UNIVERSIDADE FEDERAL DE SERGIPE
Av. Marcelo Deda Chagas, s/n, - Bairro Rosa Elze, São Cristóvão/SE, CEP 49107-230

TERMO DE REFERÊNCIA

ANEXO I - MODELO DE ORDEM DE SERVIÇO

1. MODELO DE ORDEM DE SERVIÇO PARA SERVIÇOS DESENVOLVIMENTO, MANUTENÇÃO E SUSTENTAÇÃO DE SOFTWARE)

MODELO DE ORDEM DE SERVIÇO				
1 – IDENTIFICAÇÃO				
Nº da OS	xxxx/aaaa	**Data de emissão**		<dd/mm/aaaa a>
Contrato nº	xx/aaaa			
Objeto do Contrato	<Descrição do objeto do contrato>			
Contratado	<Nome do contratado>	**CNPJ**	99.999.999/9999-99	
Preposto	<Nome do preposto>			
Início vigência	<dd/mm/aaaa>	**Fim vigência**	<dd/mm/aaaa>	
2 - ÁREA REQUISITANTE				
Unidade	< Sigla – Nome da unidade>			
Solicitante	<Nome do solicitante>	**E-mail**	xxxxxxxxxxxx	
3 – OBJETIVO DA ORDEM DE SERVIÇO				
Exemplo: Disponibilizar o Módulo de apuração de valores a receber aos usuários externos da solução de arrecadação				
4 - QUANTITATIVO DE PROFISIONAIS DE TI A SEREM ALOCADOS				

Para Desenvolvimento e Manutenção de Software		
Perfil Profissional (*)	**Quantidade de profissionais**	**Memória de cálculo do quantitativos de profissionais / Justificativa**

(*) Informar apenas perfis a serem alocados para a execução dos serviços de Desenvolvimento e Manutenção de Software.

(**) Informar percentual de alocação dos perfis profissionais, observando os "Requisitos de Formação da Equipe" estabelecidos no termo de referência.

Para Sustentação de Software		
Perfil Profissional(*)	**Quantidade de profissionais**	**Memória de cálculo do quantitativos de profissionais / Justificativa**

(*) Informar apenas perfis a serem alocados para a execução dos serviços de sustentação de Software.

5 - META DE PRODUTIVIDADE

Exemplo:

- *0,8 ponto de função por dia por desenvolvedor,*
- *33,33 Linhas de Código por dia por desenvolvedor,*
- *6 pontos de história por Sprint por desenvolvedor.*

6 - PERÍODO DE ALOCAÇÃO

<dd/mm/aaaa> a <dd/mm/aaaa> - <número de dias corridos>

7 - PRAZO MÁXIMO PARA ALOCAÇÃO

15 dias úteis

8– ASSINATURA E ENCAMINHAMENTO DA DEMANDA

Autoriza-se a execução dos serviços correspondentes à presente OS, no período e nos quantitativos acima identificados.

<Local>, xx de xxxxxxxxx de xxxx

_____

<Nome >

<Responsável pela demanda/ Fiscal Requisitante>

Matr.: <Nº da matrícula>

_____

<Nome >

<Responsável pela demanda/ Fiscal Requisitante>

Matr.: <Nº da matrícula>

UNIVERSIDADE FEDERAL DE SERGIPE
Av. Marcelo Deda Chagas, s/n, - Bairro Rosa Elze, São Cristóvão/SE, CEP 49107-230

TERMO DE REFERÊNCIA

ANEXO II - MODELO DE TERMO DE CIÊNCIA

Contrato Nº:			
Objeto:			
Contratante:			
Gestor do Contrato:		Matr.:	
Contratado:		CNPJ:	
Preposto do Contratado:		CPF:	

Por este instrumento, os funcionários abaixo-assinados declaram ter ciência e conhecer o teor do Termo de Compromisso de Manutenção de Sigilo e as normas de segurança vigentes no Contratante.

CONTRATADO	
Funcionários	
<Nome> Matrícula: <Matr.>	<Nome> Matrícula: <Matr.>
<Nome> Matrícula: <Matr.>	<Nome> Matrícula: <Matr.>
<Nome> Matrícula: <Matr.>	<Nome> Matrícula: <Matr.>
<Nome> Matrícula: <Matr.>	<Nome> Matrícula: <Matr.>

_____, _____ de _____de

20_____. Ciência

UNIVERSIDADE FEDERAL DE SERGIPE
Av. Marcelo Deda Chagas, s/n, - Bairro Rosa Elze, São Cristóvão/SE, CEP 49107-230

TERMO DE REFERÊNCIA

ANEXO III - MODELO DE TERMO DE COMPROMISSO DE MANUTENÇÃO DE SIGILO

O <NOME DO ÓRGÃO>, sediado em <ENDEREÇO>, CNPJ n.º <CNPJ>, doravante denominado Contratante, e, de outro lado, a <NOME DA EMPRESA>, sediada em <ENDEREÇO>, CNPJ n.º <CNPJ>, doravante denominada CONTRATADO;

CONSIDERANDO que, em razão do CONTRATO N.º XX/20XX doravante denominado CONTRATO PRINCIPAL, o CONTRATADO poderá ter acesso a informações sigilosas do Contratante;

CONSIDERANDO a necessidade de ajustar as condições de revelação destas informações sigilosas, bem como definir as regras para o seu uso e proteção;

CONSIDERANDO o disposto na Política de Segurança da Informação do Contratante;

Resolvem celebrar o presente TERMO DE COMPROMISSO DE MANUTENÇÃO DE SIGILO, doravante TERMO, vinculado ao CONTRATO PRINCIPAL, mediante as seguintes cláusulas e condições:

Cláusula Primeira – DO OBJETO

Constitui objeto deste TERMO o estabelecimento de condições específicas para regulamentar as obrigações a serem observadas pelo Contratado, no que diz respeito ao trato de informações sigilosas, disponibilizadas pelo Contratante, por força dos procedimentos necessários para a execução do objeto do CONTRATO PRINCIPAL celebrado entre as partes e em acordo com o que dispõem a Lei nº 12.527, de 18/11/2011 e os Decretos nº 7.724, de 16/05/2012 e 7.845, de 14/11/2012, que regulamentam os procedimentos para acesso e tratamento de informação classificada em qualquer grau de sigilo.

Cláusula Segunda – DOS CONCEITOS E DEFINIÇÕES

Para os efeitos deste TERMO, são estabelecidos os seguintes conceitos e definições:

INFORMAÇÃO: dados, processados ou não, que podem ser utilizados para produção e transmissão de conhecimento, contidos em qualquer meio, suporte ou formato.

INFORMAÇÃO SIGILOSA: aquela submetida temporariamente à restrição de acesso público em razão de sua imprescindibilidade para a segurança da sociedade e do Estado.

CONTRATO PRINCIPAL: contrato celebrado entre as partes, ao qual este TERMO se vincula.

Cláusula Terceira – DA INFORMAÇÃO SIGILOSA

Serão consideradas como informação sigilosa, toda e qualquer informação classificada ou não nos graus de sigilo ultrassecreto, secreto e reservado. O TERMO abrangerá toda informação escrita, verbal, ou em linguagem computacional em qualquer nível, ou de qualquer outro modo apresentada, tangível ou intangível, podendo incluir, mas não se limitando a: ***know-how***, técnicas, especificações, relatórios, compilações, código fonte de programas de computador na íntegra ou em partes, fórmulas, desenhos, cópias, modelos, amostras de ideias, aspectos financeiros e econômicos, definições, informações sobre as atividades do Contratante e/ou quaisquer informações técnicas/comerciais relacionadas/resultantes ou não ao CONTRATO PRINCIPAL, doravante denominados INFORMAÇÕES, a que diretamente ou pelos seus empregados, o Contratado venha a ter acesso, conhecimento ou que venha a lhe ser confiada durante e em razão das atuações de execução do CONTRATO PRINCIPAL celebrado entre as partes.

Cláusula Quarta – DOS LIMITES DO SIGILO

As obrigações constantes deste TERMO não serão aplicadas às INFORMAÇÕES que:

I – Sejam comprovadamente de domínio público no momento da revelação, exceto se tal fato decorrer de ato ou omissão do Contratado;

II – Tenham sido comprovadas e legitimamente recebidas de terceiros, estranhos ao presente TERMO;

III – Sejam reveladas em razão de requisição judicial ou outra determinação válida do Governo, somente até a extensão de tais ordens, desde que as partes cumpram qualquer medida de proteção pertinente e tenham sido notificadas sobre a existência de tal ordem, previamente e por escrito, dando a esta, na medida do possível, tempo hábil para pleitear medidas de proteção que julgar cabíveis.

Cláusula Quinta – DOS DIREITOS E OBRIGAÇÕES

As partes se comprometem a não revelar, copiar, transmitir, reproduzir, utilizar, transportar ou dar conhecimento, em hipótese alguma, a terceiros, bem como a não permitir que qualquer empregado envolvido direta ou indiretamente na execução do CONTRATO PRINCIPAL, em qualquer nível hierárquico de sua estrutura organizacional e sob quaisquer alegações, faça uso dessas INFORMAÇÕES, que se restringem estritamente ao cumprimento do CONTRATO PRINCIPAL.

Parágrafo Primeiro – O Contratado se compromete a não efetuar qualquer tipo de cópia da informação sigilosa sem o consentimento expresso e prévio do Contratante.

Parágrafo Segundo – O Contratado compromete-se a dar ciência e obter o aceite formal da direção e empregados que atuarão direta ou indiretamente na execução do CONTRATO PRINCIPAL sobre a existência deste TERMO bem como da natureza sigilosa das informações.

I – O Contratado deverá firmar acordos por escrito com seus empregados visando garantir o cumprimento de todas as disposições do presente TERMO e dará ciência ao Contratante dos documentos comprobatórios.

Parágrafo Terceiro – O Contratado obriga-se a tomar todas as medidas necessárias à proteção da informação sigilosa do Contratante, bem como evitar e prevenir a revelação a terceiros, exceto se devidamente autorizado por escrito pelo Contratante.

Parágrafo Quarto – Cada parte permanecerá como fiel depositária das informações reveladas à outra parte em função deste TERMO.

I – Quando requeridas, as INFORMAÇÕES deverão retornar imediatamente ao proprietário, bem como todas e quaisquer cópias eventualmente existentes.

Parágrafo Quinto – O Contratado obriga-se por si, sua controladora, suas controladas, coligadas, representantes, procuradores, sócios, acionistas e cotistas, por terceiros eventualmente consultados, seus empregados e contratados, assim como por quaisquer outras pessoas vinculadas ao Contratado, direta ou indiretamente, a manter sigilo, bem como a limitar a utilização das informações disponibilizadas em face da execução do CONTRATO PRINCIPAL.

Parágrafo Sexto - O Contratado, na forma disposta no parágrafo primeiro, acima, também se obriga a:

I. Não discutir perante terceiros, usar, divulgar, revelar, ceder a qualquer título ou dispor das INFORMAÇÕES, no território brasileiro ou no exterior, para nenhuma pessoa, física ou jurídica, e para nenhuma outra finalidade que não seja exclusivamente relacionada ao objetivo aqui referido, cumprindo-lhe adotar cautelas e precauções adequadas no sentido de impedir o uso indevido por qualquer pessoa que, por qualquer razão, tenha acesso a elas;
II. Responsabilizar-se por impedir, por qualquer meio em direito admitido, arcando com todos os custos do impedimento, mesmo judiciais, inclusive as despesas processuais e outras despesas derivadas, a divulgação ou utilização das INFORMAÇÕES por seus agentes, representantes ou por terceiros;
III. Comunicar ao Contratante, de imediato, de forma expressa e antes de qualquer divulgação, caso tenha que revelar qualquer uma das INFORMAÇÕES, por determinação judicial ou ordem de atendimento obrigatório determinado por órgão competente; e
IV. Identificar as pessoas que, em nome do Contratado, terão acesso às informações sigilosas.

Cláusula Sexta – DA VIGÊNCIA

O presente TERMO tem natureza irrevogável e irretratável, permanecendo em vigor desde a data de sua assinatura até expirar o prazo de classificação da informação a que o Contratado teve acesso em razão do CONTRATO PRINCIPAL.

A vigência deste Termo independe do prazo de vigência do contrato assinado.

Cláusula Sétima – DAS PENALIDADES

A quebra do sigilo e/ou da confidencialidade das INFORMAÇÕES, devidamente comprovada, possibilitará a imediata aplicação de penalidades previstas conforme disposições contratuais e legislações em vigor

que tratam desse assunto, podendo até culminar na rescisão do CONTRATO PRINCIPAL firmado entre as PARTES. Neste caso, o Contratado, estará sujeita, por ação ou omissão, ao pagamento ou recomposição de todas as perdas e danos sofridos pelo Contratante, inclusive as de ordem moral, bem como as de responsabilidades civil e criminal, as quais serão apuradas em regular processo administrativo ou judicial, sem prejuízo das demais sanções legais cabíveis.

Cláusula Oitava – DISPOSIÇÕES GERAIS

Este TERMO de Confidencialidade é parte integrante e inseparável do CONTRATO PRINCIPAL.

Parágrafo Primeiro – Surgindo divergências quanto à interpretação do disposto neste instrumento, ou quanto à execução das obrigações dele decorrentes, ou constatando-se casos omissos, as partes buscarão solucionar as divergências de acordo com os princípios de boa-fé, da equidade, da razoabilidade, da economicidade e da moralidade.

Parágrafo Segundo – O disposto no presente TERMO prevalecerá sempre em caso de dúvida e, salvo expressa determinação em contrário, sobre eventuais disposições constantes de outros instrumentos conexos firmados entre as partes quanto ao sigilo de informações, tal como aqui definidas.

