



Ministério do Meio Ambiente e Mudança do Clima
SE/SPOA/CGTI/CSISP

ROTEIRO DE MÉTRICAS

Ministério do Meio Ambiente e

Mudança do Clima - MMA



Ministério do Meio Ambiente e Mudança do Clima
SE/SPOA/CGTI/CSISP

Controle de Versão		
Data	Responsável	Alterações
27/03/2024	Elissany	Revisão das atividades de Qualidade e testes avançados
25/09/2023	Humberto	Documento Inicial
29/09/2023	Elissany	Revisão
15/12/2022	Humberto	Ajuste na Sustentação



1. INTRODUÇÃO

As métricas de software desempenham um papel fundamental na tomada de decisões, tanto no planejamento quanto na execução dos serviços de desenvolvimento e manutenção/sustentação de software.

Nas contratações de serviços de desenvolvimento, manutenção e sustentação de software, é essencial definir métricas objetivas que possibilitem a gestão contratual, a mensuração adequada e, por consequência, a remuneração adequada dos serviços e produtos entregues pela empresa contratada, de acordo com o processo de desenvolvimento de software adotado pelo órgão ou entidade.

De acordo com a Portaria SGD/MGI nº 750, de 20 de março de 2023, independentemente da modalidade contratual, as entregas devem ser aferidas por meio de métricas e os resultados devem ser mantidos em uma base histórica. Essa norma lista algumas métricas possíveis no item 12.3, como os Pontos de Função (IFPUG, NESMA, COSMIC, Simple Function Point - SFP), Linhas de Código Implementadas e Pontos de História (Story Point).

Neste roteiro, será apresentado o método de Ponto de Função Simples (Simple Function Point - SFP) publicado pelo International Function Point Users Group (IFPUG). Esse método possibilita a aferição da entrega de produtos por meio de métricas de software objetivas, de forma rápida e simples.

O método de Ponto de Função Simples (SFP) permite avaliar a medida de tamanho funcional de maneira 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 em adotar as mesmas definições de processos elementares e arquivos lógicos da contagem de pontos de função tradicional, garantindo, portanto, as mesmas características em termos de métrica de produto de software. Além disso, ele oferece as seguintes vantagens:

- É útil no início do processo de desenvolvimento para estimar o esforço e o tamanho funcional a ser implementado;
- Independente da metodologia de desenvolvimento e tecnologia adotada;
- Oferece resultados confiáveis, repetíveis e objetivos;
- Foi projetado para ser ágil, rápido, leve e de fácil utilização, acelerando consideravelmente o processo de dimensionamento funcional.



O detalhamento do SFP neste roteiro não exclui a utilização de outras métricas como IFPUG, NESMA, Pontos de história (Story Point), Roteiro de Métricas do SISP, por exemplo, pelo MMA, sendo SFP de utilização preferencial e as demais poderão ser adotadas em casos específicos a critério da área técnica do órgão.

Ademais, este roteiro trata também da mensuração de itens que não são medidos por meio do SFP, uma vez que não representam a entrega de um incremento de software, visto a métrica supramencionada medir o tamanho funcional de um software. Assim, faz-se necessário o estabelecimento de parâmetros para mensuração dessas entregas específicas, sendo abordados nas seções finais deste documento.

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:

- a. **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).
- b. **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.



- c. **Arquivo lógico:** representa a funcionalidade fornecida ao usuário para atender requisitos de armazenamento de dados internos e externos.
- d. **Baseline:** corresponde ao tamanho funcional de uma aplicação. Este tamanho fornece uma medida de funções atuais que o aplicativo fornece ao usuário.
- e. **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.
- f. **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). ALLs, AIEs, EEs, SEs, CEs são os tipos de CFBs (BFCs).
- g. **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.
- h. **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.
- i. **Funcionalidade de conversão:** funções transacionais ou de dados fornecidas para converter dados e/ou fornecer outros requisitos de conversão especificados pelo usuário.
- j. **IFPUG:** Grupo Internacional de Usuários de Ponto de Função
- k. **Incremento de produto:** versão de um produto que pode ser liberada no final de um período (timebox).
- l. **Medição:** A tarefa de medir e seu resultado, atribuindo um valor a um atributo em de acordo com uma escala de referência.
- m. **Método de medição:** Uma sequência lógica de operações realizadas para produzir medidas.
- n. **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.

- o. **Níveis mínimos de serviço:** são regras objetivas e fixas que estipulam valores e/ou características mínimas de atendimento a uma meta a ser cumprida pelo Contratado na prestação dos serviços.
- p. **Produto de Software ou Software:** conjunto de programas, procedimentos, rotinas ou scripts, componentes, Application Programming Interface - API, webservices, incluindo os dados e documentação associada.
- q. **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.
- r. **Projeto ágil:** projeto de desenvolvimento de software que utiliza abordagem de desenvolvimento ágil.
- s. **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).
- t. **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.
- u. **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.
- v. **Requisitos do usuário:** requisitos que descrevem o que o usuário está pedindo.
- w. **Requisitos funcionais:** conjunto de requisitos do usuário que descrevem o que o software deve fazer em termos de tarefas e serviços.
- x. **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.
- y. **Simple Function Point - SFP:** Pontos de função simples.



