicação do SPP para história de média complexidade**

Etapa	Elemento Avaliado	Evidência Objetiva	Pontuação
<b>Passo 1</b>	Tipo da História	Entrada/alteração de dados (cadastro com persistência)	2
<b>Passo 2 – Bloco A</b>	A2 – Múltiplas camadas	Backend + banco de dados (persistência explícita)	+1
<b>Passo 2 – Bloco B</b>	B2 – Validações	Mais de 5 validações explícitas nos critérios	+1
<b>Passo 2 – Bloco D</b>	D1 – Consulta externa	Consulta ao serviço de CEP	+1
<b>Passo 2 – Bloco E</b>	E1 – Auditoria	Registro de alterações (log de auditoria)	+1
<b>Passo 3</b>	Soma total	2 (base) + 4 (fatores)	6
<b>Passo 4</b>	Normalização	Faixa 5-6	5 SPP

### 15.7.3.c. Exemplo 3: Aplicação do SPP para história de alta complexidade

Agora vamos com um exemplo típico de história com forte carga de regras, validações, integração e requisito não-funcional explícito:

- História do usuário: **Como** gestor do programa, **quero** que o sistema homologue automaticamente a concessão do benefício, **para** que a elegibilidade do beneficiário seja avaliada com base nas regras do programa, indicando se está apto, não apto ou pendente de análise.
- Critérios de aceitação:
  - O sistema deve verificar se os dados cadastrais, socioeconômicos e documentais do beneficiário estão preenchidos e atualizados no momento da análise.
  - Caso existam dados obrigatórios ausentes ou desatualizados, o sistema deve impedir a homologação e classificar o status como **“pendente”**, exibindo mensagem informando os campos inconsistentes.
  - O sistema deve validar o CPF com 11 dígitos numéricos válidos, a faixa etária do beneficiário, dados cadastrais preenchidos, dados cadastrais atualizados e situação cadastral ativa. Caso qualquer validação falhe, o sistema deve impedir a homologação e registrar a inconsistência.

- O sistema deve aplicar regras de elegibilidade do beneficiário, considerando, no mínimo: faixa de renda familiar, composição familiar, impedimentos legais e acúmulo indevido de benefícios para classificação do beneficiário.
- O sistema deve classificar o beneficiário como: ‘apto’, quando atender a todas as regras de elegibilidade; e como ‘não apto’, quando violar qualquer regra de elegibilidade.
- O sistema deve consultar sistema externo para verificar a existência de benefício incompatível.
- Caso o sistema externo esteja indisponível: o sistema deve classificar a análise como “pendente” e deve registrar a indisponibilidade para auditoria.
- O sistema deve registrar o resultado da análise contendo: o nome e o CPF do beneficiário, o status final (aprovado, rejeitado ou pendente), a data e a hora da análise; a regras de elegibilidade aplicadas e a justificativa da decisão.
- O sistema deve registrar log de auditoria contendo, no mínimo: o usuário responsável pela operação, a data e a hora; a ação executada e as regras avaliadas.
- O sistema deve processar a análise em até 2 segundos por solicitação.

Aplicando o método de contagem por SPP temos:

**Tabela 55 - Aplicação do SPP para história de alta complexidade**

<b>Etapa</b>	<b>Elemento avaliado</b>	<b>Evidência objetiva</b>	<b>Pontuação</b>
<b>Passo 1</b>	Tipo da História	Processamento com regras de negócio predominante	3
<b>Passo 2 – Bloco A</b>	A1 – Envolve a análise de múltiplas funcionalidades	Cadastro, análise e decisão funcional	+1
<b>Passo 2 – Bloco A</b>	A2 – Múltiplas camadas	Aplicação, processamento e persistência	+1
<b>Passo 2 – Bloco B</b>	B1 – Campos	Mais de 10 campos/atributos analisados	+1
<b>Passo 2 – Bloco B</b>	B2 – Validações	Mais de 5 validações explícitas	+1
<b>Passo 2 – Bloco B</b>	B3 – Regras de negócio independentes	Mais de 3 regras independentes de elegibilidade	+1
<b>Passo 2 – Bloco D</b>	D1 – Consulta externa	Verificação em sistema externo	+1
<b>Passo 2 – Bloco D</b>	D4 - Dependência de disponibilidade	Dependência de disponibilidade de sistema externo para execução da funcionalidade, devidamente indicada	+1

<b>Passo 2 – Bloco E</b>	E1 – Segurança/ auditoria	Log de auditoria descrito	+1
<b>Passo 2 – Bloco E</b>	E4 – Desempenho	Tempo máximo de resposta explicitamente definido	+1
<b>Passo 3</b>	Soma total	3 (base) + 9	12
<b>Passo 4</b>	Normalização	Faixa 10–12	13 SPP

### 15.7.4. Tratamento de Melhorias

As melhorias evolutivas de funcionalidades existentes devem ser estimadas utilizando o mesmo procedimento aplicado às demais histórias de usuário no SPP. Ou seja, não há uma forma diferenciada de contagem para melhorias.

A estimativa por SPP deve sempre considerar o tipo da história, de acordo com a natureza da funcionalidade implementada, e os fatores técnicos, identificados a partir dos critérios de aceitação.

Embora seja comum no dia a dia tratar itens como “melhoria”, “ajuste” ou “evolução”, essas classificações representam o contexto da mudança e não a complexidade técnica da implementação.

Criar uma forma diferenciada de estimar melhorias pode gerar distorções importantes na métrica:

**Tabela 56 - Riscos no tratamento de melhorias de forma diferenciada**

<b>Possíveis Riscos</b>	<b>Descrição</b>
<b>Perda de padronização</b>	Se melhorias forem estimadas de forma diferente, histórias com complexidade semelhante podem receber pontuações distintas, comprometendo a consistência do método.
<b>Redução da comparabilidade</b>	A métrica SPP permite comparar estimativas entre histórias, sprints e times. Ao introduzir uma regra específica para melhorias, essa comparabilidade é prejudicada.
<b>Subestimação do esforço</b>	Melhorias são frequentemente percebidas como “mais simples”, mas podem envolver: <ul style="list-style-type: none"> <li>• alterações em código legado;</li> <li>• impacto em múltiplos módulos;</li> <li>• novas validações e regras de negócio;</li> </ul>

- ajustes em integrações existentes.

Uma contagem diferenciada pode levar à subavaliação da complexidade real.

#### **Distorção na medição de produtividade**

Como o SPP também é utilizado para mensuração de produtividade, qualquer inconsistência na estimativa impacta diretamente os indicadores.

Se melhorias forem subestimadas:

- a produtividade aparenta ser maior do que realmente é;
- a capacidade de planejamento da equipe é prejudicada;
- decisões gerenciais podem ser tomadas com base em dados imprecisos.

Quando a melhoria ocorre em uma ou mais *sprints* da mesma *release* em que a história foi desenvolvida a organização pode optar por não estimar novamente a história e documentar as melhorias implementadas ao longo da *release*.

Contudo, caso a organização deseje manter o registro de todas as melhorias realizadas dentro e fora da *release*, a contagem com SPP da melhoria deve seguir o mesmo fluxo da contagem por SPP.

### **15.7.5. Tratamento de Correção de Defeitos com SPP**

As correções de defeitos (bugs) podem ou não ser estimadas utilizando o SPP, dependendo de sua natureza e impacto.

Nem todo defeito deve ser estimado com SPP; apenas aqueles que representam esforço significativo e impacto funcional devem compor a métrica. Utilizar SPP para estimar todos os defeitos pode:

- Inflar artificialmente a produtividade do time;
- Distorcer a percepção de entrega de valor;
- Reduzir a comparabilidade entre sprints.