Parágrafo Terceiro - Havendo necessidade legal devido a Programas de Governo, o Contratado assume o compromisso de assinar Termo de Sigilo (ou equivalente) adicional relacionado ao Programa, prevalecendo as cláusulas mais restritivas em benefício do Contratante.

Parágrafo Quarto – Ao assinar o presente instrumento, o Contratado manifesta sua concordância no sentido de que:

I. O Contratante terá o direito de, a qualquer tempo e sob qualquer motivo, auditar e monitorar as atividades do Contratado;
II. O Contratado deverá disponibilizar, sempre que solicitadas formalmente pelo Contratante, todas as informações requeri das pertinentes ao CONTRATO PRINCIPAL;
III. A omissão ou tolerância das partes, em exigir o estrito cumprimento das condições estabelecidas neste instrumento, não constituirá novação ou renúncia, nem afetará os direitos, que poderão ser exercidos a qualquer tempo;
IV. Todas as condições, TERMOS e obrigações ora constituídos serão regidos pela legislação e regulamentação brasileiras pertinentes;
V. O presente TERMO somente poderá ser alterado mediante TERMO aditivo firmado pelas partes;
VI. Alterações do número, natureza e quantidade das informações disponibilizadas para o Contratado não descaracterizarão ou reduzirão o compromisso e as obrigações pactuadas neste TERMO, que permanecerá válido e com todos seus efeitos legais em qualquer uma das situações tipificadas neste instrumento;
VII. O acréscimo, complementação, substituição ou esclarecimento de qualquer uma das informações disponibilizadas para o Contratado, serão incorporados a este TERMO, passando a fazer dele parte integrante, para todos os fins e efeitos, recebendo também a mesma proteção

descrita para as informações iniciais disponibilizadas, sendo necessário a formalização de TERMO aditivo ao CONTRATO PRINCIPAL;

VIII. . Este TERMO não deve ser interpretado como criação ou envolvimento das Partes, ou suas filiadas, nem em obrigação de divulgar INFORMAÇÕES para a outra Parte, nem como obrigação de celebrarem qualquer outro acordo entre si.

Cláusula Nona – DO FORO

O Contratante elege o foro da <CIDADE DO CONTRATANTE>, onde está localizada a sede do Contratante, para dirimir quaisquer dúvidas originadas do presente TERMO, com renúncia expressa a qualquer outro, por mais privilegiado que seja.

E, por assim estarem justas e estabelecidas as condições, o presente TERMO DE COMPROMISSO DE MANUTENÇÃO DE SIGILO é assinado pelas partes em 2 vias de igual teor e um só efeito.

_____, _____de _____de 20

De acordo.

CONTRATANTE	**CONTRATADO**
_____ <Nome> Matrícula: <Matr.>	_____ <Nome> <Qualificação>
Testemunhas	
Testemunha 1 _____ <Nome> <Qualificação>	Testemunha 2 _____ <Nome> <Qualificação>

UNIVERSIDADE FEDERAL DE SERGIPE
Av. Marcelo Deda Chagas, s/n, - Bairro Rosa Elze, São Cristóvão/SE, CEP 49107-230

TERMO DE REFERÊNCIA

ANEXO IV – MODELO DE PROPOSTA DE PREÇOS

Ao

Ministério da Gestão e da Inovação em Serviços

Públicos Pregão Eletrônico nº __/20__ .

Apresentamos nossa proposta para prestação de serviços técnicos especializados de desenvolvimento, manutenção, sustentação, testes e controle de qualidade de software, por alocação de perfil profissional de TI vinculado ao alcance de resultados, sem dedicação exclusiva de mão de obra, sob demanda, conforme modalidade prevista na Portaria SGD/MGI nº 750, de 20 de março de 2023, com vistas a executar atividades de projeto, construção, testes, implantação, evolução, manutenção, sustentação e garantia de qualidade relacionadas ao ciclo de vida de software, adotando-se práticas ágeis aderentes ao processo de software estabelecido neste Termo de referência, durante o período de 12 (doze) meses:

<Favor preencher somente os Lotes a que se pretende participar >

DADOS DA EMPRESA Razão Social: CNPJ (MF) nº: Endereço: Cidade: UF: CEP: Telefone: Endereço Eletrônico:
DADOS BANCÁRIOS BANCO: AGÊNCIA: CONTA: Dados do Representante para fim de apresentação da proposta e assinatura do contrato: Nome: CPF: Cargo/Função: Carteira de Identidade: Expedido por: Endereço Eletrônico:

LOTE N – DESCRIÇAO DO LOTE

ITEM	CATSER	UNIDADE	QUANTIDADE ESTIMADA (Q)	VALOR UNITÁRIO DO SALÁRIO MENSAL DO PROFISSIONAL (R)	FATOR-K (K)	CUSTO UNITÁRIO MENSAL DO PROFISSIONAL (CM = R * K)	CUSTO TOTAL DO PROFISSIONAL (CT = Q * CM * 12 MESES)
TOTAL							<Soma das células acima>

Nos valores propostos estarão inclusos todos os custos operacionais, encargos previdenciários, trabalhistas, tributários, comerciais e quaisquer outros que incidam direta ou indiretamente na prestação do serviço, apurados mediante o preenchimento do modelo de Planilha de Custos e Formação de Preços.

Nos propomos a fornecer os serviços de acordo com as especificações concordando com o prazo de pagamento e demais condições estabelecidas no Edital e seus anexos. Informamos que o prazo de validade de nossa PROPOSTA DE PREÇOS é de _(_____) dias, a contar da data de abertura da licitação (o prazo não poderá ser inferior a 60 dias).

Anexo:

Planilha de Custo e Formação de Preços, conforme modelo do Anexo XI do Termo de Referência.

UNIVERSIDADE FEDERAL DE SERGIPE
Av. Marcelo Deda Chagas, s/n, - Bairro Rosa Elze, São Cristóvão/SE, CEP 49107-230

TERMO DE REFERÊNCIA

ANEXO V – 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:

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). ALIs, 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 manutenido 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.

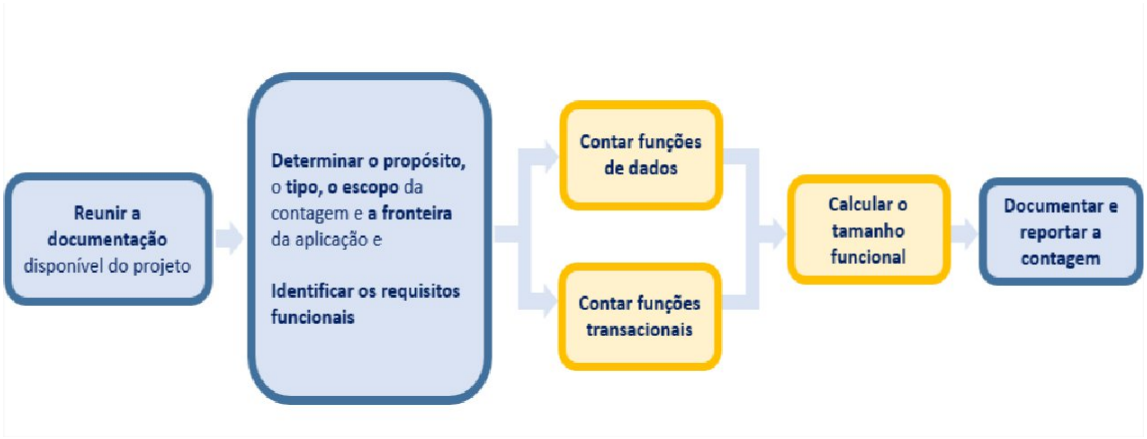
Cada função transacional corresponde a um (1) Processo elementar no SFP e cada função de dados corresponde a um (1) arquivo lógico no SFP.

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 UFS.

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 os 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

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

f) **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:

Tamanho Funcional em SFP = (QtdCE + QtdEE + QtdSE) x 4,6 + (QtdALI + QtdAIE) x 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 fórmula para conversibilidade de PF para SFP temos:

SFP = (5CE + 10EE +10SE) x 4,6 + (5ALI + 2AIE) x 7

SFP = (25) x 4,6 + (7) x 7 = 164 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:

SFP_DESENVOLVIMENTO = SFP_INCLUIDO + 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:

SFP_BASELINE = 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:

SFP_MELHORIA = SFP_INCLUIDO + (FI* x SFP_ALTERADO) + (0,30 x SFP_EXCLUIDO) + 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):

SFP_BASELINE = SFP_BASELINE + SFP_INCLUIDO - SFP_EXCLUIDO

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

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

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.

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

UNIVERSIDADE FEDERAL DE SERGIPE
Av. Marcelo Deda Chagas, s/n, - Bairro Rosa Elze, São Cristóvão/SE, CEP 49107-230

TERMO DE REFERÊNCIA

ANEXO VI - PROCESSO DE SOFTWARE

1. INTRODUÇÃO

1.1. O Processo de Desenvolvimento de Software e manutenção de software de referência baseia-se no framework **SCRUM** , observa as diretrizes de processo de ciclo de vida do **software** constantes da ABNT NBR ISO IEC IEEE 12.207:2020 e está alinhado à Portaria SGD/MGI nº 750, de 20 de março de 2023.

1.2. Conforme Portaria SGD/MGI nº 750, de 20 de março de 2023 o processo de desenvolvimento de **software** deve prever uma fase inicial para o planejamento do projeto, que envolve a captura da visão do usuário, das necessidades e regras negociais, da definição do escopo do projeto e das principais funcionalidades do produto a ser desenvolvido (**backlog** do produto).

1.3. Deve-se evitar o início da implementação do software sem o correspondente planejamento do produto a ser desenvolvido.

1.4. O processo de desenvolvimento de **software** descrito a seguir, aborda diferentes dimensões relacionadas ao ciclo de vida de construção e utilização de software, que podem ser adaptadas à realidade operacional da Contratante.

1.5. Os projetos ágeis devem ser elaborados com a participação de servidor ou profissional contratado com conhecimentos em metodologias ágeis.

2. PAPEIS E RESPONSABILIDADES

2.1. O processo de desenvolvimento de **software** é executado pelo Time ágil e deve ser observado e adaptado à realidade operacional da Contratante.

2.2. O Time ágil consiste, no mínimo, em um **Scrum Master,** um Dono do Produto e Desenvolvedores.

2.2.1 Esse time pode ser integrado também por profissionais especializados.

2.2.2 Os membros do time ágil devem:

a) Participar ativamente dos eventos do **Scrum (Sprint,** Planejamento da **Sprint,** Reuniões diárias, Revisão da **Sprint**, Retrospectiva da **Sprint**);
b) Executar a **sprint** e entregar incrementos de **software** ao final de cada **sprint**;
c) Ser responsável por todas as atividades relacionadas com os produtos, desde a colaboração dos **stakeholder**s, verificação, manutenção, operação, experimentação, investigação e desenvolvimento, assim como tudo o mais que possa ser necessário;
d) Adotar práticas de melhoria contínua;
e) Realizar testes funcionais e não funcionais concomitante ao desenvolvimento de software;
f) Apoiar na homologação das **sprints** e releases;
g) Elaborar manuais do usuário ou **help** de funcionalidades.

2.3. Participam do Time ágil, os seguintes papéis:

PAPEL		**ATIVIDADES QUE É RESPONSÁVEL OU QUE PARTICIPA COMO UM DOS RESPONSÁVEIS PELA EXECUÇÃO**
Scrum Master	Profissional com conhecimento aprofundado em técnicas ágeis.	● Garantir que o **Scrum** seja entendido e aplicado; ● Assegurar que todos os eventos do **Scrum** têm lugar e são positivos, produtivos e mantidos dentro tempo previsto; ● Apoiar o Dono do Produto e a organização na adoção de práticas ágeis; ● Buscar melhoria contínua do time; ● Facilitar a colaboração dos **stakeholders** conforme solicitado ou necessário; ● Atualizar Gráfico de **Burndown**; ● Remover impedimentos para a equipe de desenvolvimento durante a execução das **Sprints.**
Dono do Produto (**Product Owner –** PO)	Servidor representante da área demandante de soluções de **software**, designado por autoridade competente	● Responsável por ordenar o trabalho a ser realizado pelo time, criando, mantendo e priorizando o(s) **backlog**(s) do(s) produto(s); ● Criar e compartilhar a Visão do Produto; ● Planejar o **Roadmap**; ● Construir o **Backlog** do Produto; ● Expressar claramente os itens do **Backlog** do Produto; ● Ordenar e priorizar os itens do **Backlog** do Produto; ● Garantir que o time de desenvolvimento entenda os itens do **Backlog** do Produto no nível necessário; ● Apoiar no planejamento do **Release**; ● Validar Incremento de **Software**; ● Validar s**oftware** ao final de cada **sprint e release**; ● Reportar a Avaliação de Satisfação do Dono do Produto.
Desenvolvedores de software	Desenvolvedores que fazem parte do time ágil	● Criar o **Sprint Backlog**; ● Construir o(s) produto(s) de software.
Profissionais Especializados	Demais profissionais que integram o time com especialidades definidas a exemplo de Arquitetos, Analistas de Teste/Qualidade, Analista de Dados, etc.	● Contribuir para a construção dos produtos de software.
Partes interessadas (Stakeholders)	Profissionais impactados pela solução ou que possuam interesse na entrega da solução	● Opinar, influenciar, contribuir para o planejamento e tomadas de decisão do negócio ou projeto; ● Esclarecer dúvidas; ● Se necessário, apoiar o PO na validação da **sprint ou release**.

Analistas de Teste e Qualidade	Analistas de Teste e Qualidade que fazem parte do time ágil.	• Garantir a qualidade dos sistemas durante todo o ciclo do processo de software até a sua implantação, minimizando a ocorrência de erros no ambiente de produção; • Realizar a revisão de código, realização de testes avançados e revisão da qualidade da documentação produzida; • Apoiar a fiscalização técnica dos contratos de desenvolvimento, manutenção e sustentação de software na revisão técnica dos critérios de aceitação e de qualidade dos produtos entregues.