- z. **SISP:** Sistema de Administração dos Recursos de Tecnologia da Informação do Poder Executivo Federal.
- aa. **SPM:** Manual de práticas de contagem do método Ponto de Função Simples.
- bb. **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.
- cc. **Tamanho funcional (ISO 14143-1: 2007 (R2019)):** Tamanho do software derivado da quantificação os Requisitos Funcionais do Usuário.
- dd. **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.

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:

Componentes funcionais básicos do IFPUG SFP	Pontuação
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:

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

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 SF.

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:

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

Componentes funcionais básicos do IFPUG SFP	Pontuação
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

Depois de estimado o tamanho do projeto, é possível estimar o esforço, prazo, custo e recursos para o desenvolvimento do software. Todas essas definições são feitas baseadas no Roteiro de métricas do SISF, versão mais atualizada.

A estimativa de prazo inicial do projeto é baseada no escopo, o qual é composto por um conjunto de requisitos funcionais genéricos. Parte-se do pressuposto que nessa fase a concepção do sistema é vaga e que a lista apresentada pelo PO não é exaustiva. Além disso, o ponto de função não mensura aspectos de requisitos não funcionais.

10.1 Dos Prazos

Na execução das atividades, tarefas e serviços de **Desenvolvimento e Manutenção de software** deverão ser observados os seguintes prazos máximos:

Atividade, Tarefa ou Serviço	Prazo máximo de início da execução serviço/tarefa/atividade	Prazo Máximo de término da execução do serviço/tarefa/atividade
Executar a Sprint	1 dia útil	até 4 semanas (Conforme cálculo de prazo).
Construir a visão do negócio	1 dia útil	até 5 dias úteis
Planejar o roadmap do produto	1 dia útil	até 3 dias úteis
Elaborar backlog do produto	1 dia útil	até 3 dias úteis
Planejar o release	1 dia útil	até 2 dias úteis
Elaborar Help / Manual de Usuário de Sistema em	1 dia útil	até 3 dias úteis



Desenvolvimento (por funcionalidade)		
--------------------------------------	--	--

Na execução das atividades de **Sustentação de software** deverão ser observados os seguintes prazos máximos:

Atividade, Tarefa ou Serviço	Prazo Máximo de início do atendimento	Prazo Máximo de solução do problema
Correção de não conformidades identificadas como de baixo impacto, que não impede o uso do sistema e não provoque prejuízo significativo ao negócio.	4 horas úteis	24 horas úteis
Correção de não conformidades identificadas como de impacto moderado, que compromete o uso do sistema (sem impedimento) e não provoque prejuízo significativo ao negócio.	2 horas corridas	24 horas corridas
Correção de não conformidades identificadas como de alto impacto, que impede o uso do sistema ou provoque prejuízo significativo ao negócio ao ocorrer.	30 minutos corridos	8 horas corridas



Realização de atualização ou modificação de código não classificados como correção.	4 horas úteis	24 horas úteis
---	---------------	----------------

Destaca-se que esses prazos são aplicáveis às correções dada a necessidade de maior celeridade de sua implementação, porém, as pequenas evolutivas que são classificadas como sustentação, aplicam-se os prazos de manutenção e desenvolvimento.

Na execução dos serviços de **Qualidade e Testes avançados** deverão ser observados os seguintes prazos máximos:

Serviço Técnico	Descrição do Serviço	Unidade de Medida	Entregável / Produto	Meta de Produtividade em horas
Elaborar caso de testes funcionais ou não funcionais	Elaborar casos de teste funcionais ou não funcionais de uma aplicação.	por caso de teste	Caso de Teste	10h
Executar, monitorar, controlar e analisar testes funcionais e não funcionais	Desenvolver atividades para executar, monitorar, controlar e analisar testes funcionais e não funcionais, utilizando técnicas de teste manuais e automatizadas. A exemplo de testes funcionais de sistema, testes exploratórios, testes de integração, testes de desempenho, testes de carga, testes de estresse, etc	por relatório	Relatório de testes	40h
Executar, monitorar, controlar e analisar testes de Segurança e Vulnerabilidades	Executar procedimentos de testes de segurança. Testes de segurança estáticos (SAST – Static Application Security Testing) e testes de segurança dinâmicos (DAST – Dynamic Application Security Testing).	por relatório	Relatório de testes	24h
Elaborar Plano de Testes	Desenvolver atividades para criação do Plano Testes de um projeto, usado para planejamento e gerenciamento das atividades de teste. Descreve as decisões tomadas durante o planejamento inicial do projeto e deve apresentar	por plano de testes do projeto	Plano de testes do projeto	16h