A decisão deve ser documentada e deve considerar o tipo da correção:

**Tabela 57 – Tratamento de Correção de defeitos com SPP**

Tipos de Correção	Como tratar	Motivo	Exemplos
<b>Representa um ajuste pontual de baixa complexidade</b>	<b>✗ NÃO</b> estimar com SPP	Esses itens não representam entrega funcional significativa e, portanto, não devem impactar a métrica de produtividade baseada em SPP.	<ul style="list-style-type: none"> <li>ajuste de layout (ex: alinhamento de botão);</li> <li>correção de texto ou mensagem;</li> <li>correção de erro simples de validação;</li> <li>ajuste de configuração sem impacto funcional relevante.</li> </ul>
<b>Representam esforço significativo e impacto funcional</b>	<b>☑</b> Estimar com SPP	<p>Correções relevantes de defeitos devem ser estimadas com SPP quando:</p> <ul style="list-style-type: none"> <li>exigirem análise mais aprofundada;</li> <li>envolverem múltiplas partes do sistema;</li> <li>alterarem regras de negócio;</li> <li>impactarem integrações;</li> <li>demandarem esforço semelhante ao de uma nova história.</li> </ul>	<ul style="list-style-type: none"> <li>correção de cálculo incorreto de benefício;</li> <li>ajuste em regra de validação com múltiplas condições;</li> <li>correção que exige alteração em backend e frontend;</li> <li>erro em integração com sistema externo.</li> </ul>

Quando aplicável, a correção deve seguir o mesmo fluxo da contagem por SPP.

### 15.7.6. Tratamento de Atividades Técnicas na contagem com SPP

As atividades técnicas — como refatoração, melhorias de infraestrutura ou ajustes internos no sistema — devem ser avaliadas com cuidado quanto à utilização do SPP.

Como já foi mencionado, o SPP deve ser utilizado prioritariamente para mensurar entregas funcionais. As atividades técnicas devem ser consideradas apenas quando estiverem diretamente associadas a uma funcionalidade (ver item 15.5.1)

Para esses casos, a atividade técnica deve estar relacionada e incorporada dentro de uma história de usuário funcional, e o esforço técnico é considerado nos fatores técnicos.

Dentre os riscos de incluir todas as atividades técnicas em uma contagem por SPP, destacam-se:

- distorção da produtividade;
- perda de foco em entrega de valor;
- dificuldade de comparação entre times;
- mistura de esforço técnico com valor entregue.

### 15.7.7. O que contar e o que não contar com SPP

O SPP deve medir apenas entregas funcionais percebidas pelo usuário ou que impactem diretamente o comportamento do sistema:

**Tabela 58 – O que contar e o que não contar com SPP**

Tipo de item	Entra no SPP?	Como tratar	Exemplos
<b>Nova funcionalidade</b>	<input checked="" type="checkbox"/> SIM	Aplicar fluxo completo do SPP	Novo cadastro, nova consulta, nova regra de negócio
<b>Melhoria evolutiva (com impacto funcional)</b>	<input checked="" type="checkbox"/> SIM	Estimar normalmente conforme tipo da história	Inclusão de campo, nova validação, melhoria de fluxo
<b>Correção de defeito com esforço significativo e impacto funcional</b>	<input checked="" type="checkbox"/> SIM	Tratar e/ou associar a uma história e aplicar SPP	Correção de cálculo, ajuste em regra de negócio, erro em integração
<b>Correção de defeito (ajuste pontual de baixa complexidade)</b>	<input checked="" type="checkbox"/> NÃO	Tratar fora do SPP (ex: esforço operacional)	Ajuste de layout, correção de texto, erro simples
<b>Atividade técnica com impacto funcional</b>	<input checked="" type="checkbox"/> SIM	Associar/ incorporar a história do usuário	Otimização que melhora tempo de resposta percebido
<b>Refatoração sem impacto funcional</b>	<input checked="" type="checkbox"/> NÃO	Não estimar com SPP	Reorganização interna de código
<b>Infraestrutura</b>	<input checked="" type="checkbox"/> NÃO	Tratar fora do SPP	Configuração de servidor, deploy, ajustes de ambiente
<b>Atualização de bibliotecas ou frameworks</b>	<input checked="" type="checkbox"/> NÃO	Tratar fora do SPP	Upgrade de versão de framework, atualização de dependências
<b>Ajustes internos no sistema</b>	<input checked="" type="checkbox"/> NÃO	Tratar fora do SPP	Melhorias internas sem impacto no usuário

<b>Integração com sistemas externos</b>	<input checked="" type="checkbox"/> SIM	Estimar como tipo de história (integração)	Consumo de API, envio de dados
<b>Requisitos não funcionais com valor para o usuário</b>	<input checked="" type="checkbox"/> SIM	Considerar como fator técnico associado a uma história	Segurança, desempenho, disponibilidade
<b>Requisitos não funcionais sem valor para o usuário</b>	<input checked="" type="checkbox"/> NÃO	Tratar fora do SPP	Ajustes técnicos não perceptíveis
<b>Testes, homologação e documentação</b>	<input checked="" type="checkbox"/> NÃO	Já fazem parte do esforço do desenvolvimento da história	Testes, homologação e elaboração de documentação associada não devem ser estimados separadamente
<b>Atividades administrativas ou de gestão</b>	<input checked="" type="checkbox"/> NÃO	Fora do escopo da métrica	Reuniões, planejamento, alinhamentos
<b>Cerimônias do Scrum, pontos de controle diários, etc</b>	<input checked="" type="checkbox"/> NÃO	Fora do escopo da métrica	Sprint Planning (planejamento da spint), Daily (reunião diária), Sprint Review (revisão da sprint) e Sprint Retrospective (retrospectiva da sprint)

### 15.7.8. Boas práticas na estimativa por SPP

A seguir listam-se boas práticas - que devem ser observadas pelos times - ao realizar a contagem por SPP ou mesmo, *story points* tradicional :

- 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 uma pontuação máxima para as histórias (Ex: 21 é a pontuação máxima);
- Definir diretrizes em relação a pontuação das histórias e documentar. Por exemplo, o que torna uma história mais complexa para a organização ou time? 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 tenha segurança para marcar os fatores técnicos de uma história, o time deve verificar se a história foi bem compreendida por todos, se os critérios de aceitação não estão claros, se deve ser refinada, etc;
- Verificar quais histórias foram subestimadas ou superestimadas em estimativas anteriores e assim, estimar as próximas histórias com maior precisão.

### 15.7.9. 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 0) 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.

### 15.7.10. Erros a serem evitados na contagem por SPP

A aplicação da métrica *Story Points* Padronizados (SPP) deve seguir critérios objetivos e verificáveis. No entanto, alguns equívocos recorrentes podem comprometer a consistência, a comparabilidade e a confiabilidade da mensuração.

Os principais erros foram listados a seguir e agrupados nas seguintes categorias:

**Tabela 59 - Erros a serem evitados na contagem por SPP**

**Exemplos de erros a serem evitados**

<p><b>Erros relacionados ao entendimento incorreto do propósito e da natureza do SPP</b></p>	<ul style="list-style-type: none"> <li>• Utilizar SPP como métrica exclusivamente para pagamento;</li> <li>• Associar SPP diretamente a horas ou tempo de execução;</li> <li>• Confundir esforço/complexidade com valor entregue;</li> <li>• Utilizar SPP para medidas milimétricas;</li> <li>• Considerar esforço técnico interno como fator técnico funcional;</li> <li>• Utilizar prazo, urgência ou pressão do projeto como critério de pontuação.</li> </ul>
<p><b>Erros relacionados à forma de identificação e aplicação dos fatores técnicos</b></p>	<ul style="list-style-type: none"> <li>• Marcar fatores técnicos sem evidência explícita nos critérios de aceitação;</li> <li>• Inferir ou assumir complexidade com base em suposições ou conhecimento tácito;</li> <li>• Não esclarecer dúvidas sobre a história antes da estimativa;</li> <li>• Realizar a contagem com base em entendimento incompleto da funcionalidade;</li> <li>• Não analisar adequadamente a complexidade e utilizar aproximações simplificadas (ex: média);</li> <li>• Aplicar fatores de forma inconsistente entre histórias ou equipes;</li> <li>• Contabilizar o mesmo fator mais de uma vez na mesma história;</li> <li>• Aplicar a métrica sem considerar a qualidade e completude dos critérios de aceitação.</li> </ul>
<p><b>Erros relacionados à forma como a métrica é utilizada e gerida na organização</b></p>	<ul style="list-style-type: none"> <li>• Ajustar a pontuação para “encaixar” no planejamento ou capacidade do time;</li> <li>• Utilizar SPP para comparações entre times, projetos ou organizações sem padronização;</li> <li>• Não manter histórico e registro das contagens realizadas;</li> <li>• Deixar de manter referências consistentes entre itens do backlog (histórias, épicos etc.).</li> </ul>
<p><b>Erros relacionados à utilização inadequada da métrica no contexto ágil</b></p>	<ul style="list-style-type: none"> <li>• Priorizar o aumento da pontuação entregue em detrimento do valor ao usuário;</li> <li>• Utilizar SPP como instrumento de negociação ou pressão sobre o time;</li> <li>• Confundir a medição do fluxo de trabalho (entrega) com o fluxo de valor (resultado).</li> </ul>