3. PLANEJAMENTO DO PRODUTO

3.1. A área requisitante do projeto de **software**, representada pelo dono do produto e **stakeholders**, fornece a compreensão do negócio, das necessidades, dos objetivos de negócio e dos requisitos do produto de **software.**

3.2. A equipe designada pelo Contratado apoia nas atividades de planejamento do produto.

3.3. O planejamento compreende atividades anteriores a execução da primeira **sprint** do projeto.

3.4. É composto pelas seguintes atividades, que podem ser adaptadas à realidade operacional da Contratante:

a) Construir a Visão do Produto

b) Construir o **Roadmap** do Produto

Etapa	Atividades	Descrição	Atores	Entregáveis
PLANEJAMENTO DO PRODUTO	Construir a Visão do Produto	Nessa atividade, entende-se que existe um problema a ser resolvido ou uma oportunidade a ser aproveitada.	PO e profissional (is) designado (s) pelo Contratado para apoiar na construção da visão do produto.	Documento de Visão
	Construir o Roadmap do Produto	Dividir os objetivos de negócio e as características-chaves ou macro funções do produto em partes entregáveis, por ordem de prioridade. As partes são os **releases** que, por sua vez, são construídas a partir das características-chaves do produto priorizadas e ordenadas.	PO e profissional (is) designado(s) pelo Contratado para apoiar na construção do **roadmap.**	Roadmap do Produto

4. PLANEJAMENTO DO RELEASE

4.1. Neste grupo de atividades será construído o planejamento de um ou mais **releases,** conforme o plano cronológico definido no **roadmap.**

4.2. Cada release poderá ser construído em uma ou mais iterações (**Sprints**). A equipe designada pelo Contratado apoia nas atividades de planejamento do **release**.

4.3. É composto pelas seguintes atividades, que podem ser adaptadas à realidade operacional da Contratante:

a) Construir o **Backlog** do Produto;

b) Estimar o tamanho e esforço do **backlog** do produto;

c) Planejar o **Release**.

Etapa	Atividades	Descrição	Atores	Entregáveis
PLANEJAMENTO DO RELEASE	Construir o **Backlog** do Produto	● Construir e disponibilizar o **backlog** do produto, que é a lista priorizada dos itens necessários para o desenvolvimento e entrega do produto de software. ● O **Backlog** do Produto representa tudo que é necessário para desenvolver e lançar um produto de valor agregado ao negócio. É uma lista de todos os requisitos (funcionais e não funcionais), funções, tecnologias, melhorias e correções de defeitos que constituem as mudanças que serão efetuadas no produto para versões futuras. ● Os requisitos do software a serem desenvolvidos serão decompostos em histórias de usuários que, por sua vez, poderão ser subdivididas em tarefas. Este refinamento será feito no decorrer do projeto de acordo com a prioridade dos requisitos do software. ● Os requisitos do **software**, as histórias de usuários e as tarefas compõem o **backlog** do produto. ● As eventuais manutenções corretivas e adaptativas que venham a ser necessárias no software podem ser incluídas no **backlog** do produto. ● O **backlog** do produto será priorizado pelo **Product Owner** (PO). **Nota:** Nas atividades de planejamento do produto caso seja identificado um grande conhecimento da solução a ser produzida (objetivos de negócio, metas e características-chaves), com pequenas possibilidades de mudanças, a elaboração do backlog do produto poderá ser logo após a concepção do Documento de Visão. Essa estratégia possibilita que o planejamento do roadmap, com o plano cronológico de entrega dos releases seja estimado com maior precisão.	PO e profissional(is) designado(s) pelo Contratado para apoiar na construção do **backlog** do produto	**Backlog** do Produto

	Estimar o tamanho e esforço do backlog do produto	● Atividade para estimar o tamanho do backlog do produto e esforço necessário para sua construção, utilizando técnicas de mensuração. ● Apoia no planejamento dos releases e quantidade de iterações. Nota: a estimativa de tamanho e esforço pode ser feita por meio do método de ponto de função simplificado (SFP) descrito no ANEXO V – ROTEIRO DE MÉTRICAS.	Profissional designado pela Contratante para apoiar na estimativa do tamanho do backlog do produto	Estimativa do tamanho e esforço do Backlog do produto

	Planejar o Release	● Atividade para definir o plano do release com a meta a ser alcançada em função dos objetivos de negócio e características-chaves do produto. O planejamento do release parte do pressuposto que o backlog do release já foi definido a partir dos objetivos de negócio e características-chaves do release. ● A fragmentação do release em sprints (de duração fixa) ocorrerá conforme a priorização e complexidade dos itens do backlog, o tempo disponível para a construção do release ou do projeto, estimativa de tamanho/esforço dos itens de backlog, histórico de produtividade da instituição e a estratégia de desenvolvimento estabelecida para o produto. Nota: Podem ser inseridos no plano do release premissas, impedimentos e riscos envolvidos no release, além de prever atividades prévias ao início das iterações para que a equipe execute a criação/disponibilização dos ambientes de desenvolvimento e de testes necessários. Devem ser definidos, para cada projeto, parâmetros para a execução das sprints, tais como: ● Configuração mínima do time que irá executar o conjunto de sprints, indicando perfis profissionais mínimos e nível de compartilhamento aceitável para determinados perfis; ● Duração máxima da sprint; ● Meta de velocidade da sprint, como a quantidade de histórias de usuário e pontos de função; ● Meta de escopo planejado x realizado, que indica o percentual realizado a cada sprint em comparação ao escopo planejado; 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.	PO e profissional(is) designado(s) pelo Contratado	Plano do Release

5. SPRINTS

- Corresponde a um grupo de atividades para transformar os itens de backlog em um incremento de software.
- As Iterações/**Sprints** são ciclos de execução do projeto, com duração de 1 a 4 semanas, de acordo com as características de cada projeto.
- A duração das **sprints** e o dia e a duração das reuniões de planejamento, revisão e retrospectiva das sprints serão definidas pela Contratante no início do projeto. Estas definições poderão ser alteradas posteriormente, a critério da Contratante, mediante comunicação prévia à equipe de desenvolvimento do Contratado.
- O critério para aceitar cada sprint como "pronta" (finalizada, **done**) será definido para cada projeto e, quando necessário, de forma particular para cada tarefa.

5.1. PLANEJAMENTO DA SPRINT

5.5.1. Com base no **backlog** do produto priorizado, o time realiza o planejamento da **sprint**, selecionando os itens possíveis de serem desenvolvidos no tempo da **sprint.**

5.5.2. O dono do produto estabelece os critérios de definição de pronto, ou seja, o que deve ser entregue ao final da **sprint**.

5.5.3. O time, então, se compromete a entregar o incremento de **software**, ao final da **sprint**.

5.5.4. Esse planejamento pode levar até 4 horas, dependendo do tamanho da **sprint.**

5.5.5. Os itens de **Product Backlog** selecionados para a **Sprint** e o plano de entrega do incremento, são agrupados no **Backlog da Sprint.**

5.5.6. Na definição do **backlog** da **sprint**, deve-se monitorar a relação quantitativa entre itens planejados e itens não planejados, com vistas a assegurar que o maior esforço esteja sendo empreendido na entrega de valor.

5.2. EXECUÇÃO DA SPRINT E REUNIÃO DIÁRIA

5.2.1. O time de desenvolvimento então inicia a execução da sprint, construindo o incremento de software a partir do **backlog da sprint.**

5.2.2. Durante esse período, o time também se reúne diariamente, por 15 minutos, para responder a três perguntas:

a) O que eu fiz desde a última reunião?

b) O que eu vou fazer até a próxima reunião?

c) Há algum impedimento para o trabalho ser realizado?

5.2.3. O **Sprint Backlog** é atualizado durante a execução da **Sprint**, à medida que se vai aprendendo mais.

5.2.4. O **Sprint Backlog** deve estar detalhado de forma que o seu progresso possa ser inspecionado nas reuniões diárias.

5.3. REVISÃO DA SPRINT

5.3.1. Após o último dia da execução da **sprint**, a equipe se reúne para realizar a revisão do produto.

5.3.2. Nesse evento, o time de desenvolvimento apresenta o incremento de **software** construído para o Dono do Produto e os **stakeholders.**

5.3.3. O time revisa o que foi realizado na **sprint** e avalia o que fazer a seguir.

5.3.4. O Dono do Produto analisará a resolução de cada história apresentada e decidirá se está "pronta" ou não, de acordo com o critério previamente estabelecido.

5.3.5. Nesse momento, o **backlog** do produto pode ser atualizado para refletir as novas demandas e oportunidades.

5.3.6. Caso alguma história seja reprovada, ela será reinserida no **Backlog,** ficando disponível para uma próxima **sprint.**

5.3.7. Essa revisão pode levar até 4 horas para **sprints** de 4 semanas.

5.4. RETROSPECTIVA DA SPRINT

5.4.1. O último evento antes da conclusão da **sprint** é chamado de retrospectiva da **sprint**.

5.4.2. O propósito da retrospectiva da **sprint** é planejar maneiras de aumentar a qualidade e a eficácia do trabalho.

5.4.3. O time discute como foi a última **sprint** em relação a indivíduos, interações, processos, ferramentas e a Definição de Pronto.

5.4.4. Nesse evento responde-se basicamente a dois questionamentos:

a) O que deu certo?

b) O que pode ser melhorado?

5.4.5. O time, então, identifica as mudanças mais úteis para melhorar sua eficácia e procura aplicar essas mudanças na **sprint** seguinte.

5.4.6. A retrospectiva pode levar até 4 horas para **sprints** de 4 semanas.

5.4.7. Finalmente, o time inicia o planejamento da próxima **sprint** e o ciclo se repete quantas vezes forem necessárias.

5.5. LIBERAÇÃO E ENTREGAS

5.5.1. A entrega dos produtos de **software** ocorre após a finalização da **sprint,** quando o incremento de

software é revisado e homologado.

5.5.2. É um evento separado da **sprint**.

6. TRANSIÇÃO DO PROJETO

6.1. Este grupo de atividades tem a finalidade de realizar algumas verificações finais do **release** ou do projeto produzido e garantir uma versão útil do produto em ambiente de produção.

6.2. É composto das seguintes atividades, que podem ser adaptadas à realidade operacional da Contratante:

a) Planejar implantação;

b) Homologação final do **release;**

c) Finalização de documentação: manual do usuário, **help** de funcionalidades, histórias do usuário;

d) Implantar **software;**

e) Reunião de encerramento do **release** ou do projeto.

7. TESTES E CONTROLE DE QUALIDADE DO SOFTWARE:

7.1. Os testes e controle da qualidade do produto que serão entregues devem permear todo o projeto e execução das sprints.

7.2 Sempre que possível, deve-se promover as seguintes atividades, que podem ser adaptadas à realidade operacional da Contratante:

a) Analisar riscos e a conformidade de processo, projetos, técnicas, práticas e ferramentas de desenvolvimento e testes das empresas que prestam serviços de desenvolvimento de sistemas em relação às normas e padrões da organização e às melhores práticas de mercado no que diz respeito a versionamento, arquitetura, padrão visual, segurança, testes e qualidade de software;

b) Verificar a atualidade e conformidade da documentação técnica dos sistemas de informação da organização em relação ao software que está operando;

c) Promover o diagnóstico de situações de gargalos e problemas de desempenho nos sistemas;

d) Verificar se os padrões de arquitetura de software foram obedecidos e se os padrões de organização de códigos, bibliotecas, componentes foram observados;

e) Detectar falhas e propor correções em processos de testes implantados na organização;

f) Emitir pareceres técnicos relacionados ao ambiente de sistemas da organização;

g) Elaborar Relatório Técnico de Análise de Qualidade.

7.3. Os testes devem ser planejados conforme práticas a seguir, que podem ser adaptadas à realidade operacional da Contratante:

a) Realizar reuniões com os usuários e/ou times de desenvolvimento para modelar e elaborar estratégias de testes;

b) Planejar testes funcionais e não funcionais de softwares;

c) Elaborar artefatos como Roteiros/Casos de Teste, Listas de Verificação, Critérios de Aceite.

7.4. A execução dos testes deve abordar as seguintes atividades, que podem ser adaptadas à realidade operacional da Contratante:

a) Elaborar artefatos de apoio a testes, como roteiros, **scripts** de testes, relatório de evidências de testes;

b) Executar testes automáticos e/ou manuais em ambiente de testes e de homologação;

c) Executar testes funcionais e não funcionais;

d) Dar suporte aos testes realizados pelo usuário.

8. DEFINIÇÃO DE PRONTO

8.1. A definição de pronto é uma descrição formal do estado do incremento, quando este cumpre as medidas de qualidade exigidas para o produto.

8.2. Quando um item do **Backlog** do produto satisfaz a Definição de Pronto, nasce um incremento.

8.3. A definição de pronto cria transparência; proporciona a todos uma compreensão do trabalho que foi concluído como parte do Incremento. Todo o time ágil deve estar em conformidade com a definição de pronto.

8.4. Se um item de **Product Backlog** não cumpre a definição de pronto, não pode ser lançado ou mesmo apresentado na Revisão da **Sprint**. Em vez disso, volta ao **Backlog** do produto para consideração futura.

8.5. A seguir, alguns critérios para aceitação dos produtos, que podem ser adaptados à realidade operacional da Contratante:

8.5.1. **Para admissibilidade do produto:**

a) Código-fonte submetido ao controle de versões do Contratado;

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.

