

## ANEXO XI – ROTEIRO DE MÉTRICAS

### 1. INTRODUÇÃO

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

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

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

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

*b) Linhas de código implementadas;*

*c) Pontos de história (Story Point);” Este roteiro apresenta o método de Ponto de Função Simples (Simple Function Point - SFP) publicado pelo International Function Point Users Group (IFPUG), com vistas possibilitar uma aferição da entrega de produtos por meio de métricas de software objetivas e de forma célere e simples.”*

O método apresentado neste roteiro permite avaliar a medida de tamanho funcional de forma simplificada, conforme descrito no Manual de Práticas de Contagem do Método - SPM v.2.1, publicado pelo IFPUG.

A aplicação do método de análise de Pontos de Função Simplificado consiste na adoção das mesmas definições de processos elementares e arquivos lógicos da contagem de pontos de função tradicional, e por conseguinte possui as mesmas características em termos de métrica de produto de software, além da agilidade conforme descrito a seguir:

- a) é útil no início do processo de desenvolvimento para estimativa do esforço e tamanho funcional a ser implementado;
- b) independe da metodologia de desenvolvimento e tecnologia adotada;
- c) oferece resultados confiáveis, repetíveis e objetivos;
- d) foi projetado para ser ágil, rápido, leve e de fácil utilização, acelerando consideravelmente o processo de dimensionamento funcional.

## 2. OBJETIVO

O objetivo principal deste documento é apresentar o método Ponto de Função Simples (SFP) para dimensionamento do tamanho funcional, com base nas regras de contagem do Manual de Práticas de Contagem do Método - SPM v.2.1. Esse roteiro está alinhado com a Portaria SGD/MGI nº 750, de 20 de março de 2023 e às recomendações dispostas no Acórdão nº 2.037/2019-TCU-Plenário e no Acórdão nº 1.508/2020-TCU-Plenário. Para as questões não abrangidas por este modelo, deve-se aplicar subsidiariamente as disposições constantes do Roteiro de Métricas do SISP, versão 2.3.

## 3. TERMOS E DEFINIÇÕES

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

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

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

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

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

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

**Componente Funcional Básico:** Unidade elementar de Requisitos Funcionais do Usuário definido e utilizado pelo método FSM para propósitos de medição (ISO/IEC 14143-1:2007). ALIs, AIEs, EEs, SEs, CEs são os tipos de CFBs (BFCs).

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

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

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

IFPUG: Grupo Internacional de Usuários de Ponto de Função

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

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

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

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

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

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

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

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

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

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

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

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

Requisitos funcionais: conjunto de requisitos do usuário que descrevem o que o software deve fazer em termos de tarefas e serviços.

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

*Simple Function Point* - SFP: Pontos de função simples.

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

SPM: Manual de práticas de contagem do método Ponto de Função Simples.

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

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

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

#### **4. CONTAGEM DE PONTOS DE FUNÇÃO COM O MÉTODO SFP**

O *Simple Function Point* (SFP) é um método de medição funcional compatível e consistente com o IFPUG APF, descrito no CPM v.4.3.1, em terminologia e definições.

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 SFP é orientado ao produto de software, pode ser aplicado em qualquer estágio do ciclo de vida de desenvolvimento do software, sendo utilizado para:

- a) apoiar análises de qualidade e produtividade e complementar métricas de equipe;
- b) estimar o tamanho funcional de software a ser desenvolvido;
- c) estimar o tamanho, custo e os recursos necessários para projetos de desenvolvimento, melhoria, manutenções etc.;
- d) contagem de pontos de função simples da baseline da aplicação.

O método tem correspondência de conceitos, tipos de medição, fórmulas de cálculo do método APF tradicional, o que leva a identificação das mesmas funções a serem medidas. A definição de um processo elementar é a mesma.

Cada função transacional corresponde a um (1) Processo elementar no SFP e cada função de dados corresponde a um (1) arquivo lógico no SFP.

O método apresentado é consistente com a estrutura da família de padrões ISO 14.143 e totalmente compatível ao método de análise de pontos de função tradicional publicado pelo IFPUG.