# PARTE III

## Horas De Serviço Técnico - HST

## 16. CONTAGEM DE HST

Horas de Serviço Técnico (HST) constituem uma métrica baseada na quantidade de horas necessárias para alcançar um resultado ou entregar um produto, a partir de atividades executadas por um ou mais perfis profissionais, aferidas com base em 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, complementada por Horas de Serviço Técnico. Nessa modalidade, a remuneração deve ser realizada por meio da métrica de Ponto de Função (método APF ou SFP, conforme definido no termo de referência), combinada, quando aplicável, ao pagamento por HST, com base em catálogos de serviços previamente definidos

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 apresentam-se:

- a descrição dos campos que compõem o catálogo de serviços técnicos (subitem 16.1);
- as orientações para o cálculo do custo da HST (subitem 0); e
- as orientações e fórmula para o cálculo do valor a ser remunerado pelo serviço técnico executado (subitem 16.3).

### 16.1. 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 (disponível em <https://www.gov.br/governodigital/pt-br/contratacoes-de-tic>) está alinhado com a portaria mencionada. Ele apresenta os campos mínimos indicados na Portaria SGD/MGI nº 750 e outros campos, alinhados com Acórdão nº 1508/2020 – TCU – Plenário, com o objetivo 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 disseminação de critérios subjetivos;
- Compartilhamento de metodologias e práticas sem a devida consistência e sem justificativas técnicas e econômicas;
- 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 60 - Descrição de campos do catálogo de serviços técnicos**

<b>Campos</b>	<b>Descrição dos campos</b>
<b>ID</b>	Identificador do serviço técnico
<b>Serviço Técnico</b>	Título/Identificação do serviço técnico
<b>Descrição do Serviço</b>	Descrição do serviço técnico.
<b>Unidade de Medida</b>	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.
<b>Descrição dos Entregáveis</b>	Produto resultante da atividade ou serviço realizado. 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.

<b>Descrição das atividades mínimas</b>	Descrição das atividades mínimas a serem realizadas para execução do serviço técnico.
<b>Complexidade</b>	Definição do grau de complexidade do serviço.
<b>Critérios de identificação do grau de complexidade</b>	<p>Descrição dos critérios objetivos utilizados na definição de complexidade do serviço, devendo-se evitar termos ou adjetivos subjetivos.</p> <p>*Os critérios de identificação do grau de complexidade devem estar associados a uma propriedade, característica ou atributo quantificável numericamente. Exemplo: quantidade de tabelas, quantidade de dimensões</p>
<b>Qualificação profissional do executor</b>	<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 à 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>
<b>Esforço (h) necessário à execução do serviço (A)</b>	Corresponde à estimativa de horas necessárias para a realização de um serviço.
<b>Memória de cálculo do esforço</b>	Corresponde ao detalhamento do esforço (em horas) para a realização das atividades envolvidas na execução do serviço técnico.
<b>Critérios de aceitação do serviço</b>	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 da tabela a seguir, com o fim de apresentar a memória de cálculo que justifique a estimativa de custo e estimativa de execução das horas de serviço técnico:

**Tabela 61 - Campos adicionais do catálogo de serviços técnicos que devem constar no ETP**

<b>Campos</b>	<b>Descrição dos campos</b>
<b>Prazo máximo de execução do serviço (em horas)</b>	Corresponde ao nível mínimo de serviço esperado, com base em histórico de produtividade da organização, relacionado à execução da atividade técnica.
<b>Quantidade estimada de execuções do serviço (B)</b>	Quantidade estimada de execuções de cada serviço técnico, a ser definido pela organização, conforme base histórica.
<b>Quantitativo estimado total de HST no contrato, para o serviço (C = A x B)</b>	Corresponde ao produto do esforço em horas (A) necessário à execução do serviço e a sua quantidade estimada de execuções (B).
<b>Memória de cálculo da estimativa de execuções do serviço</b>	Corresponde ao detalhamento do cálculo utilizado para definir a quantidade estimada de execuções do serviço técnico.
<b>Observações</b>	Notas complementares, dicas, comentários e/ou orientações gerais relacionadas ao serviço técnico, se houver.



ATENÇÃO! O catálogo de serviços técnicos apresenta alguns dos serviços técnicos complementares aos serviços de desenvolvimento e manutenção de *software*.

**O órgão pode incluir outros serviços técnicos**, não mensuráveis com a métrica de PF, seguindo o padrão apresentado neste roteiro.

## 16.2. CUSTO DA HST

O custo da HST adjudicado na licitação é utilizado para a remuneração dos serviços técnicos. Normalmente, para sua determinação, utiliza-se a seguinte fórmula:


$$\text{Custo da HST} = \text{Ct}/160$$

Onde: **Ct** corresponde ao custo total mensal do profissional de referência, incluindo encargos e demais componentes da formação de preço, dividido por 160 horas mensais.

Ressalta-se que o custo total do profissional de referência (Ct) é composto pelo custo mensal do perfil profissional mais os custos adicionais previstos na proposta de preços vencedora. Desta forma, uma hora de serviço técnico do profissional de referência corresponde a 1 HST.

Para melhor compreensão, recomenda-se consultar a tabela de cálculo da estimativa de custo mensal do time de referência, apresentada no item 9.5.1 deste roteiro, na qual se detalha a composição do custo e a obtenção do valor da HST.

## 16.3. VALOR A SER REMUNERADO PELO SERVIÇO

Como já mencionado, 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 baseia-se em 3 elementos principais:

- **natureza** do serviço técnico;
- **esforço** (tempo estimado para realização da atividade técnica);
- **perfil profissional:** perfil profissional que irá executar o serviço técnico.

Considerando que os profissionais que atuam nas ordens de serviço remuneradas por HST possuem perfil igual ou compatível com aqueles utilizados nas ordens de serviço remuneradas por pontos de função, adotam-se, para fins de demonstração do cálculo do valor unitário da HST, os perfis profissionais e os valores salariais de referência estabelecidos pela Portaria SGD/MGI nº 750, de 20 de março de 2023, e suas atualizações.



**ATENÇÃO!** Os valores constantes na tabela 62 são meramente ilustrativos, devendo-se utilizar os valores vigentes na Portaria SGD/MGI nº 750 e suas atualizações, ou os valores de novos perfis incluídos pelo órgão observando as diretrizes desta portaria.