8.5.2. **Para aceitação da demanda:** após realizar a inspeção do produto quanto à sua admissibilidade, o Contratado deverá:

a) Executar testes funcionais automatizados que tenham sido solicitados e, consequentemente, verificar se estão corretamente implementados ou mesmo se existem, além de observar os resultados da execução;

b) Executar testes unitários ou verificar relatórios de execução destes que possam envolver porções críticas do produto;

c) Realizar alguns testes funcionais, pelo menos nos principais fluxos do produto entregue.

8.5.3. **Após a realização dos testes,** a organização deve proceder a uma das ações a seguir:

a) **Rejeição:** caso sejam percebidos defeitos de natureza impeditiva em alguma história implementada ou não tenha coberto o escopo planejado de tal forma que a entrega não seja passível de aceitação;

b) **Aceitação parcial**: caso a demanda possua alguns defeitos significativos de natureza não-impeditiva ou não tenha coberto o escopo planejado de tal forma que ainda seja passível de aceitação;

c) **Aceitação integral:** caso a demanda esteja em nível de qualidade tal que não sejam percebidos defeitos significativos, bem como envolva cumprimento do escopo planejado.

8.6. O Contratado deve registrar todos os aspectos relevantes. Os defeitos percebidos nos casos de rejeição ou aceitação parcial da **sprint** devem fazer parte de um item de **backlog** da próxima **sprint**.

UNIVERSIDADE FEDERAL DE SERGIPE
Av. Marcelo Deda Chagas, s/n, - Bairro Rosa Elze, São Cristóvão/SE, CEP 49107-230

TERMO DE REFERÊNCIA

ANEXO VII - REQUISITOS MÍNIMOS DE QUALIDADE E PADRONIZAÇÃO DOS ASPECTOS TÉCNICOS DA CODIFICAÇÃO

1. Os requisitos mínimos de qualidade e padronização de código descritos nesse Anexo devem ser integralmente observados pelo Contratado, salvo quando autorizado pela Contratante.
2. O desenvolvimento e manutenção de softwares deverá seguir as diretrizes constantes do Roteiro de Codificação (ANEXO XVII).
3. Cada indicador de qualidade do código fonte possui meta a ser alcançada que estão relacionadas no quadro exemplificativo a seguir:

Grupo	**Indicador**	**Unidade**	**Meta**
Projeto	Complexity / file ou equivalente	média total	<= 10
	Complexity / class ou equivalente	média total	<= 10
	Complexity /function ou equivalente	média total	<= 3
	Duplications ou equivalente	%	<= 4%
	Security Issue Tags ou equivalente	unidades	=0
	Technical Debt ratio ou equivalente	%	<= 2,5%
	SQALE RATING ou equivalente	Nota	=A
Violações de código (possíveis bugs, estilo de codificação, más práticas de codificação)	Critical Issues ou equivalente	unidades	=0
	Blocker Issues ou equivalente	unidades	=0
Indicadores relacionados a testes	Unit Tests Coverage - camada negócio / Impl ou equivalente	%	>=70%
	Unit Test Success ou equivalente	%	>=100%
	Skipped Tests ou equivalente	unidades	=0

4. As metas e indicadores constantes do quadro acima poderão ser ajustados pela CONTRANTE e adequados a cada Ordem de Serviço. Caso não seja informado novo quadro na Ordem de Serviço ou no aditivo da Ordem de Serviço, deverá ser considerado o quadro acima.

5. O Indicador de qualidade de código (IQC) será calculado dividindo a Quantidade de requisitos de qualidade de código atendidos (ΣQrc) pela Quantidade total de requisitos de qualidade de código avaliados (ΣQtr):

IQC = 100 * (ΣQrc / ΣQtr)

Onde:

IQC = Indicador de qualidade de código;

ΣQrc = Somatório da Quantidade de requisitos de qualidade de código atendidos;

ΣQtr = Somatório da Quantidade total de requisitos de qualidade de código avaliados.

6. Caso algum indicador seja depreciado ou removido em futura versão da ferramenta de análise de código que venha a ser adotada pela Contratante e não seja substituído por indicador equivalente nesta mesma ferramenta, ele deixará de ser considerado para fins de aferição do IQC e substituído por um equivalente presente na nova versão ou ferramenta.

7. Problemas de qualidade no código-fonte do software pré-existentes à abertura da OS serão desconsiderados também na aferição do indicador IQC.

8. Havendo impossibilidade do uso da ferramenta deve-se promover a verificação manual por meio de aplicação de técnicas de amostragem conforme Roteiro de Amostragem (ANEXO VIII).

TERMO DE REFERÊNCIA

ANEXO VIII - ROTEIRO DE AMOSTRAGEM

1. O processo de amostragem para fins de verificação dos níveis mínimos de serviço deve observar ao menos as seguintes etapas:

a) Definição do Tamanho amostral, que deve ser superior a 30 amostras.
b) Seleção das amostras em diferentes funcionalidades ou módulos, orientando-se por aquelas de maior risco ou valor esperado ao usuário.
c) Realização dos testes e análise sobre as amostras coletadas.
d) Cálculo do valor percentual de atendimento do nível de serviço sobre o conjunto de amostras.

2. Se a diferença entre o indicador aferido no processo de amostragem for igual ou inferior a 5% ao limite do nível mínimo de serviço, deve-se ampliar a quantidade de amostras em no mínimo 30 unidades com vistas a certificar-se que a aferição não está situada na margem de erro.

UNIVERSIDADE FEDERAL DE SERGIPE
Av. Marcelo Deda Chagas, s/n, - Bairro Rosa Elze, São Cristóvão/SE, CEP 49107-230

TERMO DE REFERÊNCIA

ANEXO IX – ROTEIRO DE MENSURAÇÃO POR PONTOS DE HISTÓRIA (Story Point).

O roteiro medição utilizando métrica por Pontos por História (Story Points) para fins de verificação dos níveis mínimos de serviço deve observar ao menos as seguintes etapas:

1. Mapeamento de Histórias do usuário

a. **Objetivo**: Identificar e registrar todas as histórias do usuário (User Stories) a partir dos casos de uso ou requisitos de alto nível.

b. **Ações**:

i. Reunir o time para revisar os casos de uso.

ii. Transformar cada caso de uso em histórias de usuários entregáveis.

iii. Garantir que cada história de usuário gere valor ao usuário final.

2. Refinamento de Histórias do usuário

a. **Objetivo**: Detalhar histórias de alto nível em histórias menores e mais precisas.

b. **Ações**:

i. Utilizar técnicas como *INVEST* (Independente, Negociável, Valiosa, Estimável, Pequena(Small), Testável).

ii. Validar que cada história seja pequena o suficiente para ser entregue dentro de um sprint.

3. Atualização de Estimativas

a. **Objetivo**: Estimar o esforço necessário para cada história.

b. **Ações**:

i. Utilizar a escala de **Fibonacci** (1, 2, 3, 5, 8, 13, 21...) para estimar os Story Points.

ii. Realizar sessões de *Planning Poker* com o time para consenso nas estimativas.

4. Régua de Pontuação

a. **Objetivo**: Padronizar a atribuição de Story Points.

b. **Referência**: i. **1 CRUD completo = 3 Story Points**

ii. O time deve ajustar os valores conforme a complexidade, integração, regras de negócio, etc.

5. Cálculo da Velocidade do time(Velocity)

a. **Objetivo**: Medir a capacidade média de entrega do time por sprint por Story Points.

b. **Ações**:

i. Após cada sprint, registrar o total de Story Points concluídos.

ii. Calcular a média dos últimos 4 Sprints para obter a **velocity**.

iii. Utilizar essa métrica para planejamento de futuras sprints.

iv. O valor utilizado na **velocity** pode variar de acordo com a quantidade de desenvolvedores no time e o seu nível de experiência.

6. Acompanhamento e Métricas

a. **Objetivo**: Monitorar a evolução e produtividade do time.

b. **Indicadores**: i. **Velocity** por sprint. (Utilizada para fins de nível mínimo de serviço como variável Pp (*produtividade prevista no período*) no cálculo do IPA - Indicador de Produtividade Ágil).

ii. **Burndown Chart**.

iii. **Percentual de histórias entregues vs planejadas**.

iv. **Lead Time e Cycle Time** (se aplicável).

7. Ferramentas de Apoio

a. Redmine, Trello, ou outra ferramenta de gestão ágil.

b. Planilhas para controle manual, se necessário

Pontos de história (**Story Point**).

UNIVERSIDADE FEDERAL DE SERGIPE
Av. Marcelo Deda Chagas, s/n, - Bairro Rosa Elze, São Cristóvão/SE, CEP 49107-230

TERMO DE REFERÊNCIA

ANEXO X - QUESTIONÁRIO DE AVALIAÇÃO INDIVIDUAL DO PERFIL PROFISSIONAL

QUESTIONÁRIO DE AVALIAÇÃO INDIVIDUAL DO PERFIL PROFISSIONAL	
CONTRATO nº _____/__	Período de Referência: _____/____/____a _____/____/____
OS nº _____	
1.IDENTIFICAÇÃO	
Nome do profissional:	
Identificação do perfil profissional de referência:	
2.CRITERIOS DE AVALIAÇÃO	
2.1 - Assiduidade	Compareceu aos eventos previstos no processo de desenvolvimento ágil, reuniões e convocações: [ ] 4 – Todas [ ] 3 – Maioria [ ] 2 – Metade [ ] 1 – Minoria [ ] 0 – Nenhuma
2.2 - Demonstração de conhecimento técnico;	O profissional mostrou conhecimento e domínio técnico em sua área de competência, aplicando-os de maneira eficiente no atendimento das demandas e soluções dos problemas? [ ] 4 – Demonstrou integralmente [ ] 3 – Demonstrou a maior parte do tempo [ ] 2 – Demonstrou parcialmente [ ] 1 – Demonstrou a menor parte do tempo [ ] 0 – Não demonstrou em nenhum momento

2.3 - Habilidades comportamentais	Apresentou as habilidades comportamentais (comunicação oral e escrita clara e objetiva, capacidade de trabalho em equipe, proatividade, relacionamento interpessoal profissional) no atendimento das demandas e soluções dos problemas? [ ] 4 – Demonstrou integralmente [ ] 3 – Demonstrou a maior parte do tempo [ ] 2 – Demonstrou parcialmente [ ] 1 – Demonstrou a menor parte do tempo [ ] 0 – Não demonstrou em nenhum momento
Pontuação obtida	(Valor do Item 1 + Valor do Item 2 + Valor do Item 3)
Pontuação máxima	12
3 - ASSINATURA	
_____ <Nome do Fiscal Técnico do Contrato>	

UNIVERSIDADE FEDERAL DE SERGIPE
Av. Marcelo Deda Chagas, s/n, - Bairro Rosa Elze, São Cristóvão/SE, CEP 49107-230

TERMO DE REFERÊNCIA

ANEXO XI - MODELO DE PLANILHA DE CUSTOS E FORMAÇÃO DE PREÇOS

1. A planilha apresentada neste anexo poderá ser adaptada às características do serviço contratado e da proposta realizada pela LICITANTE, inclusive no que concerne às rubricas e suas respectivas provisões e/ou estimativas, desde que haja justificativa.

2. Deverá ser elabora uma planilha para cada perfil profissional de TI (ITEM) a ser ofertado pela LICITANTE conforme requerido em cada LOTE.

PLANILHA DE CUSTOS E FORMAÇÃO DE PREÇOS		
	Nº PROCESSO	
	LICITAÇÃO Nº	
	CNPJ	
	NOME DA EMPRESA	
	LOTE	
	ITEM	

Discriminação dos Serviços (dados referentes à contratação)		
A	Data da Apresentação da Proposta (dia/mês/ano)	
B	Município/UF	
C	Ano acordo, convenção ou Sentença Normativa em Dissídio Coletivo	
D	Número de Meses de Execução do Contrato	
E	Número de registro da convenção coletiva de trabalho	
F	Regime Tributário da Empresa:	

Dados complementares para composição dos custos referentes ao profissional alocado		
1	Tipo de Serviço (mesmo serviço com características distintas)	
2	Remuneração do profissional	

3	Categoria Profissional (vinculada à execução contratual)	CBO:	
4	Data Base da Categoria (dia/mês/ano)		

MÓDULO 1: COMPOSIÇÃO DA REMUNERAÇÃO			
1	**Composição da Remuneração**		**Valor (R$)**
A	Salário Base		R$
B	Adicional de Periculosidade	%	R$
C	Adicional de Insalubridade	%	R$
D	Adicional Noturno		R$
E	Hora Noturna Adicional		R$
F	Adicional de Hora Extra no feriado trabalhado		R$
G	Outros (especificar)		R$
TOTAL DO MÓDULO 1			R$

MÓDULO 2: ENCARGOS E BENEFÍCIOS ANUAIS, MENSAIS E DIÁRIOS			
2.1	**Submódulo 2.1 - 13° (décimo terceiro) Salário, Férias e Adicional de Fé- rias**		**Valor (R$)**
A	13° Salário	%	R$
B	Férias e Adicional de Férias	%	R$
Subtotal			**R$**
Incidência do Submódulo 2.2		**%**	**R$**
Total			**R$**

2.2	**Submódulo 2.2 - Encargos Previdenciários (GPS), FGTS e Outras Contri- buições**				**Valor (R$)**
A	INSS			%	R$
B	Salário Educação			%	R$
C	Seguro Acidente de Trabalho	RAT	FAP	%	R$
D	SESI ou SESC			%	R$

E	SENAI ou SENAC				%	R$
F	SEBRAE				%	R$
G	INCRA				%	R$
H	FGTS				%	R$
I	Outras Contribuições (especificar)				%	R$
Total					**%**	**R$**
2.3	**Submódulo 2.3 - Benefícios Mensais e Diários**					**Valor (R$)**
A	Transporte:	Nº Vales	Valor do Vale	Nº dias úteis	Desc. Empregado	R$
					R$	
B	Auxílio Alimentação (Vales, cestas básicas, etc.):		Valor do Vale	Nº dias úteis	Desc. Empregado	R$
					R$	
C	Assistência Médica e Familiar/Odontológica					R$
D	Auxílio Creche					R$
E	Seguro de vida, invalidez e funeral					R$
F	Auxílio cesta básica					R$
I	Outros (especificar)					R$
Total						**R$**
MÓDULO 2: RESUMO						
2.1	13º (décimo terceiro) Salário, Férias e Adicional de Férias					R$