	uma descrição detalhada dos objetivos a serem alcançados. O plano de testes do projeto inclui a estratégia de teste específica do projeto e o agendamento dos testes.			
Atualizar Plano de Testes	Desenvolver atividades para atualização do Plano Testes Mestre de um projeto; documentação usada para planejamento e gerenciamento das atividades de teste.	por plano de testes do projeto	Plano de testes do projeto atualizado	8h
Apoiar a fiscalização técnica dos contratos de desenvolvimento, manutenção e sustentação de software	Apoiar a fiscalização técnica dos contratos de desenvolvimento, manutenção e sustentação de software, atuando para garantir que os softwares entregues tenham sido testados de acordo com as melhores práticas de mercado e os padrões previstos no Termo de Referência, apontando as falhas e/ou oportunidades de melhoria nos processos de desenvolvimento e testes, bem como nos produtos resultantes destes processos.	por relatório	Relatório Técnico de Análise de Qualidade	16h
Emitir pareceres técnicos	Emitir pareceres técnicos relacionados ao ambiente de sistemas da organização incluindo diagnóstico de situações de gargalos e problemas de desempenho	por relatório	Relatório Técnico de Análise de Qualidade	16h
Elaborar Artefatos /Templates	Elaborar Artefatos tais como Listas de Verificação, Critérios de Aceite.	por artefato	Artefatos tais como Listas de Verificação, Critérios de Aceite.	4h
Suporte ao usuário na execução de testes	Apoiar o usuário na execução de testes do sistema.	por relatório	Relatório de testes	4h
Executar Testes de Acessibilidade	Executar teste de Acessibilidade de Software para testar o grau de facilidade de uso do software	por relatório	Relatório de testes	4h



	por pessoas com determinadas deficiências ou limitações em decorrência da idade avançada.			
Revisão de Código e documentação associada	Verificar partes de código em busca de erros e bugs com o fim de melhorar a qualidade do código e, conseqüentemente, a qualidade do software. A lista de verificação de revisão inclui, mas não está limitada a: testes de depuração, verificações de segurança do código, legibilidade, documentação associada, testes automatizados, etc.	por software	Lista de verificação de código	40h

Na execução das atividades de **Desenvolvimento e Manutenção de painéis e ambientes de analytics** deverão ser observados os seguintes prazos máximos:

Tipo de Atividade	Descrição da Atividade	Unidade de medida	Esforço de Referência (horas)
DW e Analytics	Construção do job ETL	Por Job	10
DW e Analytics	Construção de job para geração de bases de dados para treinamento, validação e testes	Por Job	1
Big Data	Realizar suporte técnico em Analytics	Por atendimento	8
Big Data	Instalar serviços/componentes	Por serviço ou componente	16
Big Data	Configurar serviços/componentes	Por serviço ou componente	8
Big Data	Realizar pesquisa técnica de componentes	Por relatório	24



Ministério do Meio Ambiente e Mudança do Clima
SE/SPOA/CGTI/CSISP

Big Data	Executar testes	Por Teste	24
Big Data	Elaborar roteiro de instalação/configuração	Por serviço ou componente	24
Big Data	Elaborar script de automação	Por tarefa	24
Análise e exploração de dados	Construir/alterar relatório utilizando ferramentas de visualização de dados	Por relatório	2
Análise e exploração de dados	Construir/alterar gráfico utilizando ferramentas de visualização de dados	Por gráfico	2
Análise e exploração de dados	Construir/alterar Indicador utilizando ferramentas de visualização de dados	Por indicador	1
Análise e exploração de dados	Construir/alterar Dashboard utilizando ferramentas de visualização de dados	Por dashboard	8
Análise e exploração de dados	Mapear Objeto de Dados	Por objeto/tabela	1
Análise e exploração de dados	Construir/alterar funções, scripts ou métricas calculadas utilizadas em ferramentas de visualização de dados	Por função, script ou métrica	2



Web Analytics	Construir ou alterar script para a criação de imagens de containers	Por arquivo de script	16
Web Analytics	Realizar pesquisa técnica de componentes	Por relatório	24
Inteligência Artificial	Elaborar query para Inteligência Artificial	Objeto/Tabela	3
Machine Learning	Planejamento da solução	Por relatório	40
Machine Learning	Treinamento/Retreinamento de Modelo e Análise de Resultados	Por Iteração de Treinamento	52
Machine Learning	Parametrização/Configuração de Modelo	Por modelo	20
Machine Learning	Pesquisa técnica de Modelos e Métodos	Por Modelo Avaliado	20
Machine Learning	Pré-processamento de dados	Por script	30
Machine Learning	Realizar exploração ou modelagem de dados	Por relatório	20

10.2 Da Produtividade

A produtividade é a relação da capacidade de produção com o tempo. Conforme a Portaria nº750, item 8.7.6, alínea a), considera-se que, no geral, os projetos ágeis têm sua produtividade de 10hh/PF. O Roteiro de Métricas do SISP usa como parâmetro de produtividade a complexidade do projeto.