**Tabela 62 - Mapa de Pesquisa Salarial de Referência (Portaria SGD/MGI nº 750)**

<b>PERFIL</b>	<b>Descrição do Perfil</b>	<b>Salário de Referência</b>	<b>Fator de Ajuste</b>
<b>ARQSOF-01</b>	Arquiteto de Software - Pleno	R\$ 15.580,79	1,99
<b>ARQSOF-02</b>	Arquiteto de Software - Sênior	R\$ 19.078,14	1,98
<b>ATQ-01</b>	Analista de Testes/Qualidade - Júnior	R\$ 5.568,74	2,15
<b>ATQ-02</b>	Analista de Testes/Qualidade - Pleno	R\$ 8.616,67	2,06
<b>ATQ-03</b>	Analista de Testes/Qualidade - Sênior	R\$ 13.883,33	2,00
<b>DESENV-01</b>	Desenvolvedor de Software - Júnior	R\$ 6.687,85	2,11
<b>DESENV-02</b>	Desenvolvedor de Software - Pleno	R\$ 11.023,35	2,03
<b>DESENV-03</b>	Desenvolvedor de Software - Sênior	R\$ 16.205,18	1,99
<b>ANR-01</b>	Analista de Negócios/Requisitos Júnior	R\$ 6.757,02	2,10
<b>ANR-02</b>	Analista de Negócios/Requisitos Pleno	R\$ 9.561,11	2,04
<b>ANR-03</b>	Analista de Negócios/Requisitos Sênior	R\$ 12.193,99	2,02
<b>ABI-01</b>	Analista de BI Júnior	R\$ 7.588,47	2,08
<b>ABI-02</b>	Analista de BI Pleno	R\$ 11.071,12	2,02
<b>ABI-03</b>	Analista de BI Sênior	R\$ 14.490,34	2,00
<b>ADADOS-02</b>	Administrador de Dados Pleno	R\$ 7.936,98	2,07
<b>ADADOS-03</b>	Administrador de Dados Sênior	R\$ 11.673,56	2,02
<b>LDESENV</b>	Líder Técnico de Desenvolvimento	R\$ 19.065,28	1,98
<b>SCRUM</b>	Scrum Master	R\$ 12.312,50	2,01
<b>GERPRO</b>	Gerente de projetos de tecnologia da informação	R\$ 17.138,03	1,99
<b>AUX/UI-01</b>	Analista de UX/UI Pleno	R\$ 7.877,75	2,07
<b>AUX/UI-02</b>	Analista de UX/UI Sênior	R\$ 12.878,40	2,01
<b>CDADOS-01</b>	Cientista de Dados Júnior	R\$ 8.466,82	2,06
<b>CDADOS-02</b>	Cientista de Dados Pleno	R\$ 13.554,04	2,01

<b>CDADOS-03</b>	Cientista de Dados Sênior	R\$ 19.549,10	1,98
<b>ARQDADOS-01</b>	Arquiteto de Dados Júnior	R\$ 9.299,20	2,04
<b>ARQDADOS-02</b>	Arquiteto de Dados Pleno	R\$ 13.982,75	2,00
<b>ARQDADOS-03</b>	Arquiteto de Dados Sênior	R\$ 18.520,20	1,98
<b>IA-ENG-01</b>	Engenheiro de IA Júnior	R\$ 8.745,65	2,06
<b>IA-ENG-02</b>	Engenheiro de IA Pleno	R\$ 14.370,30	2,00
<b>IA-ENG-03</b>	Engenheiro de IA Sênior	R\$ 16.976,85	1,99
<b>METRICA-01</b>	Analista de Métricas - Júnior	R\$ 5.568,74	2,15
<b>METRICA-02</b>	Analista de Métricas - Pleno	R\$ 8.616,67	2,06
<b>METRICA-03</b>	Analista de Métricas - Sênior	R\$ 13.883,33	2,00

Inicialmente, antes de apresentar a fórmula para o cálculo do valor a ser remunerado pelo serviço, calcula-se o fator de ajuste indicado na tabela anterior (última coluna). Para isso, este roteiro adota como referência o salário do perfil profissional ATQ-01 (linha destacada em amarelo).

Para os demais perfis, com salários superiores ao perfil profissional ATQ-01, o fator de ajuste é superior a um (1). Para fins de padronização, os fatores de ajuste foram arredondados para duas casas decimais. Assim, o fator de ajuste dos demais perfis profissionais é obtido por meio da fórmula a seguir:

$$\text{Fator de Ajuste} = \frac{\text{Salário do Perfil Profissional Y} + \text{Salário do Perfil Profissional de Referência}}{\text{Salário do Perfil Profissional de Referência}}$$

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 9.5.1), conforme fórmula a seguir:



**Valor HST do perfil profissional =**  
Custo da HST X Fator de Ajuste do perfil profissional



Assim, o valor a ser remunerado para o serviço técnico executado é calculado conforme a seguinte expressão:



**Valor a ser remunerado para o serviço =**  
Valor HST do perfil profissional X Esforço (h) X Qtd. de Execuções



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 técnicos de referência, disponível em <https://www.gov.br/governodigital/pt-br/contratacoes-de-tic>; e
- **Qtd. de Execuções** corresponde à quantidade de vezes que o serviço técnico a ser remunerado foi executado.

Por fim, destacam-se as seguintes diretrizes para utilização da métrica de Horas de Serviço Técnico (HST):

- A composição do catálogo de serviços técnicos a serem remunerados pela métrica HST deve observar as diretrizes estabelecidas no item 16.1 deste roteiro, bem como o catálogo de referência de serviços técnicos, disponível em <https://www.gov.br/governodigital/pt-br/contratacoes-de-tic>, alinhado à Portaria SGD/MGI nº 750 e ao Acórdão nº 1508/2020 – TCU – Plenário.
- Caso o serviço seja executado por profissional com qualificação superior ao mínimo exigido, o valor da HST não deve ser alterado, devendo ser considerado o valor correspondente ao perfil de referência do serviço.

- Não deve haver sobreposição entre serviços remunerados por Ponto de Função (PF) e por Horas de Serviço Técnico (HST), devendo cada serviço ser classificado de acordo com sua natureza.
- No que se refere à atomicidade, a menor unidade de medição é de 1 (uma) HST, não sendo permitido o fracionamento inferior a esse valor.



**Atenção! 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 técnicos do órgão.

Caso o serviço seja realizado por profissional com qualificação superior ao mínimo exigido, o valor da HST do perfil profissional não deve ser alterado.



# PARTE IV

## Linhas de Código Funcionais (LoCF)

## 17. MÉTRICA DE LINHAS DE CÓDIGO FUNCIONAIS (LoCF)

As métricas baseadas em Linhas de Código (LoC), historicamente, figuram entre as mais antigas formas de mensuração de *software*. Contudo, seu uso indiscriminado demonstrou limitações relevantes, tais como dependência da linguagem de programação, estímulo a soluções excessivamente complexas e fraca correlação direta com valor funcional. Por essa razão, este roteiro apresenta o conceito de **Linhas de Código Funcionais (LoCF)**, que representa uma evolução conceitual das métricas tradicionais de LoC.

Diferentemente da simples contagem volumétrica de linhas, a LoCF busca mensurar apenas o código efetivamente necessário para implementar funcionalidades que agregam valor ao sistema, desconsiderando elementos acessórios, redundantes ou meramente estruturais, e observando critérios mínimos de qualidade e arquitetura previamente definidos. Além disso, o mecanismo de cálculo da LoCF foi desenhado para estimular a refatoração e otimização de código útil e funcional, inclusive em contextos de desenvolvimento apoiado por ferramentas de Inteligência Artificial, nos quais a produtividade técnica não deve ser aferida pela simples geração automática de código, mas pela capacidade da equipe de revisar, integrar, ajustar, testar, refatorar, proteger e manter código efetivamente incorporado ao produto de software.

A métrica Linha de Código Funcional (LoCF) caracteriza-se por:

- Contribuir diretamente para o comportamento observável do software em execução;
- Estar associada a uma funcionalidade, regra de negócio ou requisito claramente identificado;
- Ser tecnicamente necessária, evitando duplicação, excesso de complexidade ou fragmentação artificial;
- Atender aos padrões de qualidade, arquitetura e de codificação definidos pelo órgão;
- Estar vinculada a uma entrega aceita, validada ou homologada conforme o processo de desenvolvimento definido pelo órgão;
- Integrar código executável, lógica funcional ou componente técnico necessário à entrega da funcionalidade, observados os critérios de rastreabilidade, qualidade e aceitação.