2.2	GPS, FGTS e outras contribuições			R$
2.3	Benefícios Mensais e Diários			R$
TOTAL DO MÓDULO 2				**R$**

MÓDULO 3: PROVISÃO PARA RESCISÃO			
3.1	**Provisão para Rescisão**		**Valor (R$)**
A	Aviso Prévio Indenizado	%	R$
B	Incidência do FGTS sobre o Aviso Prévio Indenizado	%	R$
C	Aviso Prévio Trabalhado	%	R$
D	Incidência do Submódulo 2.2 sobre o Aviso Prévio Trabalhado	%	R$
E	Multa sobre o FGTS sobre o Aviso Prévio Indenizado e Trabalhado	%	R$
TOTAL DO MÓDULO 3		**%**	**R$**

MÓDULO 4: CUSTO DE REPOSIÇÃO DO PROFISSIONAL AUSENTE					
4.1	**Submódulo 4.1 - Ausências Legais**				**Valor (R$)**
A	Férias e Terço Constitucional de Férias				R$
B	Ausências Legais			%	R$
C	Licença Paternidade	Licenças/ano:		%	R$
		Incidência:			
D	Ausência por Acidente de Trabalho	Licenças/ano:		%	R$
		Incidência:			
E	Afastamento Maternidade	Licenças/ano:		%	R$
		Incidência:			
F	Outros (especificar)			%	R$
Total					**R$**

4.2	**Submódulo 4.2 - Intrajornada**		**Valor (R$)**
A	Substituto no Intervalo para repouso ou alimentação	%	R$

Total		**R$**
MÓDULO 4: RESUMO		
4.1	Substituto nas Ausências Legais	R$
4.2	Substituto na Intrajornada	R$
Subtotal do Módulo 4		**R$**
Incidência do submódulo 2.2	**%**	**R$**
TOTAL DO MÓDULO 4		**R$**
MÓDULO 5: INSUMOS DIVERSOS		
5	**Insumos Diversos (valores mensais por empregado)**	**Valor (R$)**
A	Uniformes (valor em parte não renovável)	R$
B	Materiais	R$
C	Microcomputador utilizado por profissional	R$
D	Outros (especificar)	R$
TOTAL DO MÓDULO 5		**R$**

MÓDULO 6: CUSTOS INDIRETOS, TRIBUTOS E LUCRO				
6	**Custos Indiretos, Tributos e Lucro**	**Base**	**%**	**Valor (R$)**
A	Custos Indiretos	R$	%	R$
B	Lucro	R$	%	R$
Subtotal - Base de Cálculo de Tributos				**R$**
Subtotal B - Base de Cálculo de Tributos por dentro ou racional				**R$**
C.1	Tributos federais (COFINS)		%	R$
C.	Tributos Federais (PIS)		%	R$

2			
C.3	INSS (Desoneração)	%	R$
D	Tributos Estaduais (especificar)	%	R$
E.1	Tributos Municipais (ISS)	%	R$
E.2	Outros Tributos Municipais (especificar)	%	R$
F	**Total dos Tributos**	**%**	**R$**

MÓDULO 6: RESUMO		
6.A	Custos Indiretos	R$
6.B	Lucro	R$
6.F	Tributos	R$
TOTAL DO MÓDULO 6		**R$**

QUADRO RESUMO DO CUSTO DO PERFIL PROFISSIONAL		
Mão-de-Obra vinculada à execução contratual (valor por empregado)		**Valor (R$)**
A	MÓDULO 1: COMPOSIÇÃO DA REMUNERAÇÃO	R$
B	MÓDULO 2: ENCARGOS E BENEFÍCIOS ANUAIS, MENSAIS E DIÁRIOS	R$
C	MÓDULO 3: PROVISÃO PARA RESCISÃO	R$
D	MÓDULO 4: CUSTO DE REPOSIÇÃO DO PROFISSIONAL AUSENTE	R$
E	MÓDULO 5: INSUMOS DIVERSOS	R$
	Subtotal (A + B + C + D + E)	R$
F	MÓDULO 6: CUSTOS INDIRETOS, TRIBUTOS E LUCRO	R$

VALOR TOTAL DO PERFIL PROFISSIONAL	**R$**
QUANTIDADE DE PROFISSIONAIS	
CUSTO TOTAL MENSAL DO PERFIL PROFISSIONAL	**R$**
FATOR-K	
TOTAL ANUAL DO PERFIL PROFISSIONAL	**R$**
TOTAL GLOBAL DO PERFIL PROFISSIONAL	**R$**

UNIVERSIDADE FEDERAL DE SERGIPE
Av. Marcelo Deda Chagas, s/n, - Bairro Rosa Elze, São Cristóvão/SE, CEP 49107-230

TERMO DE REFERÊNCIA

ANEXO XII - TECNOLOGIAS

As informações consolidadas acerca das tecnologias utilizadas pela contratante, estão apresentadas e detalhadas na tabela a seguir:

Categoria	Tecnologia	Descrição / Aplicação
Linguagens de Programação	Java	Linguagem de programação utilizada no desenvolvimento de sistemas corporativos e aplicações web.
	Python	Linguagem de programação de alto nível, empregada em automações, integração de sistemas e serviços backend.
Frameworks e APIs	Spring Boot	Framework para criação de aplicações Java com configuração simplificada e suporte a arquitetura de microsserviços.
	REST	Padrão de arquitetura para construção de APIs na Web.
Frontend / Interface	HTML	Linguagem de marcação utilizada para estruturação de páginas e interfaces web.
	CSS	Folhas de estilo utilizadas para definição de layout, aparência e responsividade das interfaces web.
	JSP	JavaServer Pages, tecnologia utilizada para geração dinâmica de conteúdo web em aplicações Java.
	Thymeleaf	Template engine para aplicações Java, utilizada para renderização dinâmica de páginas HTML.
Banco de Dados	PostgreSQL	Sistema gerenciador de banco de dados relacional de código aberto, utilizado para armazenamento e consulta de dados das aplicações.

Categoria	Tecnologia	Descrição / Aplicação
Linguagens de Programação	Java	Desenvolvimento de sistemas corporativos robustos e aplicações backend.
	Python	Linguagem líder para automação, integração de sistemas e Inteligência Artificial.
	TypeScript	Essencial para garantir a qualidade e manutenibilidade do código Frontend moderno.
Frameworks e APIs	Spring Boot	Framework Java para arquiteturas de microsserviços escaláveis.

Categoria	Tecnologia	Descrição / Aplicação
	FastAPI / Flask	Frameworks Python para criação ágil de APIs de alta performance (essenciais para IA).
	REST	Padrão arquitetural para integração de sistemas e serviços Web.
Frontend / Interface	React ou Angular	Bibliotecas/Frameworks modernos para interfaces ricas, rápidas e responsivas (SPA).
	HTML5 / CSS3	Padrões fundamentais para estruturação e estilização de interfaces.
	JSP / Thymeleaf	Tecnologias para manutenção de sistemas legados com renderização no servidor.
Inteligência Artificial (IA)	PyTorch / TensorFlow	Frameworks de Deep Learning para criação e treinamento de redes neurais complexas.
	Scikit-Learn / Pandas	Bibliotecas para Machine Learning clássico, estatística e análise de dados.
	LangChain / LlamaIndex	Frameworks para orquestração de IA Generativa e integração com LLMs.
Bancos de Dados e Vetores	PostgreSQL	Banco de dados relacional principal
	Redis	Armazenamento em memória para cache de alta performance e sessões.
	MongoDB	Banco NoSQL para documentos e dados não estruturados.
	Pinecone / Milvus / ChromaDB / extensão pgvector (PostgreSQL)	Bases de dados especializadas em armazenamento e busca por similaridade de *embeddings*, essenciais para arquiteturas RAG (Retrieval-Augmented Generation) e memória de longo prazo para LLMs.
Infraestrutura e Nuvem	AWS / Azure / GCP	Provedores de nuvem para hospedagem e serviços gerenciados de IA/GPU.
	Docker / Kubernetes	Containerização e orquestração para garantir portabilidade e escalabilidade.
DevOps e Mensageria	Git	Controle de versão e gestão de código-fonte.
	RabbitMQ / Kafka	Sistemas de mensageria para comunicação assíncrona entre microsserviços.

UNIVERSIDADE FEDERAL DE SERGIPE
Av. Marcelo Deda Chagas, s/n, - Bairro Rosa Elze, São Cristóvão/SE, CEP 49107-230

TERMO DE REFERÊNCIA

ANEXO XIII – TERMO DE RECEBIMENTO PROVISÓRIO

INTRODUÇÃO
O Termo de Recebimento Provisório trata-se de termo detalhado que declarará que os serviços foram prestados e atendem às exigências de caráter técnico, sem prejuízo de posterior verificação de sua conformidade com as exigências contratuais, baseada nos requisitos e nos critérios de aceitação definidos no Modelo de Gestão do Contrato. **Referência: Inciso XXI, art. 2º, e alínea "i", inciso II, art. 33 da IN SGD/ME Nº 94/2022.**

1 – IDENTIFICAÇÃO			
CONTRATO Nº	xx/aaaa		
CONTRATADO	<Nome do Contratado>	**CNPJ**	xxxxxxxxxxxx
Nº DA OS	<xxxx/aaaa>		
DATA DA EMISSÃO	<dd/mm/aaaa>		

2 – ESPECIFICAÇÃO DOS SERVIÇOS E VOLUMES DE EXECUÇÃO			
SOLUÇÃO DE TIC			
<Descrição da solução de TIC solicitada relacionada ao contrato anteriormente identificado>			
ITEM	**DESCRIÇÃO DO BEM OU SERVIÇO**	**MÉTRICA**	**QUANTIDADE**
1	<Descrição igual ao da OS de abertura>	<Ex.: PF>	<n>
...	...	...	...
...	...	...	...

...	...	...	...
TOTAL DE ITENS			

3 – RECEBIMENTO

Para fins de cumprimento do disposto no art. 33, inciso II, alínea "i", da IN SGD/ME nº 94/2022, por este instrumento ATESTO que os serviços correspondentes à <OS> acima identificada, conforme definido no Modelo de Execução do contrato supracitado, foram executados e

<atende(m)/atende(m) parcialmente/não atende(m)> às respectivas exigências de caráter técnico discriminadas abaixo. Não obstante, estarão sujeitos à avaliação específica para verificação do atendimento às demais exigências contratuais, de acordo com os Critérios de Aceitação previamente definidos no Modelo de Gestão do contrato.

Ressaltamos que o recebimento definitivo desses serviços ocorrerá somente após a verificação desses requisitos e das demais condições contratuais, desde que não se observem inconformidades ou divergências quanto às especificações constantes do Termo de Referência e do Contrato acima identificado que ensejem correções por parte do **Contratado**. Por fim, reitera-se que o objeto poderá ser rejeitado, no todo ou em parte, quando estiver em desacordo com o contrato.

ITEM	ESPECIFICAÇÃO TÉCNICA	ATENDIMENTO	OBSERVAÇÃO
1	<exigências técnicas definidas no TR>	...	.....
...	...	...	.....
...	...	...	.....
...	...	...	.....

4 – ASSINATURA

FISCAL TÉCNICO
_____ <Nome do Fiscal Técnico do Contrato> **Matrícula:** xxxxxx <Local>, <dia> de <mês> de <ano>.

PREPOSTO
_____ <Nome do Preposto do Contrato> **Matrícula**: xxxxxx <Local>, <dia> de <mês> de <ano>.

UNIVERSIDADE FEDERAL DE SERGIPE
Av. Marcelo Deda Chagas, s/n, - Bairro Rosa Elze, São Cristóvão/SE, CEP 49107-230

TERMO DE REFERÊNCIA

ANEXO XIV – MODELO DE TERMO DE RECEBIMENTO DEFINITIVO

TERMO DE RECEBIMENTO DEFINITIVO – SERVIÇOS DE TIC

INTRODUÇÃO
O Termo de Recebimento Definitivo declarará formalmente ao Contratado que os serviços prestados ou que os bens fornecidos foram devidamente avaliados e atendem às exigências contratuais, de acordo com os requisitos e critérios de aceitação estabelecidos. **Referência: Inciso XXII, Art. 2° e alínea "h" inciso I do art. 33, da IN SGD/ME N° 94/2022.**

1 – IDENTIFICAÇÃO			
CONTRATO/NOTA DE EMPENHO N°	xx/aaaa		
CONTRATADO	<Nome da Contratado>	**CNPJ**	xxxxxxxxxxxx
N° DA OS/OFB	<xxxx/aaaa>		
DATA DA EMISSÃO	<dd/mm/aaaa>		

2 – ESPECIFICAÇÃO DOS PRODUTO(S)/BEM(S)/SERVIÇOS E VOLUMES DE EXECUÇÃO				
SOLUÇÃO DE TIC				
<descrição da solução de TIC solicitada relacionada ao contrato anteriormente identificado>				
ITEM	**DESCRIÇÃO DO BEM OU SERVIÇO**	**MÉTRICA**	**QUANTIDADE**	**TOTAL**
1	<descrição igual à da OS/OFB de abertura>	<Ex.: Perfil Alocado>	<n>	<total>

...				
TOTAL DE ITENS				

3 – ATESTE DE RECEBIMENTO

Para fins de cumprimento do disposto no art. 33, inciso II, alínea "h", da IN SGD/ME nº 94/2022, por este instrumento ATESTO/ATESTAMOS que o(s) <serviço(s)/ bem(s)> correspondentes à <OS/OFB> acima identificada foram <prestados/entregues> pelo **CONTRATADO** e ATENDEM às exigências contratuais, discriminadas abaixo, de acordo com os Critérios de Aceitação previamente definidos no Modelo de Gestão do Contrato acima indicado.