## 5. PROCESSO DE CONTAGEM DO MÉTODO SFP

A figura a seguir representa o procedimento de contagem do método SFP, que é idêntico ao procedimento de contagem do método de Análise de Pontos de Função Tradicional. A diferença principal entre os dois métodos está na execução das atividades destacadas (em amarelo), descritas nos itens subsequentes:

**Figura 1: 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, o SPM v.2.1. A seguir, o detalhamento de cada etapa do procedimento de contagem de pontos de função usando o método SFP:

a) 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.

b) determinar:

- O propósito da contagem fornece uma resposta para uma questão de negócio a ser resolvida, por exemplo: necessidade de dimensionar um projeto de um novo sistema para auxiliar o processo de contratação do mesmo. Com base no propósito da contagem são definidos o escopo e o tipo de contagem.
- O tipo da contagem identifica se o projeto é de desenvolvimento, de melhoria ou aplicação instalada.
- O escopo da contagem identifica quais funcionalidades serão incluídas na contagem de pontos de função. O escopo pode incluir mais de um aplicativo de software.

A fronteira da aplicação é determinada pela identificação das aplicações e seus limites são orientados por princípios lógicos, não técnicos, focado no ponto de vista do usuário (em qualquer nível de abstração). Vale destacar que, assim como no

método de APF, no método SFP o posicionamento da fronteira influencia fortemente a contagem de pontos de função, portanto, em editais para contratação de projetos de manutenção é fortemente recomendado a definição das fronteiras de todas as aplicações a serem contratadas.

c) 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 no SFP não há diferenciação entre os arquivos lógicos que são apenas consultados e aqueles que são consultados e armazenam dados. No 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.

d) 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 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 1: Componentes funcionais básicos do método SFP**

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

e) Calcular o tamanho funcional: o cálculo do tamanho funcional para o método SFP requer apenas a identificação de dois componentes funcionais básicos: (1) Processo Elementar e (2) Arquivo Lógico. Para o método IFPUG SFP não são diferenciados os tipos de processo elementar e os tipos de arquivo lógico, tampouco as suas complexidades, não é necessária a identificação de DERs, ALRs ou RLRs, tampouco a identificação da “intenção primária”. Ainda assim, convém realçar que as regras de contagem de pontos de função simples - descritas no SPM 2.1 - devem ser observadas e seguidas. As pontuações a serem atribuídas para os dois componentes funcionais básicos são:

**Tabela 2: Pontuação dos Componentes funcionais básicos do método SFP**

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

d) Documentar e reportar a contagem: a última etapa do processo de medição é realizar a documentação da contagem com todas as premissas e decisões de medição tomadas, os padrões utilizados, as diretrizes adotadas e os links para a documentação do projeto. No SPM 2.1 é apresentado um conjunto mínimo de informações necessárias para a tarefa.

## 6. 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 PF e o método IFPUG SFP encontrou as seguintes evidências:

- a) correspondência dos conceitos de aplicação, escopo, limite e finalidade;
- b) correspondência de Tipos de Medição;
- c) correspondência dos componentes funcionais básicos;
- d) correspondência de fórmulas de cálculo: as fórmulas são semelhantes; há diferença apenas na fórmula usada para atualizar as linhas de base após aprimoramento funcional, uma vez que não há mudança na complexidade.

Os processos elementares aferidos por meio do método SFP podem ser classificados como Entrada Externa (EE), Consulta Externa (CE) ou Saída Externa (SE) e os arquivos lógicos no método SFP podem ser classificados como Arquivo Lógico Interno (ALI) ou Arquivo de Interface Externa (AIE).

A contabilização dos pontos de função por meio do método SFP consiste na aplicação da seguinte fórmula:

$$\text{Tamanho Funcional em SFP} = (\text{QtdCE} + \text{QtdEE} + \text{QtdSE}) \times 4,6 + (\text{QtdALI} + \text{QtdAIE}) \times 7$$

Onde

- Qtd = quantidade de PE- CE, EE e/ou SE contadas com o método de PF tradicional – para as quais é atribuída a pontuação de 4,6 SFP;
- Qtd = quantidade de AL - ALI e/ou AIE contados com o método de PF tradicional – para os quais é atribuída a pontuação de 7 SFP.