Elemento central do conceito de Linhas de Código Funcionais é a definição explícita dos critérios de não contabilização, cuja finalidade é evitar distorções métricas e comportamentos oportunistas. Assim, não devem ser contabilizadas como LoCF as linhas de código que não agreguem valor funcional direto ou que não sejam estritamente necessárias à implementação das funcionalidades contratadas.

Não são consideradas Linhas de Código Funcionais, em especial, as seguintes categorias:

- a) Linhas de comentário, independentemente de sua relevância documental;

- b) Linhas em branco ou utilizadas exclusivamente para formatação visual;
- c) Código gerado automaticamente por ferramentas, *frameworks* ou *scaffolding*, salvo quando houver modificação manual relevante e tecnicamente justificável;
- d) Código gerado por ferramenta de Inteligência Artificial que não tenha sido revisado, integrado, testado e assumido tecnicamente pela equipe responsável;
- e) Código provisório, experimental ou descartável que não componha a versão aceita da entrega;
- f) Código que não possua rastreabilidade mínima com requisito, história, defeito, tarefa técnica autorizada ou critério de aceitação.
- g) Código duplicado, redundante ou reutilizado sem adaptação funcional;
- h) Código morto, desativado, comentado ou não referenciado em fluxo de execução;
- i) Trechos criados apenas para inflar métricas, sem impacto funcional comprovável.

Adicionalmente, não devem ser contabilizadas:

- a) linhas de código cuja existência decorra de escolhas técnicas que contrariem os padrões de qualidade definidos pelo órgão, tais como fragmentação artificial de métodos, complexidade excessiva sem justificativa funcional, violação de princípios de coesão e acoplamento, ou desrespeito a guias de nomenclatura, arquitetura e padrões de codificação.
- b) linhas de código pertencentes a arquivos de configuração, *scripts* auxiliares, artefatos de infraestrutura, rotinas de *build* ou automação, exceto quando tais elementos incorporarem lógica funcional explícita e prevista no escopo da ordem de serviço.
- c) linhas de código que violem decisões arquiteturais registradas, padrões de camadas, modularidade, separação de responsabilidades, baixo acoplamento, alta coesão, testabilidade, segurança, manutenibilidade, legibilidade e demais critérios de qualidade de código definidos pelo órgão.
- d) linhas de código pertencentes a métodos, funções, procedimentos ou unidades lógicas executáveis cuja complexidade ciclomática exceda o limite máximo definido pelo órgão, recomendando-se, como referência inicial, o limite de complexidade ciclomática igual ou inferior a 10, conforme itens 8.118 e 8.134 da ISO/IEC 5055:2021, salvo quando formalmente justificadas, registradas e aceitas pelo órgão contratante.

A LoCF deve permitir a mensuração de diferentes formas de contribuição técnica sobre o código-fonte, distinguindo código novo, código alterado, código refatorado, código otimizado e código removido com preservação do comportamento esperado. Essa distinção é necessária para evitar que a métrica estimule apenas a expansão do volume de código, em prejuízo da simplificação, da redução de dívida técnica e da melhoria da manutenibilidade.

Para esse fim, a apuração da quantidade de LoCF engloba as seguintes categorias:

a) **LoCF-N**: linhas funcionais novas, correspondentes ao código novo implementado, aceito e aderente aos critérios de qualidade definidos;

b) **LoCF-A**: linhas funcionais alteradas, correspondentes ao código existente modificado para atendimento de requisito, correção de defeito ou adaptação técnica autorizada;

c) **LoCF-R**: linhas refatoradas ou otimizadas, correspondentes ao código reestruturado sem alteração funcional externa, desde que haja ganho demonstrável de legibilidade, manutenibilidade, desempenho, segurança, testabilidade, redução de complexidade ou aderência arquitetural;

d) **LoCF-E**: linhas excluídas por otimização, correspondentes à remoção de código morto, duplicado, redundante, desnecessário ou substituído por solução mais simples, desde que preservado o comportamento esperado e comprovada a não regressão.

Obtendo-se a seguinte forma de cálculo:



$$\text{Quantidade de LoCF} = \text{LoCF-N} + \text{LoCF-A} + \text{LoCF-R} + \text{LoCF-E}$$



Para fins de apuração, uma mesma linha de código não poderá ser contabilizada simultaneamente em mais de uma categoria de LoCF. Nos casos em que uma alteração envolver inclusão, modificação, refatoração ou exclusão no mesmo trecho lógico de código, deverá prevalecer a classificação predominante, conforme a natureza principal da intervenção técnica realizada e a evidência registrada no processo de desenvolvimento.

A apuração da LoCF deve observar, ainda, filtros mínimos de aceitabilidade. Apenas poderá ser contabilizada como LoCF a linha de código que: esteja vinculada a requisito, história de usuário, item de backlog, defeito, dívida técnica autorizada ou critério de aceitação; integre o fluxo executável ou logicamente necessário da aplicação; esteja presente em entrega aceita; tenha sido submetida aos mecanismos de revisão, teste e validação definidos pelo órgão; e não se enquadre nas hipóteses de exclusão previstas neste roteiro.

A métrica de Linhas de Código Funcionais não deve, em nenhuma hipótese, ser utilizada como base direta para cálculo de pagamento, remuneração, faturamento ou reajuste contratual. Seu uso limita-se à verificação da produtividade do time, à formação de base histórica, ao acompanhamento da evolução da produtividade técnica,

à identificação de tendências de qualidade de código e ao apoio à gestão interna de equipes e processos de desenvolvimento.

## 17.1. EXEMPLO DA APLICAÇÃO DA MÉTRICA LoCF

Considere para fins exemplificativas que determinado órgão emitiu uma ordem de serviço para evolução de uma rotina PL/SQL responsável pela consolidação mensal de dados orçamentários. A demanda envolveu a inclusão de nova regra de validação, alteração de procedimento existente, refatoração de função com alta complexidade, exclusão de código redundante e uso de apoio de ferramenta de IA generativa durante o desenvolvimento.

**Tabela 63 – Exemplo de memória de cálculo da métrica.**

<b>Categoria</b>	<b>Quantidade de linhas</b>	<b>Quantidade contabilizada</b>	<b>Justificativa</b>
Código gerado por IA generativa e não incorporado à entrega	35	0	<b>Não contabilizadas - código sugerido por ferramenta de IA, mas não revisado, integrado, testado e assumido tecnicamente pela equipe responsável.</b>
LoCF-N - novas linhas funcionais em package PL/SQL	160	160	<b>Contabilizada - linhas novas implementadas em package PL/SQL para inclusão de regra de validação vinculada a requisito aceito na ordem de serviço.</b>
Comentários técnicos e cabeçalhos de procedures	42	0	<b>Não contabilizadas - comentários, cabeçalhos e anotações documentais não representam código executável nem lógica funcional diretamente incorporada ao produto.</b>
LoCF-E - exclusão de código redundante	75	75	<b>Contabilizada - remoção de trechos duplicados em procedures PL/SQL, com preservação do comportamento esperado e comprovação de não regressão.</b>
Script de testes do código PL/SQL	95	0	<b>Não contabilizadas - scripts de teste utilizados para validação técnica da rotina não integram o código funcional entregue como produto da aplicação.</b>
LoCF-A - alteração de linhas funcionais existentes	90	90	<b>Contabilizada - linhas existentes foram modificadas para adequar regra de cálculo em procedure já existente, com rastreabilidade para requisito funcional autorizado.</b>
Linhas em branco e formatação visual	28	0	<b>Não contabilizadas - linhas utilizadas apenas para espaçamento, organização visual ou</b>

			<b>formatação do código-fonte não agregam valor funcional direto.</b>
LoCF-R - refatoração de função PL/SQL	<b>110</b>	<b>110</b>	<b>Contabilizada - função PL/SQL foi reestruturada sem alteração do comportamento externo, com redução de complexidade ciclomática, melhoria de legibilidade e maior testabilidade.</b>
Código do prompt de IA generativa	<b>18</b>	<b>0</b>	<b>Não contabilizadas - texto de prompt utilizado para apoiar a geração de sugestões pela ferramenta de IA não constitui código-fonte executável nem componente funcional do software.</b>
Código provisório de diagnóstico removido antes da entrega	<b>22</b>	<b>0</b>	<b>Não contabilizadas - comandos temporários de diagnóstico e depuração não compuseram a versão aceita da entrega.</b>
<b>Total de linhas contabilizadas como LoCF</b>		<b>435</b>	

Assim, embora tenham sido movimentadas 675 linhas no processo de desenvolvimento, apenas 435 linhas foram contabilizadas como Linhas de Código Funcionais, por atenderem aos critérios de contabilização da métrica.