ITEM	EXIGÊNCIA CONTRATUAL	ATENDIMENTO	OBSERVAÇÃO
1	<exigência contratual estabelecida no TR >	...	.....
...	...	...	.....
...	...	...	.....
...	...	...	.....

4 – DESCONTOS EFETUADOS E VALOR A LIQUIDAR

De acordo com os critérios de aceitação e demais termos contratuais, <não> há incidência de descontos por desatendimento dos indicadores de níveis de serviços definidos.

<Não foram / Foram> identificadas inconformidades técnicas ou de negócio que ensejam indicação de glosas e sanções, <cuja instrução corre em processo administrativo próprio (nº do processo)>.

Por conseguinte, o valor a liquidar correspondente à <OS/OFB> acima identificada monta em R$ <valor> (<valor por extenso>).

Referência: <Relatório de Fiscalização nº xxxx>.

5 – ASSINATURA

GESTOR DO CONTRATO

_____

<Nome do Gestor do Contrato>
Matrícula: xxxxxxxx

<Local>, <dia> de <mês> de <ano>.

<As seções seguintes podem constar em documento diverso, pois dizem respeito à autorização para o faturamento, a cargo do Gestor do Contrato, e a respectiva ciência do preposto quanto a esta autorização>.

5 – AUTORIZAÇÃO PARA FATURAMENTO
GESTOR DO CONTRATO
Nos termos da alínea "n", inciso I, art. 33, da IN SGD/ME nº 94/2022, AUTORIZA-SE o **CONTRATADO** a <faturar os serviços executados / apresentar as notas fiscais dos bens entregues> relativos à supracitada <OS/OFB>, no valor discriminado no item 4, acima. _____ <Nome do Gestor do Contrato> **Matrícula:** xxxxxxxx <Local>, <dia> de <mês> de <ano>

7 – CIÊNCIA
PREPOSTO
_____ <Nome do Preposto do Contrato> **Matrícula:** xxxxxxx <Local>, <dia> de <mês> de <ano>

UNIVERSIDADE FEDERAL DE SERGIPE
Av. Marcelo Deda Chagas, s/n, - Bairro Rosa Elze, São Cristóvão/SE, CEP 49107-230

TERMO DE REFERÊNCIA

ANEXO XV – RELATÓRIO MENSAL DE FISCALIZAÇÃO

MODELO DE RELATÓRIO MENSAL DE FISCALIZAÇAO PARA CONTRATO DE SERVIÇOS DE DESENVOLVIMENTO, MANUTENÇÃO E SUSTENTAÇÃO DE SOFTWARE

1 – INTRODUÇÃO

O contrato nº <xx/aaaa>, processo <nº do processo>, objeto deste relatório, é relativo à prestação de serviços de desenvolvimento, manutenção e sustentação de software na modalidade de remuneração por Alocação de Profissionais de TI, padronizada pela **Portaria SGD/MGI nº 750, de 2023**.

Os serviços são prestados pela empresa <Nome da Contratado>, CNPJ: <número do CNPJ>, iniciado em dd/mm/2023 (Pregão Eletrônico nº XX/202X).

A fiscalização técnica executada pelo(s) servidor(es) <Nome do(s) Servidor(es)>, iniciou-se em <dd/mm/aaaa>, após a publicação da Portaria Nº <número da portaria>, de <dd/mm/aaaa>.

2 – REGISTRO DE OCORRÊNCIAS

Conforme **§1º do art. 117 da Lei 14.133/21**, a execução do contrato deverá ser acompanhada e fiscalizada por representantes da Administração, que deverão anotar em registro próprio todas as ocorrências relacionadas com a execução do contrato, determinando o que for necessário à regularização das faltas ou defeitos observados.

Dessa forma, a partir da nossa nomeação do(s) fiscal(is) técnico(s), mantém-se registro em planilha eletrônica sobre todas as ocorrências relacionadas à execução deste contrato (tabela a seguir):

Data	Tipo	Ocorrência	Sistema

3 – AFERIÇÃO DOS NÍVEIS DE SERVIÇO

A verificação da adequação da prestação do serviço é realizada com base em Níveis de Serviço definidos nesta seção.

A aferição dos níveis de serviço é realizada mensalmente pelos fiscais do contrato, referente às Ordens de Serviço encerradas no mês anterior.

A ferramenta adotada pelo órgão para controle de demandas é <nome da ferramenta utilizada>. Nos itens

seguintes, serão apurados indicadores:

a. Para os **serviços de desenvolvimento e manutenção de software:**

- Indicador de Aceitação da Sprint/Entrega (IAS);
- Indicador de Produtividade Ágil (IPA);
- Indicador de avaliação individual do Perfil Profissional (IPP);
- Indicador de Qualidade de Código (IQC);
- Indicador de Desmobilização de Equipe (IDE);
- Indicador de Satisfação do dono do Produto (ISP).

b. Para os **serviços de sustentação de software**:
- Indicador de Atendimento aos Prazos de chamados de Sustentação (IAPS),
- Indicador de avaliação individual do Perfil Profissional (IPP);
- Indicador de Qualidade de Código (IQC);
- Indicador de Cobertura de Testes (ICT);

3.1. Indicador de Aceitação da Sprint/Entrega (IAS)

Este indicador tem o objetivo de verificar se as demandas planejadas nas *sprints* foram executadas no *timebox*

e com qualidade.

Para aferição do indicador é feita uma relação de proporção entre a quantidade de sprints aceitas integral- mente e parcialmente junto ao total, chegando a um valor percentual:

A seguir, o mecanismo de cálculo do indicador:

$$\textbf{IAS} = (Qi + Qp/3) \times 100$$

$$Qt$$

Onde:

- **IAS =** Indicador de Aceitação da Sprint/Entrega;
- **Qi =** Quantidade de sprints aceitas integralmente;
- **Qp =** Quantidade de sprints aceitas parcialmente;
- **Qt =** Quantidade total de sprints enviadas para aceite.

Para efeitos desse indicador, não são contabilizadas **sprints** rejeitadas, pois não atendem aos critérios mínimos de aceitação previamente estabelecidos.

O IAS deve ser igual ou superior a 75%.

Qi	**Qp**	**Qt**	**IAS aferido** = $\frac{(Qi + Qp/3) \times 100}{Qt}$	**Situação**

3.2. Indicador de Produtividade Ágil (IPA)

Este indicador visa garantir a produtividade das equipes ágeis, em termos do alcance de metas aferidas por meio de métricas de **software**, observando os critérios de qualidade e de aceitação definidos, bem como mensuração em termo de produto ou resultado entregue.

Para cálculo do indicador, afere-se a produtividade realizada no período, considerando as metas de produtividade previamente estabelecidas na ordem de serviço, conforme descrito no item “Metas de Produtividade” do Termo de Referência.

A seguir, o mecanismo de cálculo do indicador:

IPA=100*(Pr/ Pp)

Onde:

- **IPA =** Indicador de Produtividade Ágil;
- **Pr =** Produtividade realizada no período para os perfis profissionais alocados, em função da métrica de **software** previamente estabelecida;
- **Pp =** Produtividade prevista no período para os perfis profissionais alocados, em função da métrica de **software** previamente estabelecida.

A produtividade é aferida por meio de metas de produtividade previstas em Ordem de Serviço, a critério da Contratante.

Como as metas de produtividades variam de acordo com as atividades desempenhadas pelos profissionais alocados, esse indicador poderá ser calculado mais de uma vez, por período.

O IPA deve ser igual ou superior a 90%

Cálculo de IPA para equipes de Desenvolvimento e Manutenção de SW			
Métrica estabelecida em OS = *<informe a métrica utilizada; ex: 0,58PF/dia, 10h/PF>*			
Pr	**Pp**	**IPA aferido =** 100*(Pr/ Pp)	**Situação**

Cálculo de IPA para Analistas de Dados e Analistas de BI			
Métrica estabelecida em OS = *< ex: Catálogo de serviços de produtos de dados>*			
Pr	**Pp**	**IPA aferido =** 100*(Pr/ Pp)	**Situação**

3.3. Indicador de avaliação individual do Perfil Profissional (IPP)

Este indicador visa assegurar que os profissionais alocados agreguem valor ao time por meio de contribuições técnicas e participação ativa no processo.

A avaliação é realizada mensalmente, POR PERFIL ALOCADO, em projetos de desenvolvimento, manutenção e sustentação de software, por meio de questionário estruturado (Anexo X do TR) baseado em critérios e pontuações previamente definidas com enfoque nas seguintes dimensões:

a) Demonstração de conhecimento técnico;
b) Assiduidade;
c) Comunicação e interação com a equipe.

A seguir, o mecanismo de cálculo do indicador:

IPP = 100 * (ΣPafr / ΣPtot)

Onde:

IPP = Indicador de avaliação individual do Perfil Profissional

ΣPafr = Somatório de Pontuação aferida.

ΣPtot = Somatório de Pontuação total máxima possível para todos os critérios estabelecidos. O IPP

mínimo é de 80%.

Σ Pafr	Σ Ptot	**IPP aferido =** 100 * (ΣPafr / ΣPtot)	**Situação**

3.4. Indicador de Qualidade de Código (IQC)

Este indicador visa assegurar a qualidade do código em projetos de desenvolvimento, manutenção e sustentação e diminuir a ocorrência de defeitos e medir o nível de adequação do código fonte a características de qualidade determinadas pela contratante.

A aferição deste indicador é realizada pela ferramenta <nome da ferramenta automatizada utilizada>. Na ausência ou indisponibilidade da ferramenta, será adotado procedimento de amostragem conforme ANEXO VIII - Roteiro de Amostragem do Termo de Referência.

A seguir, o mecanismo de cálculo do indicador:

IQC = 100 * (ΣQrc / ΣQtr)

Onde:

- **IQC =** Indicador de qualidade de código;
- **ΣQrc =** Somatório da Quantidade de requisitos de qualidade de código atendidos;
- **ΣQtr =** Somatório da Quantidade total de requisitos de qualidade de código avaliados. Os requisitos mínimos de qualidade de código-fonte estão disponíveis no ANEXO VII.

O iQC deve ser maior ou igual a 90% (>=90%)

IQC para projetos de desenvolvimento e manutenção			
ΣQrc	**ΣQtr**	**IQC aferido** = 100 * (ΣQrc / ΣQtr)	**Situação**

IQC para projetos de sustentação			
ΣQrc	**ΣQtr**	**IQC aferido** = 100 * (ΣQrc / ΣQtr**)**	**Situação**

3.5. Indicador de Desmobilização de Equipe (IDE)

Este indicador visa incentivar que o Contratado assegure a manutenção da equipe alocada na execução da **sprint**, ou que crie mecanismos e estratégias para realizar uma substituição transparente (sem prejuízos à execução da **sprint**), promover a comunicação e transferência de conhecimento efetivas.

Projetos que sofrerem desligamento/mudança de integrantes de times ágeis e subsequente insucesso total ou parcial na aceitação de **Sprints** estarão sujeitos ao Índice de Desmobilização de Equipe. O índice somente é aplicado para as sprints que têm aceitação parcial e/ou rejeição, sendo ignorado para sprints aceitas.

Para aferição do indicador, é apurado o somatório de desligamentos de profissionais das equipes ágeis para cada projeto que teve uma **sprint** rejeitada ou aceita parcialmente.

Para cada insucesso parcial ou total de **Sprint**, será observado se houve desligamento de algum membro da equipe em até 2 (duas) **Sprints** imediatamente anteriores (independente da aceitação ou não dessas **Sprints** prévias). Não é considerado o número de desligamentos da **Sprint** corrente.

A seguir, o mecanismo de cálculo do indicador:

IDE = ((ΣQsr * 0,05) + (ΣQsp * 0,025)) *100

Onde:

- **IDE=** Indicador de desmobilização de equipe;

- **ΣQsr =** Somatório do número de desligamentos de pessoal (por projeto) da respectiva equipe ágil nas últimas 2 **Sprints**, anteriores à **sprint** atual rejeitada;
- **ΣQsp =** Somatório do número de desligamentos de pessoal (por projeto) da respectiva equipe ágil nas últimas 2 **Sprints**, anteriores à **sprint** atual aceita parcial- mente.

O IDE deve ser igual a 0 (zero).

ΣQsr	**ΣQsp**	**IDE aferido** = ((ΣQsr * 0,05) + (ΣQsp * 0,025)) *100	**Situação**

3.6. Indicador de Satisfação do dono do Produto (ISP)

Este indicador visa assegurar a qualidade na execução dos processos de entrega dos produtos em termos de satisfação das partes interessadas, segundo critérios pré-estabelecidos.

Avaliação é realizada mensalmente junto aos donos de produtos por meio de questionários estruturado baseado em critérios e pontuações previamente definidas.

A seguir, o mecanismo de cálculo do indicador:

ISP = 100 * (ΣPafr / ΣPtot)

Onde:

- ISP = Indicador de satisfação do Dono de Produto;
- **Σ**Pafr = Somatório de Pontuação aferida;
- **Σ**Ptot = Somatório de Pontuação total máxima possível para todos os critérios estabelecidos.

O ISP deve ser igual ou superior de 80%.

ΣPafr	**ΣPtot**	**ISP aferido** = 100 * (ΣPafr / ΣPtot)	**Situação**

3.7. Indicador de Atendimento aos Prazos de chamados de Sustentação (IAPS)

Este indicador visa assegurar a resposta tempestiva aos chamados relacionados à SUSTENTAÇÃO das aplicações e incentivar a atuação preventiva na execução dos serviços de sustentação.

