

**UNIVERSIDADE SÃO JUDAS TADEU**  
**BACHARELADO EM SISTEMA DE INFORMAÇÃO**

**Geovane Augusto Costa dos Santos**

**Natan Fernandes Araujo Ibiapina**

**Henryk Bagdanovicius Roza**

**Olivia Frankiw de Carvalho**

**Gabriela Alves**

**João Luiz Santana Borean**

**MÉTRICAS DE SOFTWARE**

**São Paulo**

# 1. Resumo: Principais Métricas de Software

Fonte: <https://www.computerweekly.com/br/tip/23-metricas-de-desenvolvimento-desoftware-que-devem-ser-monitoradas>

As organizações de TI utilizam Indicadores-Chave de Desempenho (KPIs) para compreender o progresso no desenvolvimento de software, a qualidade do produto final, seu desempenho em operação e a satisfação dos usuários. Essas métricas podem ser agrupadas em quatro categorias principais:

## Métricas de Produtividade do Desenvolvedor

Focam na eficiência da equipe e no resultado do trabalho.

### Exemplos:

- **Tempo de entrega (Lead Time):** Tempo total desde a concepção de uma tarefa até sua conclusão.
- **Trabalho em Progresso (WIP):** Quantidade de tarefas iniciadas e ainda não finalizadas.
- **Velocidade Ágil:** Quantidade de trabalho (pontos de história/histórias de usuário) que a equipe consegue concluir em um sprint.

## Métricas de Desempenho do Software

Medem atributos não funcionais que indicam a eficiência e estabilidade do software.

### Exemplos:

- **Tempo de Resposta:** Tempo que o sistema leva para responder a uma solicitação.
- **Desempenho (Throughput):** Quantidade de dados processada em um determinado período.
- **Disponibilidade:** Percentual de tempo que o software está operacional e acessível.

## Métricas Baseadas em Defeitos e Segurança

Avaliam a quantidade de falhas, a eficácia dos testes e a segurança do software.

### Exemplos:

- **Densidade de Defeitos:** Número de defeitos por unidade de código (ex: KLOC).
- **Dívida Técnica:** Esforço futuro implícito causado por escolhas de curto prazo no desenvolvimento.
- **Tempo Médio para Reparo (MTTR):** Tempo médio necessário para corrigir uma falha após sua detecção.

## Métricas de Usabilidade e Experiência do Usuário (UX)

Avaliam a facilidade de uso do software e a percepção geral dos usuários.

### Exemplos:

- **Métricas de Usabilidade:** Componentes como facilidade de aprendizado, eficiência e satisfação do usuário.
- **Net Promoter Score (NPS):** Mede a probabilidade de os clientes recomendarem o software a outros.

Embora haja um vasto conjunto de métricas, é crucial que cada organização de TI identifique e priorize aquelas que melhor se alinham aos seus requisitos e objetivos estratégicos para monitorar a qualidade e o sucesso de seus projetos de software.

## 2. Resumo: Guia de Métricas de Software para FINEP

Fonte:

[http://www.finep.gov.br/images/licitacoes/2017/Consulta012017/II\\_GuiaDeMetricasDeSoftware.pdf](http://www.finep.gov.br/images/licitacoes/2017/Consulta012017/II_GuiaDeMetricasDeSoftware.pdf)

Este Guia tem como objetivo apresentar as métricas de software a serem aplicadas no desenvolvimento e manutenção de software pela Finep, visando apoiar o relacionamento com fornecedores e subsidiar mediações ou arbitragens. O documento estabelece uma ordem de precedência para aplicação de regras, priorizando este Guia,

seguido pelo Roteiro de Métricas de Software do SISP e, por último, o Manual de Práticas de Contagem (CPM) do IFPUG.