A aferição de Linhas de Código Funcionais deve ser apoiada, sempre que possível, por ferramentas automatizadas de análise e verificação, devidamente homologadas ou aceitas pelo órgão contratante. O uso de ferramentas automatizadas visa assegurar objetividade, reprodutibilidade, rastreabilidade e transparência ao processo de aferição, reduzindo subjetividade e assimetria de informação entre as partes.

Por fim, ressalta-se que a métrica de Linhas de Código Funcionais (LoCF) possui caráter complementar à gestão das equipes de desenvolvimento. A LoCF não substitui nem integra a mensuração funcional realizada por Pontos de Função, devendo ser utilizada exclusivamente para acompanhamento de produtividade técnica, sem interferir nos critérios formais de estimativa, medição e remuneração estabelecidos neste roteiro.



# PARTE V

## Seleção e Utilização das Métricas

## 18. COMO SELECIONAR A MÉTRICA A SER ADOTADA

Cada métrica descrita neste Roteiro possui campos de aplicação distintos e complementares seja em busca de implementar mecanismos de governança dos serviços de construção, aprimoramento e sustentação de sistemas da informação, seja no fornecimento de insumos para tomada de decisão sobre investimentos, alocação de recursos, continuidade de projetos e revisão de práticas operacionais para maior entrega de valor.

Portanto, as métricas descritas nesse roteiro devem ser entendidas como um conjunto de diferentes tipos de ferramentas de transparência e gestão das atividades de construção de sistemas de informação

Nesse sentido, saber escolher bem uma ferramenta é fundamental para aprimorar a capacidade de gestão do processo de engenharia de software da instituição. O primeiro passo para a seleção do conjunto de métricas a serem adotadas no apoio a essa gestão é ter consciência de que não há uma “bala de prata” ou uma métrica que seja a melhor independente da realidade e perfil de ambientes e processos dos órgãos, mas sim a combinação de diferentes métricas aplicadas em contextos específicos.

Para auxiliar os gestores de tecnologia da informação na seleção do conjunto de métricas e respectivos escopos de aplicação, apresenta-se a seguir uma tabela de referência capaz de auxiliar nesta seleção. Contudo, reforça-se que cada realidade, escopo, tipo de sistema e contexto organizacional pode influenciar nessa seleção, sendo necessário que o gestor de tecnologia avalie e considere esses fatores no momento da seleção das métricas não se pautando exclusivamente na referência a seguir:

**Tabela 64 - Apoio na seleção de métricas**

<b>Métrica</b>	<b>Objeto da medição</b>	<b>Cenários mais adequados de uso</b>	<b>Restrições de uso</b>	<b>Premissas para a adoção da métrica</b>	<b>Outros benefícios</b>
<b>APF (Análise de Pontos de Função)</b>	Funcionalidades de software percebidas pelo usuário e implementadas pelo sistema.	Projetos de desenvolvimento, melhoria e baseline com requisitos funcionais suficientemente definidos. Adequada quando se exige alta rastreabilidade, comparabilidade histórica e	Não mede diretamente requisitos não funcionais, esforço técnico interno, infraestrutura, suporte ou atividades sem entrega funcional.	Fronteira da aplicação, escopo e propósito da contagem previamente definidos. Documentação funcional disponível. Conhecimento das regras do CPM/IFPUG e do roteiro do órgão. Profissionais capacitados em APF.	Favorece a comparabilidade entre projetos, a independência tecnológica, a rastreabilidade das entregas, a realização de benchmarking e a gestão contratual baseada em resultados e maior

		independência tecnológica.			acurácia na medição.
<b>SFP (Simple Function Point)</b>	Funcionalidades de software percebidas pelo usuário e implementadas pelo sistema.	Projetos que demandam contagem funcional mais simples, rápida e previsível. Adequada para estimativas iniciais, projetos ágeis, releases, manutenção evolutiva e contextos de independência tecnológica.	Não mede diretamente requisitos não funcionais, esforço técnico interno, infraestrutura, suporte ou atividades sem entrega funcional. Não permite derivar automaticamente uma contagem APF completa a partir do SFP.	Fronteira, escopo e propósito da contagem definidos. Identificação objetiva de Processos Elementares e Arquivos Lógicos. Conhecimento Adoção do CPM/IFPUG. SPF/IFPUG e das regras complementares do roteiro.	Simplifica o processo de contagem, reduz custos de medição e simplifica o processo de adoção institucional da métrica.
<b>SPP (Story Points Padronizados)</b>	Histórias de usuário dimensionadas por critérios técnicos padronizados.	Ambientes com fluência em práticas ágeis	Não deve ser utilizada exclusivamente para pagamento. Não deve ser associada diretamente a hora ou prazo de entrega. Não é adequada para histórias de usuário que não possuam critérios de aceitação verificáveis e padrões de construção definidos.	Padrão de construção de histórias de usuário definido. Critérios de aceitação claros, verificáveis e completos. Tipos base e fatores técnicos padronizados. Definição de Preparado e Definição de Pronto estabelecidas.	Facilita o planejamento de sprints, melhora a previsibilidade da capacidade produtiva, reduz divergências entre equipes e aumenta a transparência dos critérios de estimativa.
<b>LocF (Linhas de Código Funcionais)</b>	Código-fonte funcional efetivamente desenvolvido, mantido ou evoluído em sistemas legados.	Sistemas legados baseados em abordagem de codificação estruturada ou procedural.	Não é adequada para ambientes baseados em no-code, low-code, microserviços e cloud native.	Padrões de codificação e de arquitetura definidos. Critérios explícitos de contagem e de não contagem. Repositório versionado, rastreabilidade entre requisito e código, ferramenta de análise estática, revisão técnica e validação de qualidade.	Possibilita o aproveitamento de métricas históricas, facilita a medição de sistemas com documentação limitada e apoia a gestão da evolução técnica de aplicações legadas.
<b>HST (Horas de Serviços Técnicos)</b>	Produtos e serviços técnicos especializados executados ao longo do ciclo de vida da solução.	Atividades complementares ao desenvolvimento de software que não sejam adequadamente mensuráveis por	Não deve substituir métricas orientadas a produto quando houver entrega funcional mensurável.	Catálogo de Serviços Técnicos definido contendo: Produtos/entregáveis, perfis profissionais, esforço estimado, critérios de aceitação, níveis mínimos de serviço, memória de	Amplia a capacidade de contratação de serviços especializados, proporciona maior flexibilidade operacional e permite mensurar

APF, SFP, SPP ou LoCF.	cálculo e forma de remuneração previamente estabelecidos.	entregas técnicas que não são adequadamente representadas por métricas funcionais.
------------------------	---	--

A seleção das métricas deve considerar a adoção isolada ou combinada das métricas apresentadas neste roteiro, com vistas a melhor aferir, acompanhar e comparar a produtividade dos serviços, a qualidade das entregas e a aderência dos resultados aos objetivos institucionais.

Essa escolha deve observar a natureza do serviço contratado, o grau de maturidade dos requisitos, o modelo de desenvolvimento adotado, o perfil dos sistemas, o ambiente tecnológico, a disponibilidade de documentação, a existência de base histórica e demais especificidades do órgão ou entidade que possam influenciar a eficiência, a objetividade e a rastreabilidade da mensuração.