Para sua aferição é apurada a quantidade de chamados atendidos dentro do prazo máximo estabelecido em relação a quantidade total de chamados atendidos no período de referência.

A seguir, o mecanismo de cálculo do indicador:

IAPS = 100 * (ΣQcap / ΣQctot)

Onde:

- **IAPS =** Indicador de atendimento aos prazos de chamados de sustentação;

- **Qcap =** Quantidade de chamados de sustentação atendidos no prazo máximo estabelecido no TR com previsão de encerramento para o período de referência;
- **Qctot =** Quantidade total de chamados de sustentação registrados com previsão de encerra- mento para o período de referência.

O IAPS deve ser igual ou superior a 90%.

ΣQcap	**ΣQctot**	**IAPS aferido** = 100 * (ΣQcap / ΣQctot)	**Situação**

3.8. Indicador de Cobertura de Testes (ICT)

Este indicador tem o objetivo de incentivar ações proativas de testes de qualidade do código em projetos de sustentação.

A aferição deste indicador é realizada pela ferramenta <nome da ferramenta automatizada utilizada>. A seguir, o mecanismo de cálculo do indicador:

ICT = (I / Tlic)*100

Onde:

- **ICT= I**ndicador de cobertura de testes;
- **I =** número de itens executados (instruções, ramificações e caminhos de código, pontos de decisão do estado de dados ou nomes de elementos de dados);

- **Tlic =** é o número total de itens no código.

O ICT deve ser igual a 100%.

I	**Tlic**	**ICT aferido** = (I / Tlic)*100	**Situação**

4 – CONCLUSÃO

O presente relatório apresentou o resultado da fiscalização do Contrato <número do contrato> referente ao mês de <mês/ano>.

Registre-se ainda que a lista das OS encerradas em <mm/aaaa>, utilizadas para a aferição dos níveis de serviço, foram previamente enviadas por e-mail <dd/mm/aaaa> ao(s) preposto(s) da Contratada para possível manifestação.

A empresa respondeu em <dd/mm/aaaa> e os fiscais técnicos analisaram cada justificativa apresentada.

Essa análise foi enviada por e-mail em <dd/mm/aaaa> ao(s) preposto(s) e ao gestor do contrato, sendo que este Relatório já contempla a análise realizada.

A aferição dos níveis mínimos de serviço das Ordens de Serviço entregues no mês de <mm/aaaa> evidenciou que os seguintes indicadores NAO foram atendidos:

// A tabela a seguir deve ser preenchida apenas com o(s) indicador(es) não atendido(s) no período.

?? Todas as glosas vão ser aplicadas em cima do valor mensal a ser pago na OS de alocação dos profissionais?

Indicadores	**Projetos de**	**Situação**	**Valor aferido**	**Valor da OS**	**Glosas/faixas de ajuste:**	**Valor da Glosa**
Indicador de Aceitação da	Desenvolvimento e manutenção	Não atendido			IAS >= 75%: sem descontos sobre o valor da OS.	
Sprint/Entrega (IAS);					IAS >= 65% e < 75%: 10% de desconto sobre o valor da OS. IAS >= 55% e < 65%: 20% de desconto sobre o valor da OS. IAS < 55%: 30% de desconto sobre o valor da OS.	
Indicador de Produtividade Ágil (IPA)	Desenvolvimento e manutenção	*Não atendido*			IPA>= 90%: sem descontos sobre o valor mensal apurado da OS IPA >= 80% e < 90%: 10% de desconto sobre o valor mensal apurado da OS. IPA >= 70% e < 80%: 20% de desconto sobre o valor mensal apurado da OS. IPA >= 60% e < 70%: 30% de desconto sobre o valor mensal apurado da OS. IPA >= 50% e < 60%: 40% de desconto sobre o valor mensal apurado da OS. IPA < 50%: 50% de desconto sobre o valor mensal apurado da OS.	
Indicador de avaliação individual do Perfil Profissional (IPP)	Desenvolvimento, manutenção e Sustentação	*Não atendido*			IPP >= 80%: sem descontos sobre o valor mensal apurado da OS. IPP >= 70% e < 80%: 1% de desconto sobre o valor mensal apurado da OS. IPP >= 60% e < 70%: 2% de desconto sobre o valor mensal apurado da OS. IPP < 60%: 3% de desconto sobre o valor mensal apurado da OS.	
Indicador de	Desenvolvimento, manutenção e Sustentação				IQC>= 90%: sem descontos sobre o valor mensal apurado da OS IQC >= 80% e < 90%: 10% de desconto sobre o valor mensal apurado da OS. IQC >= 70% e < 80%: 20% de desconto sobre o valor mensal apurado da OS. IQC >= 60% e < 70%: 30% de desconto sobre o valor mensal apurado da OS.	

Qualidade de Código (IQC)		*Não atendido*			IQC >= 50% e < 60%: 40% de desconto sobre o valor mensal apurado da OS. IQC < 50%: 50% de desconto sobre o valor mensal apurado da OS.	
Indicador de Desmobilização de Equipe (IDE)	Desenvolvimento e manutenção	*Não atendido*			O índice IDE representa diretamente o percentual de desconto sobre a fatura mensal.	
Indicador de Satisfação do dono do Produto (ISP)	Desenvolvimento e manutenção	*Não atendido*			ISP >= 80%: sem descontos sobre o valor da OS. ISP >= 70% e < 80%: 2% de desconto sobre o valor da OS. ISP >= 60% e < 70%: 5% de desconto sobre o valor da OS. ISP < 60%: 10% de desconto sobre o valor da OS.	
Indicador de Atendimento aos Prazos de chamados de Sus- tentação (IAPS)	Sustentação	*Não atendido*			IAPS >= 90%: sem descontos sobre o valor da fatura mensal. IAPS >= 80% e < 90%: 10% de desconto sobre o valor da fatura mensal. IAPS >= 70% e < 80%: 20% de desconto sobre o valor da fatura mensal. IAPS < 70%: 30% de desconto sobre o valor da fatura mensal.	
Indicador de Cobertura de Testes (ICT)	Sustentação	*Não atendido*			ICT = 100%: sem descontos sobre o valor da OS. ICT >= 90% e < 100%: 2% de desconto sobre o valor da OS. ICT >= 80% e < 90%: 4% de desconto sobre o valor da OS. ICT >= 70% e < 80%: 6% de desconto sobre o valor da OS. ICT >= 60% e < 70%: 8% de desconto sobre o valor da OS. ICT < 60%: 10% de desconto sobre o valor da OS.	

Os demais indicadores de nível de serviço (informar os indicadores atendidos) foram atendidos.

Dessa forma, encaminhamos este Relatório de Fiscalização para análise do(a) Gestor(a) do Contrato e providências cabíveis.

Este é o relatório.

Documento assinado eletronicamente

<NOME DO FISCAL TECNICO>

Fiscal Técnico

Documento assinado eletronicamente

<NOME DO FISCAL TECNICO>

Fiscal Técnico

UNIVERSIDADE FEDERAL DE SERGIPE
Av. Marcelo Deda Chagas, s/n, - Bairro Rosa Elze, São Cristóvão/SE, CEP 49107-230

TERMO DE REFERÊNCIA

ANEXO XVI – PROCEDIMENTO DE CÁLCULO DO VALOR A SER LIQUIDADO

1. ORIENTAÇÕES PARA CÁLCULO

1.1. A Contratada será remunerada pela quantidade de ordens de serviços entregues no período.

1.2. Quando não houver ordem de serviço aberta, não deverá ocorrer a alocação de profissionais de TI pela Contratada e, consequentemente, não haverá pagamento.

1.3. Para cada ordem de serviço deve-se analisar se a alocação dos profissionais de TI que foram disponibilizados durante a sua execução está de acordo com os perfis, quantitativos e o percentual de alocação solicitados na abertura da ordem de serviço.

1.3.1. Se durante o período de execução da ordem de serviço houver atraso na disponibilização da alocação dos profissionais deverá ser calculado o valor proporcional do custo dos profissionais à partir do período em que efetivamente iniciou- se a alocação dos mesmos, sem prejuízo da aplicação de glosas e sanções de- correntes do inadimplemento dos prazos de alocação.

1.3.2. O custo apurado da alocação dos perfis profissionais para cada ordem de serviço entregue deverá ser calculado conforme fórmula descrita a seguir:

$$Ca = \sum_{1}^{n} (Cp\ x\ q)$$, sendo:

Ca = Custo apurado dos profissionais de TI alocados;

Cp = Custo proporcional dos profissionais de TI alocados durante a execução da ordem de serviço;

n = Quantidade de itens (tipos de perfis) profissionais previstos na ordem de serviços. (Ex.: Desenvolvedor Júnior, Pleno e Sênior; Líder Técnico)

q = Quantidade de profissionais alocados durante a execução da ordem de serviço;

1.3.3. O Custo proporcional do profissional de TI alocado durante a execução da ordem de serviço (**Cp**) deve ser calculado conforme fórmula descrita a seguir:

Cp = Pae x Cdp, sendo:

Pae = Período de alocação efetivo do profissional durante a execução da ordem de serviço, que deverá ser a soma dos dias corridos contabilizados a partir da data de

início da disponibilização do profissional, limitado ao período máximo previsto na ordem de serviço.

Cdp = Custo diário do profissional, que deverá ser calculado observando o Custo total do perfil profissional (**Ctp**) vezes a taxa de alocação (**Ta**) dividido por 30 (trinta) dias. A formula de cálculo será**: Cdp = (Ctp x Ta)/30**, sendo a taxa de alocação considerada como 1 (100%) nos casos em que o perfil profissional alocado seja para prestação dos serviços de sustentação, qualidade e testes avançados de software.

1.3.4. Para melhor compreensão do cálculo do custo diário do profissional (**Cdp**) cita-se os seguintes exemplos:

a) No caso hipotético de uma ordem de serviço de desenvolvimento e manutenção de software executada, observando o período de 28 (vinte e oito) dias e contenha um perfil profissional que custe R$ 20.000,00 (vinte mil reais) com taxa de alocação de 30% (trinta por cento) e período efetivo de alocação de 25 (vinte e cinco) dias, deve-se fazer o seguinte cálculo:

- Cálculo do Custo diário do profissional:

Cdp = (Ctp x Ta)/30, logo:

Cdp = (R$ 20.000,00 x 0,3)/30

Cdp = R$ 200,00

- Cálculo do Custo proporcional do profissional de TI alocado durante a execução da ordem de serviço:

Cp = Pae x Cdp, logo:

Cp = 25 x R$ 200,00 = **R$ 5.000,00**

b) No caso hipotético de uma ordem de serviço de sustentação de software executada, observando o período de 30 (trinta) dias e contenha um perfil profissional que custe R$ 15.000,00 (quinze mil reais) e período efetivo de alocação de 27 (vinte e sete) dias, deve-se fazer o seguinte cálculo:

- Cálculo do Custo diário do profissional:

Cdp = (Ctp x Ta)/30, logo:

Cdp = (R$ 15.000,00 x 1)/30

Cdp = R$ 500,00

- Cálculo do Custo proporcional do profissional de TI alocado durante a execução da ordem de serviço:

Cp = Pae x Cdp, logo:

Cp = 27 x R$ 500,00 = R$ **13.500,00**

1.4. Para cada ordem de serviço, após calcular o custo apurado da alocação dos profissionais de TI (**Ca**) deve-se verificar a adequação da prestação dos serviços através da aferição dos Níveis Mínimos de Serviços (**Dnms**), apurando-se os indicadores estabelecidos na Seção 8.14 do Termo de Referência.

1.4.1. O valor da ordem de serviço será calculado conforme fórmula descrita a seguir:

Vos = Ca – Dnms, sendo:

Vos = Valor da ordem de serviço apurada;

Ca = Custo apurado dos profissionais de TI alocados;

Dnms = Descontos sobre o não atendimento aos níveis mínimos de serviços;

1.5. Para cada conjunto de ordens de serviços apuradas (**∑Vos**) a serem pagas, deve- se observar nos casos de inadimplemento na execução do objeto, a aplicação das Sanções Administrativas e Procedimentos para retenção ou glosa no pagamento estabelecidas na Seção 8.19 do Termo de Referência.

1.5.1. O valor da nota fiscal a ser paga será calculada conforme fórmula descrita a seguir:

Vnf = ∑Vos - ∑Gos, sendo:

Vnf = Valor da nota fiscal a ser paga;

∑Vos = Soma dos valores das ordens de serviços apuradas;

∑Gos = Soma dos valores das glosas sobre as ordens de serviços apuradas;

UNIVERSIDADE FEDERAL DE SERGIPE
Av. Marcelo Deda Chagas, s/n, - Bairro Rosa Elze, São Cristóvão/SE, CEP 49107-230

TERMO DE REFERÊNCIA

ANEXO XVII – ROTEIRO DE CODIFICAÇÃO

1. INTRODUÇÃO

1.1 O presente Roteiro dispõe sobre as melhores práticas de codificação com vistas a assegurar uma codificação enxuta, limpa, clara e eficiente, observando as diretrizes de codificação segura publicadas pela Secretaria de Governo Digital.