O guia é dividido em duas partes principais:

## Parte I – Métrica Ponto de Função (PF)

Esta parte detalha a aplicação da Análise de Pontos de Função (APF), aderente ao CPM do IFPUG, como complemento para mensurar o tamanho funcional de projetos de software. Aborda:

- **Níveis de Contagem:** Define contagens Indicativa, Estimada (usando NESMA para fases iniciais) e Detalhada (para pagamento após validação da entrega).
- **Orientações para Contagens Detalhadas:** Fornece regras específicas da Finep para situações como:
  - Alterações em funções existentes (diferenciando projetos tradicionais e ágeis, com conceitos de Melhoria e Refinamento);
  - Manutenção de componentes reutilizáveis e templates;
  - Contagem de consultas (implícitas, filtros diferentes, saídas diferentes);
  - Subdivisão de funcionalidades (abas, telas encadeadas, salvamento de rascunho);
  - Integração entre aplicações (via serviços, fronteiras distintas ou iguais, dados técnicos);
  - Gravação de dados acessórios (histórico, trilha de auditoria, logs – estes últimos não são mensurados);
  - Business Intelligence (segue roteiro específico do SISP).
- **Itens Não Mensuráveis Diretamente por PF:** Define fatores de conversão ou abordagens específicas para:
  - Web Design (tabelas com equivalência em PF para estilo, layout, menus, wireframes, ilustrações, etc.);
  - CMS Joomla (tabela com equivalência em PF para desenvolvimento/alteração de componentes, plugins, conteúdo, formulários, configuração, migração);
  - Demais Serviços (conversão baseada em horas: 1 PF = 10 horas).

## Parte II – Métrica Unidade de Serviço Técnico (UST)

Esta parte introduz a métrica UST, destinada exclusivamente a medir o desenvolvimento e manutenção de funcionalidades de Middleware (BPM, SOA, ECM).

- **Cálculo:** A UST é calculada pela fórmula  $UST = COMP \times BPM \times ECM \times SERV \times PI$ .
- **Componentes do Cálculo:**
  - Agregador: Módulo ou subsistema que agrupa cenários;
  - Cenário: Funcionalidade a ser medida (ex: “Cadastrar Funcionário”);
  - COMP: Fator de Complexidade do cenário (Baixa, Média, Alta) baseado nos elementos de interface;
  - SERV, BPM, ECM: Fatores que indicam se o cenário implementa Serviços (SOA), fluxo BPM ou integra com ECM;
  - PI (Pontos de Interface): Soma dos Pontos de Regras de Negócio (PRN), Pontos de Regras de Apresentação (PRA) e Pontos de Integração (INT), cujos valores são atribuídos com base na quantidade de regras/integrações identificadas no cenário.

O guia visa, portanto, oferecer um conjunto de regras e procedimentos para a medição objetiva dos serviços de software contratados pela Finep, adaptando métricas padrão como APF e introduzindo a UST para contextos específicos como middleware.

### 3. Aplicação de Métricas Ágeis no Desenvolvimento de Software

Fonte: <https://ateliware.com/blog/metricas-ageis-no-desenvolvimento-de-software>

A página está fora do ar.

### 4. Resumo: Roteiro de Métricas de Software do SISP

Fonte: <https://www.gov.br/governodigital/pt-br/sisp/documentos/arquivos/roteiro-demetricas-do-sisp-v2-3.pdf>

O Roteiro de Métricas de Software do SISP – Versão 2.3 tem como objetivo estabelecer diretrizes para mensuração do tamanho funcional de projetos de software por meio da métrica Ponto de Função (PF), conforme definido pelo Manual de Práticas de Contagem (CPM 4.3) do IFPUG.

O documento adapta o uso da métrica às necessidades dos órgãos integrantes do Sistema de Administração dos Recursos de Tecnologia da Informação (SISP), promovendo maior objetividade na contratação de serviços de desenvolvimento e manutenção de sistemas.

São abordados os procedimentos de contagem funcional, estimativas de esforço, custo e prazo, bem como critérios para classificação de projetos em categorias como:

- Desenvolvimento;
- Melhoria;
- Manutenção corretiva;
- Migração de dados;
- Mudanças de plataforma;
- Atualização de versões.

O roteiro também propõe metodologias para uso da métrica em ambientes com métodos ágeis e descreve atividades que não devem ser consideradas na contagem. Além disso, recomenda a documentação adequada dos requisitos e fronteiras das aplicações, com vistas à padronização e consistência na medição.

O documento contribui para a profissionalização das contratações públicas de software ao substituir modelos baseados em homem-hora por métricas funcionais, promovendo maior controle, previsibilidade e transparência. A versão 2.3 incorpora ajustes com base em recomendações da CGU e reforça a importância da capacitação de profissionais na aplicação correta das regras de contagem.