Assim, recomenda-se que a métrica ou o conjunto de métricas adotado em cada contratação seja expressamente justificado nos artefatos de planejamento e de gestão contratual, com a indicação de seu escopo de aplicação, premissas, limitações e mecanismos de controle.

Dessa forma, evita-se o uso genérico ou indiscriminado de métricas fora do contexto para o qual são mais adequadas, fortalecendo a governança, a transparência e a capacidade de gestão dos serviços de desenvolvimento, manutenção e sustentação de software.

## 19. PRÁTICAS DE ADOÇÃO E MONITORAMENTO DAS MÉTRICAS

A adoção de métricas deve estar alinhada aos objetivos da contratação, às características do ambiente de desenvolvimento e ao nível de maturidade da organização. A seleção da métrica mais adequada deve considerar fatores como: aderência ao processo de desenvolvimento de software do órgão, características tecnológicas dos ambientes desenvolvimento de softwares do órgão, capacitação da equipe de mensuração, objetivos da mensuração para a gestão dos projetos.

Independentemente da métrica adotada, recomenda-se que a organização estabeleça procedimentos formais para sua aplicação, acompanhamento e evolução contínua. Tais procedimentos devem considerar no mínimo os elementos de planejamento da adoção, forma de implantação, meios de monitoramento, mecanismos de controle/governança e procedimentos para aprimoramento do uso.

Figura 28 - Adoção e monitoramento das métricas



## 19.1. PLANEJAMENTO DA ADOÇÃO

Antes da adoção de uma métrica, recomenda-se:

- Definir claramente os objetivos da medição;
- Identificar os processos, produtos ou serviços que serão mensurados;
- Avaliar a aderência da métrica ao contexto organizacional;
- Estabelecer critérios de contagem, estimativa e aceite;
- Capacitar as equipes envolvidas;
- Definir responsabilidades pela medição, validação e auditoria.

## 19.2. IMPLANTAÇÃO FORMAL

Toda métrica a ser aplicada no escopo da execução de serviços contratados de desenvolvimento, manutenção e sustentação de software deve estar formalmente estabelecida e ser amparada por instrumentos que assegurem sua objetividade, rastreabilidade e previsibilidade.

Essa abordagem permite:

- Segurança jurídica;
- Previsibilidade na formação do preço;
- Objetividade na execução e mensuração dos serviços;
- Reduzir riscos de desvios de interpretação;
- Construir uma base histórica de referência.

## 19.3. MONITORAMENTO E ACOMPANHAMENTO

O monitoramento das métricas deve ocorrer de forma contínua, permitindo avaliar a efetividade da sua utilização e apoiar a tomada de decisão.

Recomenda-se acompanhar, entre outros:

- Volume de entregas realizadas;
- Produtividade das equipes;
- Evolução do backlog;
- Qualidade das entregas;
- Retrabalho identificado;
- Tendências de crescimento ou redução da demanda;
- Desvios entre estimativas e resultados obtidos.

## 19.4. GOVERNANÇA DA MEDIÇÃO

No contexto de contratos de desenvolvimento, manutenção e sustentação de software, a governança da medição busca garantir que a quantificação de software produzido ou mantido seja objetiva, comparável e útil à tomada de decisão, prevenindo interpretações divergentes, reduzindo assimetrias de informação entre contratante e contratada e promovendo a melhoria contínua dos processos, produtos e serviços de software.

Assim a organização deve estabelecer mecanismos de governança que assegurem:

- Aplicação uniforme dos critérios de medição;
- Registro das contagens e estimativas realizadas;
- Rastreabilidade dos resultados;
- Tratamento de divergências de interpretação;
- Atualização periódica dos procedimentos internos.

A aplicação uniforme dos critérios de medição pode ser alcançada com a institucionalização do Roteiro de Métricas, capacitação da equipe técnica do órgão, previsão objetiva das métricas em instrumentos contratuais e padronização do processo de aferição de software.

Recomenda-se a adoção de ferramenta para registro e acompanhamento das contagens e estimativas, incluindo funcionalidades e dashboards para apoio à tomada de decisão em diferentes níveis da organização. Na ausência de ferramentas, pode-se adotar repositório próprio com versionamento de documentos e definir procedimentos de registro e controle das contagens.

Para assegurar o alinhamento entre o desempenho das atividades e os respectivos objetivos alcançados, o órgão ou entidade deve associar cada aferição ao escopo e entrega efetivamente realizado indicando a ordem de serviços e informações do contrato objeto da aferição.

O Tratamento de divergências de interpretação, por sua vez, podem ser implementado por meio do estabelecimento de processo formal de contestação, bem como a definição de instâncias de revisão, incluindo a previsão de submissão de eventual divergência ao órgãos central do SISP para posicionamento independente, assegurando nos instrumentos contratuais a responsabilidade da Contratante pelo posicionamento definitivo exaurida das etapas de análise e assegurado o direito de argumentação e defesa de ambas as partes.

Por fim, a governança da medição deve contar com instrumentos passíveis e aprimoramento e atualização com base nos registros de lições aprendidas e observando o aprimoramento das normas e padrões.

## 19.5. MELHORIA CONTÍNUA

As métricas adotadas devem ser periodicamente avaliadas quanto à sua efetividade, aderência ao contexto organizacional e capacidade de produzir informações úteis para gestão.

## 20. USO DE FERRAMENTAS DE IA NA MENSURAÇÃO DE SOFTWARE

As ferramentas de IA devem ser tratadas como instrumentos de apoio técnico, podendo auxiliar na análise documental, organização de requisitos, identificação preliminar de funcionalidades, revisão de consistência, geração de hipóteses de contagem e elaboração de artefatos de apoio. Contudo, a decisão final sobre a mensuração permanece sob responsabilidade de profissional capacitado ou equipe técnica designada.

Ao adotar ferramentas de IA no processo de mensuração, deve-se assegurar:

**a) Supervisão humana** - Toda sugestão da IA deve ser revisada e validada por analista responsável;

**b) Rastreabilidade** - Entradas, premissas, prompts relevantes, respostas e decisões devem ser registradas quando impactarem a contagem;

**c) Reprodutibilidade** - a mensuração deve poder ser refeita com base nos requisitos, regras e decisões documentadas;

**d) Segurança da informação** - o uso deve observar as restrições de acesso, o sigilo, a classificação da informação, as diretrizes da LGPD e normas internas do órgão;

**e) Transparência** - o uso de IA deve ser declarado quando influenciar artefatos de medição.

Admite-se o uso de IA, por exemplo, para:

a) apoiar a leitura e sumarização de documentos de requisitos, atas, histórias de usuário, critérios de aceitação e especificações técnicas;

b) identificar possíveis processos elementares, arquivos lógicos, histórias de usuário, tarefas técnicas ou unidades candidatas de mensuração;

c) sugerir agrupamentos de requisitos, dependências, ambiguidades, lacunas e inconsistências;

d) apoiar a elaboração preliminar de memórias de cálculo;

e) comparar contagens com bases históricas, quando houver dados autorizados;

f) revisar aderência textual entre requisitos, critérios de aceitação, definição de pronto e itens mensurados;

g) apoiar a classificação preliminar de itens em APF, SFP, SPP, HST ou LoCF;

h) auxiliar na identificação de riscos de superestimativa da contagem, subestimativa da contagem, duplicidade ou ausência de evidência funcional.



Atenção! As saídas produzidas por ferramentas de IA não substituem a aplicação dos métodos de mensuração definidos neste Roteiro, tampouco a análise crítica e a responsabilidade técnica do profissional ou equipe responsável pela contagem.

## 21. CONCLUSÃO

O Roteiro de Métricas de Software do SISP, instituído pela Portaria SGD/MGI nº 3656, de 2026, em sua versão 3.0, foi revisto para considerar práticas contemporâneas de desenvolvimento de software, inclusive a codificação assistida por Inteligência Artificial, assegurando-se que a mensuração permaneça orientada ao produto efetivamente entregue, aceito e aderente aos requisitos funcionais e não funcionais aplicáveis.