2. DIRETRIZES GERAIS

2.1 Além das diretrizes constantes deste Roteiro, deve-se adotar no que couber as orientações constantes dos:

a. **Guia de Requisitos e Obrigações quanto a Privacidade e à Segurança da Informação**, disponível em: https://www.gov.br/governodigital/pt-br/seguranca-e-protecao-de-dados/ppsi/guia_requisitos_obrigacoes.pdf
b. **Guia de Segurança em Aplicações Web**, disponível em: https://www.gov.br/governodigital/pt-br/seguranca-e-protecao-de-dados/guias/guia_seguranca_aplicacoesweb.pdf.
c. **Guia de Requisitos Mínimos de Segurança e Privacidade para APIs**, disponível em: https://www.gov.br/governodigital/pt-br/seguranca-e-protecao-de-dados/guias/guia_seguranca_apis.pdf.
d. **Guia de Requisitos Mínimos de Segurança e Privacidade para Aplicativos Móveis**, disponível em https://www.gov.br/governodigital/pt-br/seguranca-e-protecao-de-dados/guias/guia_seguranca_apps.pdf

3. ESCRITA DE CÓDIGO

3.1 Nomes Significativos

3.1.1 Os nomes utilizados para variáveis, funções, classes e métodos devem ser autoexplicativos, ou seja deve responder as seguintes perguntas: "Por que ele existe?", "o que ele faz?" e "como é usado?".

3.1.2 Não se deve incluir no código informações inverídicas ou falas para mascarar o código ou dificultar sua manutenção.

3.1.3 Os nomes devem ser pronunciáveis e ter significado adequado ao contexto em língua portuguesa.

3.1.4 Evite codificações ou encriptações em nomes de variáveis, classes, funções e métodos.

3.1.5 Os nomes devem ser de fácil entendimento associado ao contexto do domínio do negócio.

3.2 Funções e Classes

3.2.1 As funções e classes devem ter o mínimo de linhas possível, mantendo-se a clareza e implementando o código necessário para traduzir o objetivo esperado da função ou da classe.

3.2.2 As funções e classes devem ter objetivos específicos possuindo um nível de abstração (deve ter apenas uma responsabilidade ou objetivo, ou seja, deve fazer coisa).

3.2.3 A escrita do código deve seguir uma narrativa top-down.

3.2.4 As funções e classes não devem possuir efeitos escondidos ou colaterais, ou seja, não devem produzir efeitos distintos daqueles que se propõe a realizar.

3.2.5 Deve-se evitar níveis de herança exagerados, superior a 4 (quatro).

3.3 Comentários

3.3.1 O código útil deve ser autoexplicativo, limpo e eficiente, devendo possuir apenas os mínimos de comentário possível.

3.3.2 Evite comentários redundantes ou óbvios, que descrevem o que o código faz sem acrescentar nenhuma informação útil.

3.4 Estruturas

3.4.1 Não se deve criar estruturas de código redundantes ou sem efetiva geração de valor à finalidade pretendida.

3.4.2 Não se deve replicar linhas de códigos ou estruturas tornando-as redundantes.

3.4.3 Deve-se evitar duplicar ou replicar funções ou classes que possuam a mesma responsabilidade ou objetivo.

3.5 Identação

3.5.1 Deve-se adotar a identação adequada para melhor entendimento e clareza do código, evitando-se identações exageradas ou nenhuma identação.

3.6 Complexidade ciclomática

3.6.1. Deve-se adotar a menor complexidade ciclomática possível por algoritmo.

3.6.2 Deve-se evitar o aninhamento excessivo de instruções de condições (If's dentro de If''s).

3.6.3 Deve-se evitar o exagero de condições ("Case") dentro de instruções de escolha ("switch")

3.6.4 Deve-se remover construções desnecessárias, ou seja, aquelas que podem ser removidas sem alterar a semântica e a finalidade do código.

4. DIRETRIZES DE SEGURANÇA DA INFORMAÇÃO

4.1 Armazenamento de dados

4.1.1 Nos casos de armazenamento de dados abertos (Dados públicos), não se deve utilizar meio de armazenamento que não possua acesso para escrita restrito por senha.

4.1.2 Nos casos de armazenamento de dados fechados (Dados cujo acesso é restrito a um grupo específico de pessoas), não se deve utilizar meio de armazenamento que não possua acesso para leitura e escrita restrito por senha.

4.1.3 Não se deve disponibilizar à aplicações acesso à algum banco de dados utilizando login de usuário com permissões de root.

4.1.4 Não se deve utilizar o armazenamento de senhas em código-fonte.

4.1.5 Não se deve utilizar as mesmas senhas para ambientes de desenvolvimento ou homologação e produção

4.1.6 Deve-se armazenar de forma segura os dados de usuários e os sistemas que utilizam cada senha fornecida.

4.2 Gerenciamento de Acessos e Permissões de Usuários

4.2.1 Não se deve armazenar senhas em texto plano sem utilizar um algoritmo de hash seguro e salt.

4.2.2 Deve-se utilizar controle de usuário e senha nominais para determinar a identidade do usuário.

4.2.3 Deve-se utilizar HTTPS para controle de autenticação ao menos nas telas de login do sistema web.

4.3 Comunicação

4.3.1 Deve-se empregar canal de comunicação com controle de duplicação e perda de informações/mensagens.

4.3.2 Deve-se empregar canal de comunicação com controle de autenticação (eg, HTTPS, certificados digitais gerados por autoridades confiáveis, VPNs).

4.3.3 Deve-se empregar canal de comunicação que provenha confidencialidade dos dados transmitidos (eg, HTTPS, VPNs).

4.4 Medidas contra-ataques

4.4.1 Deve-se prevenir ataques de injeção de SQL (SQL Injection), a exemplo das seguintes medidas mínimas:

a) Não se deve criar SQLs concatenando parâmetros textuais de origem não-segura, como parâmetros preenchidos pelo usuário ou mesmo armazenados no banco de dados.

b) Deve-se restringir permissões de acesso ao banco de dados para o usuário da aplicação.

c) Deve-se, sempre que possível, passar parâmetros em comandos SQL (DML ou DDL) utilizando prepared statements.

d) Consultas que não podem ser parametrizadas deverão receber tratamento especial, como escapes ou codificação em hexadecimal.

4.4.2 Deve-se prevenir ataques de injeção de HTML e Javascript.

4.4.3 Deve-se prevenir ataques do tipo cross-site scripting (XSS).

4.4.4 Deve-se prevenir ataques de quebra de autenticação e gerenciamento de sessão (Broken Authentication and Session Management).

4.5 Auditoria, rastreamento e Logs

4.5.1 Permitir que eventos sejam registrados a exemplo de:

a) operações de login e logout;

b) acessos a determinadas telas ou seções do sistema;

c) acesso a informações com alguma restrição (eg, documentos sigilosos, processos em segredo de justiça, dados pessoais ou bancários);

d) operações de inclusão, alteração ou exclusão de registros no banco de dados;

e) alteração de perfil de acesso (para sistemas que possuem acesso com diferentes perfis); e -

f) execução de jobs e tarefas automatizadas.

4.5.2 Permitir que informações sejam auditáveis para cada tipo de evento selecionado para monitoramento:

a) data e hora;
b) usuário que efetuou a operação;
c) endereço IP;
d) identificador da sessão do usuário (quando aplicável, eg, cookie);
e) tela (página) do sistema de onde a operação foi realizada;
f) identificador da instância (para sistemas clusterizados);
g) para operações de inserção, alteração ou exclusão, o tipo da operação, nome da tabela que foi manipulada, ID do registro e, se for o caso, valores anterior e atual de cada campo;
h) parâmetros informados pelo usuário (eg, parâmetros GET ou POST), tomando cuidado de não armazenar dados sensíveis, como senhas;
i) tempo de resposta do sistema;
j) para execução de jobs e tarefas automatizadas, armazenar o resultado da operação; falha, sucesso, cancelada, etc.

4.5.3 Permitir formas de captur dos dados para auditoria, a exemplo:

a) alterações aplicadas no banco de dados podem ser auditadas via triggers;
b) auditar as alterações a partir da própria aplicação, sendo que algumas informações poderão não ser registradas (eg, operações SQL realizadas por fora da aplicação).
c) Em sistemas web desenvolvidos em Java, um Filtro pode interceptar as requisições feitas à aplicação.

4.5.3 Deve-se definir no documento de especificação de requisitos do sistema quais informações deverão ser registradas e o local de armazenamento dos dados da auditoria.

4.6 Cópias de Segurança (Backups)

4.6.1 Deve-se incluir no plano de projeto a especificação da necessidade e a atribuição da responsabilidade de realização de backups do banco de dados e dos códigos-fonte do sistema, bem como as políticas de acesso a este backup.

4.6.2 Deve-se definir um procedimento estruturado para a restauração de backups.

4.6.3 Deve-se definir e capacitar responsáveis pela recuperação dos backups.

4.7 Testes

4.7.1 Deve-se realizar testes manuais de segurança antes de cada versão do software que modifique sua estrutura (telas de login, serviços não autenticados, novos formulários com interação com o usuário, etc.).

4.7.2 Deve-se garantir, através de testes automatizados, que os serviços e dados sigilosos estão protegidos e disponíveis apenas para os usuários detentores das informações.

4.7.3 Deve-se elaborar uma política de testes, automatizados ou não, visando a garantia de não vulnerabilidade aos principais ataques conhecidos em sistemas.

4.7.4 Deve-se definir cenários de testes voltados à garantia dos requisitos não funcionais do software, preferencialmente realizado por uma equipe de testes diferente da equipe de desenvolvimento do software, com intuito de se evitar vícios.

4.7.5 Deve-se definir cenários de testes, principalmente nos aspectos de segurança, para os casos de atualizações na arquitetura do sistema (servidores de aplicação, banco de dados, versões de browser, versões de sistema operacional, etc.).

4.8 Medidas contra falhas de segurança

4.8.1 Deve-se manter procedimento planejado para imediata indisponibilização do sistema e realização de manutenção corretiva.

4.8.2 Deve-se definir uma política de acompanhamento pós-correção de ocorrências de falha de segurança.

4.9 Diretrizes para a instalação, configuração e gerenciamento de ambientes de desenvolvimento de sistemas

4.9.1 Deve-se utilizar um controle de versão distribuído, que mantém um repositório completo em cada máquina de desenvolvimento, Eg, Git , Mercurial .

4.9.2 Deve-se utilizar bancos de dados distintos para cada ambiente (Desenvolvimento, produção).

4.9.3 Deve-se utilizar servidores de aplicação/web distintos para cada ambiente (Desenvolvimento, produção).

4.9.4 Deve-se prover acesso ao ambiente de desenvolvimento/testes/homologação apenas aos integrantes da equipe de desenvolvimento e aos interessados no projeto (stakeholders).

4.9.5 Deve-se realizar testes periódicos para assegurar a segurança do ambiente de desenvolvimento/testes/homologação.

4.10 Diretrizes para a configuração de proteção a dados sensíveis

4.10.1 Deve-se utilizar um método criptográfico que siga o princípio de Kerckhoffs; o método de encriptação e seus parâmetros devem ser públicos e estar documentados, somente a chave criptográfica deve ser mantida em sigilo.

4.10.2 Não se deve utilizar um cifrador que admita um método conhecido para quebra da chave criptográfica melhor do que a força bruta, baseada em tentativa e erro.

4.10.3 Não se deve utilizar o modo de cifrador de bloco electronic codebook (ECB) ou modos menos seguros.

4.10.4 Não se deve utilizar um tamanho da chave menor que 128 bits (cifrador simétrico) ou 1024 bits (cifrador assimétrico).

4.10.5 Não se deve utilizar função de hash sem algum tipo de salt.

4.10.6 Não se deve utilizar algoritmos considerados obsoletos para criptografia e hash criptográfico. Exemplos: MD5, SHA1, DES/3DES, RC2, RC4, MD4.

4.10.7 Não se deve utilizar um tamanho da chave menor que 192 bits (cifrador simétrico) ou 2048 bits (cifrador assimétrico).

4.10.8 Não se deve distribuir chaves criptográficas sem a utilização de uma infraestrutura de chave pública e, portanto, sem a utilização de um cifrador assimétrico.

4.11 Geração de Senhas

4.11.1 Não se deve utilizar senhas com menos de 8 caracteres.

4.11.2 Não se deve utilizar somente um tipo de caractere (letras, dígitos e símbolos).

4.11.3 Não se deve utilizar palavras comumente utilizadas para senhas (ou variantes destas), como, por exemplo: nome do animal de estimação, membro da família ou pessoa significativa; datas de aniversário; nome do feriado favorito; algo relacionado ao time esportivo favorito; e as palavras “senha” e “password”.

4.11.4 Não se deve utilizar periodicidade de troca superior a 1 ano.

4.11.5 Não se deve armazenar senhas em claro.

4.11.6 Deve-se armazenar ao menos o hash criptográfico com salt.

4.11.7 Não se deve usar um canal em claro para a transmissão da senha ou elemento correspondente.

4.11.8 Não se deve utilizar método de conferência menos seguro que desafios baseados em hash ou o uso de hashes armazenados.

4.11.9 Não se deve permitir uma taxa de tentativas de validação de senha superior a 5 tentativas por minuto.

4.11.10 Não se deve mostrar diretamente a senha quando esta necessita ser digitada pelo usuário; deve haver opção de habilitar e desabilitar a visualização da senha digitada até então.

4.12 Diretrizes para reforço da segurança de software nas fases de especificação, projeto e design

4.12.1 Deve-se empregar modelo de projeto de software que contemple, no mínimo:

a) a etapa de modelagem de ameaças;

b) definição clara dos riscos de segurança; e

c) nível de severidade que o comprometimento de dados sensíveis traria ao sistema e à instituição.

4.12.2 Não se deve omitir, durante o projeto de desenvolvimento de sistema e sua execução, a definição de responsabilidades pela segurança de dados do sistema e como essa responsabilidade será verificada.

4.12.3 Deve-se utilizar cronograma de projeto que contemple pontos de verificação de segurança do sistema desenvolvido ao longo de sua construção.

4.13 Atualização de softwares

4.13.1 Não se deve habilitar as atualizações automáticas de software ou componentes utilizados na construção de um sistema, sob pena de introdução indevida de falhas de segurança.

Somente utilize código de terceiros que sejam confiáveis.