A produtividade de 0,58 PF/dia equivale a 14hh/PF, padrão utilizado na contratação centralizada de Desenvolvimento e Manutenção de software.

No MMA a meta de produtividade seguirá a complexidade do projeto, conforme quadro abaixo:



Complexidade do projeto	Meta de produtividade mínima por desenvolvedor
Baixa	7hh/PF
Média	10hh/PF
Alta	14hh/PF

Ademais, é importante observar que em alguns casos será necessária a alocação de um profissional que não entrega o time ágil, para realização de uma entrega específica. Assim, a produtividade desse profissional será fracionada conforme o quadro de esforço abaixo para macroatividades isoladas:

Macroatividades do Processo de Software	Percentual de esforço
Engenharia de requisitos	25%
Design/Arquitetura	10%
Implementação	40%
Testes	15%
Homologação	5%
Implantação	5%

10.3 Definição da complexidade de projeto

Em geral, projetos típicos devem ser estimados com a complexidade média. Os projetos de manutenções que a equipe possua conhecimento do sistema podem ser estimados com a complexidade baixa. Os projetos com requisitos não funcionais complexos podem ser estimados com a complexidade alta.

A tabela abaixo é uma referência para definição da complexidade do projeto e a sua finalidade é fornecer alguns parâmetros a serem considerados por impactarem tanto a produtividade quanto o prazo. Essas questões são relevantes e devem ser consideradas no início do projeto.

Fator	Descrição	1	2	3	4	5
--------------	------------------	----------	----------	----------	----------	----------



Requisitos Ambíguos	Requisitos pouco claros ou mal definidos dificultam o planejamento e a implementação.					
Escopo Vago	Escopo pouco definido ou em constante mudança pode levar a incertezas e retrabalho.					
Integrações Externas	Integração com sistemas externos pode gerar desafios de compatibilidade e segurança.					
Requisitos Não Funcionais	Necessidades de desempenho, segurança e usabilidade podem aumentar a complexidade.					
Tecnologias Emergentes	O uso de tecnologias novas ou pouco conhecidas pode requerer mais pesquisa e aprendizado.					
Regulamentações	Projetos que devem cumprir normas e regulamentos específicos podem ser mais complexos.					
Conformidade	A necessidade de aderir a padrões de desenvolvimento pode aumentar a complexidade.					
Escalabilidade	Projetos que precisam lidar com grande volume de dados ou usuários requerem soluções escalonáveis.					
Restrições de Tempo	Prazos apertados podem levar a decisões apressadas e aumentar a chance de erros.					
Necessidades Específicas	Projetos que atendem a requisitos exclusivos ou personalizados são mais complexos.					
Mudanças Frequentes	Alterações constantes no projeto podem dificultar o gerenciamento e a estabilidade.					
Segurança Crítica	Projetos com exigências rigorosas de segurança podem apresentar desafios adicionais.					
Interação com Hardware	Projetos que envolvem interação com dispositivos de hardware podem ser mais complexos.					
Integração com outros sistemas	Projetos que envolvem integração com outros sistemas podem ser mais complexos.					

10.4 Indicador de Complexidade



O indicador tem finalidade de ser uma referência com base nos intervalos definidos abaixo.

- Baixo – 13 a 31
- Médio - 32 a 48
- Alto – 49 a 65

Essa classificação fica a critério do Líder do projeto que, nos momentos iniciais do projeto, com base no documento de visão e reunião com as partes interessadas, por exemplo, define o projeto como de complexidade baixa, média ou alta.

Isso não significa que este rótulo seja estático, um projeto inicialmente classificado como de baixa complexidade, por exemplo, após expansão das funcionalidades do projeto e inclusão de requisitos de segurança de alta criticidade, pode ser reclassificado como de alta complexidade.

O intuito não é a exatidão, mas buscar que a característica do projeto reflita razoavelmente os prazos e produtividade esperados.

11. CONCLUSÃO

Este documento apresentou um roteiro para dimensionamento de tamanho funcional de software utilizando o método IFPUG SFP, bem como a mensuração de itens que não são medidos por meio do 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.

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-do-sisp-v2-3.pdf>.



Ministério do Meio Ambiente e Mudança do Clima
SE/SPOA/CGTI/CSISP

- **Simple Function Point: a new Functional Size Measurement Method fully compliant with IFPUG 4.x.** Disponível em: <https://www.researchgate.net/publication/258209664>