Nesse contexto, buscou-se ampliar as práticas e técnicas de mensuração de produtos e serviços de software no âmbito da Administração Pública Federal, ao incorporar métodos já consagrados no campo da engenharia de software, como a Análise de Pontos de Função (APF), ao introduzir alternativas funcionais simplificadas, como o Simple Function Point (SFP), ao trazer alternativas ágeis, como os Story Points Padronizados (SPP), ao complementar a mensuração funcional com aferição de produtos complementares por meio da Hora de Serviço Técnico (HST) e ao reposicionar a métrica Linhas de Código Funcionais Implementadas (LoCF) como alternativa para mensuração de produtividade em sistemas legados, com foco na vinculação a aspectos funcionais.

A literatura de engenharia de software reforça que a medição é condição necessária para a gestão da qualidade na produção de software, já que métricas bem definidas subsidiam o processo de decisão técnica e operacional dos gestores de TI do órgão, além de serem capazes de identificar desvios, reduzir retrabalho, aprimorar estimativas, fortalecer a previsibilidade contratual e apoiar a evolução dos processos de entrega de sistemas e produtos de software no contexto da transformação digital do setor público. Assim, a adoção de métricas adequadas contribui para que os investimentos públicos em software sejam planejados, contratados, executados, acompanhados e avaliados com maior racionalidade, transparência e foco em resultados.

Além das diferentes métricas, o escopo de mensuração foi ampliado e passou a incluir, além dos aspectos funcionais tradicionais dos sistemas transacionais, elementos como recursos de Inteligência Artificial, Chatbots, Geoprocessamento, Painéis Analíticos, Data Lake e Data Warehouse. Com isso, o roteiro contribui para que os órgãos e entidades integrantes do SISP disponham de parâmetros mais adequados para dimensionar, acompanhar e avaliar soluções digitais contemporâneas, cada vez mais relevantes para a modernização do Estado, para a melhoria dos serviços públicos e para a entrega de valor à sociedade.

A incorporação da métrica de Horas de Serviço Técnico (HST), acompanhada do Catálogo de Referência de Serviços Técnicos publicados e atualizados pela Secretaria de Governo Digital, contribui para mitigar riscos de antieconomicidade, subjetividade, ausência de padronização e assimetria de informações nas contratações públicas de TIC. Ao estabelecer critérios mais objetivos para a mensuração de serviços técnicos complementares, o roteiro favorece maior equilíbrio nas relações entre a Administração Pública e o mercado, fortalece a transparência na formação de preços e apoia a adequada remuneração dos produtos e serviços efetivamente entregues.

Ao também contemplar a métrica de Story Points Padronizados (SPP), o roteiro reforça relevância dos métodos ágeis no desenvolvimento de software e propõe uma forma mais objetiva, verificável e comparável de mensurar histórias de usuário, estimar esforço e acompanhar produtividade. De forma complementar, a métrica de Linhas de Código Funcionais (LoCF) amplia o conjunto de instrumentos disponíveis para contextos em que seja necessário avaliar incrementos funcionais de software associados à produção de código útil em sistemas legados.

Dessa forma, este roteiro estimula a aplicação padronizada de metodologias objetivas, promove maior comparabilidade de preços, reduz riscos contratuais, fortalece a governança da medição e apoia a melhoria contínua da gestão de serviços de desenvolvimento, manutenção e sustentação de software. Ao consolidar práticas de mensuração alinhadas às normas internacionais ISO/IEC/IEEE 15939 e ISO/IEC/IEEE 12207, bem como às normas de contratação de Tecnologia da Informação, como a IN SGD/ME nº 94/2022 e a Portaria SGD/MGI nº 750/2023, o Roteiro de Métricas de Software do SISP 3.0 contribui para a melhoria da qualidade do gasto público, para o aumento da eficiência na execução contratual e para a entrega de produtos de software mais seguros, úteis, sustentáveis e aderentes às necessidades da sociedade brasileira.

## 22. REFERÊNCIAS BIBLIOGRÁFICAS

1. BRASIL. Ministério da Gestão e da Inovação em Serviços Públicos. Portaria SGD/MGI nº 750, de 20 de março de 2023. Disponível em: <https://www.gov.br/governodigital/pt-br/contratacoes-de-tic>. Acesso em: 3 out. 2025.
2. BRASIL. Ministério da Gestão e da Inovação em Serviços Públicos. Portaria SGD/MGI nº 6.040, de 11 de agosto de 2025. Acesso em: 3 out. 2025.
3. BRASIL. Ministério do Planejamento, Desenvolvimento e Gestão. Secretaria de Tecnologia da Informação e Comunicação – Setic. Roteiro de Métricas de Software do SISP: versão 2.3. Brasília: MP, 2018. 96 p. BRASIL. Tribunal de Contas da União. Acórdão nº 1508/2020 – Plenário.
4. CAROLI, P.; AGUIAR, R. Product Backlog Building: um guia prático para criação e refinamento de backlog para produtos de sucesso.
5. COHN, M. Agile Estimating and Planning. Prentice Hall, 2005.
6. DEKKERS, C. Measuring the “logical” or “functional” Size of Software Projects and Software Application. Spotlight Software, ISO Bulletin, p. 10-13, Maio 2003.
7. ENGENHARIA de Software e Sistemas — Teste de Software — Parte 6: Diretrizes para o uso da ISO/IEC/IEEE 29119 em Projetos Ágeis. ISO/IEC/IEEE 29119-6:2017.
8. GARTNER. User Story Essentials.
9. GUIAS SCRUM. O Guia Scrum 2020.
10. INTERNATIONAL FUNCTION POINT USERS GROUP (IFPUG). Counting Practices Manual. Version 4.3.1. Jan. 2010.
11. INTERNATIONAL FUNCTION POINT USERS GROUP (IFPUG). SFP Counting Practices Manual. Release 2.1. Oct. 2021.
12. ISO/IEC/IEEE 12207:2017. Engenharia de sistemas e software — Processos do ciclo de vida de software.
13. ISO/IEC/IEEE 24765:2017. Engenharia de sistemas e software — Vocabulário.
14. ISO/IEC/IEEE 26515:2018. Engenharia de sistemas e software — Desenvolvendo informações para usuários em um ambiente ágil.
15. ISO/IEC/TR 24587:2021. Engenharia de software e sistemas — Desenvolvimento ágil — Considerações de adoção ágil.
16. JEFFRIES, R.; ANDERSON, A.; HENDRICKSON, C. Extreme Programming Installed. Addison-Wesley, 2000.
17. MANIFESTO ÁGIL. Disponível em: <https://agilemanifesto.org/iso/ptbr/manifesto.html>. Acesso em: 3 out. 2025.
18. PMI – PROJECT MANAGEMENT INSTITUTE. Um guia do conhecimento em gerenciamento de projetos (Guia PMBOK). 6. ed. Estados Unidos: PMI, 2017.

19. PMI – PROJECT MANAGEMENT INSTITUTE. Um guia do conhecimento em gerenciamento de projetos (Guia PMBOK). 7. ed. Estados Unidos: PMI, 2021.
20. PRINCÍPIOS por trás do Manifesto Ágil. Disponível em: <https://agilemanifesto.org/iso/ptbr/principles.html>. Acesso em: 3 out. 2025.
21. SCHWABER, K.; SUTHERLAND, J. The Scrum Guide™ – O Guia Definitivo do Scrum: As Regras do Jogo. 2020.
22. SERVIÇO FEDERAL DE PROCESSAMENTO DE DADOS (SERPRO). Guia de Referência de Contagem de Pontos de Função. Versão 5.3. 2025.



gov.br

MINISTÉRIO DA  
GESTÃO E DA INOVAÇÃO  
EM SERVIÇOS PÚBLICOS

GOVERNO DO  
**BRASIL**  
DO LADO DO POVO BRASILEIRO