## 7. EXEMPLO DE APLICAÇÃO DA CONVERSIBILIDADE

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

- a) quantidade e tipos de processo elementar: 5 CE, 10 EE, 10 SE;
- b) quantidade e tipos de arquivo lógico: 5 ALI e 2 AIE. Aplicando a fórmula para conversibilidade de PF para SFP temos:

$$\text{SFP} = (5\text{CE} + 10\text{EE} + 10\text{SE}) \times 4,6 + (5\text{ALI} + 2\text{AIE}) \times 7$$

$$\text{SFP} = (25) \times 4,6 + (7) \times 7 = 164 \text{ SFP}$$

Convém destacar novamente que, para o método SFP, a contagem das funções transacionais, funções de dados e o cálculo do tamanho funcional:

- a) requer apenas a identificação de dois componentes funcionais básicos: Processo Elementar (PE) e Arquivo Lógico (AL);
- b) não é necessária a identificação da “intenção primária”;
- c) não é necessário diferenciar os tipos de processo elementar (EE, CE, SE) e os tipos de arquivo lógico (ALI, AIE), tampouco a sua complexidade;
- d) não é necessária a identificação de DERs, ALRs ou RLRs;
- e) a pontuação funcional do PE é de 4,6 SFP;
- f) a pontuação funcional do AL é de 7,0 SFP.

## **8. CÁLCULO DE PONTOS DE FUNÇÃO COM O MÉTODO SFP**

Conforme SPM 2.1, as fórmulas dos dois métodos são semelhantes (APF e SFP); a diferença está apenas na fórmula usada para atualizar a baseline após um projeto de melhoria, uma vez que não há mudança na complexidade dos componentes funcionais.

Nos subitens a seguir, serão apresentadas fórmulas para:

- Projetos de desenvolvimento;
- Linha de base após desenvolvimento inicial;
- Projeto de melhoria;
- Atualizar a baseline após um projeto de melhoria.

Para os demais tipos de projetos (migração de dados, manutenções, mudanças de plataforma, atualização de versão etc.), aplica-se o roteiro de métricas do SISP, versão 2.3, considerando as pontuações funcionais do método SFP (processo elementar igual a 4,6 SFP e arquivo lógico igual a 7,0 SFP)

### **8.1. Projeto de Desenvolvimento**

É o projeto para desenvolver e entregar a primeira versão de uma aplicação de software.

Ao criar um novo aplicativo de software, dois componentes devem ser considerados: as funções incluídas (SFP\_INCLUIDO) e as funções de conversão (SFP\_CONVERSÃO) que apoiam o início do uso do software, com a população de arquivos lógicos por exemplo.



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

$$\text{SFP\_DESENVOLVIMENTO} = \text{SFP\_INCLUIDO} + \text{SFP\_CONVERSÃO}$$

## 8.2. Linha de Base após Desenvolvimento Inicial

Ao final de um projeto de desenvolvimento inicial, a medição da linha de base do software vai considerar apenas as funções de transação e dados incluídas (SFP\_INCLUIDO); não vai considerar as funções de conversão (SFP\_CONVERSÃO).

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

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

$$\text{SFP\_BASELINE} = \text{SFP\_INCLUIDO}$$

## 8.3. Projeto de Melhoria

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

Segundo o CPM 4.3 e SPM 2.1, esta manutenção é considerada um tipo de projeto de manutenção adaptativa com mudanças em requisitos funcionais da aplicação, ou seja, com funcionalidades incluídas, alteradas ou excluídas na aplicação,

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

$$\text{SFP\_MELHORIA} = \text{SFP\_INCLUIDO} + (\text{FI}^* \times \text{SFP\_ALTERADO}) + (0,30 \times \text{SFP\_EXCLUIDO}) + \text{SFP\_CONVERSÃO}$$

Onde o FI\* (Fator de Impacto) pode variar de 50% a 90% conforme condições abaixo:

- FI = 50% para funcionalidade de sistema desenvolvida ou mantida por meio de um projeto de melhoria pela empresa contratada.
- FI = 75% para funcionalidade de sistema não desenvolvida ou mantida por meio de um projeto de melhoria pela empresa contratada e sem necessidade de redocumentação da funcionalidade.
- FI = 90% para funcionalidade de sistema não desenvolvida ou mantida por meio de um projeto de melhoria pela empresa contratada e com necessidade de redocumentação da funcionalidade. FI igual a 90% representa a adição de 15% como fator de redocumentação ao Fator de Impacto anterior (75%). Nesse caso, o Contratado deve redocumentar a funcionalidade mantida, gerando a

documentação completa da mesma, aderente ao processo de software da contratante. Se houver uma nova demanda de projeto de melhoria na funcionalidade em questão, será considerado que o Contratado desenvolveu a funcionalidade. Observe que o percentual de 90% apenas será considerado na primeira demanda de projeto de melhoria em cada funcionalidade.

#### 8.4. Baseline após Projeto de Melhoria

Como já exposto, o SPM 2.1 informa que as fórmulas dos dois métodos são semelhantes (APF e SFP); a diferença está apenas na fórmula usada para atualizar a baseline após um projeto de melhoria, uma vez que não há mudança na complexidade dos componentes funcionais.

Após um projeto de melhoria, a medição da baseline do aplicativo lançado (SFP\_BASELINE) será a medição baseline anterior ao projeto de melhoria, mais os novos recursos (SFP\_INCLUIDO), menos os recursos removidos (SFP\_EXCLUÍDO):

$$\text{SFP\_BASELINE} = \text{SFP\_BASELINE} + \text{SFP\_INCLUIDO} - \text{SFP\_EXCLUÍDO}$$

Para o método SFP, na contagem da baseline da aplicação não devem constar as funcionalidades alteradas em um projeto de melhoria, visto que estas já foram incluídas a partir de outro projeto e, portanto, já estão contempladas na baseline. Ademais, o método SFP não determina a complexidade das funções, não gerando assim alteração em sua pontuação. Também não devem constar funcionalidades de conversão, refinadas ou excluídas.

### 9. ESTIMATIVA DE TAMANHO FUNCIONAL COM SFP

Para realizar a estimativa de tamanho funcional de um software, deve-se realizar todas as etapas do procedimento de contagem exposto no item 5 deste roteiro.

Deve-se realizar a leitura de toda a documentação disponível buscando informações relevantes para identificação dos processos elementares e arquivos lógicos. Convém destacar que 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.

Após identificação dos processos elementares e arquivos lógicos, é atribuída a seguinte pontuação funcional:

**Tabela 2: Pontuação dos Componentes funcionais básicos do método SFP**

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

A estimativa de tamanho do projeto em SFP deve ser gerada com a totalização da pontuação de todos os processos elementares e arquivos lógicos identificados.

## **10. ESTIMATIVA DE ESFORÇO, PRAZO, CUSTO E RECURSOS**

Uma vez que o tamanho do projeto foi estimado, é possível estimar o esforço, prazo, custo e recursos para o desenvolvimento do software. Para tanto, devem ser utilizadas as fórmulas apresentadas no roteiro de métricas do SISP, versão 2.3.

## **11. CONCLUSÃO**

Este documento apresentou um roteiro para dimensionamento de tamanho funcional de software utilizando o método IFPUG SFP.

Destaca-se que este roteiro está alinhado à Portaria SGD/MGI nº 750, de 20 de março de 2023, que estabelece que independente da modalidade de contratação, deve-se aferir a entrega de produtos por meio de métricas de software.

Vale ressaltar que fica a cargo da CONTRATANTE definir o tipo de contagem final da aplicação, se via SFP ou contagem detalhada.

## **12. REFERÊNCIAS BIBLIOGRÁFICAS**

- SFP Counting Practices Manual by International Function Point Users Group (IFPUG), versão 2.1. Disponível em <https://ifpug.org/>.
- Manual de Práticas de Contagem de Pontos de Função, Versão 4.3.1 IFPUG. Disponível em: <https://ifpug.org/>.
- Roteiro de Métricas do SISP, versão 2.3. Disponível em: <https://www.gov.br/governodigital/pt-br/sisp/documentos/arquivos/roteiro-de-metricas-dosisp-v2-3.pdf>.
- Simple Function Point: a new Functional Size Measurement Method fully compliant with IFPUG
- 4.x. Disponível em: <https://www.researchgate.net/publication/258209664>